What is Next when Sequential Prediction Meets Implicitly Hard Interaction?

Kaixi Hu
Wuhan University of Technology
issac_hkx@whut.edu.cn

Lin Li*
Wuhan University of Technology
cathylilin@whut.edu.cn

Qing Xie
Wuhan University of Technology
felixxq@whut.edu.cn

Jianquan Liu
NEC Corporation
jqliu@nec.com

Xiaohui Tao
University of Southern Queensland
Xiaohui.Tao@usq.edu.au

ABSTRACT

Hard interaction learning between source sequences and their next targets is challenging, which exists in a myriad of sequential prediction tasks. During the training process, most existing methods focus on explicitly hard interactions caused by wrong responses. However, a model might conduct correct responses by capturing a subset of learnable patterns, which results in implicitly hard interactions with some unlearned patterns. As such, its generalization performance is weakened. The problem gets more serious in sequential prediction due to the interference of substantial similar candidate targets.

To this end, we propose a Hardness Aware Interaction Learning framework (HAIL) that mainly consists of two base sequential learning networks and mutual exclusivity distillation (MED). The base networks are initialized differently to learn distinctive view patterns, thus gaining different training experiences. The experiences in the form of the unlikelihood of correct responses are drawn from each other by MED, which provides mutual exclusivity knowledge to figure out implicitly hard interactions. Moreover, we deduce that the unlikelihood essentially introduces additional gradients to push the pattern learning of correct responses. Our framework can be easily extended to more peer base networks. Evaluation is conducted on four datasets covering cyber and physical spaces. The experimental results demonstrate that our framework outperforms several state-of-the-art methods in terms of top-k based metrics.

CCS CONCEPTS

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

KEYWORDS

sequential prediction, hard interaction, unlikelihood, knowledge distillation

[*] Lin Li is the corresponding author.

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482492

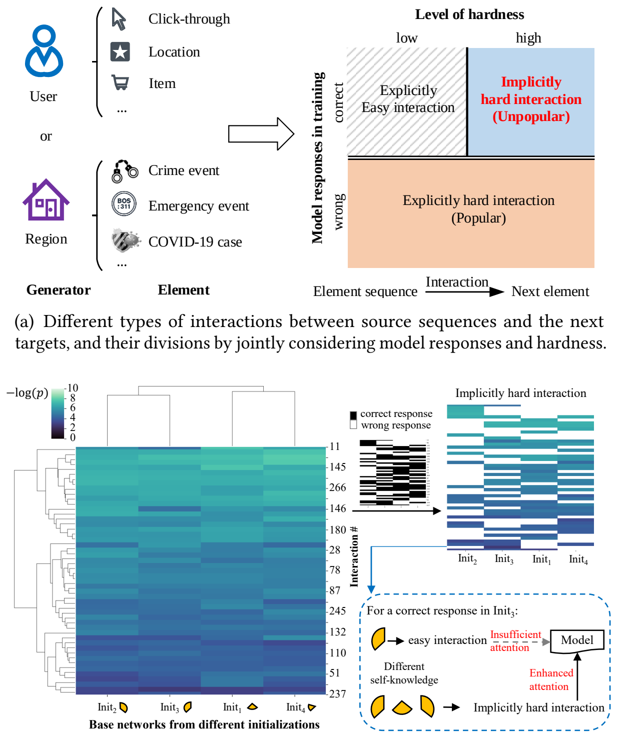
ACM Reference Format:
Kaixi Hu, Lin Li, Qing Xie, Jianquan Liu, and Xiaohui Tao. 2021. What is Next when Sequential Prediction Meets Implicitly Hard Interaction?. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482492

1 INTRODUCTION

In modern society, various sequential prediction tasks can help humans making informed decisions such as recommendation [28, 32], trajectory prediction [21], click-through rate prediction [25, 38] and region-centered event prediction [12]. For example, during the COVID-19 pandemic, precise prediction of infected cases has led to better allocation of healthcare resources [30, 39]. As shown in the left side of Figure 1(a), such tasks typically arrange a series of historical elements (e.g., items, events, locations or their counts) from a certain generator in chronological order, which constitutes a sequence. Existing studies have proved that there exist diverse kinds of interaction patterns between the element sequences and their corresponding next elements [6, 18, 21, 45]. These patterns can give humans a hint about what the future element is like to some degree. In order to provide high-quality predictions, numerous researchers are trying to model and learn the latent interaction patterns comprehensively.

To capture the interaction patterns, various deep learning models are proposed. Most works notice that some interactions involved by humans present more complex characteristics such as irregular interaction [27, 28, 32], dynamic dependency [35, 43], noise interference [12, 20], than others in nature. Therefore, the basic paradigm of their works is to design a well-adapted structure to better model sequential interactions. Recent studies indicate that an effective training strategy can also help to improve prediction performance by making use of interaction information [36, 41, 45]. Despite the effectiveness of prior methods, most of them capture interaction patterns by revising wrong responses where ground-truth is not inferred correctly.

By jointly considering model responses in training and hardness of interactions, the mining of interactions can be divided into three types. As shown in the right part of Figure 1(a), for the lower red region, the wrong responses often with relatively larger training loss are a kind of explicitly hard interactions that have caught the eye of researchers [12, 28, 45]. For the upper region, correct responses



(a) Different types of interactions between source sequences and the next targets, and their divisions by jointly considering model responses and hardness.

(b) Distinctive self-knowledge is generated from different initializations. The deeper the color, the higher likelihood of correct response.

Figure 1: An illustration of implicitly hard interactions caused by subsets of learnable patterns during training.

usually generate relatively lower loss [36]. However, among correct responses, there are kinds of interactions with some patterns unlearned by models. Low loss might make models no longer able to make specific intentional adjustments and prone to a subset of comprehensive patterns. With the limitation of training samples, how to learn such implicitly hard interactions is a burning problem to improve the generalization performance of sequential prediction models. More recently, a multi-view theory [1] proposed by Microsoft Research indicates that individual models have limited capability of capturing multiple views of data. In addition, according to the biased assessment in some psychological mechanisms [4, 16], one-sided perspective is easy to generate biased self-knowledge without communicating with others. Inspired by the above studies, multi-perspective experiences about pattern learning can be introduced to enhance the learning of implicitly hard interactions in sequential prediction. We think that some implicitly hard interactions can be identified from the inconsistent response results across different models. which might make a model easily prone to a subset of learnable patterns.

To further investigate the implicitly hard interactions, we analyze the negative log-likelihood of 50 training samples in ML-1m (a movie ratings dataset) by a hierarchically-clustered heatmap. As shown in Figure 1(b), several observations are listed as follows:

- From the left side, the likelihood of predicting positive samples varies with initializations. And, both the interactions and initializations present cluster effects by observing the dendrogram on the left and top of heatmap, respectively. It denotes that models with different initializations can generate distinctive self-knowledge.
- After masking the wrong responses with white blocks, inconsistent training results accounted for 15.42% of total samples can be observed from the right side. It is noted that some interactions are correctly predicted with high likelihood (deep color) by only a model while other models present wrong responses.

Overall, the observations are in line with the limitation of self-knowledge [1, 4, 16, 37]. And, inspired by them, multiple perspectives of interactions can be generated by distinctive initializations.

To this end, we propose a novel Hardness Aware Interaction Learning framework (HAIL). Our solution aims at exchanging mutual exclusivity knowledge, which aggregates training experiences from other perspectives for learning implicitly hard interactions. *First*, two base networks are developed from different initializations to generate distinctive information about implicitly hard interactions. *Second*, we propose mutual exclusivity distillation (MED) that subtly transfers the unlikelihood of correct responses for different interactions. We further infer that such mutual exclusivity knowledge in the form of unlikelihood can adjust the gradients of models, which can enhance the learning of implicitly hard interactions. This learning paradigm is conducive to improving generalization performance of models by enriching view patterns and can be easily extended to more peers.

Our main contributions are summarized as follows:

- We highlight the impact of implicitly hard interactions and identify their inconsistent characteristics across different perspectives. To our knowledge, it is the first time that implicitly hard interactions are mentioned in sequential prediction tasks.
- A general learning framework HAIL is developed for sequential prediction to enhance the learning of implicitly hard interactions. In particular, MED is proposed to derive mutual exclusivity knowledge, which breaks the conventional manner of mimic learning in knowledge distillation. We further infer that MED essentially introduces additional gradients to push pattern learning of implicitly hard interactions.
- With extensive experiments on two benchmark recommendation datasets from cyber space and two event datasets from physical space, the proposed framework HAIL outperforms existing state-of-the-art methods in several typical applications in terms of top-k based metrics.

2 RELATED WORK

In this section, we review some sequential prediction works with respect to hard interactions, and then investigate the advance of knowledge distillation.

2.1 Sequential Prediction

Sequential prediction is a common technique that is widely used in various domains such as sequential recommendation [28, 32], location prediction [21], click-through rate prediction [25, 38].

Incorporating More Information. Early matrix factorization based methods [7, 10, 15] are difficult to capture hard interactions. With the emerging of deep learning, substantial neural networks are proposed to learn hard interactions by incorporating more information, such as RNN [7, 19, 40] and convolution [2, 32]. Recently, since self-attention [34] shows promising performance, more advanced self-attention based methods are proposed to introduce related information in terms of different applications. These methods can be divided into two aspects: 1) For the tasks in cyber space, recommendation is one of the hotspot research areas [28, 35, 45]; 2) For the tasks in physical space, there are convolution kernels [20], sparse mechanism based Informer [44], geography-aware based GeoSAN [21] and adjacent context based DuroNet [12].

Learning Strategy. More recently, some studies indicate that effective training strategies can also enhance the learning of hard interactions and they are less affected by specific applications. MIM can well capture intrinsic data correlation to avoid overemphasizing the final performance [45]. On the premise of sufficient data, the learning of partial hard interactions can be amplified by removing noisy interactions [36].

The above methods focus on learning hard interactions under self-knowledge from a single perspective, which pays less attention to implicitly hard interactions. Different from them, our work tries to capture implicitly hard interactions no matter which specific cyber or physical spaces they are in by deriving training experience from others. To this end, we propose a novel hardness aware interaction learning framework that mainly consists of two base networks and a MED strategy.

2.2 Knowledge Distillation

Given a training model, this paper focuses on how to draw experiences about learning implicitly hard interactions from other models. Knowledge distillation is an effective means to transfer knowledge between models, which is particularly suitable in our scenario.

Model Compression. Ideas underpinning distillation can date back to model compression [5]. The current and most well-known neural distillation is proposed by Hinton et al. [11] where small student models can derive more information from the softened output of cumbersome teacher models than the ground-truth. Subsequently, a long line of papers about distillation and compression quickly emerges. In sequential prediction, some progress has been made in the *distillation object, architectures and procedures*, respectively. Specifically, some features from the teacher's hidden layer are also distilled to guide the learning of student models [26]. Self-distillation allows the teacher and students lying in a same model architecture [22]. A two-step distillation is proposed for the pre-training and fine-tuning stage, respectively [14].

Non-compression Task. Recently, distillation is proved to be feasible in other tasks that are not for model compression. Two typical works break the learning pattern from teacher models to improve performance of image classification. Born-again neural network obtains improvement from the prior model by teaching selves [9]. Deep mutual learning collects knowledge from student cohort [42]. Moreover, multilingual translation can be integrated into a unified model by distilling different language models [31].

In this work, we break the mimic learning in convention where student models try to reproduce the knowledge from their teacher models. We argue that imitation is not much appropriate to further improve performance, since the teacher models preferentially transfer selective knowledge that they are in high confidence. To this end, MED is proposed to employ mutual exclusivity knowledge that is also a kind of learning experience. In this manner, a model can acquire hints from the unlikelihood of others' correct responses to notice implicitly hard interactions.

3 FRAMEWORK

In sequential prediction, the principal entities are generators (e.g., users, regions) and elements (e.g., items, locations, events). The generator can generate a series of elements in chronological order. Given a set of generators $\mathcal{G} = \{g_1, g_2, ..., g_{|\mathcal{G}|}\}$ and a set of elements $\mathcal{E} = \{e_1, e_2, ..., e_{|\mathcal{E}|}\}$, the interaction sequence in chronological order for generator $g \in \mathcal{G}$ can be denoted as $\mathcal{X}_g = \{e_1^{(g)}, ..., e_t^{(g)}, ..., e_{n_g}^{(g)}\}$, where $e_t^{(g)} \in \mathcal{E}$ is the element that g has interacted with at time step t and n_g is the length of interaction sequence for generator g.

Sequential Prediction for Next. Based on the above notations, the task of sequential prediction can be formally defined as follows: given the historical interaction sequence $\mathcal{X}_g$, the objective is to learn a prediction model $\mathcal{M} : \mathcal{X}_g \rightarrow \mathcal{P}_g$ where $\mathcal{P}_g$ is the likelihood distribution over all elements that generator g possibly interacts with at time step $n_g + 1$. The next element $\hat{\mathcal{Y}}$ can be inferred by sorting the likelihood in descending order.

3.1 Overview

The framework of HAIL is presented in Figure 2. HAIL consists of an shared interaction embedding layer, two base networks (peer for each other) and a shared prediction layer. In the training stage, the experience of learning implicitly hard interactions can be distilled from the peer model. In prediction, either of the base networks can be removed.

The basic idea of our work is to employ the mutual exclusivity knowledge of peer's correct responses to enhance the learning of implicitly hard interactions. To this end, two base networks from different initializations are designed to generate distinctive self-knowledge. Then, MED is proposed to make base networks close to each other and exchange the unlikelihood of correct responses. Finally, each model collects the self-knowledge and the mutual exclusivity knowledge from its peer to adjust the learning weights for implicitly hard interactions.

In the following, we first introduce the components of the proposed framework HAIL. And then, we elaborate on the hardness aware learning and the proposed MED.

3.2 Interaction Embedding Layer

As shown in the lower-left region of Figure 2, a sequence of elements (a.k.a. interaction) is first embedded into fixed-length vectors by looking up a shared embedding table $\boldsymbol{U} \in \mathbb{R}^{|\mathcal{E}| \times d}$ where $|\mathcal{E}|$ is the number of elements and d is the length of vectors. Here, we train the table from scratch without introducing any pre-training parameters. To make use of positional information, we also employ a positional table $\boldsymbol{S} \in \mathbb{R}^{N \times d}$ to generate a fixed-length position vector where N is the maximum length of input sequence. Formally, the output of embedding layer can be derived by suming the element embedding and the positional embedding as $\boldsymbol{h}_i^0 = \boldsymbol{x}_i \boldsymbol{U} + \boldsymbol{s}_i$, where $\boldsymbol{x}_i$, $\boldsymbol{s}_i$ and $\boldsymbol{h}_i^0$ are one-hot input, position and embedding vector of the ith element,

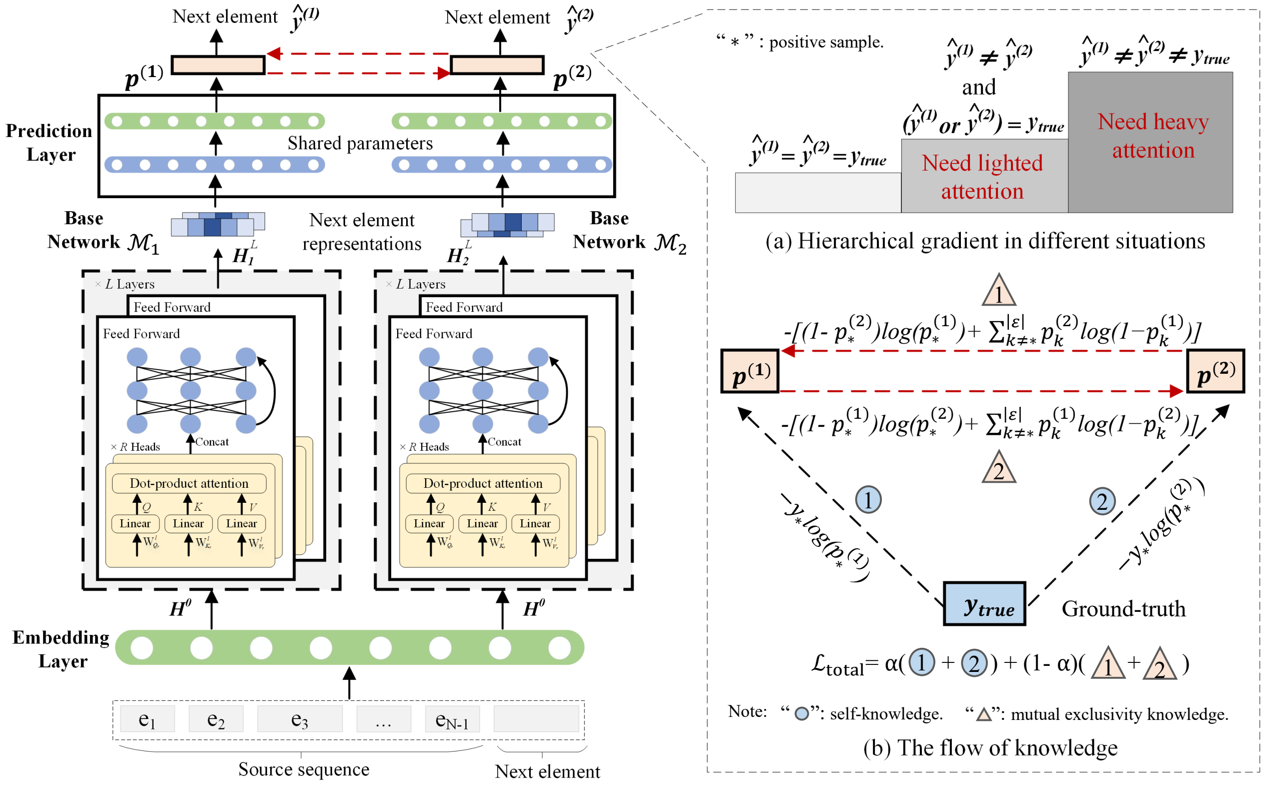

Next element $\hat{y}^{(1)}$
Next element $\hat{y}^{(2)}$
$p^{(1)}$
$p^{(2)}$
Prediction Layer
Shared parameters
Base Network $\mathcal{M}_1$
Next element representations
Base Network $\mathcal{M}_2$
H_1^L
H_2^L
× L Layers
Feed Forward
Feed Forward
× R Heads
Concat
Dot-product attention
Q
K
V
Linear
Linear
Linear
$W_{Q_r}^l$
$W_{K_r}^l$
$W_{V_r}^l$
H^0
H^0
Embedding Layer
e1
e2
e3
…
eN-1
Source sequence
Next element
" * " : positive sample.
$\hat{y}^{(1)} = \hat{y}^{(2)} = y_{true}$
$\hat{y}^{(1)} \neq \hat{y}^{(2)}$ and $(\hat{y}^{(1)}$ or $\hat{y}^{(2)}) = y_{true}$
$\hat{y}^{(1)} \neq \hat{y}^{(2)} \neq y_{true}$
Need lighted attention
Need heavy attention
(a) Hierarchical gradient in different situations
1
$-[(1-p_*^{(2)})log(p_*^{(1)}) + \sum_{k\neq *}^{|\varepsilon|} p_k^{(2)} log(1-p_k^{(1)})]$
$p^{(1)}$
$p^{(2)}$
$-[(1-p_*^{(1)})log(p_*^{(2)}) + \sum_{k\neq *}^{|\varepsilon|} p_k^{(1)} log(1-p_k^{(2)})]$
2
1
2
$-y_* log(p_*^{(1)})$
$-y_* log(p_*^{(2)})$
y_{true}
Ground-truth
$\mathcal{L}_{total}$ = α(1 + 2) + (1- α)(1 + 2)
Note: " ○": self-knowledge. " △": mutual exclusivity knowledge.
(b) The flow of knowledge

Figure 2: The framework of the proposed HAIL. Implicitly hard interactions are identified by the inconsistent results of two base networks as shown in Subfigure (a). The base networks exchange mutual exclusivity knowledge in the form of unlikelihood to enhance the learning of implicitly hard interactions as shown in Subfigure (b).

respectively. The embedding matrix $H^0 = [h_1^0, ..., h_i^0, ...h_N^0]$ will be fed into the following components.

3.3 Base Network

As shown in the middle-left region of Figure 2, to introduce additional knowledge, two base networks initialized differently are employed to independently model interactions. We attempt to capture multi-perspective interactions by employing dark knowledge [42], since models learn from different starting points can derive mutable probability distributions. Note that the architecture of base networks can be set flexibility according to specific applications. Without loss of generality, we do not distinguish network $\mathcal{M}_1$ and network $\mathcal{M}_2$ in the following description for simplicity.

3.3.1 Multi-head Self-attention Layer. In most sequential prediction tasks, interactions hide in a relatively long-time span. The conventional RNNs are easy to meet the vanishing gradient problem [29]. Hence, we adopt the self-attention that captures interactions between elements without regard to distance.

In particular, the input of the lth layer H^{l-1} is transformed into R subspaces simultaneously to derive R attention heads. Then, the heads are concatenated and transformed again to output the representations, after a residual connection. The process can be defined as follows:

$$\begin{aligned} A_0^l &= Concat(\boldsymbol{head}_1, ..., \boldsymbol{head}_r, ..., \boldsymbol{head}_R)W_O^l + H^{l-1}, \\ \boldsymbol{head}_r &= Attention(H^{l-1}W_{Q_r}^l, H^{l-1}W_{K_r}^l, H^{l-1}W_{V_r}^l), \end{aligned} \tag{1}$$

where $W_{Q_r}^l, W_{K_r}^l, W_{V_r}^l \in \mathbb{R}^{d\times d/R}$ are three projection matrices of the rth subspace. $W_O^l \in \mathbb{R}^{d\times d}$ is the output projection. R is the number of heads. A_0^l is the final representation after residual connection. The attention function is a scaled dot-product computation that can be calculated as

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d/R}})V. \tag{2}$$

3.3.2 Feed Forward layer. To introduce more nonlinearity, a two-layer feed forward network is applied on each element representations as follows:

$$\begin{aligned} A_1^l &= \tau(A_0^l W_1^l + b_1^l), \\ A_2^l &= A_1^l W_2^l + b_2^l, \end{aligned} \tag{3}$$

where $W_1^l \in \mathbb{R}^{d\times d_h}$ and $W_2^l \in \mathbb{R}^{d_h\times d}$ are weight matrixes. $b_1^l \in \mathbb{R}^{d_h}$ and $b_2^l \in \mathbb{R}^d$ are the biases. d_h is the dimension of the intermediate layer. $\tau(\cdot)$ is the activation function (GELU in our experiment). Finally, the output of the lth encoder can be derived after a residual connection as $H^l = A_2^l + A_0^l$.

3.4 Prediction Layer for Next

As shown in the upper-left region of Figure 2, the final representations of masked elements are fed into a prediction layer after iteratively computing of the latent patterns. A shared feed forward and the shared embedding table are employed to decode them. The likelihood distribution $\boldsymbol{p}^{(j)}$ of the predicted next element for the jth base network can be derived as follows:

$$\begin{aligned} \boldsymbol{z}^{(j)} &= \tau(\boldsymbol{h}^{(j,L)}\boldsymbol{W_P} + \boldsymbol{b_P})\boldsymbol{U}^T + \boldsymbol{b_U}, \\ \boldsymbol{p}^{(j)} &= \Psi(\boldsymbol{z}^{(j)}), \end{aligned} \tag{4}$$

where $\boldsymbol{h}^{(j,L)}$ denotes the final representation of the Lth layer and L is the number of layers, $\boldsymbol{W_P} \in \mathbb{R}^{d \times d}$ is the weight matrix, $\boldsymbol{b_P} \in \mathbb{R}^d$ and $\boldsymbol{b_U} \in \mathbb{R}^{|\mathcal{E}|}$ are the biases. $\boldsymbol{z}^{(j)} = [z_1^{(j)}, ..., z_{|\mathcal{E}|}^{(j)}]$ is the output of logits. $\Psi(\cdot)$ is the score function (softmax in our experiment).

3.5 Hardness Aware Learning

As shown in right part of Figure 2, for each base network, the ground-truth and the likelihood distribution of the output from its peer are vital sources of knowledge. To make use of them, self-knowledge independent learning and mutual exclusivity knowledge distillation are designed, respectively.

3.5.1 *Self-knowledge Independent Learning.*

To obtain a decent baseline and avoid model drifting arbitrarily, both base networks are designed to learn from ground-truth independently. In this way, they can acquire distinctive self-knowledge from different initial learning points. By following most existing methods [8, 28, 45], self-supervised learning is adopted in our work. In particular, for any element sequence, we randomly mask a proportion of elements with special tokens "[Mask]". This process can be repeated multiple times to generate multiple masked sequences. It is worth noting that more training sequences with final elements masked are appended to avoid fine-tuning in prediction [28].

As the blue circle shown in the lower-right region of Figure 2, the cross-entropy loss is adopted to converge the proposed model. For each input sequence, the self-knowledge based loss of the jth base model can be defined as:

$$\mathcal{L}_{SK}^{(j)} = -\frac{1}{|\mathcal{I}|}\sum_{i \in \mathcal{I}} y_{i,*} log(p_{i,*}^{(j)}), \tag{5}$$

where $\mathcal{I}$ is the index set of masked elements in the input sequence. For the ith masked element, $\boldsymbol{y_i} = [y_{i,1}, ..., y_{i,*}, ..., y_{i,|\mathcal{E}|}]$ is the corresponding ground-truth label. $\boldsymbol{p_i}^{(j)} = [p_{i,1}^{(j)}, ..., p_{i,*}^{(j)}, ..., p_{i,|\mathcal{E}|}^{(j)}]$ is the likelihood distribution of all elements. $*$ denotes the positive sample.

3.5.2 *Mutual Exclusivity Knowledge Distillation.*

When training a model, learning experiences from its peer can be introduced by knowledge distillation. For conventional mimic learning [11, 42], the student model preferentially obtains the experiences with high likelihood. However, for sequential prediction tasks, the likelihood of hard interactions is generally not on a high level. Hence, such knowledge from mimic learning is not suitable for our tasks. To address the problem, we employ another learning experience that is mutual exclusivity knowledge of correct responses to enhance the learning of implicitly hard interactions.

As the red triangle shown in the right bottom part of Figure 2, the mutual exclusivity knowledge based loss is derived by employing the posterior unlikelihood of correct responses from the peer network. For each training sequence, the loss of the jth base network can be defined as:

$$\mathcal{L}_{MEK_{pos}}^{(j)} = -\frac{1}{|\mathcal{I}|}\sum_{i \in \mathcal{I}}(1 - p_{i,*}^{(\neq j)}) log(p_{i,*}^{(j)}), \tag{6}$$

where $p_{i,*}^{(\neq j)}$ denotes the likelihood of positive label from the peer network. Note that $p_{i,*}^{(\neq j)}$ is the distillation target in conventional mimic learning [11]. Differently, in our work, $(1 - p_{i,*}^{(\neq j)})$ represents the mutual exclusivity knowledge that is distilled in the learning of positive samples.

Most sequential prediction tasks require predicting the next element from tens of thousands of candidates. It is intuitive that the highly similar elements might present more serious interference for the target. To this end, the mutual exclusivity knowledge is also introduced in the learning of negative samples. The sum of their loss is defined as:

$$\mathcal{L}_{MEK_{neg}}^{(j)} = -\frac{1}{|\mathcal{I}|}\sum_{i \in \mathcal{I}}\sum_{k \neq *}^{|\mathcal{E}|} p_{i,k}^{(\neq j)} log(1 - p_{i,k}^{(j)}), \tag{7}$$

where $p_{i,k}^{(j)}$ is the likelihood of the kth element in the likelihood distribution $\boldsymbol{p_i}^{(j)}$ and $p_{i,k}^{(\neq j)}$ is the likelihood of the kth element from the peer model. $*$ denotes the positive sample.

3.5.3 *Denoising.*

The interference of noisy interactions which do not reflect true preference [36], is an inevitable problem when enhancing the learning of implicitly hard interactions. It is harmful to make model fit them, which may hurt the generalization [3, 13]. Some works point out that the number of noisy interactions is less and their losses are larger [17, 36]. As such, the shared interactions with large losses in the self-knowledge independent learning are truncated in the failure experience mutual learning. Formally, for each positive or negative sample, the truncated hard interaction loss is defined as follows:

$$\bar{\mathcal{L}}_{MEK_{\{pos,neg\}}}^{(j)} = \begin{cases} 0, & rank(\mathcal{L}_{SK}^{(\forall j)}) < \beta \cdot S_L \\ \mathcal{L}_{MEK_{\{pos,neg\}}}^{(j)}, & otherwise, \end{cases} \tag{8}$$

where β is the proportion of truncated noisy interactions and S_L is the size of interactions. rank$(\cdot)$ denotes the rank of the loss in all interactions in descending order. In Equation (8), the noisy interactions are also identified by jointly conditioning on the losses from both networks.

3.5.4 *Loss Balance.*

To balance the knowledge between the ground-truth and the peer model, a balance factor α is introduced to derive the total loss of the jth base model as follows:

$$\mathcal{L}_{total}^{(j)} = \alpha\mathcal{L}_{SK}^{(j)} + (1 - \alpha)(\bar{\mathcal{L}}_{MEK_{pos}}^{(j)} + \bar{\mathcal{L}}_{MEK_{neg}}^{(j)}). \tag{9}$$

Eventually, the final loss function that is adopted to converge the proposed model is defined as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{total}^{1} + \mathcal{L}_{total}^{2}. \tag{10}$$

3.6 Extension to More Peers

Notwithstanding the promising exclusivity knowledge transfer between two base networks, the proposed MED can be naturally extended to more peers with different parameters or structures. More base networks are expected to introduce such knowledge from diverse perspectives to conduct learning of implicitly hard interactions. And in the follow-up deployment, the redundant networks can be removed for reducing the computation.

Given T ($T \geq 2$) base networks, the mutual exclusivity knowledge distillation can be extended as Equations (11) and (12). For each network, it can obtain $T-1$ hints that are introduced from the other peers. Such that, Equations (6) and (7) can be regarded as a special situation with $T = 2$. Here, a rescaling factor $\frac{1}{T-1}$ is introduced to ensure a balanced value of loss.

$$\tilde{\mathcal{L}}^{(j)}_{MEK_{pos}} = -\frac{1}{T-1} \sum_{c \neq j}^{T} (1 - p_*^{(c)}) log(p_*^{(j)}). \tag{11}$$

$$\tilde{\mathcal{L}}^{(j)}_{MEK_{neg}} = -\frac{1}{T-1} \sum_{c \neq j}^{T} \sum_{k \neq *}^{|\mathcal{E}|} p_k^{(c)} log(1 - p_k^{(j)}). \tag{12}$$

4 DISCUSSION

The key contribution of our work is to enhance the learning of implicitly hard interactions by employing training experience. To obtain more insight about it, we explore how MED works. Without loss of generality, the discussion focuses on the model $\mathcal{M}_1$. We aim to answer the following questions:

Question 1: *Why does MED work? What does mutual exclusivity knowledge distillation bring?*

During the exchange of training experience, the parameters are usually updated along the direction of negative gradient. Without loss of generality, for each positive sample in the base model $\mathcal{M}_1$, the gradient of cross-entroy [23] in Equation (6) with respect to base model's logits $z_*^{(1)}$ in Equation (4) can be derived as:

$$\begin{aligned} \frac{\partial \mathcal{L}^{(1)}_{MEK_{pos}}}{\partial z_*^{(1)}} &= \frac{\partial \mathcal{L}^{(1)}_{MEK_{pos}}}{\partial p_*^{(1)}} \frac{\partial p_*^{(1)}}{\partial z_*^{(1)}} = -\frac{1 - p_*^{(2)}}{p_*^{(1)}} \cdot p_*^{(1)}(1 - p_*^{(1)}) \\ &= \underbrace{(1 - p_*^{(2)})(p_*^{(1)} - y_*)}_{①} = \underbrace{p_*^{(1)} - y_*}_{self-knowledge} + \underbrace{p_*^{(2)}(y_* - p_*^{(1)})}_{②}, \end{aligned} \tag{13}$$

where $y_* = 1$ is the label of positive samples. $p_*^{(1)}$ and $p_*^{(2)}$ are the likelihood of the next element.

As shown in Equation (13), the final gradient is rewritten as term ①. In this form, $(1 - p_*^{(2)})$ can be interpreted as an importance weight of the original ground-truth label y_*. When base model $\mathcal{M}_2$ makes a serious mistake (i.e., $p_*^{(2)} \approx 0$), Equation (13) is approximate to the gradient generated by the self-knowledge based loss in Equation (5). In this situation, base model $\mathcal{M}_2$ gives base model $\mathcal{M}_1$ a hint that the current interaction is hard to learn. And, base model $\mathcal{M}_1$ will enhance the learning of the interaction with a gradient affected by another perspective. While base model $\mathcal{M}_2$ with a higher likelihood, the gradient of the interaction will be rescaled and generate less contribution. Along this line, MED can take similar effect on negative samples.

Compared with self-knowledge independent learning, mutual exclusivity knowledge distillation contains an additional term ② in Equation (13). The gradient of implicitly hard interaction is derived by jointly conditioning on the likelihood from distinctive initializations. More views of data are introduced to enhance the learning of implicitly hard interactions. As shown in the upper-right region of Figure 2, the combination of prediction results and ground-truth can be divided into different levels. In particular, if base models generate inconsistent results, a trade-off gradient can be derived. Therefore, mutual exclusivity knowledge distillation conduct a hierarchical learning of different interactions.

Question 2: *What is the difference between conventional mimic learning (learn from $p_*^{(2)}$) and MED (learn from $1 - p_*^{(2)}$)?*

It is straightforward that the proposed MED can dynamically rescale the importance weights of interactions by their wrong responses, which aims to distinguish the next target from substantial candidates. The conventional mimic learning carries information from teacher to students, which emphasises similar knowledge among similar elements. In this case, the weight $(1 - p_*^{(2)})$ in Equation (13) is replaced by $p_*^{(2)}$, which means the model prefers to easy interactions. Therefore, for substantial elements in sequential prediction, MED can effectively reduce the interference, which is more suitable in our scenario.

5 EXPERIMENTS

In this section, experiments are conducted on different datasets to validate the effectiveness of our HAIL. In particular, we aim to answer the following research questions:

- **RQ1:** How does our HAIL perform compared with the state-of-the-art sequential prediction methods?
- **RQ2:** How is the performance of HAIL variants with different combinations of terms in loss function (Equation (9))?
- **RQ3:** What is the effect of the truncation proportion β in denoising (Equation (8)) and the balance factor α in loss function (Equation (9))?

5.1 Experimental Setup

5.1.1 Dataset. Experiments are conducted on two benchmark recommendation datasets from cyber space and two crime datasets from physical space. These datasets that are involved by humans contain more implicitly hard interactions. As shown in Table 1, the size of different datasets varies with the domains.

- **ML-1m**[1]: This is a movie ratings dataset created in February, 2003. As one of the stable benchmark datasets, most recommendation algorithms are evaluated on it.
- **Toys**: Toys is a subcategory in Amazon review dataset. We obtained this dataset from [45].
- **CHI-18**[2]: This is a public crime dataset updated by Chicago Police Department in 2018. To achieve a more fine-grained prediction, an event is described by a simple event model (SEM) [33]. In particular, SEM can model basic events in various domains without domain-specific vocabularies. In this work, the fields about geographical information are extracted to describe

[1] https://grouplens.org/datasets/movielens/1m/

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

Table 1: Statistics of experimented datasets

Dataset	Generator #	Elements #	Sequence Length			
			Max.	Min.	Avg.	Std.
ML-1m	6,040	3,416	2,275	16	163.50	192.53
Toys	19,412	11,924	548	3	6.63	8.50
CHI-18	2,692	246	3,525	3	96.18	253.36
NYC-16	3,229	440	4,496	3	144.76	429.78

generators. Then, time slots are divided every 3 hours, which is smaller than the meaningful interval 6 hours [24]. The time slot and crime type are together used to describe elements.

- **NYC-16**[3]: This is a crime dataset provided by New York City Police Department in 2016. Similar to Chicago, the precincts and premises are used to describe generators. The time slots and classification codes are leveraged to describe elements.

For all datasets, the elements are grouped by generators and sorted in chronological order for each generator. The inactive generators with fewer than five elements are removed to ensure the quality of prediction. Moreover, the last element in each sequence is taken as the test data and the element before the last element as the validation set. The remaining elements are used for training. The maximum length of sequence is set as 200. To ensure the sequence within the maximum length, longer sequences will be sliced into multiple sub-sequences from right to left.

5.1.2 Metrics. Following the common assessment means [35, 45], the performance of prediction can be assessed by top-k Hit Ratio (HR@k), top-k Normalized Discounted Cumulative Gain (NDCG@k), and Mean Reciprocal Rank (MRR). In this paper, the cutoff k is set as {1,5,10}. Note that HR@1 is equal to NDCG@1 that is a harsh metric for performance evaluation. To achieve an efficient computation in a large candidate set, 99 negative elements are randomly selected to rank with the target element. For all metrics, the higher the value, the better the performance.

5.1.3 Settings. We implemented the proposed HAIL in Python with TensorFlow and conducted experiments on a commodity machine equipped with a 12GB TITAN Xp GPU. We train the model by using Adam with NOAM decay [34]. The batch size is set as 256. In base network, we set the layer number L as 2, the head number R as 2, the hidden size d as 64, the intermediate size as 256. For learning hyper-parameters, β is tuned in {0,0.01,0.02,0.03}, α is searched in {0.1,0.2,...,0.8,0.9}. The source code is available at GitHub[4].

5.2 Baselines

To validate the effectiveness and generalization of our HAIL, we conduct a comparison with eight baselines from different domains. They are elaborated as follows:

- **POP**: A non-sequential baseline that simply regards the frequency of interactions as the probability of the next element.

Recommendation Methods:

- **BERT4Rec**[28]: BERT4Rec is a session-based method adapted from the language model BERT [8]. It employs bidirection information to model interactions which are not in a rigid order.
- **R-CE**[36]: An adaptive denoising training strategy (ADT) is applied for BERT4Rec to reduce the effect of hard interactions.
- **S^3-Rec**[45]: It utilizes mutual information maximization to capture intrinsic data correlation for sequential recommendation. For fair comparison, the extra attributes of items are removed, and the MIP and SP objectives are employed.
- **HyperRec**[35]: It adopts hypergraph to model dynamic interactions between users and items in recommendation.

Event Prediction Methods:

- **DuroNet-s**[12]: It is a robust crime count prediction model that reduces the point-wise and the sequence-wise effect of noises. To adapt to our task, the spatial module is removed.

General Sequential Prediction Methods:

- **Convolutional Self-attention (CSa)**[20]: It reduces the sensitivity to anomalies in series by utilizing causal convolution.
- **Informer**[44]: An efficient transformer-based model to capture dependence in extreme long sequences.

For BERT4Rec[5], S^3-Rec[6], HyperRec[7], DuroNet-s[8], and Informer[9], we use the code released by the authors. For CSa, we reproduce it in Pytorch. To make some regression methods adapt to our tasks, we add a shared embedding layer before and after the original model and adopt cross-entropy to train them. We adjust the hidden dimension size from {32,64,128}. The other hyper-parameters are set as reported in the papers. Their results are reported under the optimal settings.

5.3 Overall Performance Comparison (RQ1)

The comparison results with all baselines are shown in Table 2. On the right side, we count the differences of accuracy between the base models $\mathcal{M}_1$ and $\mathcal{M}_2$, and the improvements of the best results relative to the suboptimal results. Several observations are summarized as follows:

For non-recommendation methods, the recommendation methods outperform them in most metrics. It is possibly caused by two reasons: 1) the non-recommendation methods generally assume a rigidly ordered sequence and design a relatively coarse-grain regression task to predict the next counts; 2) most recommendation methods conduct a cloze task [8] to pretrain their models, which generates more samples to train the models. Both DuroNet-s and CSa employ a convolution operator to reduce the effect of noises, which smooths the representations of related elements and reduces the differences. Hence, their values of HR@1 are significantly lower than other metrics. Informer is a specific method for extreme long sequences. However, it seems not suitable for recommendation scenario since it yields an inconsistent performance than POP in the recommendation datasets which lengths are short.

Among recommendation methods, BERT4Rec achieves comparable performance with S^3-Rec and HyperRec. It indicates that BERT4Rec makes use of interaction information without modeling generator. However, they do not outperform HAIL since they learn implicitly hard interactions under self-knowledge. When further changing the learning strategy of BERT4Rec with R-CE, the

[3] https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Current-Year-To-Date-/5uac-w243

[4] https://github.com/hukx-issac/HAIL

[5] https://github.com/FeiSun/BERT4Rec

[6] https://github.com/RUCAIBox/CIKM2020-S3Rec

[7] https://github.com/wangjlgz/HyperRec

[8] https://github.com/hukx-issac/DuroNet-for-crime-prediction

[9] https://github.com/zhouhaoyi/Informer2020

Table 2: Accuracy comparison with baselines on four datasets. The optimal results are denoted in bold while the suboptimal results are underlined. "∗" indicates significant improvement.

Dataset	Metric	Recommendation					Non-recommendation			HAIL(ours)		Diff.	Improv.
		POP	BERT4Rec	R-CE	S^3-Rec	HyperRec	DuroNet-s	CSa	Informer	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_1$-$\mathcal{M}_2$	
ML-1m	HR@1	0.0407	0.3695	0.3988	0.2897	0.3180	0.1321	0.1778	0.0265	**0.4291***	0.4252	+0.0039	7.60%
	HR@5	0.1603	0.6851	0.6478	0.6575	0.6631	0.3849	0.4629	0.1154	0.7202	**0.7214***	-0.0012	5.30%
	HR@10	0.2775	0.7823	0.7404	0.7911	0.7738	0.5412	0.6108	0.2023	0.8098	**0.8146***	-0.0048	2.97%
	NDCG@5	0.1008	0.5375	0.5327	0.4557	0.5014	0.2616	0.3243	0.0707	**0.5862***	0.5843	+0.0019	9.16%
	NDCG@10	0.1383	0.5690	0.5627	0.5266	0.5375	0.3121	0.3723	0.0986	**0.6155***	0.6134	+0.0021	8.17%
	MRR	0.1233	0.5108	0.5179	0.4535	0.4731	0.2615	0.3154	0.0922	0.5622	**0.5791***	-0.0169	13.37%
Toys	HR@1	0.0260	0.1390	0.1130	0.0990	0.1147	0.0465	0.0534	0.0144	**0.1783***	0.1780	+0.0003	28.27%
	HR@5	0.1046	0.3379	0.3189	0.3023	0.2875	0.1608	0.1754	0.0682	0.3751	**0.3755***	-0.0004	11.13%
	HR@10	0.1848	0.4596	0.4529	0.4393	0.3909	0.2572	0.2723	0.1286	**0.4796***	**0.4796***	+0.0000	4.35%
	NDCG@5	0.0652	0.2409	0.2179	0.2021	0.2031	0.1040	0.1148	0.0407	**0.2802***	**0.2802***	+0.0000	16.31%
	NDCG@10	0.0909	0.2802	0.2611	0.2463	0.2365	0.1350	0.1459	0.0600	**0.3138***	0.3136	+0.0002	11.99%
	MRR	0.0861	0.2444	0.2233	0.2081	0.2087	0.1211	0.1301	0.0628	**0.2812***	0.2810	+0.0002	15.06%
CHI-18	HR@1	0.0249	0.4421	0.4114	0.3978	0.1679	0.1386	0.1378	0.3507	**0.4744***	0.4703	+0.0041	7.31%
	HR@5	0.1668	0.6861	0.6349	0.6664	0.3956	0.4577	0.4499	0.5914	**0.7117***	0.7099	+0.0018	3.73%
	HR@10	0.3250	0.8024	0.7708	0.7942	0.6088	0.6356	0.6333	0.7166	**0.8243***	0.8228	+0.0015	2.73%
	NDCG@5	0.0926	0.5691	0.5253	0.5383	0.2834	0.3039	0.2981	0.4744	**0.5985***	0.5956	+0.0029	5.17%
	NDCG@10	0.1440	0.6068	0.5692	0.5799	0.3525	0.3613	0.3569	0.5148	**0.6347***	0.6321	+0.0026	4.60%
	MRR	0.1190	0.5567	0.5197	0.5243	0.2957	0.2961	0.2917	0.4650	**0.5853***	0.5823	+0.0030	5.14%
NYC-16	HR@1	0.0660	0.4339	0.4472	0.3874	0.3137	0.1871	0.1975	0.3685	**0.4772***	0.4754	+0.0018	6.71%
	HR@5	0.1994	0.6931	0.6261	0.6909	0.6358	0.5401	0.5509	0.6178	0.7166	**0.7182***	-0.0016	3.62%
	HR@10	0.3537	0.8250	0.7088	0.8287	0.7690	0.7114	0.7275	0.7461	0.8433	**0.8458***	-0.0025	2.06%
	NDCG@5	0.1332	0.5668	0.5396	0.5446	0.4816	0.3688	0.3795	0.4968	0.6014	**0.6019***	-0.0005	6.19%
	NDCG@10	0.1824	0.6095	0.5665	0.5893	0.5247	0.4244	0.4363	0.5383	0.6427	**0.6435***	-0.0008	5.58%
	MRR	0.1616	0.5527	0.5322	0.5245	0.4611	0.3511	0.3608	0.4862	**0.5893***	**0.5893***	+0.0000	6.62%

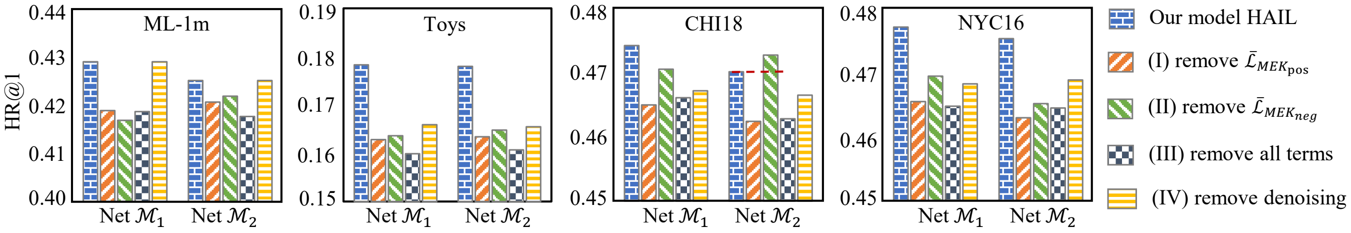


Figure 3: Ablation Study of HAIL in terms of HR@1. The higher histograms, the better the performance.

performance does not present significant improvement. It demonstrates that enhancing the learning of implicitly hard interaction is effective to enrich model patterns.

Finally, by comparing all the baselines, we can find that HAIL consistently achieves significant improvements. In terms of difference, the performance of model $\mathcal{M}_1$ is highly closed to that of model $\mathcal{M}_2$, indicating that exchanging mutual exclusivity knowledge can effectively reduce the gap between two models and improve generalization performance. For the dominant performance on datasets from different domains, HAIL is proved to be generalized to a much broader cyber or physical scenarios.

5.4 Ablation Study (RQ2)

To investigate the effectiveness of components in mutual exclusivity knowledge distillation, we remove the positive sample term (variant I), the negative sample term (variant II), both of them (variant III) in the Equation (9) and the denoising strategy (variant IV) in the Equation (8), respectively. Note that the shared embedding and prediction layers are still kept. And then, we compare their performance with the original HAIL in terms of HR@1, since it is a harsh metric for performance evaluation. The results are reported in Figure 3. We have the following findings:

- *Finding 1: Mutual exclusivity knowledge distillation effectively improves the performance.* Compared with variant III, the performance of original model improves relatively 2.48%, 11.51%, 1.76% and 2.67% on four datasets, respectively. It demonstrates that enhancing the learning of implicitly hard interaction can improve the generalization performance of models.
- *Finding 2: The contribution of positive sample term is robust.* When removing the positive sample term, the performance of variant I drops obviously. For example, compared with the original model, the performance of variant I decreases relatively 2.43%, 9.52%, 2.47% and 2.51% on four datasets. Meanwhile, the negative sample term shows a little fluctuation, particularly for CHI18 dataset (the red dash line). The observation demonstrates that distilling

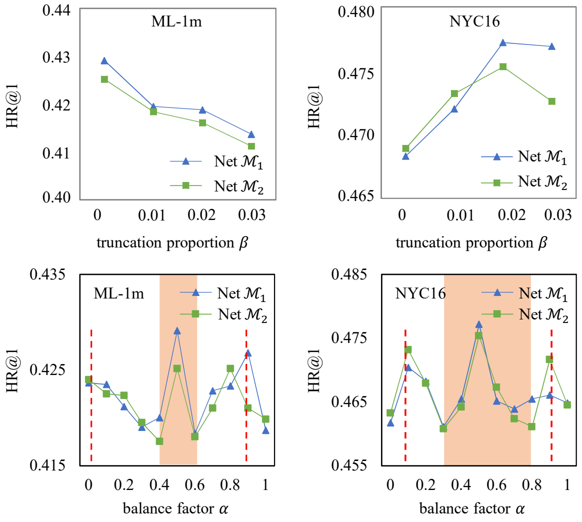


Figure 4: The analysis of parameter sensitivity (HR@1).

mutual exclusivity knowledge in positive samples is more efficient. The negative sample term might be affected by the size of elements, which denotes less candidates in prediction.

- *Finding 3: The denoising strategy proves to be helpful.* Except for the benchmark ML-1m dataset, denoising is imposed on the other datasets at different degrees. Compared with the original model, using denoising strategy improves the performance by 7.16%, 1.50% and 1.87% on the three datasets. This is because denoising can effectively avoid the interference of noisy interactions.

5.5 Parameter Sensitivity (RQ3)

To investigate the effect of different parameters, we tune the value of truncation proportion β from 0 to 0.03 with a step 0.01 and the value of balance factor α from 0 to 1 with a step 0.1.

As shown in the first row of Figure 4, the bigger truncation proportion β, the greater strength of denoising. It can be observed that the performance of model directly falls or first rises and then falls with the growth of the value β. This is because the fitting of noisy interactions might mislead models. And, the performance can be improved if noises are removed. However, if limiting the noises too much, the meaningful hard interactions might be damaged and the performance decreases.

As shown in the second row of Figure 4, the bigger balance factor α, the smaller proportion of failure experience based loss. The red dash line indicates a meaningful region with a shape like "W". It implies that HAIL performs better when the ratio of two types of knowledge based learning is balanced or either of them achieves a dominated ratio. In fact, with the increase of failure experience based on loss (α varies from 1 to 0), the learning will step into three independent stages. To further explain this observation, the curve can be divided into three regions.

- For the first stage in the right region, the proportion of self-knowledge based loss is dominated. Here, mutual exclusivity knowledge is similar to regularization, since some extra information about parameter learning can be introduced to lightly adjust the risk of overfitting.

Table 3: The comparison between conventional knowledge distillation (CKD) and the proposed MED.

Dataset	Mutual mimic learning [42]			MED		
	HR@1	NDCG@5	MRR	HR@1	NDCG@5	MRR
ML-1m	0.3952	0.5656	0.5386	0.4291	0.5862	0.5622
Toys	0.1693	0.2761	0.2767	0.1783	0.2802	0.2812
CHI-18	0.4699	0.5932	0.5815	0.4744	0.5985	0.5853
NYC-16	0.4660	0.5956	0.5806	0.4772	0.6014	0.5893

- For the center red region, both losses get into a balance period. In this stage, the base models can obtain a decent baseline from the self-knowledge and enhance the learning of implicitly hard interactions by employing mutual exclusivity knowledge. However, the performance decreases after the balance is broken.
- As shown in the left region, the models mainly learn from mutual exclusivity knowledge. This stage is similar with the first stage where self-knowledge based loss is like a regularization. However, if its weight is set to 0, both the models might mislead each other.

5.6 Evaluation w.r.t. Mutual Distillation

To investigate the difference between mutual mimic learning [42] and MED, two types of knowledge (i.e., likelihood distribution) mentioned in §3.5.2 are transferred between two base models, respectively. The results from either of base models are randomly selected as the final output. As shown in Table 3, the proposed MED is better than mutual mimic learning. It is because mutual mimic learning makes models prone to the learning of easy interactions while hard interactions might be taken for noises. However, sequential prediction is a challenging task with complex interaction patterns and substantial candidates. The proposed MED can effectively enhance the learning of implicitly hard interactions, which is more suitable for our task.

It is fair to discuss the increase of parameters and runtime. As shown in the left part of Figure 2, this work tries to reduce parameters by introducing more shared layers. Parameters are only doubled in the base networks. In terms of the benchmark ML-1m dataset, the training time of mutual distillation is about 17 seconds for each batch while the individual model is about 9 seconds.

6 CONCLUSION AND FUTURE WORK

In this paper, we highlight the effect of implicitly hard interactions. To this end, a hardness aware interaction learning framework called HAIL is proposed to enhance the learning of them. In particular, based on the proposed MED, implicitly hard interactions are identified from different perspectives. And, both base models derive training experience from each other to adjust themselves learning strategy. Extensive experiments are conducted on four datasets covering cyber and physical spaces. The results show that HAIL outperforms several state-of -the-art methods. For future work, we are interested in extending the framework to more complex sequential prediction tasks, such as multi-modal prediction, and sequence-to-sequence prediction.

ACKNOWLEDGMENTS

This work is partially supported by the China Scholarship Council (LiuJinMei [2020] 1509, 202106950041).

REFERENCES

[1] Zeyuan Allen-Zhu and Yuanzhi Li. 2020. Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning. arXiv:cs.LG/2012.09816

[2] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv:cs.LG/1803.01271

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*. PMLR, 41–48.

[4] Jonathon D Brown. 1991. Accuracy and bias in self-knowledge. *Handbook of social and clinical psychology: The health perspective* (1991), 158–178.

[5] Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *KDD*. ACM, 535–541.

[6] Taotao Cai, Jianxin Li, Ajmal S Mian, Ronghua li, Timos Sellis, and Jeffrey Xu Yu. 2020. Target-aware Holistic Influence Maximization in Spatial Social Networks. *IEEE Trans. Knowl. Data Eng.* (2020), 1–1.

[7] Chen Cheng, Haiqin Yang, Michael R. Lyu, and Irwin King. 2013. Where You Like to Go Next: Successive Point-of-Interest Recommendation. In *IJCAI*. Morgan Kaufmann, 2605–2611.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. NAACL, 4171–4186.

[9] Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-Again Neural Networks. In *ICML*. PMLR, 1602–1611.

[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. ACM, 173–182.

[11] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2014. Distilling the Knowledge in a Neural Network. In *NeurIPS*. MIT Press, 1–9.

[12] 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), 1–24.

[13] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. In *ICML*. PMLR, 2309–2318.

[14] 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 *EMNLP*. ACL Press, 4163–4174.

[15] Yu-Chin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware Factorization Machines for CTR Prediction. In *RecSys*. ACM, 43–50.

[16] Samuel C Karpen. 2018. The social psychology of biased self-assessment. *Am. J. Pharm. Educ.* 82, 5 (2018), 441–448.

[17] Buyu Li, Yu Liu, and Xiaogang Wang. 2019. Gradient Harmonized Single-Stage Detector. In *AAAI*. AAAI Press, 8577–8584.

[18] Jianxin Li, Taotao Cai, Ke Deng, Xinjue Wang, Timos Sellis, and Feng Xia. 2020. Community-diversified influence maximization in social networks. *Inf. Syst.* 92 (2020), 101522–101533.

[19] Ranzhen Li, Yanyan Shen, and Yanmin Zhu. 2018. Next Point-of-Interest Recommendation with Temporal and Multi-level Context Attention. In *ICDM*. IEEE, 1110–1115.

[20] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. 2019. Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. In *NeurIPS*. MIT Press, 5244–5254.

[21] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-Aware Sequential Location Recommendation. In *KDD*. ACM, 2009–2019.

[22] Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. FastBERT: a Self-distilling BERT with Adaptive Inference Time. In *ACL*. ACL Press, 6035–6044.

[23] Kevin P. Murphy. 2012. *Machine learning - a probabilistic perspective.* MIT Press.

[24] Apoorva Nitsure, Rajesh Bordawekar, and Jose Neves. 2020. Unlocking New York City Crime Insights using Relational Database Embeddings. arXiv:cs.DB/2005.09617

[25] 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 *KDD*. ACM, 2671–2679.

[26] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for Thin Deep Nets. In *ICLR*. 1–13.

[27] Satya Narayan Shukla and Benjamin M. Marlin. 2019. Interpolation-Prediction Networks for Irregularly Sampled Time Series. In *ICLR*. 1–14.

[28] 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 *CIKM*. ACM, 1441–1450.

[29] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NeurIPS*. MIT Press, 3104–3112.

[30] H Swapnarekha, Himansu Sekhar Behera, Janmenjoy Nayak, and Bighnaraj Naik. 2020. Role of intelligent computing in COVID-19 prognosis: A state-of-the-art review. *Chaos, Solitons & Fractals* 138 (2020), 109947–109961.

[31] Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019. Multilingual Neural Machine Translation with Knowledge Distillation. In *ICLR*. 1–14.

[32] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM*. ACM, 565–573.

[33] 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.

[34] 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 *NeurIPS*. MIT Press, 5998–6008.

[35] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item Recommendation with Sequential Hypergraphs. In *SIGIR*. ACM, 1101–1110.

[36] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising Implicit Feedback for Recommendation. In *WSDM*. ACM, 373–381.

[37] Timothy D Wilson and Elizabeth W Dunn. 2004. Self-knowledge: Its limits, value, and potential for improvement. *Annu. Rev. Psychol.* 55 (2004), 493–518.

[38] 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 *SIGIR*. ACM, 1905–1908.

[39] Li Yan, Hai-Tao Zhang, Jorge Goncalves, Yang Xiao, Maolin Wang, Yuqi Guo, Chuan Sun, Xiuchuan Tang, Liang Jing, Mingyang Zhang, et al. 2020. An interpretable mortality prediction model for COVID-19 patients. *Nat. Mach. Intell.* 2, 5 (2020), 283–288.

[40] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A Dynamic Recurrent Model for Next Basket Recommendation. In *SIGIR*. ACM, 729–732.

[41] Fajie Yuan, Xiangnan He, Haochuan Jiang, Guibing Guo, Jian Xiong, Zhezhao Xu, and Yilin Xiong. 2020. Future Data Helps Training: Modeling Future Contexts for Session-based Recommendation. In *WWW*. ACM, 303–313.

[42] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. 2018. Deep Mutual Learning. In *CVPR*. IEEE, 4320–4328.

[43] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep Interest Evolution Network for Click-Through Rate Prediction. In *AAAI*. AAAI Press, 5941–5948.

[44] 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 *AAAI*. AAAI Press, 11106–11115.

[45] 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 *CIKM*. ACM, 1893–1902.