On the Effectiveness of Self-supervised Pre-training for Modeling User Behavior Sequences

Yiping Liao
yiping@unity3d.com
Unity Technologies
Helsinki, Finland

ABSTRACT

Modeling the temporal dependency in user historical behavior is crucial to improve the conversion prediction in mobile game advertising. One common approach is to encode the time-dependent behavior sequence into meaningful representations which can enhance expressiveness of the conversion prediction model. In this work, we propose a self-supervised learning (SSL) scheme for pre-training such representations with a sequential network. An SSL pretext task is introduced to model the correlation between past and future events without labels. The pre-trained sequential network can then be transferred to perform the downstream task, i.e. conversion prediction, along with a dense network that models the feature interaction between the target ads and their context. We assess the proposed models on a real-world dataset collected from our online advertising system. From the experiments, we observe that the models with the proposed pre-training scheme (1) achieve lower test log-losses and higher AUC values, and (2) require fewer labels to achieve the similar prediction accuracy than those without in the various scenarios where the models have access to the limited or full labels. Accordingly, the proposed pre-training scheme enhances the downstream models in terms of generalization ability and label efficiency, facilitating the deployment of the sequential model at scale in the online advertising system.

KEYWORDS

Online advertising, recurrent neural networks, self-supervised learning, sequential representation learning

ACM Reference Format:
Yiping Liao. 2020. On the Effectiveness of Self-supervised Pre-training for Modeling User Behavior Sequences. In *Proceedings of ACM Conference (AdKDD'20).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The mobile gaming market has tremendous growth in recent years. A large number of users play varying types of games, including the hyper-casual, puzzle, and strategy games etc. With this perceived growing trend, game advertising has become an important monetization tool. To satisfy the diverse interests of gamers and enable the ad network to succeed, providing personalized recommendations is very curial. Specifically, by better identifying a user's preference toward different types of games, the mobile game ad network can improve its conversion prediction system by recommending the most relevant games to users, and optimize the ad revenue.

The history of a user interacting with the advertising system has a strong connection to her interests. The user behavior sequence contains rich information such as the frequency of a user viewing or clicking ads, the history of items that have been shown to the user, the most recent actions generated by the users and so on. Many studies utilize sequential models, such as the recurrent neural networks (RNNs) to capture temporal dependency of user behavior sequences [13, 19, 20, 23]. In those studies, the temporal information from user behavior sequences is usually encoded into a condense representation which enhances the predicting ability of a model.

However, training the industrial sequential model at scale is challenging. First of all, maintaining and updating the large volume of user's behavior sequences and deriving the corresponding context and outcomes create storage and computation burdens. Second, many of the user's historical sequences are of varying lengths and the labels are unbalanced, increasing the difficulties of the model training. The learned representation can be biased toward the label with overwhelmingly more abundant samples, e.g. non-conversion ones. Third, the representations learned from predicting a single task might not generalize well to other tasks [13]. In addition to the predicting target, the funnel stage of the user's journey to conversion contains essential information and should be encoded into the learned representation. For example, the probability of a user viewing or clicking the next item can be important extra information to enrich the representations.

To address the aforementioned issues, we resort to transfer learning and model pre-training. It has been shown in various applications that a well pre-trained model adapts to the new task faster and better than those trained from scratch [22]. However, the transferability of such a model hinges on that the model can learn the sufficiently expressive representations on a pre-text task where large amounts of labeled data are usually required. Inspired by the recently developed self-supervised learning (SSL) methodology, we propose a pre-training scheme that does not require labels (e.g. whether a user is converted) for enhancing the conversion prediction models. Specifically, we propose an SSL pre-text task in which the objective is to model the correlation between the past event sequences and the possible future observations. Based on the past sequence, the model learns to distinguish the correct future event from the wrong ones with a contrastive loss [14]. Instead of directly

AdKDD'20, August 23, 2020, San Diego, CA, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

predicting or reconstructing the future events, the model maximizes the mutual information between the encoded representations of the past and the future event sequences. It enables the model to learn shared high-level latent information from input sequences, such as the user's preference of the items and the patterns of interacting with ads, while discard the low-level information and noise.

The main contributions of this work are, first, we explore and investigate the benefits of pre-training the representations for event-based sequences with SSL which provides more diverse targets for a model to learn from. Second, we demonstrate that the proposed SSL pre-training scheme can effectively enhance the performance of the downstream task, e.g. conversion prediction. Empirically, we found that the SSL pre-trained models consistently outperform those not pre-trained given varying levels of availability of the labeled data. Alongside the numerous SSL works for applications in, e.g. natural language processing, computer vision, and audio domains, our work could be considered as one of the earliest exploratory research that studies the effectiveness of the SSL methodology for ad-related prediction tasks.

2 RELATED WORK

2.1 Recommender Systems for Online Advertising

In an advertising system, the recommendation is often made by considering the conversion rate of showing different items to users. Thus, accurately predicting conversion probability is important and the approaches to it are heavily researched in both industry and academia. Notably in the recent years, deep neural networks (DNNs) have been widely applied to predict click-through rate (CTR) in many online advertising systems [2, 7, 11, 16, 21, 25]. The shared architecture of those systems is to have the embedding layers that transform the sparse categorical features into the dense vectors and a deep network that follows to learn the interactions between features with minimal feature engineering. A more complex architecture, such as Cross Network [21] or Factorization Machine [7], was integrated with the deep network to further improve the model performance.

Besides the ad-related features used in the aforementioned works, considering the history of a user interacting with the ad system can enhance the understanding of user's preferences. For instance, Ouyang et al. [15] utilized auxiliary data including user history on both clicked and unclicked ads to account for user interests. In [23, 24], the dependency and evolution in user's sequential behaviors are modeled with RNNs to make the click prediction. Similar sequential models also benefit the item recommendation task. In [19, 20], RNNs are employed to predict user interests on the future items based on items that a user has interacted with. Ni et al. [13] learned the universal user representations across multiple tasks with a sequential model. Since trained for multi-tasking, the learned representations are shown to be more generic and effective for a wide range of tasks.

Motivated by the models introduced above, the proposed architecture, which consists of a dense network and a sequential network (introduced in Secs. 3.1 and 3.2, respectively), is designed to capitalize on different types of features available in our advertising system.

2.2 Self-supervised Learning

In SSL, the goal is to extract useful information from the data itself without any labels. It is achieved by designing the self-supervised tasks, also known as the pretext tasks in which the learning objectives are based on the proxy labels generated from the data. Guided by these objectives, the model learns to encode the underlying structure of the data into the representations which can be used in the downstream tasks.

SSL is widely applied in visual, audio, and natural language domains due to the availability of vast amounts of unlabeled data. Many self-supervised tasks were studied, e.g. predicting image rotations [6], the relative position of patches [3] and identifying uncorrelated element from a set of relevant ones [5]. Oord et al. [14] proposed a contrastive learning methodology dubbed as Contrastive Predictive Coding (CPC) which learns to predict the future signals in the latent space. The same methodology is shown to be effective in varying applications. Saeed et al. [18] proposed to learn generic representations from the sensory data in the multi-tasking way. The representations, which were transferred to the human activity recognition task, led to the improved recognition rate.

From those works where SSL are successfully applied, it is evidently a well-designed pre-text task can provide strong supervisory signals for learning effective representations. This inspires us to study and explore the SSL methodology tailored to the recommender systems for online advertising.

3 CONVERSION PREDICTION MODELS

The goal of the conversion prediction model in the mobile game advertising system is to optimize the predicting accuracy of the conversion probability. Based on the predicted probabilities, the system selects the ads that are the most valuable to present to wide variety of users for maximizing the gain of the advertising revenue. Our conversion prediction framework (see Figure 1) consists of two networks: (1) a dense network that takes different types (i.e. *numerical* and *categorical*) of features. (2) a sequential network that takes *history events* of users interacting with the ads. The outputs of the two networks are fused and fed into a fully-connected layer from which the conversion predictions are made. We describe the specifics (i.e. the example inputs and architecture choices) of these two networks respectively in Sections 3.1 and 3.2. The proposed self-supervised pre-training scheme, which is the focus of this work, is described in Section 3.3.

3.1 Dense Network

Many sophisticated network architectures, e.g. *Wide & Deep Network* [2], *Deep & Cross Network* [21], and *Deep Factorization Machine* [7], have been proposed to model complex interactions between features used in the recommender systems to optimize the CTR. We consider those as the justifiable candidates for the dense network, however, we do not use them directly and focus on assessing the proposed self-supervised learning scheme with a simpler dense network (see Figure 1, *Dense Network*) described next.

Given the sparse *categorical* features $x_i^{c_1}, x_i^{c_2}, ..., x_i^{c_K}$ and *numerical* features $x_i^{n_1}, x_i^{n_2}, ..., x_i^{n_L}$ (for i^{th} sample), $x_i^{c_k}$ is mapped to a dense embedding $\mathbf{z}_i^{c_k}$ with a transformation matrix W^{c_k}. The categorical embeddings $\{\mathbf{z}_i^{c_k} | k = 1, ..., K\}$ and the numerical features

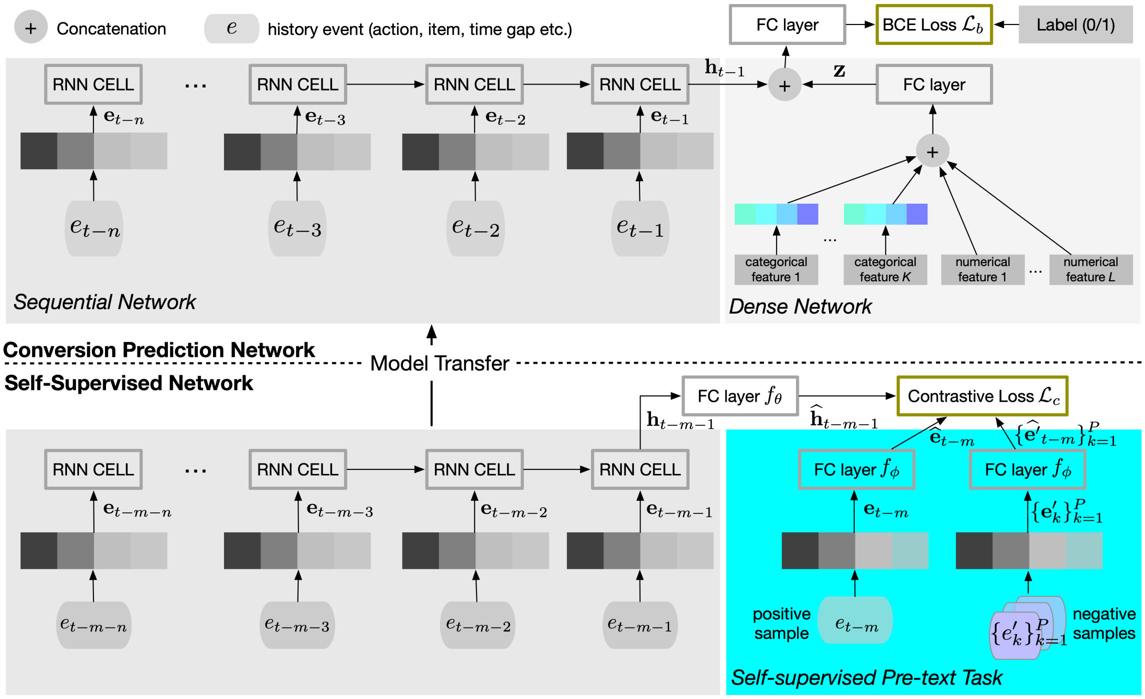

Figure 1: The overview of the proposed conversion prediction and the self-supervised networks. The former consists of the sequential and the dense networks. The latter is learned by the guidance of the proposed pre-text task and it serves as the pre-trained model for the sequential network.

$\{x_i^{n_l}|l = 1, ..., L\}$ are concatenated together to construct the context features $\mathbf{c}_i$. The dense network finnaly outputs $\mathbf{z}_i$ from a fully-connected layer which takes $\mathbf{c}_i$ as input. The example inputs to the dense network are the target *item*[1], and the item context (e.g. geo-location, device type, game type, etc.). We intentionally leave the choice of input features open since the interest of this work is not to exhaustively involve as many collectable features as possible.

3.2 Sequential Network

While the dense network capitalizes only on the non-sequential data, the sequential network models the evolution in user behavior and interest.

Our sequential network (see Figure 1, *Sequential Network*) adopts RNN to model *history events* from the users interacting with the advertising system over time. The history events defined in this work for the i^{th} sequence are $e_{i,j} = \{x_{i,j}^a, x_{i,j}^g, d_{i,j}\}_{j=t-n}^{t-1}$ in which an event $e_{i,j}$ at the j^{th} step encapsulates (1) the type of *actions* $x_{i,j}^a$ in response to the presented item (2) the *history items* $x_{i,j}^g$, and (3) the *time gap* $d_{i,j}$ between two successive events. t denotes the index of the latest sample in the sequence. n is the maximum sequence length. Note that for notational brevity we ignore the varying n and t across sequences (See Table 1 for the average length of the sequences in the benchmark dataset). Similar to that in the dense network, $x_{i,j}^a$ and $x_{i,j}^g$ are firstly mapped to the dense embeddings $\mathbf{z}_{i,j}^a$ and $\mathbf{z}_{i,j}^g$, respectively. To account for the time duration between events, we learn the *time masks* $\mathbf{m}_{i,j}$ [10],

$$\mathbf{m}_{i,j} = \sigma(\mathbf{c}_{i,j} W^c + \mathbf{b}^c), \tag{1}$$

$$\mathbf{c}_{i,j} = (\log d_{i,j}) \cdot w^d + b^d, \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function and W^c, b^c are the parameter matrix and the biases which transform the *time context vector* $\mathbf{c}_{i,j}$ into $\mathbf{m}_{i,j}$. w^d and b^d are the parameter vector and the bias term for projecting the scalar $\log d_{i,j}$ to a higher dimension. The time-dependent representation $\mathbf{e}_{i,j}$ input to each RNN cell is then

$$\mathbf{e}_{i,j} = (\mathbf{z}_{i,j}^a \oplus \mathbf{z}_{i,j}^g) \odot \mathbf{m}_{i,j}, \tag{3}$$

where $\oplus$ and $\odot$ denotes concatenation and element-wise multiplication, respectively. Each RNN step updates the hidden states $\mathbf{h}_{i,j}$.

Next, we combine the outputs $\mathbf{z}_i$ and $\mathbf{h}_{i,n}$ from the dense and the sequential networks, respectively, to define the loss $\mathcal{L}_b$ for the conversion prediction model with the standard binary cross entropy function

$$\mathcal{L}_b = -y_i \log \sigma(\mathbf{g}_i) - (1 - y_i) \log \left(1 - \sigma(\mathbf{g}_i)\right), \tag{4}$$

$$\mathbf{g}_i = (\mathbf{h}_{i,n} \oplus \mathbf{z}_i) W^z + \mathbf{b}^z, \tag{5}$$

where $\mathbf{w}^z, b^z$ are the parameter vector and the scalar bias term, respectively.

[1] We refer *item* to be the recommended item, e.g. game or product, presented in an ad.

3.3 Self-supervised Pre-training

3.3.1 Challenges and Motivations. In an online advertising system, it is crucial that the models can be updated as frequently as possible because the new data is rolling in and new ad campaigns are created in real-time. While a model consisting of both the dense and sequential networks can lead to better predictions than that with only the dense network, it can be more difficult to train and requires longer training time owing to the more complex architecture. Thereby, such a joint model could be less capable of keeping up with the most updated trends in user intentions or preferences.

We propose a self-supervised learning (SSL) scheme to pre-train the sequential network without the supervision from the labels in the downstream task. The advantages that SSL brings are two-fold. First, the sequential network can be pre-trained and initialized with the most updated history events of the users without waiting for the conversion windows. Second, the sequential network is then no longer trained from scratch, so it could converge better when training with the dense network in the downstream task.

Inspired by contrastive predictive coding (CPC) [14], we propose an SSL pre-text task that models the representations of future events based on past sequences. The task is performed by contrasting the encoded representations of the possible future events from those of the less likely events with a contrastive loss. Instead of explicitly predicting the next event based on the past, we are interested in training the model to distinguish what kind of events are more likely to happen in the future. The reasons for this are, first, directly predicting the next event is difficult because the candidate future items can be many. Second, the item history in the user event sequence could be loosely correlated with the user actions and affected by different strategies that an ad system takes.

In short, we pre-train the sequential network on the proposed SSL task prior to training it along with the dense network for the conversion prediction task. Learning on a different objective from that in the downstream task, the sequential network can learn extra information on the evolution in user's intentions and preferences.

3.3.2 Self-supervised Learning. Here we describe our SSL pre-text task (see Figure 1, *Self-supervised Pre-text Task*). The task is to distinguish the true future events from the negative events given the past events. The negative events are sampled drawn from other user behavior sequences. This pre-text task prompts the network to model the relative probability of the ad system showing certain items that respond to the action sequences of the users. Accordingly, the self-supervised network learns to capture the abstraction of user's preferences and interests by modeling the co-occurrence between the user behavior sequences and the possible future events.

Specifically, we sample user behavior sequence $\{e_{i,j}\}_{j=t-m-n}^{t-m-1}$ of length n, where $m > 0$, to construct the input for the pre-training task. t denotes the index of the latest sample in the sequence. The objective here is to classify the representation of the future event $e_{i,t-m}$ given history events $\{e_{i,j}\}$ among P negative events $\{e'_k\}_{k=1}^{P}$. The same RNN described in Section 3.2 processes $\{e_{i,j}\}$ and outputs the temporal representation $\{\mathbf{h}_{i,j}\}$. The representations of $e_{i,t-m}$ and $\{e'_k\}_{k=1}^{P}$ are $\mathbf{e}_{i,j}$ and $\{\mathbf{e}'_k\}$ obtained through Eqs. (1-3), respectively. The model then contrasts $\mathbf{e}_{i,t-m}$ with $\{\mathbf{e}'_k\}_{k=1}^{P}$ by the contrastive loss [14]

$$\mathcal{L}_c = -\log P(e_{i,t-m} \mid e_{i,t-m-1}, \{e'_k\}_{k=1}^{P}) \tag{6}$$

$$= -\log \frac{\exp(\widehat{\mathbf{h}}_{i,t-m-1} \cdot \widehat{\mathbf{e}}_{i,t-m})}{\exp(\widehat{\mathbf{h}}_{i,t-m-1} \cdot \widehat{\mathbf{e}}_{i,t-m}) + \sum_{k=1}^{P} \exp(\widehat{\mathbf{h}}_{i,t-m-1} \cdot \widehat{\mathbf{e}}'_k)}, \tag{7}$$

$$\widehat{\mathbf{h}}_{i,t-m-1} = f_\theta(\mathbf{h}_{i,t-m-1}), \tag{8}$$

$$\widehat{\mathbf{e}}_{i,t-m} = f_\phi(\mathbf{e}_{i,t-m}), \tag{9}$$

$$\widehat{\mathbf{e}}'_k = f_\phi(\mathbf{e}'_k). \tag{10}$$

$f_\theta(\cdot)$ and $f_\phi(\cdot)$ are two embedding layers (implemented by the fully-connected layers that respectively map $\mathbf{e}_{i,t-m}$ and $\mathbf{h}_{t-m-1}$ (as well as $\mathbf{e}'_k$) to the representations of the same length.

To summarize, the proposed SSL pre-trained conversion prediction model can be obtained by, first, pre-training the sequential network on the pre-text task with the contrastive loss $\mathcal{L}_c$ (see Eq. (7)) and second, jointly training the sequential and dense networks for conversion prediction with the loss $\mathcal{L}_b$ (see Eq. (4)). We assess the proposed model in the next section.

4 EXPERIMENTS

The experiments investigate the conversion prediction models with and without SSL pre-training in several facets: (1) the representation learned in the pre-training, (2) the convergence of the models, and (3) the prediction accuracy. The dataset, metrics, model training details, and experiment results are described in the following sections.

4.1 Experiment Settings

Table 1: Data statistics.

training data size	20,000,000
validation data size	4,000,000
test data size	4,000,000
average sequence length	23.3
number of unique target items	2974

4.1.1 Data Preparation. The statistics of the dataset in this experiment are shown in Table 1. For the dense network, despite that technically it can take both the categorical and numerical features as input (as shown in Figure 1), we use the target item as the only input. For the sequential network, we collect up to one month of the behavior history sequence from each user and truncate each sequence to the most recent 30 events to construct the inputs. We zero-pad the sequences shorter than 30 events.

4.1.2 Model Architecture. In the dense network, the fully connected layer is of 16 hidden units. In the sequential network, the sizes of the target action and the item embeddings ($\mathbf{z}^{a}_{i,j}$, $\mathbf{z}^{g}_{i,j}$ in Eq. 3) are set to be 12 and 32, respectively. We use Gated Recurrent Unit (GRU) of 64 hidden units as the RNN cells for the SSL pre-training and the conversion prediction tasks.

4.1.3 Training Details. We use Adam [8] as the optimizer with learning rate 10^{-3}. The batch size is set as 5,000 samples. We train the self-supervised and conversion prediction networks for maximally 100 epochs. The training of the conversion prediction network is early stopped once it has been ten epochs without perceived improvement in validation log-loss. We select the models with the best validation log-loss for testing.

4.1.4 Evaluation Methods.
Linear Evaluation. One common way to evaluate the representations learned with SSL is to apply the linear evaluation [1, 9, 14]. When training a linear evaluation model, the sequential network is firstly pre-trained on the proposed SSL task. All the parameters in the embedding layers and GRU are frozen and only those in the dense network are updated.
Varying Proportions of Labeled Samples. We assess the proposed SSL pre-trained models trained with different proportions of labeled samples to consider the scenarios in which the access to the labeled data is limited.

4.1.5 Evaluation Metrics. We consider (1) the log-loss calculated by Eq. (4), (2) the area under the ROC curve (AUC) on the test dataset, and (3) as in [24], the relative improvement in AUC (*RelaImpr*) of a measured model over the baseline. The model used for the linear evaluation is considered as the baseline.

$$RelaImpr = \left(\frac{AUC(\text{measured model}) - 0.5}{AUC(\text{baseline model}) - 0.5} - 1 \right) \times 100\% \qquad (11)$$

4.2 Experimental Results

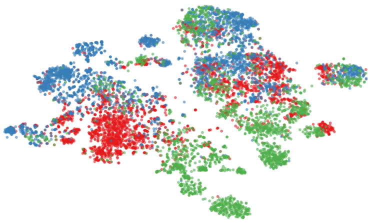

Figure 2: Visualizing sequence representations learned from the proposed SSL pre-text task using t-SNE [12] for a subset of four target items with the largest traffics. 2,000 samples are randomly selected for each target item. Target items are presented in different colors.

4.2.1 Sequence Representation. To assess if the model can extract meaningful information from the SSL task, we visualize the sequential representations from the four most common target items using t-SNE [12] as illustrated in Figure 2. One can observe that even learned from behavior sequences of varying lengths without explicit labels of the future items, the projected sequence representations contain clusters of the target items and the clusters within each target item. It indicates that the learned representations encode the information, i.e. the possible future item that might interest the user, to the downstream task.

4.2.2 Model Convergence. We examine how the proposed pre-training scheme affects the convergence of the models during training. Figure 3 demonstrates that given 10% of labels (or full labeled set), the pre-trained models show much better and faster convergence than those not pre-trained, e.g. around five epochs (or more) faster to reach the same log-loss with the lower validation log-loss during training.

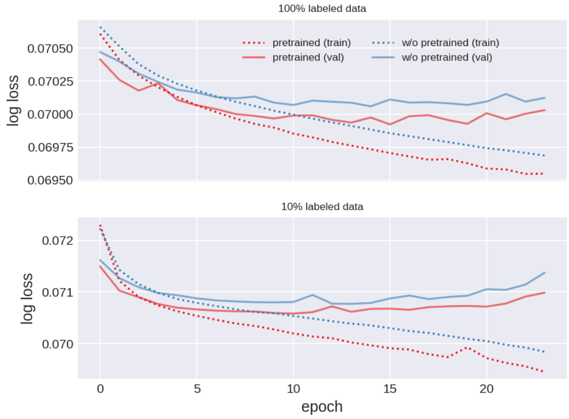

Figure 3: Training and validation curves of models (with and without pre-training) trained on 10% and 100% of labeled samples.

Table 2: Comparison of models (with and without pre-training) trained with all labeled samples in terms of the test log-loss and AUC. The reference baseline used for calculating Impr and RelaImpr is the linear evaluation model without pre-training (described in the first row).

Test case	Pre-trained	Log loss	Impr	AUC	RelaImpr
Linear evaluation		0.07188		0.7701	
	✓	0.07105	1.15%	0.7831	4.81%
Full labeled data		0.06934	3.53%	0.8048	12.85%
	✓	0.06924	3.67%	0.8062	13.37%

4.2.3 Results of Linear Evaluation. The results of linear evaluation is shown in Table 2. We include the results of models trained with the full labeled data, so we can compare the effectiveness of learned representation from the self-supervised and supervised task. The linear model without pre-training is trained by freezing the randomly initialized sequential network and is seen as the baseline in this experiment. From the relative improvement in log-loss and AUC in the downstream task, one can observe that the linear model with pre-training learns the meaningful representation without using any labels and brings significant improvement comparing to the baseline. The representation learned from the full labeled data shows that the user behavior sequence is an important feature that benefits the prediction accuracy a lot. Besides, learning the representation directly from labels are more specific toward the downstream task comparing to that from the self-supervised task. Moreover, the self-supervised pre-training task helps to extract non-overlapping information to the representations and provides a better initialization for the downstream task training, so that the model trained with full label data and with pre-training can outperform the previous test cases.

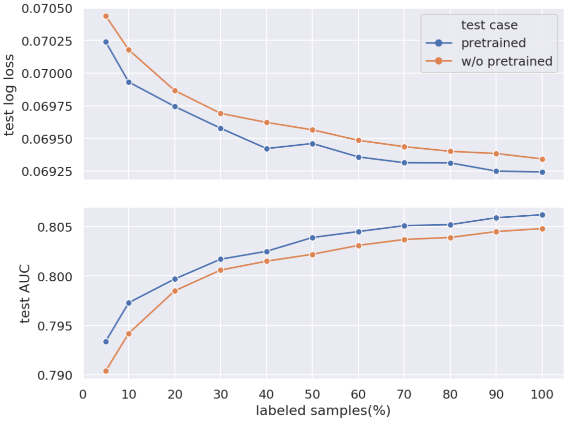

Figure 4: Comparing models (with and without pre-training) trained on different proportions of labeled samples in terms of log-loss and AUC.

4.2.4 Training with Different Proportions of Labeled Samples. Figure 4 illustrates the log-loss and AUC of the models trained with different proportions of labeled samples. One can the consistent improvements of log-loss and auc across those models. In the situation when labeled data is very sparse, i.e. 5% or 10%, the information gain from the pre-training task is large, but it quickly saturates when having enough labeled samples. It is also observed that to achieve the comparable log-loss and AUC, the models with pre-training require less labeled data comparing to the model without pre-training.

5 CONCLUSION AND FUTURE WORK

In this work, we first introduced a conversion prediction model consisting of a sequential network and a dense network to model varying types of inputs in our online mobile game advertising system. Next, we proposed a pre-training scheme based on self-supervised learning which models the temporal evolution in user intentions and preferences. Empirically, we showed that the models pre-trained with the proposed self-supervised task not only converged quicker but also were able to attain better test log-losses and AUC scores. The conversion prediction models were consistently improved by the proposed pre-training scheme across varying levels of availability of the labeled samples. For instance, the models were improved at best 0.39% when trained with 10% of labeled data.

While the introduced self-supervised approach has demonstrated its effectiveness, we consider the following studies as the future directions to explore the full potential of the SSL methodology for the online advertising systems. First, we are interested in studying more self-supervised pre-text tasks. This would allow one to discover more diverse learning targets from which the models could benefit the most in the relevant downstream tasks. Second, it could be interesting to study multi-task self-supervised learning [4, 17, 18] which jointly learns multiple pre-text tasks in the pre-training phase.

REFERENCES

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709* (2020).

[2] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems.* 7–10.

[3] Carl Doersch, Abhinav Gupta, and Alexei A Efros. 2015. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision.* 1422–1430.

[4] Carl Doersch and Andrew Zisserman. 2017. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision.* 2051–2060.

[5] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. 2017. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 3636–3645.

[6] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728* (2018).

[7] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[8] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[9] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. 2019. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.* 1920–1929.

[10] Yang Li, Nan Du, and Samy Bengio. 2017. Time-dependent representation for neural event sequence prediction. *arXiv preprint arXiv:1708.00065* (2017).

[11] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 1754–1763.

[12] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[13] Yabo Ni, Dan Ou, Shichen Liu, Xiang Li, Wenwu Ou, Anxiang Zeng, and Luo Si. 2018. Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 596–605.

[14] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[15] Wentao Ouyang, Xiuwu Zhang, Li Li, Heng Zou, Xin Xing, Zhaojie Liu, and Yanlong Du. 2019. Deep spatio-temporal neural networks for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2078–2086.

[16] Yanru Qu, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, Yong Yu, and Xiuqiang He. 2018. Product-based neural networks for user response prediction over multi-field categorical data. *ACM Transactions on Information Systems (TOIS)* 37, 1 (2018), 1–35.

[17] Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, Pawel Swietojanski, Joao Monteiro, Jan Trmal, and Yoshua Bengio. 2020. Multi-task self-supervised learning for Robust Speech Recognition. *arXiv preprint arXiv:2001.09239* (2020).

[18] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. 2019. Multi-task Self-Supervised Learning for Human Activity Detection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 2 (2019), 1–30.

[19] Elena Smirnova and Flavian Vasile. 2017. Contextual sequence modeling for recommendation with recurrent neural networks. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems.* 2–9.

[20] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems.* 17–22.

[21] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17.* 1–7.

[22] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3, 1 (2016), 9.

[23] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu. 2014. Sequential click prediction for sponsored search with recurrent neural networks. In *Twenty-Eighth AAAI Conference on Artificial Intelligence.*

[24] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5941–5948.

[25] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 1059–1068.