PRELIMINARY PREPRINT VERSION: DO NOT CITE
The AAAI Digital Library will contain the published
version some time after the conference.

Cross-Task Knowledge Distillation in Multi-Task Recommendation

Chenxiao Yang[1], Junwei Pan[2], Xiaofeng Gao[1], Tingyu Jiang[2], Dapeng Liu[2], Guihai Chen[1]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] Tencent Inc.
chr26195@sjtu.edu.com, jonaspan@tencent.com, gao-xf@cs.sjtu.edu.cn, travisjiang@tencent.com, rocliu@tencent.com, gchen@cs.sjtu.edu.cn

Abstract

Multi-task learning has been widely used in real-world recommenders to predict different types of user feedback. Most prior works focus on designing network architectures for bottom layers as a means to share the knowledge about input features representations. However, since they adopt task-specific binary labels as supervised signals for training, the knowledge about how to accurately rank items is not fully shared across tasks.

In this paper, we aim to enhance knowledge transfer for multi-task personalized recommendat optimization objectives. We propose a Cross-Task Knowledge Distillation (CrossDistil) framework in recommendation, which consists of three procedures. 1) Task Augmentation: We introduce auxiliary tasks with quadruplet loss functions to capture cross-task fine-grained ranking information, which could avoid task conflicts by preserving the cross-task consistent knowledge; 2) Knowledge Distillation: We design a knowledge distillation approach based on augmented tasks for sharing ranking knowledge, where tasks' predictions are aligned with a calibration process; 3) Model Training: Teacher and student models are trained in an end-to-end manner, with a novel error correction mechanism to speed up model training and improve knowledge quality. Comprehensive experiments on a public dataset and our production dataset are carried out to verify the effectiveness of CrossDistil as well as the necessity of its key components.

1 Introduction

Online recommender systems often need to model and predict various types of user feedback such as clicking and purchasing. Multi-Task Learning (MTL) (Caruana 1997) is widely adopted for predicting different types of user feedback using a unified model (Ma et al. 2018b; Lu, Dong, and Smyth 2018; Wang et al. 2018).

Common MTL models consist of a low-level *shared network* and several high-level *individual networks* for each task, as shown in Fig. 1(a). The shared network either learns task-invariant representations or enforces similarity on parameters of different tasks (Ruder 2017) as a way to transfer the knowledge about "how to represent the input features". Most prior works (Ma et al. 2018a; Tang et al. 2020a; Ma et al. 2019) put efforts on designing different shared network architectures with ad hoc parameter-sharing mechanisms including branching, gating, etc., to enhance the effectiveness of knowledge transfer. In these models, each task is trained under the supervision of its own binary ground-truth label (1 or 0), attempting to rank positive items above negative ones. However, using such binary labels as training signals, the task may fail to accurately capture user's preference for items with the same label. Learning the auxiliary knowledge about these items' relations may benefit the overall recommendation performance.

To address this limitation, we observe that the predictions of other tasks may contain useful information about how to rank same-labeled items. For example, given two tasks predicting 'Buy' and 'Like', and two items labeled as 'Buy:0, Like:1' and 'Buy:0, Like:0', the task 'Buy' may not accurately distinguish their relative ranking since both of their labels are 0. In contrast, another task 'Like' will identity the former item as positive with larger probability (e.g. 0.7), the latter with smaller probability (e.g. 0.1). Based on the fact that a user is more likely to purchase the item she likes [1], we could somehow take advantage of these predictions as a means to transfer ranking knowledge.

Knowledge Distillation (KD) (Hinton, Vinyals, and Dean 2015) is a teacher-student learning framework where the student is trained using the predictions of the teacher. As revealed by theoretical analysis in several studies (Tang et al. 2020b; Phuong and Lampert 2019), the predictions of the teacher, also known as *soft labels*, are more informative training signals than binary *hard labels*, since they could reflect 'whether the sample is true positive (negative)'. On the perspective of backward gradient, KD can adaptively rescale student model's training dynamics based on the values of soft labels. Specially, in the above example, we could incorporate predictions 0.7 and 0.1 in the training signals for task 'Buy'. Consequently, the gradients w.r.t the sample labeled 'Buy:0 & Like:0' in the example will be larger, indicating it is a more confident negative sample. Through this process, the task 'Buy' could hopefully give accurate rankings of same-labeled items.

Motivated by the above observations and theoretical justifications, we proceed to design a new knowledge transfer

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]The same applies to other types of user feedback, e.g., click, collect, forward.

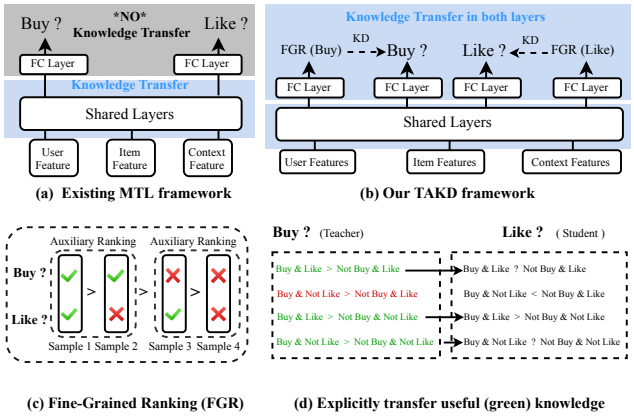


Figure 1: Illustration of the motivation of CrossDistil.

paradigm on the optimization level of MTL models by leveraging KD. It is non-trivial due to three critical and fundamental challenges:

- **How to address the task conflict problem during distillation?** Not all knowledge from other tasks is useful (Yu et al. 2020). Specially, in online recommendation, the target task may believe that a user prefers item$_A$ since she bought item$_A$ instead of item$_B$, while another task may reversely presume she prefers item$_B$ since she puts it in the collection rather than item$_A$. Such conflicting ranking knowledge may be harmful for the target task and could empirically cause significant performance drop.
- **How to align the magnitude of predictions for different tasks?** Distinct from vanilla KD where teacher and student models have the same prediction target, different tasks may have different magnitude of positive ratio. Directly using another task's predictions as training signals without alignment could mislead the target task to yield biased predictions (Zhou et al. 2021).
- **How to enhance training when teacher and student are synchronously optimized?** The vanilla KD adopts asynchronous training where the teacher model is well-trained beforehand. However, MTL inherently requires synchronous training, where each task is jointly learned from scratch. This indicates the teacher may be poorly-trained and provide inaccurate or even erroneous training signals, causing slow convergence and local optima (Wen, Lai, and Qian 2019; Xu et al. 2020).

In this paper, we propose a novel framework named as Cross-Task Knowledge Distillation (CrossDistil). Different from prior MTL models where knowledge transfer is achieved by sharing representations in bottom layers, CrossDistil also facilitates transferring ranking knowledge on the top layers, as shown in Fig. 1(b). To solve the aforementioned challenges: **First**, we introduce augmented tasks to learn the knowledge of the ranking orders of four types of samples as shown in Fig. 1(c). New tasks are trained based on a quadruplet loss function, and could fundamentally avoid conflicts by only preserving the useful knowledge and discarding the harmful one, as shown in Fig. 1(d). **Second**, we consider a calibration process that is seamlessly integrated in the KD procedure to align predictions of different tasks, which is accompanied with a bi-level training algorithm to optimize parameters for prediction and calibration respectively. **Third**, teachers and students are trained in an end-to-end manner with a novel error correction mechanism to speed up model training and further enhance knowledge quality. We conduct comprehensive experiments on a large-scale public dataset and a real-world production dataset that is collected from our platform. The results demonstrate that CrossDistil achieves state-of-the-art performance. The ablation studies also thoroughly dissect the effectiveness of its modules.

2 Preliminaries and Related Works

Knowledge distillation (Hinton, Vinyals, and Dean 2015) is a teacher-student learning framework where the student is trained according to the outputs of the teacher. For binary classification, the hint loss function for distillation is formulated as

$$\mathcal{L}^{KD} = CE(\sigma(r_T/\tau), \sigma(r_S/\tau)), \tag{1}$$

where $CE(y, \hat{y}) = y \log(\hat{y}) + (1-y)\log(1-\hat{y})$ is binary cross-entropy, r_T and r_S denote logits of the teacher and the student model, respectively, and τ is temperature.

KD has been developed for various applications apart from model compression, e.g., intelligent label smoothing (Yuan et al. 2020), self-distillation (Zhang and Sabuncu 2020). Still, most works in recommender systems adopt KD in a traditional way for model reduction, where teacher and student are differently sized models targeting *the same task* (Tang and Wang 2018; Xu et al. 2020; Zhu et al. 2020). Distinct from theirs or other works using KD in machine learning community, this paper serves as the first attempt to leverage KD to transfer knowledge *across different ranking tasks*. Achieving this is non-trivial due to the aforementioned three major challenges, and calls for deep and fundamental understanding of how KD works and its relation with ranking tasks in recommendation.

Multi-task Learning (Zhang and Yang 2021) is a machine learning framework that learns a task-invariant representation of an input data in a bottom network, while each individual task is solved in one's respective task-specific network. MTL has received increasing interests in recommender systems (Ma et al. 2018b; Lu, Dong, and Smyth 2018; Wang et al. 2018; Pan et al. 2019) due to its ability to share knowledge among different tasks. A series of works seek to improve on it by designing different types of shared layer architectures. These works either introduce constraints on task-specific parameters (Duong et al. 2015; Misra et al. 2016; Yang and Hospedales 2016) or separate shared and task-specific parameters (Ma et al. 2018a; Tang et al. 2020a; Ma et al. 2019)as a means to share knowledge about how to represent the input feature. Different from theirs, we resort to knowledge distillation to transfer ranking knowledge across tasks on task-specific networks. Notably, our model is a general framework and could be leveraged as extension for most off-the-shelf MTL models.

3 Proposed Model

3.1 Task Augmentation for Ranking

We focus on multi-task learning for predicting different user feedback (e.g. click, like, purchase, look-through). To simplify illustration, we consider two tasks denoted as task A and task B in this paper (one for student and another for teacher). First, training samples are split into multiple subsets according to permutations of multiple tasks' labels. As shown in Fig. 2, they are defined as:

$$\begin{aligned} \mathcal{D}^{+-} &= \{\mathbf{x}_i \in \mathcal{D} | y_i^A = 1, y_i^B = 0\}, \\ \mathcal{D}^{-+} &= \{\mathbf{x}_i \in \mathcal{D} | y_i^A = 0, y_i^B = 1\}, \\ \mathcal{D}^{-\cdot} &= \mathcal{D}^{--} \cup \mathcal{D}^{-+},\ \mathcal{D}^{+\cdot} = \mathcal{D}^{+-} \cup \mathcal{D}^{++}, \\ \mathcal{D}^{\cdot-} &= \mathcal{D}^{--} \cup \mathcal{D}^{+-},\ \mathcal{D}^{\cdot+} = \mathcal{D}^{-+} \cup \mathcal{D}^{++}, \end{aligned} \tag{2}$$

where $\mathbf{x}$ is an input feature vector, y^A and y^B denote hard labels for task A and task B respectively. The goal of the traditional task is to rank positive samples before negative ones. Formally, such bipartite order is represented as $\mathbf{x}_{+\cdot} \succ \mathbf{x}_{-\cdot}$ for task A and $\mathbf{x}_{\cdot+} \succ \mathbf{x}_{\cdot-}$ for task B, where $\mathbf{x}_{+\cdot} \in \mathcal{D}^{+\cdot}$ and so forth. Note that bipartite orders may be contradictory across different tasks, e.g., $\mathbf{x}_{+-} \succ \mathbf{x}_{-+}$ for task A while $\mathbf{x}_{+-} \prec \mathbf{x}_{-+}$ for task B. Such conflicts would provide inconsistent signals to backward gradients of shared parameters, leading to a negative affect on the overall prediction performance. Empirically, directly conducting KD by treating one task as the teacher and another task as the student fails to work due to these conflicts.

To prepare for subsequent KD, we introduce auxiliary ranking-based tasks that could naturally preserve useful cross-task knowledge and avoid task conflicts. Given a sample quadruplets $(\mathbf{x}_{++}, \mathbf{x}_{+-}, \mathbf{x}_{-+}, \mathbf{x}_{--})$, we consider a multipartite order $\mathbf{x}_{++} \succ \mathbf{x}_{+-} \succ \mathbf{x}_{-+} \succ \mathbf{x}_{--}$ for task A. We refer such order as *fine-grained ranking* since it reveals additional informative orders $\mathbf{x}_{++} \succ \mathbf{x}_{+-}$ and $\mathbf{x}_{-+} \succ \mathbf{x}_{--}$ and has no contradiction with the original bipartite order $\mathbf{x}_{+\cdot} \succ \mathbf{x}_{-\cdot}$. Based on this, we introduce a new ranking-based task called *augmented task* $A+$ for enhancing knowledge transfer by additionally maximizing the following objective:

$$\begin{aligned} &\ln p(\Theta | \succ) \\ &= \ln p(\mathbf{x}_{++} \succ \mathbf{x}_{+-} | \Theta) \cdot p(\mathbf{x}_{-+} \succ \mathbf{x}_{--} | \Theta) \cdot p(\Theta) \\ &= \sum_{\substack{(\mathbf{x}_{++}, \mathbf{x}_{+-}, \\ \mathbf{x}_{-+}, \mathbf{x}_{--})}} \ln \sigma(\hat{r}_{++ \succ +-}) + \ln \sigma(\hat{r}_{-+ \succ --}) - Reg(\Theta), \end{aligned} \tag{3}$$

where r is the logit value before activation in the last layer, $\hat{r}_{++ \succ +-} = \hat{r}_{++} - \hat{r}_{+-}$, and $\sigma(x) = 1/(1 + exp(-x))$ is the sigmoid function. The loss function for augmented task $A+$ is

$$\begin{aligned} \mathcal{L}^{A+} = &\sum_{\substack{(\mathbf{x}_{++}, \mathbf{x}_{+-}, \\ \mathbf{x}_{-+}, \mathbf{x}_{--})}} -\beta_1^A \ln \sigma(\hat{r}_{++ \succ +-}) - \beta_2^A \ln \sigma(\hat{r}_{-+ \succ --}) \\ &+ \sum_{(\mathbf{x}_{+\cdot}, \mathbf{x}_{-\cdot})} - \ln \sigma(\hat{r}_{+\cdot \succ -\cdot}). \end{aligned} \tag{4}$$

The loss function consists of three terms that correspond to three pair-wise relations. Coefficients β_1, β_2 balance the importance of each pair-wise relation. The loss function for augmented task $B+$ could be defined in a similar spirit.

The computational graph for augmented tasks are highlighted in blue and red in Fig. 2. These augmented ranking-based tasks are stacked and jointly trained with original regression-based tasks in MTL framework. Recall that the original regression-based loss function is formulated as:

$$\begin{aligned} \mathcal{L}^A = &CE(y^A, \hat{y}^A), \qquad \mathcal{L}^B = CE(y^B, \hat{y}^B), \\ CE(y, \hat{y}) = &\sum_{\mathbf{x}_i \in \mathcal{D}} -y_i \ln \hat{y}_i - (1 - y_i) \ln(1 - \hat{y}_i), \end{aligned} \tag{5}$$

where $\hat{y} = \sigma(r)$ is the predicted probability.

The introduced auxiliary tasks could avoid task conflicts, and thus are prerequisites for knowledge transfer through KD. Besides, task augmentation itself is beneficial (Hsieh and Tseng 2021), since introducing more related tasks for training could enhance the generalizability of main tasks (Standley et al. 2020; Liu, Davison, and Johns 2019). Empirical results also show the auxiliary ranking tasks could help to improve recommendation performance, presumably because they could provide hints about what shall be learned and transferred in shared layers.

3.2 Calibrated Knowledge Distillation

To address the limitation of mainstream MTL frameworks, we seek to design a cross-task knowledge distillation approach that can transfer fine-grained ranking knowledge on optimization objective level. Since the prediction results of another task may contain the information about unseen rankings between samples of the same label, a straightforward approach is to use soft labels of another task to teach the current task by the vanilla hint loss (i.e. distillation loss) as in Eqn. (1). Unfortunately, such naive approach may be problematic and even imposes negative effects in practice. This is because the labels of different tasks may have contradictory ranking information that would harm the learning of other tasks as mentioned in last subsection. The treatment is to only transfer the unconflicted ranking knowledge which is captured by the augmented tasks. Specifically, we treat augmented ranking-based tasks as teachers, original regression-based tasks as students, and adopt the following distillation loss functions:

$$\begin{aligned} \mathcal{L}^{A-KD} &= CE(\sigma(\hat{r}^{A+}/\tau), \sigma(\hat{r}^A/\tau)), \\ \mathcal{L}^{B-KD} &= CE(\sigma(\hat{r}^{B+}/\tau), \sigma(\hat{r}^B/\tau)). \end{aligned} \tag{6}$$

Note that soft labels $\hat{y}^{A+} = \sigma(\hat{r}^{A+}/\tau)$ and $\hat{y}^{B+} = \sigma(\hat{r}^{B+}/\tau)$ are invariant when training the student model as shown in Fig. 2, such that the student will not mislead the teacher. The loss functions for students are formulated as

$$\begin{aligned} \mathcal{L}^{A-Stu} &= (1 - \alpha^A)\mathcal{L}^A + \alpha^A \mathcal{L}^{A-KD}, \\ \mathcal{L}^{B-Stu} &= (1 - \alpha^B)\mathcal{L}^B + \alpha^B \mathcal{L}^{B-KD}, \end{aligned} \tag{7}$$

where $\alpha^A \in [0, 1]$ is the hyper-parameter to balance two losses. The soft labels output by augmented ranking-based

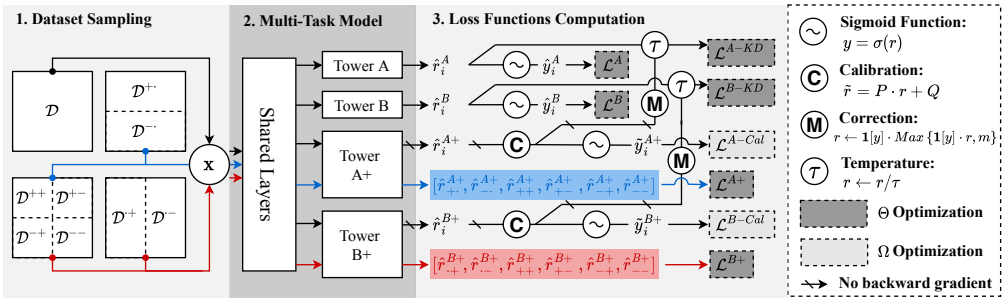


Figure 2: Illustration of computational graph for CrossDistil.

tasks are more informative than hard labels. As an example, for samples $\mathbf{x}_{++}, \mathbf{x}_{+-}, \mathbf{x}_{-+}, \mathbf{x}_{--}$, the teacher model for augmented task $A+$ may give predictions 0.9, 0.8, 0.2, 0.1 which intrinsically contains auxiliary ranking orders $\mathbf{x}_{++} \succ \mathbf{x}_{+-}$ and $\mathbf{x}_{-+} \succ \mathbf{x}_{--}$ that are not revealed in hard labels. Such knowledge is then explicitly transferred through the distillation loss and can meanwhile regularize task-specific layers from over-fitting the hard labels.

However, an issue of the aforementioned approach is that augmented tasks are optimized with pair-wise loss functions and thus are not predicting a probability, i.e., the prediction $\sigma(\hat{r}^{A+})$ does not agree with the actual probability that the input sample is a positive one. Directly using the soft labels of teachers may mislead students and cause performance deterioration. To solve this problem, we propose to calibrate the predictions so as to provide numerically sound and unbiased soft labels. Platt Scaling (Niculescu-Mizil and Caruana 2005; Platt et al. 1999) is a classic probability calibration method. We adopt it for calibration in this work. Still, one can replace it with any other more complex methods in practice. Formally, to get calibrated probabilities, we transform the logit values of teacher models through the following equation:

$$\tilde{r}^{A+} = P^A \cdot \hat{r}^{A+} + Q^A, \quad \tilde{y}^{A+} = \frac{1}{1 + \exp \tilde{r}^{A+}} \tag{8}$$

where $\tilde{r}$ and $\tilde{y}$ are the logit value and probability after calibration, respectively. The same process is also used for task $B+$. P, Q are learnable parameters specific to each task. They are trained by optimizing the calibration loss

$$\mathcal{L}^{Cal} = \mathcal{L}^{A-Cal} + \mathcal{L}^{B-Cal} = CE(y^A, \tilde{y}^{A+}) + CE(y^B, \tilde{y}^{B+}). \tag{9}$$

We fix MTL model parameters when optimizing $\mathcal{L}^{Cal}$ as shown in Fig. 2. Since the calibrated outputs of the teacher model are linear projections of the original outputs, the ranking result is unaffected so that the latent fine-grained ranking knowledge in soft labels is preserved during the calibration process. Distillation losses in Eqn. (6) are then revised by replacing $\hat{r}^{A+}, \hat{r}^{B+}$ with $\tilde{r}^{A+}, \tilde{r}^{B+}$.

3.3 Model Training

For traditional KD, a two-stage training process is a common setting where the teacher model is trained in advance and its parameters are fixed when training the student model (Hinton, Vinyals, and Dean 2015). However, such asynchronous training procedure is not favorable for industrial applications such as online advertising. Instead, because of simplicity and easy maintenance, synchronous training procedure where teacher and student models are trained in an end-to-end manner is more desirable as done in (Xu et al. 2020; Anil et al. 2018; Zhou et al. 2018). In our framework, there are two sets of parameters for optimization, namely, parameters in MTL backbone for prediction (denoted as Θ) and parameters for calibration including P^A, P^B, Q^A and Q^B (denoted as Ω). To jointly optimize prediction parameters and calibration parameters, we propose a bi-level training procedure where Θ and Ω are optimized in turn for each iteration as shown in the training algorithm. For sampling, it is impractical to enumerate every combination of samples as in Eqn. (4). Instead, We adopt bootstrap sampling strategy as used in (Rendle et al. 2012; Shan, Lin, and Sun 2018) as unbiased approximation.

3.4 Error Correction Mechanism

In KD-based methods, the student model is trained according to predictions of the teacher model, without considering if they are accurate or not. However, inaccurate predictions of the teacher model that is contradictory with the hard label could harm the student model's performance in two aspects. First, at early stage of training when the teacher model is not well-trained, frequent errors in soft labels may distract the training process of the student model, causing slow convergence (Xu et al. 2020). Second, even at later stage of training when the teacher model is relatively well-trained, it is still likely that the teacher model would occasionally provide mistaken predictions that may cause performance deterioration (Wen, Lai, and Qian 2019). A previous work (Xu et al. 2020) adopts a warm-up scheme by removing distillation loss in the earliest k steps of training. However, it is not clear how to choose an appropriate hyper-parameter k, and it cannot prevent errors after k steps.

In this work, we propose to adjust predictions of the teacher model $\tilde{y}$ to align with the hard label y. Specifically, we clamp logit values for the teacher model as follows:

$$r^{Teacher}(\mathbf{x}) \leftarrow \mathbb{1}[y] \cdot Max\left\{\mathbb{1}[y] \cdot r^{Teacher}(\mathbf{x}), m\right\} \tag{10}$$

where $r^{Teacher}$ could be $\tilde{r}^{A+}$ or $\tilde{r}^{B+}$, $\mathbb{1}[y]$ is an indicator function that returns 1 is $y = 1$ else returns -1, and m is the error correction margin, a hyper-parameter. The proposed error correction mechanism has the following properties. 1) For correct predictions of the teacher model (that predicts

Algorithm 1: Training Algorithm for CrossDistil

Input: Training dataset $\mathcal{D}$, learning rate γ_1 and γ_2, initial parameters Θ and Ω.

Construct set $\mathcal{D}^{++}, \mathcal{D}^{+-}, \mathcal{D}^{-+}, \mathcal{D}^{--}, \mathcal{D}^{+\cdot}, \mathcal{D}^{-\cdot}, \mathcal{D}^{\cdot+}, \mathcal{D}^{\cdot-}$;
while *Not converged* **do**
  Sample $\mathbf{x}$ uniformly at random from $\mathcal{D}$;
  Sample $\mathbf{x}_{++}, \mathbf{x}_{+-}, \mathbf{x}_{-+}, \mathbf{x}_{--}$ uniformly at random from $\mathcal{D}^{++}, \mathcal{D}^{+-}, \mathcal{D}^{-+}, \mathcal{D}^{--}$ respectively;
  Sample $\mathbf{x}_{+\cdot}, \mathbf{x}_{-\cdot}, \mathbf{x}_{\cdot+}, \mathbf{x}_{\cdot-}$ uniformly at random from $\mathcal{D}^{+\cdot}, \mathcal{D}^{-\cdot}, \mathcal{D}^{\cdot+}, \mathcal{D}^{\cdot-}$ respectively;
  Model parameter Θ optimization:
  Calculate $\mathcal{L}^{A+}(\mathbf{x}_{+\cdot}, \mathbf{x}_{-\cdot}, \mathbf{x}_{++}, \mathbf{x}_{+-}, \mathbf{x}_{-+}, \mathbf{x}_{--}; \Theta)$;
  Calculate $\mathcal{L}^{B+}(\mathbf{x}_{\cdot+}, \mathbf{x}_{\cdot-}, \mathbf{x}_{++}, \mathbf{x}_{+-}, \mathbf{x}_{-+}, \mathbf{x}_{--}; \Theta)$;
  Calculate $\mathcal{L}^{A-Stu}(\mathbf{x}; \Theta), \mathcal{L}^{B-Stu}(\mathbf{x}; \Theta)$;
  $\mathcal{L}^{Model} \leftarrow wightedSum(\mathcal{L}^{A+}, \mathcal{L}^{B+}, \mathcal{L}^{A-Stu}, \mathcal{L}^{B-Stu})$;
  $\Theta \leftarrow \Theta - \gamma_1 \nabla_\Theta \mathcal{L}^{Model}$;
  Calibration parameter Ω optimization:
  Calculate $\mathcal{L}^{Cal}(\mathbf{x}; \Omega)$;
  $\Omega \leftarrow \Omega - \gamma_2 \nabla_\Omega \mathcal{L}^{Cal}$;
end

the true label with at least probability $\sigma(m)$), this operation does not modify the result. Only incorrect predictions below the threshold are revised. 2) Adjustment operation is only carried out for calculating distillation loss with no backward gradient for teacher models as shown in Fig. 2, which indicates that it does not affect the training process of teachers. The proposed error correction mechanism is easy to implement and has the merits of accelerating convergence and enhancing knowledge quality to improve student model's performance.

4 Experiments

We conduct experiments on real-world datasets to answer the following research questions: **RQ1**: How do CrossDistil performs compared with the state-of-the-art multi-task learning frameworks; **RQ2**: Are the proposed modules in CrossDistil effective for improving the performance; **RQ3**: Does error correction mechanism help to accelerate convergence and enhance knowledge quality; **RQ4**: Does the student model really benefit from auxiliary ranking knowledge; **RQ5**: How do the hyper-parameters influence the performance?

4.1 Datasets

We conduct experiments on a publicly accessible dataset TikTok [2] and our WechatMoments dataset. Tiktok dataset is collected from a short-video app with two types of user feedback, i.e., 'Finish watching' and 'Like'. WechatMoments dataset is collected through sampling user logs during 5 consecutive days with two types of user feedback, i.e., 'Not interested' and 'Click'. For Tiktok, we randomly choose 80% samples as training set, 10% as validation set and the rest as test set. For WechatMoments, we split the data according to days and use the data of the first four days for training and the last day for validation and test. The statistics of datasets are given in Table 1.

[2]https://www.biendata.xyz/competition/icmechallenge2019/data/

Table 1: Statistics of two datasets.

Datasets	#Samples	#Fields	#Features	Density(A)	Density(B)
WechatMoments	9,381,820	10	447,002	1.510%	9.975%
TikTok	19,622,340	9	4,691,483	37.994%	1.101%

4.2 Evaluation Metrics

We use two widely adopted metrics, i.e., AUC and Multi-AUC, for evaluation. AUC indicates the bipartite ranking (i.e., $\mathbf{x}_+ \succ \mathbf{x}_-$) performance of the model.

$$\text{AUC} = \frac{1}{N^+ N^-} \sum_{\mathbf{x}_i \in D^+} \sum_{\mathbf{x}_j \in D^-} (\mathcal{I}(p(\mathbf{x}_i) > p(\mathbf{x}_j))) \quad (11)$$

where $p(\mathbf{x})$ is the predicted probability of $\mathbf{x}$ being a positive sample and $\mathcal{I}(\cdot)$ is the indicator function.

Multi-Class Area Under ROC Curve (Multi-AUC) The vanilla formulation of AUC only measures the performance of bipartite ranking where a data point is labeled either as a positive sample or a negative one. However, we are also interested in multipartite ranking performance since samples have multiple classes with an order $\mathbf{x}_{++} \succ \mathbf{x}_{+-} \succ \mathbf{x}_{-+} \succ \mathbf{x}_{--}$ (for task A). Therefore, following (Shan, Lin, and Sun 2018; Shan et al. 2017), we adopt Multi-AUC to evaluate multipartite ranking performance on test set. Note that we use the weighted version which considers the class imbalance problem (Hand and Till 2001) and is defined as:

$$\text{Multi-AUC} = \frac{2}{c(c-1)} \sum_{j=1}^{c} \sum_{k>j}^{c} p(j \cup k) \cdot AUC(k, j), \quad (12)$$

where c is the number of classes, $p()$ is the prevalence-weighting function as described in (Ferri, Hernández-Orallo, and Modroiu 2009), $AUC(k, j)$ is the AUC score with class k as the positive class and j as the negative class.

4.3 Baseline Methods

We choose the following MTL models with different shared network architectures for comparison: Shared-Bottom (Caruana 1997), Cross-Stitch (Misra et al. 2016), MMoE (Ma et al. 2018a), PLE (Tang et al. 2020a). We use two variants of our method: TAUG incorporates augmented tasks on top of MTL models, and CrossDistil extends TAUG by conducting calibrated knowledge distillation. Despite that Both TAUG and CrossDistil could be implemented on most state-of-the-art MTL models, *we choose the best competitor (i.e. PLE) as the backbone* in this work.

4.4 RQ1: Performance Comparison

Table 2 and 3 show the experiment results of our methods versus other competitors on WechatMoments and TikTok datasets respectively. The bold value marks the best one

Table 2: Experiment results of CrossDistil and competitors on WechatMoments dataset.

Methods	TaskA-Student		TaskB-Student		TaskA-Teacher		TaskB-Teacher	
	AUC	Multi-AUC	AUC	Multi-AUC	AUC	Multi-AUC	AUC	Multi-AUC
Single-Model	0.7528	0.6270	0.7597	0.6024	0.7535	0.6708	0.7604	0.6705
Shared-Bottom	$0.7540_{(+0.0012)}$	$0.6378_{(+0.0108)}$	$0.7587_{(-0.0010)}$	$0.6145_{(+0.0121)}$	-	-	-	-
Cross-Stitch	$0.7582_{(+0.0054)}$	$0.6360_{(+0.0090)}$	$0.7600_{(+0.0003)}$	$0.6195_{(+0.0171)}$	-	-	-	-
MMoE	$0.7619_{(+0.0091)}$	$\underline{0.6431}_{(+0.0161)}$	$0.7605_{(+0.0008)}$	$0.6226_{(+0.0202)}$	-	-	-	-
PLE	$\underline{0.7625}_{(+0.0097)}$	$0.6394_{(+0.0124)}$	$\underline{0.7607}_{(+0.0010)}$	$\underline{0.6240}_{(+0.0216)}$	-	-	-	-
TAUG	$0.7632_{(+0.0104)}$	$0.6432_{(+0.0162)}$	$0.7612_{(+0.0015)}$	$0.6394_{(+0.0370)}$	$\mathbf{0.7625}_{(+0.0090)}$	$0.6853_{(+0.0145)}$	$0.7608_{(+0.0004)}$	$0.6768_{(+0.0063)}$
CrossDistil	$\mathbf{0.7644}_{(+0.0116)}$	$\mathbf{0.6879}_{(+0.0609)}$	$\mathbf{0.7618}_{(+0.0021)}$	$\mathbf{0.6861}_{(+0.0837)}$	$0.7618_{(+0.0083)}$	$\mathbf{0.6910}_{(+0.0202)}$	$\mathbf{0.7609}_{(+0.0005)}$	$\mathbf{0.6850}_{(+0.0145)}$

Table 3: Experiment results of CrossDistil and competitors on TikTok dataset.

Methods	TaskA-Student		TaskB-Student		TaskA-Teacher		TaskB-Teacher	
	AUC	Multi-AUC	AUC	Multi-AUC	AUC	Multi-AUC	AUC	Multi-AUC
Single-Model	0.7456	0.6335	0.9491	0.7966	0.7453	0.7140	0.9481	0.8297
Shared-Bottom	$0.7375_{(-0.0081)}$	$0.6344_{(+0.0009)}$	$0.9489_{(-0.0002)}$	$\underline{0.8101}_{(+0.0135)}$	-	-	-	-
Cross-Stitch	$0.7468_{(+0.0012)}$	$0.6445_{(+0.0110)}$	$0.9488_{(-0.0003)}$	$0.7985_{(+0.0019)}$	-	-	-	-
MMoE	$0.7479_{(+0.0023)}$	$\underline{0.6474}_{(+0.0139)}$	$0.9490_{(-0.0001)}$	$0.7980_{(+0.0014)}$	-	-	-	-
PLE	$\underline{0.7485}_{(+0.0029)}$	$0.6464_{(+0.0129)}$	$\underline{0.9495}_{(+0.0004)}$	$0.7983_{(+0.0017)}$	-	-	-	-
TAUG	$0.7491_{(+0.0035)}$	$0.6743_{(+0.0408)}$	$0.9498_{(+0.0007)}$	$0.8081_{(+0.0115)}$	$0.7485_{(+0.0032)}$	$\mathbf{0.7408}_{(+0.0268)}$	$0.9501_{(+0.0020)}$	$\mathbf{0.8335}_{(+0.0038)}$
CrossDistil	$\mathbf{0.7494}_{(+0.0038)}$	$\mathbf{0.7411}_{(+0.1076)}$	$\mathbf{0.9513}_{(+0.0022)}$	$\mathbf{0.8341}_{(+0.0375)}$	$\mathbf{0.7487}_{(+0.0034)}$	$0.7403_{(+0.0263)}$	$\mathbf{0.9502}_{(+0.0021)}$	$0.8324_{(+0.0027)}$

Table 4: Ablation analysis for Task A on TikTok dataset.

Variants	AUC	Multi-AUC
w/o AuxiliaryRank	0.7488 (-0.0006)	0.6510 (-0.0901)
w/o Calibration	0.7478 (-0.0016)	0.7396 (-0.0015)
w/o Correction	0.7486 (-0.0008)	0.7399 (-0.0012)
KD (same task)	0.7489 (-0.0005)	0.6901 (-0.0510)
KD (cross task)	0.7269 (-0.0225)	0.6120 (-0.1291)
Baseline	**0.7494**	**0.7411**

Table 5: Ablation analysis for Task B on TikTok dataset.

Variants	AUC	Multi-AUC
w/o AuxiliaryRank	0.9501 (-0.0012)	0.8005 (-0.0336)
w/o Calibration	0.9504 (-0.0009)	0.8312 (-0.0029)
w/o Correction	0.9508 (-0.0005)	0.8310 (-0.0031)
KD (same task)	0.9505 (-0.0008)	0.8014 (-0.0327)
KD (cross task)	0.9184 (-0.0329)	0.7520 (-0.0821)
Baseline	**0.9513**	**0.8341**

in one column, while the underlined value corresponds to the best one among all the baselines. To show improvements over the single-task counterpart, we report results of Single-Model which uses a separate network for learning each task. As is shown in the tables, the proposed CrossDistil achieves the best performance improvements over Single-Model in terms of AUC and Multi-AUC [3]. These results manifest that CrossDistil could indeed better leverage the knowledge from other tasks to improve both bipartite and multipartite ranking abilities on all tasks. Also, TAUG model alone, without calibrated KD, achieves better performance compared with the backbone model PLE, which validates the effectiveness of task augmentation.

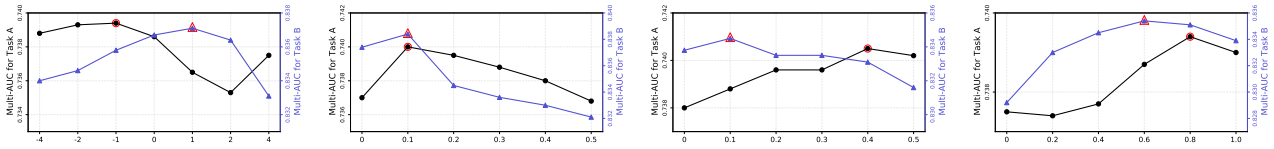
Besides, there are several observations in comparison tables. First, Single-Model on augmented ranking-based tasks (teacher) achieves better results in Multi-AUC compared with Single-Model on original regression-based task (student). It verifies that the proposed augmented tasks are capable of capturing task-specific fine-grained ranking information. Second, the student model exceeds the teacher model both in AUC and Multi-AUC performance in most cases, which is not strange since the student benefits from additional training signals that could act as label smoothing regularization and the teacher does not have such advantage. The same phenomenon is observed in many other works (Yuan et al. 2020; Tang et al. 2020b; Zhang and Sabuncu 2020)

[3]For large-scale datasets in online advertisement, the improvements of AUC in the table is considerable because of its hardness.

4.5 RQ2,3,4: Ablation Study

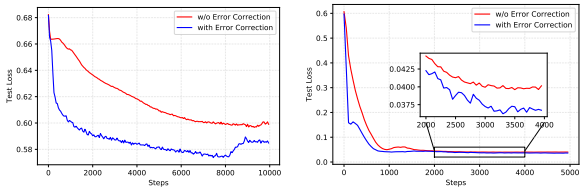
We design a series of ablation studies to investigate the effectiveness of some key components. Four variants are considered to simplify CrossDistil by: i) removing BPR losses for learning auxiliary ranking relations, ii) directly employing the teacher model outputs for knowledge distillation without any calibration, iii) not applying the error correction mechanism, vi) using regression-based teacher models that learn the same task as students and using the vanilla knowledge distillation that is similar with (Zhou et al. 2018), v) directly using the predictions of another task for distillation. Table 4 and 5 show the results for these variants on TikTok dataset and performance drops compared with the baseline (i.e. CrossDistil).

For the first variant, teacher loss function degrades to traditional BPR loss with no auxiliary ranking information. Such auxiliary ranking information that contains cross-task knowledge is a key factor for good performance in AUC and Multi-AUC. The second variant without calibration may produce unreliable soft labels and result in performance deterioration. Also, it is worth mentioning that the calibration process could significantly improve the performance of LogLoss, which is a widely used regression-based metric. Concretely, LogLoss reduces from 0.5832 to 0.5703 for task A, and 0.0623 to 0.0337 for task B by using calibration. The results of the third variant indicate that the error correction mechanism can also bring up improvements for AUC and Multi-AUC. Another benefit of error correction is to accelerate model training, which will be further discussed. For



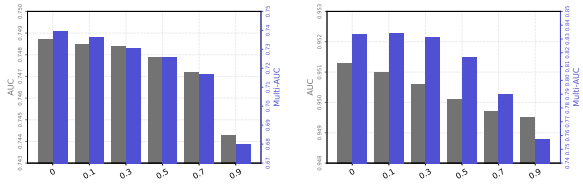
(a) Error correction margin m (b) Coefficient β_1 (c) Coefficient β_2 (d) Distillation loss weight α

Figure 3: Multi-AUC performance on TikTok dataset for TaskA and Task B w.r.t. different hyper-parameters.



(a) Task A (Finish Watching). (b) Task B (Like).

Figure 4: Learning curves of CrossDistil with and without error correction mechanism on TikTok dataset.



(a) Task A (Finish Watching). (b) Task B (Like).

Figure 5: Impact of corrupted auxiliary ranking information on the student model performance for TikTok dataset.

the fourth variant, we can see that the proposed CrossDistil is better than the vanilla KD since it transfers fine-grained ranking knowledge across tasks. For the last variant, directly conducting KD could cause performance drop because of the ranking conflicts of tasks.

RQ3: Does Error Correction Mechanism Help to Accelerate Convergence and Enhance Knowledge Quality? To answer this question, we plot the learning curves of test loss with (blue line) and without (red line) error correction in Fig. 4. As we can see, for both tasks, the test loss of CrossDistil with error correction significantly goes down faster at the beginning of the training process when the teacher is not well-trained. Plus, at later stage of training when the teacher becomes well-trained, the test loss of CrossDistil with error correction slowly keeps going down and achieves better optimal results compared with the variant, indicating that the proposed error correction mechanism could indeed help to improve knowledge quality.

RQ4: Does the Student Model Really Benefit from Auxiliary Ranking Knowledge from Other Tasks? To answer this question, we conduct the following experiment: For a target task A, we randomly choose a certain ratio of positive samples of task B, and then exchange their task B's label with the same number of randomly selected negative samples, to create a corrupted training set. Note that such data corruption process only has negative effects on the reliability of the auxiliary ranking information, so that we can investigate its impact on the student model's performance. Figure 5 shows the results of performance change when increasing the ratio from 10% to 90%. The results indicate that flawed auxiliary information has considerable negative effects on the overall performance, which again verifies CrossDistil could effectively transfer knowledge across tasks.

4.6 RQ5: Hyper-parameter Study

This subsection studies the performance variation of CrossDistil w.r.t. some key hyper-parameters (i.e. error correction margin m, auxiliary ranking loss coefficient β_1 and β_2, distillation loss weight α). Figure 3(a) shows the Multi-AUC performance with error correction margin ranges from -4 to 4. As we can see, the model performance first increases and then decreases. This is because extremely small m is equivalent to not conducting error correction, while extremely large m makes the soft labels degrade to hard labels. The results in Fig. 3(b) and Fig. 3(c) indicate a proper setting for β can help to capture the correct underlying fine-grained ranking information. The results in Fig. 3(d) reveal that a proper α from 0 to 1 can bring the best performance, which is reasonable since the distillation loss plays the role of label smoothing regularization and could not replace hard labels.

5 Conclusion

In this paper, we propose a cross-task knowledge distillation framework for multi-task recommendation. First, augmented ranking-based tasks are designed to capture fine-grained ranking knowledge, which could avoid conflicted information to alleviate negative transfer problem and prepare for subsequent knowledge distillation. Second, calibrated knowledge distillation is adopted to transfer knowledge from augmented tasks (teacher) to original tasks (student). Third, an additional error correction method is proposed to speed up the convergence and improve knowledge quality in the synchronous training process.

CrossDistil could be incorporated in most off-the-shelf multi-task learning models, and is easy to be extended or modified for industrial applications such as online advertising. The core idea of CrossDistil could inspire a new paradigm for solving domain-specific task conflict problem and enhancing knowledge transfer in broader areas of data mining.

6 Acknowledgments

This work was supported by the National Key R&D Program of China [2020YFB1707903]; the National Natural Science Foundation of China [61872238, 61972254], Shanghai Municipal Science and Technology Major Project [2021SHZDZX0102], the Tencent Marketing Solution Rhino- Bird Focused Research Program [FR202001], the CCF-Tencent Open Fund [RAGR20200105], and the Huawei Cloud [TC20201127009]. Xiaofeng Gao is the corresponding author.

References

Anil, R.; Pereyra, G.; Passos, A.; Ormandi, R.; Dahl, G. E.; and Hinton, G. E. 2018. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*.

Caruana, R. 1997. Multitask learning. *Machine learning*, 28(1): 41–75.

Duong, L.; Cohn, T.; Bird, S.; and Cook, P. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *ACL*, 845–850.

Ferri, C.; Hernández-Orallo, J.; and Modroiu, R. 2009. An experimental comparison of performance measures for classification. *PRL*, 30(1): 27–38.

Hand, D. J.; and Till, R. J. 2001. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45(2): 171–186.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hsieh, M.-E.; and Tseng, V. 2021. Boosting Multi-task Learning Through Combination of Task Labels-with Applications in ECG Phenotyping. In *AAAI*, volume 35, 7771–7779.

Liu, S.; Davison, A. J.; and Johns, E. 2019. Self-supervised generalisation with meta auxiliary learning.

Lu, Y.; Dong, R.; and Smyth, B. 2018. Why I like it: multi-task learning for recommendation and explanation. In *RecSys*, 4–12.

Ma, J.; Zhao, Z.; Chen, J.; Li, A.; Hong, L.; and Chi, E. H. 2019. Snr: Sub-network routing for flexible parameter sharing in multi-task learning. In *AAAI*, volume 33, 216–223.

Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; and Chi, E. H. 2018a. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *SIGKDD*, 1930–1939.

Ma, X.; Zhao, L.; Huang, G.; Wang, Z.; Hu, Z.; Zhu, X.; and Gai, K. 2018b. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *SIGIR*, 1137–1140.

Misra, I.; Shrivastava, A.; Gupta, A.; and Hebert, M. 2016. Cross-stitch networks for multi-task learning. In *CVPR*, 3994–4003.

Niculescu-Mizil, A.; and Caruana, R. 2005. Predicting good probabilities with supervised learning. In *ICML*, 625–632.

Pan, J.; Mao, Y.; Ruiz, A. L.; Sun, Y.; and Flores, A. 2019. Predicting different types of conversions with multi-task learning in online advertising. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2689–2697.

Phuong, M.; and Lampert, C. 2019. Towards understanding knowledge distillation. In *ICML*, 5142–5151.

Platt, J.; et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3): 61–74.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Ruder, S. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Shan, L.; Lin, L.; and Sun, C. 2018. Combined regression and tripletwise learning for conversion rate prediction in real-time bidding advertising. In *SIGIR*, 115–123.

Shan, L.; Lin, L.; Sun, C.; Wang, X.; and Liu, B. 2017. Optimizing ranking for response prediction via triplet-wise learning from historical feedback. *International Journal of Machine Learning and Cybernetics*, 8(6): 1777–1793.

Standley, T.; Zamir, A.; Chen, D.; Guibas, L.; Malik, J.; and Savarese, S. 2020. Which tasks should be learned together in multi-task learning? In *ICML*, 9120–9132.

Tang, H.; Liu, J.; Zhao, M.; and Gong, X. 2020a. Progressive layered extraction (PLE): A novel multi-task learning model for personalized recommendations. In *RecSys*, 269–278.

Tang, J.; Shivanna, R.; Zhao, Z.; Lin, D.; Singh, A.; Chi, E. H.; and Jain, S. 2020b. Understanding and improving knowledge distillation. *arXiv preprint arXiv:2002.03532*.

Tang, J.; and Wang, K. 2018. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *SIGKDD*, 2289–2298.

Wang, N.; Wang, H.; Jia, Y.; and Yin, Y. 2018. Explainable recommendation via multi-task learning in opinionated text data. In *SIGIR*, 165–174.

Wen, T.; Lai, S.; and Qian, X. 2019. Preparing lessons: Improve knowledge distillation with better supervision. *arXiv preprint arXiv:1911.07471*.

Xu, C.; Li, Q.; Ge, J.; Gao, J.; Yang, X.; Pei, C.; Sun, F.; Wu, J.; Sun, H.; and Ou, W. 2020. Privileged features distillation at Taobao recommendations. In *SIGKDD*, 2590–2598.

Yang, Y.; and Hospedales, T. 2016. Deep multi-task representation learning: A tensor factorisation approach. *arXiv preprint arXiv:1605.06391*.

Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; and Finn, C. 2020. Gradient surgery for multi-task learning. *NeurIPS*.

Yuan, L.; Tay, F. E.; Li, G.; Wang, T.; and Feng, J. 2020. Revisiting knowledge distillation via label smoothing regularization. In *CVPR*, 3903–3911.

Zhang, Y.; and Yang, Q. 2021. A survey on multi-task learning. *TKDE*.

Zhang, Z.; and Sabuncu, M. R. 2020. Self-distillation as instance-specific label smoothing. *arXiv preprint arXiv:2006.05065*.

Zhou, G.; Fan, Y.; Cui, R.; Bian, W.; Zhu, X.; and Gai, K. 2018. Rocket launching: A universal and efficient framework for training well-performing light net. In *AAAI*, volume 32.

Zhou, H.; Song, L.; Chen, J.; Zhou, Y.; Wang, G.; Yuan, J.; and Zhang, Q. 2021. Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective. *ICLR*.

Zhu, J.; Liu, J.; Li, W.; Lai, J.; He, X.; Chen, L.; and Zheng, Z. 2020. Ensembled CTR prediction via knowledge distillation. In *CIKM*, 2941–2958.