# TRAINING VIA CONFIDENCE RANKING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Model evolution and constant available data are two common phenomenon in large-scale real-world machine learning application, e.g. ads and recommendation system. To adapt, real-world system typically operates both *retraining* with all available data and *online-learning* with recent data to update models periodically with the goal of better serving performance for future data. However, if model and data evolution results in a vastly different training manner, it may induce negative impact on online A/B platform. In this paper, we propose a novel framework, named *Confidence Ranking*, to design optimization objective as a ranking function with two different models. Our confidence ranking loss allows directly optimizing the logits output for different convex surrogate function of metrics, e.g. *AUC* and *Accuracy* depending on the target tasks and datasets. Armed with our proposed methods, our experiments show that the confidence ranking loss can outperform for test-set performance on CTR prediction and model compression with various setting against the knowledge distillation baselines.

## 1 INTRODUCTION

To alleviate the discrepancy between delayed training and test distribution, typically, the real-world ads and recommendation system widely operate machine learning pipeline as follows: (1) collect user-clicks periodical data (every $\Delta = 24$ hours for daily data and $\Delta < 10$ minutes for online data); (2) train the model on collected data then deploy the model for serving until next new retrained model is produced. Due to the non-stationary data distribution and constant model evolution, the periodic retraining methodology motivate fast adaption and better generalization for approximating the recent decision boundary. Although retraining with new or all data does make the online model more current and new model more powerful, the improvement only originates from the alleviation of train-test discrepancy or promotion of model capability. We investigate if modeling previous prediction distribution can improve the performance even further. For short, as we train the model at a time $t$ with data $\mathcal{D}^t$, we only know that the model will be optimized by ERM in this period while ignoring how the data are produced and influenced by the base model. Our goal in this paper is to improve test-set performance of retrained model when deployed in the next period.

**Machine learning Pipeline in Real-World System.** We take the click-through rate (CTR) prediction task in real-world commodity ranking system as example and consider it over an input space $\mathcal{X}$ and ground truth space $\mathcal{Y}$ where the joint distribution is $\mathcal{P}(\mathcal{X}, \mathcal{Y})$ that always evolve with time. It means the hypothesis space of data is not fixed but evolving. Thus, given a prediction model, it should be trained conditioned on each time split, i.e. minimize the $\ell(y|x, t)$. During training, all of the given data are collected from $T$ time snapshots $t_1 \leq t_2 \cdots \leq t_T$. Collect these data as $\mathcal{D}_{old} \triangleq \left\{\mathcal{D}^1, \mathcal{D}^2, \cdots, \mathcal{D}^T\right\}$, we train our base model for deploying on next time interval in the future denoted as $t_{T+1}$. **Problem Statement.** Due to the expectation of maximizing the performance on $t_{T+1}$, the machine learning pipeline mostly follow practical online learning setup by iterative collecting data followed by training and deploying models which is different from the original setting of online learning for minimizing the total regret. Though we only care about the performance on next time interval, real-world production system typically releases model through an interactive improvement process by launching new models via collecting additional data and proposing empirically beneficial candidate model to replace the model of old deployed version(Jiang et al., 2021). In other, long-standing model training by empirical risk minimization (ERM) may not guarantee the online performance due to the *over-fitting* problem. These problematical issues make the continual training mode in real-world system much more complex than time-series prediction and classification, since

the unknown data distribution and model discrepancy. Despite our ultimate goal to improve the performance of online deployment stage, it can be split into two fold: improve *retraining* (with almost all of data) and *online learning* (with recently produced data) stage. Thus, an important question remains open regarding machine learning pipeline of real-world system:

*How can we train a model better than the deployed in retraining and online-learning stage?*

In this paper, we provide an affirmative answer to this question. Our solution is to optimize a novel framework of loss function for machine learning application (MLA) instead of ERM (e.g. optimize cross-entropy loss for classification). In particular, we choose to maximize the ranking score between base model and retrained model for MLA. There are several benefits of maximizing the ranking score over minimizing the cross-entropy loss. First, maximizing the ranking score is naturally suitable for classification task in real-world application (i.e. image recognition, ctr prediction and etc) where train-test distribution is nearly identically independent distributed. Directly maximizing the ranking score of different models can approximate improving the model performance relatively. Second, this framework is more suitable for handling various model complexity since maximizing ranking score aims to learn better decision boundary compared to baseline. The foremost challenge in this paper is to determine the surrogate loss for this setting. In our study, a naive approach of exploiting ranking-based pair-wise surrogate loss can be efficiently optimized in various tasks with different backbone networks. Our works is related to knowledge distillation(Hinton et al., 2015) that usually distill the output of *teacher* model to the *students*. This technique has been proved to be successful in deep learning combined with other standard ERM. Despite its success, KD and it's successor follow the same principle of aligning the student with the teacher differing to the purpose we want. Additional to the comparison with standard ERM loss, we also experiment with knowledge distillation methods in ctr prediction on real-world system and model compression for image recognition. In this paper, we show that our proposed method is not only suitable for real-world application but also beneficial on the model compression. For better understanding the diversity, we provide mathematical comparison for knowledge distillation and confidence ranking. In this paper, our contribution are summarized as follows:

- We highlight that in the real-world MLA retrained with all data and new arriving data can make model more current and capable while it keeps unknown of the performance of the retrained candidate model compared to the deployed baseline. To address it, we propose a novel loss framework for ranking the retrained model with the deployed one, namely confidence ranking. The novel loss only need the logit output of the deployed model which is efficient for real-world setting. Beyond ranking for point-wise logit output, we apply the framework to rank the bipartite distance in CTR prediction and the class margin in model compression for multi-class classification.
- Despite our real-world engineering setting, knowledge distillation can still achieve great success induced by lower logit variance of the *teacher* (Menon et al., 2021). Our confidence ranking approach is very different to this series of loss function. We present both theoretical and experimental results showing the superiority over distillation on supervised learning and model compression.

## 2 METHODS

### 2.1 PRELIMINARIES

Formally, as demonstrated in section 1, a CTR prediction pipeline in real-world system can be split to three parts: **(1)** Offline training: given old dataset $\mathcal{D}_{old}$ that consists of continuous $T$ days data with corresponding input sample $x$ ad ground truth labels $y$, we build a machine learning model $f$ with parameters $\theta$ for the aim to optimize in a sequential manner. We note the output logits with $z \triangleq h(x;\theta)$ and the corresponding probability with $f(x;\theta) \triangleq \text{sigmoid}(h(x;\theta))$. The goal of first part is to optimize $f$ on the $\mathcal{D}_{old}$ where standard pipeline follows the ERM optimization:

$$\arg\min_{\theta} \mathcal{L}_{old}, \quad \text{where} \quad \mathcal{L}_{old} \triangleq \mathbb{E}_{(x,y)\sim\mathcal{D}_{old}}[\ell(y, f(x;\theta))] \tag{1}$$

**(2)** Online serving: after the loss $\mathcal{L}_{old}$ converge smaller than $\delta$, where $\delta$ is specified by cross-validation on $\mathcal{D}_{old}$, we deploy the machine learning model $f_\theta$ on the real-world system for the aim to serve the new arriving dataset $\mathcal{D}_{new}$ where the ground truth $y_{new}$ is unknown until the user clicks the

item or leave the browser. Until now, we have demonstrate the pipeline of real world deployment applications. **(3)** Online learning: As we get new arriving data, we retrain previous deployed model $f_{old}$ to overcome the challenge of inconsistency between $\mathcal{D}_{old}$ and $\mathcal{D}_{new}$, Benefited from mitigation on distributional drift, the strategy of online learning with recent data can efficiently improve the generalization on the next serving stage. Despite effectiveness of the pipeline, this strategy does not take previous model's outputs as auxiliary information which covers the underlying relationship between prediction and ground truth.

The goal in this paper is to modify the training target of retrained and online-learning model with the online deployed one's outputs so that our training strategy can be more effective on improving test-time generalization. The main intuition behind our model is to learn better classifier and representation that encode the paired information between retrained model and the deployed model. A natural idea is to induce the past *experience* by knowledge distillation (Hinton et al., 2015; Buzzega et al., 2020) together with cross entropy loss. One of the challenge of distillation-based methods is that the past *experience* is not always right and only gets marginal improvement through its *dark* knowledge. The online deployed model's prediction may give over-estimated and under-estimated confidence due to the uncertainty of users' evolving interest. The *dark* knowledge of wrong response achieves great success for distilling big model to a small model but at the same time constrain the potential ability that the student can perform better than teacher(Dao et al., 2021). The failure mode of vanilla KD has two fold. Assume our teacher is trained insufficiently or restrictively leading to *under-fitting*, since the vanilla KD only exploits the strategy that student learns from the teacher without access to the original labels, we could expect the student to be inaccurate due to the teacher under-fitting even the student and teacher belong to same complexity model classes. Second, if the complexity of teacher model has a large critical radius(Wainwright, 2019), error bound of the student suffers due to potential teacher *over-fitting*. These reasons suggest directly optimizing KL-divergence of retrained model and previous deployed model may not be optimal for the purpose of better performance on test data. When learning the past experience we instead follow a simple idea motivated by AUC loss (Freund et al., 2003; Mohri et al., 2018) that maximize the difference of a random selected congruent pairs $\{y^+, y^-\}$ so that $y^+$ is scored closed to ground truth $y$ than $y^-$. The functional form of AUC risk is defined as: $R_{auc}(f) \triangleq \mathbb{E}_{\{x^+,x^-\}\sim\mathcal{P}^+,\mathcal{P}^-}\left[\mathbf{1}_{\{y^+>y^-\}}\right]$.

### 2.2 Confidence Ranking

In this paper, we devise confidence-ranking loss to directly utilize previous learned or served experiences (knowledge). Motivated by AUC loss, we borrow the idea of optimizing the ranking performance for maximizing the expectation of how often current model produces more confident results than previous model. This expectation only need comparison between congruent pairs thus we can use point-wise convex surrogate loss function $\phi(\hat{y}, y_{old}) = \phi(f(x;\hat{\theta}) - f(x;\theta_{old}))$ with the aim to minimize confidence-ranking $\phi$-risk:

$$R_p(f) \triangleq \mathbb{E}_{\{\hat{y},y_{old}\}\sim\mathcal{P}(x)}\left[y(\phi(f(x;\hat{\theta}) - f(x;\theta_{old})))\right] \tag{2}$$

The common approach to optimize the bayes risk of the score function is adopting possible surrogate function (e.g. square loss, hinge loss, exponential loss, and logistic loss).

**Relational Confidence Ranking (RCR):** The point-wise loss ranking the output of current model with old deployed one ensures the network gradually perform better. To further improve the bipartite ranking performance of binary classification, we follow (Freund et al., 2003) in optimizing bipartite ranking performance. In short, we aim to maximize the pos/neg distance of current model taking old generated prediction as margin. The similar idea can be extended into maximizing the similarity of samples from same classes. Thus, we define two *relational confidence ranking* $\phi$-risk in total:

$$R_{re}(f) \triangleq \begin{cases} \mathbb{E}_{\{x^+,x^-\}\sim\{\mathcal{P}^+,\mathcal{P}^-\}}\left[\phi(d_f(x^+,x^-) - d_{f_{old}}(x^+,x^-))\right] \\ \mathbb{E}_{\mathcal{P}_c\sim\{\mathcal{P}^+,\mathcal{P}^-\}}\mathbb{E}_{\{x^i,x^{-i}\}\sim\mathcal{P}_c}\left[\phi(d_{f_{old}}(x^i,x^{-i}) - d_f(x^i,x^{-i}))\right] \end{cases} \tag{3}$$

where function $d_f(x,z) = f(x) - f(z)$ performs calculating distance of different samples from various classes and same class, respectively. And we random select congruent pairs$\{x^i, x^{-i}\}$ from same distribution(class) $\mathcal{P}_c$ for maximizing similarity. Empirically, we find maximize the pos/neg

distance works well for binary classification (e.g. CTR prediction in ads and recommendation system) while maximize the similarity of samples from same classes fails on imbalanced datasets and slow down the training speed with large batch size. When we try to pull closer for samples of same classes, this loss doesn't guarantee the decision boundary between positive and negative classes gets more separate. Thus, we only construct relational confidence ranking loss for samples from various classes.

**Margin-based Relational Confidence Ranking (MRCR):** The proposed relational confidence ranking function is devised for maximizing the bipartite ranking performance in essence. However, in multi-class classification, maximizing this objective often leads to unstable optimization. Instead, we propose a margin-based confidence ranking loss for maximizing the distance of the logit output from different classes. Given a label space $[l] = \{1, \cdots, l\}$, we define *margin-based relational confidence ranking* $\phi$-risk as:

$$R_{m\_re} \triangleq \mathbb{E}_{\mathcal{P}_c \sim \mathcal{P}_{[l]}} \mathbb{E}_{x \sim \mathcal{P}_c} \mathbb{E}_{i \sim [l]} \left[ \phi(s_{f,c,i}(x) - s_{f_{old},c,i}(x)) \right] \tag{4}$$

where function $s_{f,c,i}(x) = \text{logit}(f(x))_c - \text{logit}(f(x))_i$ performs calculating difference of logit outputs of class $c$ and $i$. In this way, we aim to maximize the sample-wise class margin(Elsayed et al., 2018; Koltchinskii & Panchenko, 2002) for obtaining better inter-class separability.

### 2.3 APPLICATIONS

**CTR prediction in real-world MLA:** Our architecture comprises of two parts: (1) collect outputs $y_{old}$ of $f(x; \theta_{old})$ as a online deployed prediction probability which directly decides which item will be exposed and (2) impose a ranking-based loss to encourage the network to learn better than the online deployed one. We make the network pctr-sensitive by both taking prediction as an additional input concatenated with $x$. Similar to some prior distillation-based work (Tang & Wang, 2018; Cai et al., 2022; Hinton et al., 2015; Buzzega et al., 2020), we want our surrogate loss to be general as vanilla loss (e.g. cross-entropy loss and mean square loss) which can be trained end-to-end. Until now, we have stated how we propose to learn against old deployed models. In this paper, we suggest that margin-based square loss and logistic loss can effectively exploited for our proposed methods. Formally, the point-wise confidence ranking loss for ctr prediction is defined as:

$$\ell_{p\text{-}se}(f) \triangleq \mathbb{E}_{y \sim \mathcal{P}(x)} \left[ y(m - (f(x) - f_{old}(x)))^2 + (1 - y)(m - (f_{old}(x) - f(x)))^2 \right] \tag{5}$$

As known to literature of knowledge distillation, the optimization of KL divergence can be equivalent to minimizing the Euclidean distance between the corresponding logits under mild assumptions. The proposed margin-based square loss can be opt for match logits closed to the one-hot label. Another popular selection is to adopt logistic loss which is defined as:

$$\ell_{p\text{-}log}(f) \triangleq \mathbb{E}_{y \sim \mathcal{P}(x)} \left[ -y log \left[ \sigma(f(x) - f_{old}(x)) \right] - (1 - y) log \left[ 1 - \sigma(f(x) - f_{old}(x)) \right] \right] \tag{6}$$

where $\sigma$ is the sigmoid function. These two convex surrogate function can be easily extended to rank relational performance. The final loss function for ctr prediction is defined as:

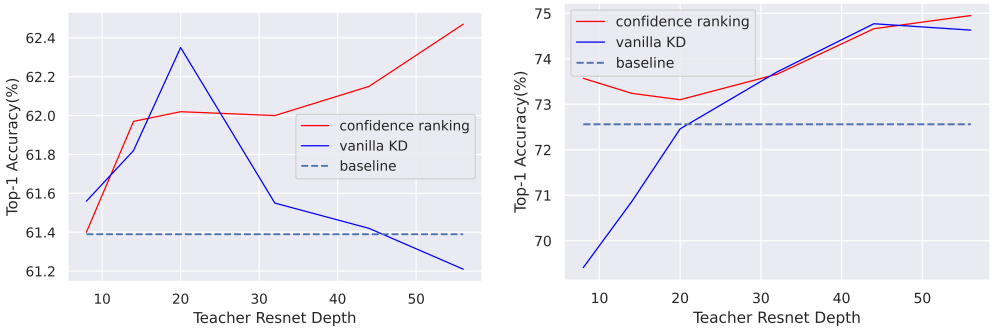
$$\ell_{ctr} = \alpha \cdot \ell_{ce} + \beta_1 \cdot \ell_p + \beta_2 \cdot \ell_{re} \tag{7}$$

where $\alpha$, $\beta_1$ and $\beta_2$ are the hyper-parameter controlling the CE, CR and RCR weight. Note that our proposed method only use logistic loss function without specified.

**Supervised Model Compression for image classification:** Model Compression can be seen as the retraining stage mentioned in above where the *teacher* is the online deployed baseline and the *student* is the new retrained model. In this setting, we directly follow the implementation of original KD(Hinton et al., 2015) without extra memory buffer for recording the *teacher* logit. Our point-wise confidence ranking loss is the same as the loss for ctr prediction except that $\sigma$ is softmax function. We further apply logistic surrogate loss function to maximize the sample-wise class margin. Empirically, we define combination of overall loss functions as :

$$\ell_{mc} = \alpha \cdot \ell_{ce} + \beta_1 \cdot \ell_p + \beta_2 \cdot \ell_{m\_re} \tag{8}$$

The combination of point-wise and relational loss is not necessary for all applications and heavily depends on datasets and metrics. In Section 4, we conduct experiments on CR and MRCR respectively.




(a) CR vs KD with ResNet8 as student

(b) CR vs KD with ResNet56 as student

Figure 1: Illustration of the test-set probability of students for vanilla knowledge distillation and confidence ranking of various depths. For Cifar100, We train teachers that are ResNets of various depths and the student is fixed depth of 8 and 56 respectively. We only use the logistic loss function for confidence ranking. Note that CR always outperform the baseline results while vanilla KD fails when discrepancy of teacher-student model complexity is large. The results are averaged over 3 runs.

## 3 Bias-Variance Perspective for Confidence Ranking

Previous work(Menon et al., 2021) gives a statistical perspective of the success of knowledge distillation. In their theoritical analysis, a bayes-distilled loss can improve generalization over population loss by producing lower variance outputs. However, a teacher's prediction results is usually seen as an imperfect estimate of the true bayes label in realistic setting. They answer the success of imperfect teacher by inducing a fundamental *bias-variance* bound 9 which shows that trade-off complexity of the teacher can have a low MSE with Bayes probability $p^*$ leading to better student's generalization error.

**Proposition 1** ***(Bias-Variance bound for knowledge distillation)*** *Pick any bounded loss $\ell$. Suppose we have a teacher model $p^t$ with corresponding distilled empirical risk $\widetilde{R}(f) = \frac{1}{N}\sum_{n\in N} p^t(x_n)\ell(f(x_n))$ and population risk $R(f) = \mathbb{E}_x\left[p^*(x)\ell(f(x))\right]$ where $p^t(x_n)$ is the teacher output confidence. For any predictor $f$: $\mathcal{X} \to \mathbb{R}^L$,*

$$\mathbb{E}\left[(\widetilde{R}(f) - R(f))^2\right] \leq \frac{1}{N}\cdot\mathbb{V}\left[p^t(x)\ell(f(x))\right] + \mathcal{O}(\mathbb{E}\left[\|p^t(x) - p^*(x)\|_2\right])^2 \tag{9}$$

In practice, we expect the distilled risk to generalize better than classification risk for knowledge distillation tasks, i.e. for $\widetilde{R}(f)$ to be smaller than $R(f)$. This is because the classification risk is computed on "hard" labels which can be seen as over-confident teacher. whereas the distilled risk is computed on "soft" labels from the base model or teacher model. This view impose that even imperfect teacher(base) model may aid in better generalization of the student(retrained) model and accurate teacher may lead to worse generalization due to large variance. To this end, we apply this loose *bias-variance* bound to the confidence ranking risk.

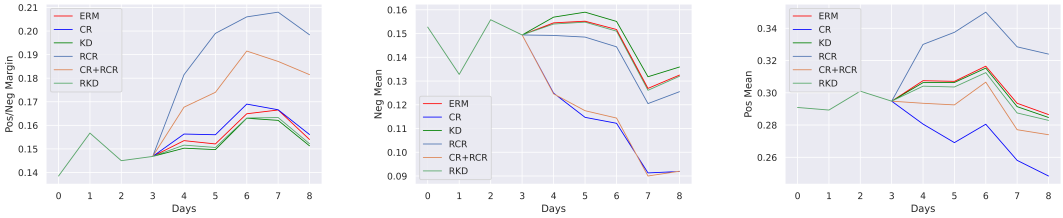
**Proposition 2** ***(Bias-Variance bound for point-wise confidence ranking)*** *Pick any convex loss $\ell$. Suppose we have a teacher model $p^t$ with corresponding empirical confidence ranking risk $\widehat{R}(f) = \frac{1}{N}\sum_{n\in N} y(x_n)\ell(f(x_n) - f_t(x_n)))$ and population risk $R(f) = \mathbb{E}_x\left[p^*(x)\ell(f(x))\right]$ where $f_t(x_n)$ is the teacher output. For any predictor $f$: $\mathcal{X} \to \mathbb{R}^L$,*

$$\mathbb{E}\left[(\widehat{R}(f) - R(f))^2\right] \leq \mathbb{E}\left[(R(f_t))^2\right] \tag{10}$$

Different to knowledge distillation, the fidelity of the confidence-ranking risk only depends one on factor: how well the teacher model estimates approximates the true $p*$ in a logistic sense. Since $p^* = \mathbb{P}(y|x)$ is a constant, 10 implies that confidence-ranking risk performs better when teacher(base) model is over-confident. We have stated a statistical perspective on confidence ranking, resting on the observation that confidence ranking offers a bound which always approximating Bayes probabilities

Table 1: AUC(%) of test-set performance on Avito, Avazu and Industrial datasets with various backbone and training strategy. We conduct KD, RKD$_l$, SC, CR, RCR and both with same DeepFM. * denotes one-pass learning. The results are averaged over 3 runs. Std $\leq 0.1\%$.

| Dataset | DNN | DCN | PNN | DeepFM | KD | RKD$_l$ | SC | CR | RCR | Both |
|---|---|---|---|---|---|---|---|---|---|---|
| Avazu | 75.05 | 74.99 | 75.06 | 75.24 | 75.41 | 75.34 | 75.36 | 75.63 | 75.59 | **75.66** |
| Avito | 77.71 | 77.66 | 77.80 | 77.73 | 78.01 | 77.83 | 77.78 | 78.33 | 78.59 | **78.62** |
| Avazu* | 74.32 | 74.30 | 74.49 | 74.69 | 74.83 | 74.90 | 74.85 | 74.98 | 75.05 | **75.14** |
| Avito* | 77.50 | 77.58 | 77.57 | 77.50 | 77.54 | 77.58 | 77.53 | 77.70 | **77.73** | 77.70 |
| Industrial* | 75.92 | 76.02 | n/a | n/a | 76.18 | 76.10 | 76.14 | 76.25 | 76.20 | **76.32** |




(a) Sample Margin (b) Negative Mean (c) Positive Mean

Figure 2: Sample margin, prediction mean of negative and positive samples on Avazu in one-pass setting. We use the data of first 4 days to train the base model. And then, imitate *online-learning* in a cycle serving-and-training process with constant daily data.

based on the performance of teacher model. However, this bound is not well qualified on deep learning architecture and may be loose and unstable for real-world application especially for the logistic confidence ranking loss. We note the comprehensive bound of confidence ranking requires specifying necessary conditions. Nonetheless, this qualitative bound can still hold majority conditions in practice. For example, Figure1 illustrates that distillation suffers from the large discrepancy of teacher-student model complexity while confidence ranking still yields satisfactory results. However, our proposed confidence ranking is not always better than KD. As shown in Figure 1a, KD achieve best accuracy when student is ResNet20 better than CR. As shown in Figure 1b, when *teacher* gets closer to student, the performance of KD improves as well as CR. This result shows KD may potentially benefit CR by carefully designed combination. We leave it into future study.

## 4 Experiments

Here we experimentally show our proposed learning objective can flexibly leverage the knowledge from previous models. We first evaluate our methods on Industrial, Avazu and Avito datasets with various controllable setting on CTR prediction. To additionally demonstrate the advantage of CR, we also include experiments on a well-studied task: supervised model compression. Our core algorithm is easy to implement with various machine learning platform. For industrial datasets, we develop it with TensorFlow while we release our code with PyTorch implementation for public dataset. All of our experiments are conducted on one P40 GPU for public datasets and 8 A100 GPUs for industrial dataset. Please refer to the Appendix for additional details.

### 4.1 CTR prediction

**Datasets.** We perform our experiments on three datasets with two training setting. Industrial search ads dataset contains 59 numerical and categorical feature fields. All of the fields data are discretized and transformed into sparse anonymous features. This dataset has more than ten billion instances range over one month with hundreds millions of active users and items. For speeding up offline experiments, we sample half of it for evaluating our method. Avazu is display recommendation dataset released on Kaggle that contain 40428967 samples with 22 feature fields. Avito is also released as ads click datasets on Kaggle containing 190107687 samples but only 16 feature fields. We

Table 2: Top-1 test accuracy(%) of student networks on CIFAR-100 with variaous distillation networks. Our methods are denoted by CR and MRCR. (↑) denotes outperformance over KD and (↓) denotes underperformance. Note that CR always outperform KD as well as the other baseline other than CRD. Our proposed CR outperforms CRD in 3 out of 6 benchmarks. Combined with vanilla KD, we show that our proposed method is compatible with KD. The results is averaged over 5 runs.

| Teacher<br>Student | WRN-40-2<br>WRN-16-2 | WRN-40-2<br>WRN-40-1 | resnet56<br>resnet20 | resnet110<br>resnet20 | resnet110<br>resnet32 | vgg13<br>vgg8 |
|---|---|---|---|---|---|---|
| Teacher | 75.61 | 75.61 | 72.34 | 74.31 | 74.31 | 74.64 |
| Student | 73.26 | 71.98 | 69.06 | 69.06 | 71.14 | 70.36 |
| KD | 74.92 | 73.54 | 70.66 | 70.67 | 73.08 | 72.98 |
| FitNet | 73.58(↓) | 72.24 (↓) | 69.21(↓) | 68.99(↓) | 71.06(↓) | 71.02(↓) |
| AT | 74.08(↓) | 72.77(↓) | 70.55(↓) | 70.22(↓) | 72.31(↓) | 72.68(↓) |
| SP | 73.83(↓) | 72.43(↓) | 69.67(↓) | 70.04(↓) | 72.69(↓) | 72.68(↓) |
| CC | 73,56(↓) | 72.21(↓) | 69.63(↓) | 69.48(↓) | 71.48(↓) | 70.71(↓) |
| VID | 74.11(↓) | 73.30(↓) | 70.38(↓) | 70.16(↓) | 72.61(↓) | 71.23(↓) |
| RKD | 73.35(↓) | 72.22(↓) | 69.61(↓) | 69.25(↓) | 71.82(↓) | 71.48(↓) |
| PKT | 74.54(↓) | 73.45(↓) | 70.34(↓) | 70.25(↓) | 72.61(↓) | 72.88(↓) |
| AB | 72.50(↓) | 72.38(↓) | 69.47(↓) | 69.53(↓) | 70.98(↓) | 70.94(↓) |
| FT | 73.25(↓) | 71.59(↓) | 69.84(↓) | 70.22(↓) | 72.37(↓) | 70.94(↓) |
| FSP | 72.91(↓) | n/a | 69.95(↓) | 70.11(↓) | 71.89(↓) | 70.23(↓) |
| NST | 73.68(↓) | 72.24(↓) | 69.60(↓) | 69.53(↓) | 71.96(↓) | 71.53(↓) |
| CRD | 75.48(↑) | 74.14(↑) | 71.16(↑) | **71.46**(↑) | **73.48**(↑) | **73.94**(↑) |
| MRCR | 74.65(↓) | 73.07(↓) | 70.01(↓) | 69.41(↓) | 72.15(↓) | 72.24(↓) |
| CR | **75.79**(↑) | **74.53**(↑) | **71.59**(↑) | 71.32(↑) | 73.44(↑) | 73.62(↑) |
| CR+KD | 75.67(↑) | 74.10(↑) | 71.93(↑) | 70.81(↑) | 73.66(↑) | 74.03(↑) |
| MRCR+KD | 75.63(↑) | 74.14(↑) | 71.82(↑) | 71.34(↑) | 73.23(↑) | 73.56(↑) |

Table 3: Top-1 test accuracy(%) of student networks on CIFAR-100 with various distillation teahcer-student architecture. CR outperforms KD and all other methods except CRD. Our proposed CR outperforms CRD on 5 out of 6 benchmarks. However, MRCR is not always better than KD. It's because the ranking objective is not directly corresponding to the accuracy. Average over 3 runs.

| Teacher<br>Student | vgg13<br>MobileNetV2 | ResNet50<br>MobileNetV2 | ResNet50<br>vgg8 | resnet32x4<br>ShuffleNetV1 | resnet32x4<br>ShuffleNetV2 | WRN-40-2<br>ShuffleNetV1 |
|---|---|---|---|---|---|---|
| Teacher | 74.64 | 79.34 | 79.34 | 79.42 | 79.42 | 75.61 |
| Student | 64.6 | 64.6 | 70.36 | 70.5 | 71.82 | 70.5 |
| KD | 67.37 | 67.35 | 73.81 | 74.07 | 74.45 | 74.83 |
| FitNet | 64.14(↓) | 63.16 (↓) | 70.69(↓) | 73.59(↓) | 73.54(↓) | 73.73(↓) |
| AT | 59.40(↓) | 58.58(↓) | 71.84(↓) | 71.73(↓) | 72.73(↓) | 73.32(↓) |
| SP | 66.30(↓) | 68.08(↑) | 73.34(↓) | 73.48(↓) | 74.56(↑) | 74.52(↓) |
| CC | 64.86(↓) | 65.43(↓) | 70.25(↓) | 71.14(↓) | 71.29(↓) | 71.38(↓) |
| VID | 65.56(↓) | 67.57(↑) | 70.30(↓) | 73.38(↓) | 73.40(↓) | 73.61(↓) |
| RKD | 64.52(↓) | 64.43(↓) | 71.50(↓) | 74.10(↑) | 73.21(↓) | 72.21(↓) |
| PKT | 67.13(↓) | 66.52(↓) | 73.01(↓) | 73.55(↓) | 74.69(↑) | 73.89(↓) |
| AB | 66.06(↓) | 67.20(↓) | 70.65(↓) | 71.75(↓) | 74.31(↓) | 73.34(↓) |
| FT | 61.78(↓) | 60.99(↓) | 70.29(↓) | 74.12(↓) | 72.50(↓) | 72.03(↓) |
| NST | 58.16(↓) | 64.96(↓) | 71.28(↓) | 75.11(↓) | 74.68(↑) | 74.89(↑) |
| CRD | **69.73**(↑) | 69.11(↑) | 74.30(↑) | 75.11(↑) | 75.65(↑) | 76.05(↑) |
| MRCR | 68.30(↑) | 68.34(↑) | 72.32(↓) | 72.98(↓) | 75.31(↑) | 73.35(↓) |
| CR | 68.50(↑) | **69.38**(↑) | **74.58**(↑) | **75.28**(↑) | **76.11**(↑) | **76.45**(↑) |
| CR+KD | 69.06(↑) | 69.33(↑) | 74.10(↑) | 75.07(↑) | 76.09(↑) | 75.88(↑) |
| MRCR+KD | 68.62(↑) | 69.04(↑) | 73.55(↑) | 74.98(↑) | 75.40(↑) | 75.64(↑) |

construct public datasets by split into training/validation/test set by timestamp where the samples of last day is set for testing and penultimate day's data is set for validation and others are set for training. To split industrial dataset, we use traffic samples of previous 15 days as training set and the last day

as test set. All these datasets is constructed to predict click through rate which can be seen as binary classification. We summarize statistic of datasets in Table 4 .

**Experiments setup**. In real-world application, the prediction naturally influence the impression of items (i.e. items that have high confidence are more prone to be exposed to users.) and multiples times for training will cause severe *over-fitting* issues and much more computation cost. Thus we adopt two various settings for evaluate our methods. The details of configuration are summarized as follows. **One-Pass Setting:** we adopt one-pass training strategy for avoid *over-fitting* that each sample is only accessed once. For industrial dataset, it's common setting for evaluating performance of our methods. However, for public dataset, all of them only contain item and user features without any information of the deployed model. To overcome, we first train a one-pass model on training set and record its prediction on validation set followed by training with our method. In this setting, the only difference between ERM and CR is the confidence of validation set because our training set contains validation set. We denote our experiments on industrial dataset only adopt one-pass training strategy. **Standard Setting:** standard supervised learning. In this sense, all of our experiments first train on training set and early stop according to validation results. We use the outputs of previous round as input of our proposed method.

**Baselines.** Though our main motivation of our work is to utilze the confidence of online deployed model on target items, we still include several baselines under standard supervised learning and one-pass learning in order to benchmark state-of-the-art results. The simplest approach is (1) ERM: we train our networks with binary cross-entropy loss under two various settings. For majority of real-world recommendation and ads system, ctr prediction models are preferred to be trained with ERM; (2) various commonly adopted CTR prediction network architectures designed for recommendation and ads system, DNN, PNN(Qu et al., 2016), DCN(Wang et al., 2017), DeepFM(Guo et al., 2017); (3) SC(Cai et al., 2022) integrates self-correction module into CTR prediction networks. Together training with ERM, it achieves state-of-the-art results on multiple CTR prediction datasets with minimal computation cost. (4) Knowledge distillation methods: since dark knowledge can induce useful gradients for model compression, we also adapt KD(Hinton et al., 2015) and RKD(Park et al., 2019) to our experimental setting. In this paper, we modify the feature-based RKD to logit-based method for aligning inter-sample distance of the logits output of the base model and current model.

**Main Results.** Table 1 compare the test-set AUC of our method on Click-Through-Rate prediction task. On Table 1, we first investigate the improvement brought by different feature interaction methods. We observe that PNN achieve best performance with marginal improvement on standard supervised learning setting but fails compared to DeepFM and DCN on one-pass setting. For convenience, we adopt DeepFM as backbone for our experiments. We can observe that the propose method CR outperforms all baselines no matter which setting is adopted. For standard supervised learning, it is also striking to see that on Avazu and Avito, our proposed CR and RCR both can outperform baselines by a large margin after trained with multiple epochs. We denote 0.1% improvement of AUC on Avazu and Avito is significant. For one-pass learning, we still observe that our proposed methods outperforms the backbone model but the margin is smaller than standard setting. It's because one-pass learning may not completely fit on the two public datasets. For industrial dataset, we carefully tune our proposed method with DCN due to its succinct implementation. Not surprisingly, it works as well. Compared to vanilla distillation, our methods improve 0.25/0.61/0.25/0.26/0.14% of the AUC respectively. Additional to offline experiments, we conduct online experiments on A/B platform. We observe over 5% improvement on CTR and apply it to serve main traffic in our system.

**Inter-class margin Visualization.** In Figure 2, we show how sample margin, prediction mean value of positive and negative samples vary along time. The relational confidence ranking loss outperforms all the other method by a large margin. We can observe RCR both decrease the negative mean and increase the positive mean in Figure 2b and 2c leading to best bipartite ranking performance among all baseline methods in Table 1. We find CR both decreases negative and positive mean resulting in marginal improvement on sample margin. We demonstrate it's because CTR prediction dataset is usually dominated by negative samples and our loss function tends to depress the negative prediction. In Figure 2, we find KD and $\text{RKD}_l$ give more smooth curve compared to our methods which may constrain the model's learning ability.

### 4.2 MODEL COMPRESSION FOR IMAGE CLASSIFICATION

**Setup.** Knowledge distillation is widely used for model compression. In this setting, we conduct our experiments on CIFAR-100(Krizhevsky et al., 2009) with various teacher-student architectures (e.g. ResNets(He et al., 2016), Wide-ResNets(Zagoruyko & Komodakis, 2016b) and etc). Different to knowledge distillation, our proposed loss is used for learning better than teacher which is prone to suffer *over-fitting*. To overcome, we simply combine our point-wise confidence ranking loss with cross-entropy loss. Experimentally, we follow the implementation of CRD(Tian et al., 2019) and our code is available on the supplementary materials.

**Results on CIFAR-100.** Table 2 and Table 3 compare top-1 accuracy of our method on model compression on CIFAR-100. We first investigate that students and teachers are both the same architectural style. We observe that our proposed CR consistently outperforms KD and other logit-based distillation, yet CR is on par with CRD in some situation. However, our proposed CR is far different from the feature-based contrastive learning distillation methods (e.g CRD) which shows potential improvement combined with these methods. Surprisingly, we find when trained together with KD, CR can still get marginal improvement for some combinations. This might because that the discrepancy of train-test distribution need a fine-grained balance ratio between CR and KD while we set same ratio for all experiments. Compared to CR, we observe that the loss function MRCR is not on par with KD while outperforms KD once trained together with KD. It may demonstrate our proposed framework can be further improved by distillation-based methods.

## 5 RELATED WORK

Our work mainly focus on two engineering stages: *retraining* and *online-learning* in real-world machine learning system which covers multiple sub-areas of machine learning.

**Retraining.** The motivation to retrain a model has many folds (e.g model evolution, data noisy and shift) where it needs to validate a new candidate model through online A/B tests. Recent deep learning techniques achieve great success on various applications by superior model architectures(Dosovitskiy et al., 2021; Devlin et al., 2018), data augmentation methods(Zhang et al., 2017; Cubuk et al., 2019), effective loss function(Elsayed et al., 2018; Khosla et al., 2020; Musgrave et al., 2020) and etc. However, most methods techniques only build upon learning from scratch but ignore the difference among trained models. The most related work to *retraining* is supervised model compression for which knowledge distillation (KD) has attracted tremendous attention in various areas (Tang & Wang, 2018; Kang et al., 2021; Liu et al., 2020; Hinton et al., 2015; Park et al., 2019; Furlanello et al., 2018; Tian et al., 2019; Li, 2018; Romero et al., 2014; Bagherinezhad et al., 2018). The goal of KD is to produce better student model than trained directly for model compression. Typically, they assume the capacity and complexity of student model is weaker than the teacher's under the resource constraints.

**Online learning.** Online learning in real-world system is similar to continuous domain generalization(Wang et al., 2020) and continual learning(Parisi et al., 2019). The methods of continuous domain generalization aim to generalize to target domain given $T$-shot unordered domain data. CIDA(Wang et al., 2020) simply takes time as input for building time-invariant feature vector followed by same classifier. They build two-party adversarial training framework which is challenging in large-scale application. Another related field is continual learning which aims to address the issues of catastrophic forgetting. However, in real-world system, online learning cares more on the future's performance.

## 6 CONCLUSION

Starting from view of real-world application, we identify the problem of learning model for better generalization when online deployed on *retraining* and *online-learning* stage. For this aim, we propose a confidence ranking based method which is agnostic of the network architecture and further extend it to relational and margin-based structure for maximizing the bipartite distance and sample-wise class margin. Furthermore, we give a comprehensive theoretical understanding of CR and knowledge distillation. Our extensive experiments on CTR prediction and model compression demonstrate the superiority of CR. We hope this paper can contribute to more applications and researches.

## REFERENCES


Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9163–9171, 2019.

Hessam Bagherinezhad, Maxwell Horton, Mohammad Rastegari, and Ali Farhadi. Label refinery: Improving imagenet classification through label progression. *arXiv preprint arXiv:1805.02641*, 2018.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Guohao Cai, Jieming Zhu, Quanyu Dai, Zhenhua Dong, Xiuqiang He, Ruiming Tang, and Rui Zhang. Reloop: A self-correction continual learning loop for recommender systems. *arXiv preprint arXiv:2204.11165*, 2022.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pp. 191–198, 2016.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019.

Tri Dao, Govinda M Kamath, Vasilis Syrgkanis, and Lester Mackey. Knowledge distillation as semiparametric inference. *arXiv preprint arXiv:2104.09732*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR 2021: The Ninth International Conference on Learning Representations*, 2021.

Gamaleldin Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. Large margin deep networks for classification. *Advances in neural information processing systems*, 31, 2018.

Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov):933–969, 2003.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pp. 1607–1616. PMLR, 2018.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3779–3787, 2019.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

Heinrich Jiang, Harikrishna Narasimhan, Dara Bahri, Andrew Cotter, and Afshin Rostamizadeh. Churn reduction via distillation. *arXiv preprint arXiv:2106.02654*, 2021.

SeongKu Kang, Junyoung Hwang, Wonbin Kweon, and Hwanjo Yu. Topology distillation for recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 829–839, 2021.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. *Advances in neural information processing systems*, 31, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Vladimir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Haitong Li. Exploring knowledge distillation of deep neural networks for efficient hardware solutions. *University Of Stanford: CS230 course report*, 2018.

Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. A general knowledge distillation framework for counterfactual recommendation via uniform data. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 831–840, 2020.

Aditya K Menon, Ankit Singh Rawat, Sashank Reddi, Seungyeon Kim, and Sanjiv Kumar. A statistical perspective on distillation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7632–7642. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/menon21a.html`.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pp. 681–699. Springer, 2020.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3967–3976, 2019.

Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 268–284, 2018.

Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5007–5016, 2019.

Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 1149–1154, 2016.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2820–2828, 2019.

Jiaxi Tang and Ke Wang. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2289–2298, 2018.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2019.

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1365–1374, 2019.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Hao Wang, Hao He, and Dina Katabi. Continuously indexed domain adaptation. *arXiv preprint arXiv:2007.01807*, 2020.

Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, pp. 12, 2017.

Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4133–4141, 2017.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016a.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016b.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856, 2018.

Table 4: The statistic of CTR prediction datasets

| Datasets | Users | Items | Fields | Feature size | Instances | Pos Ratio |
|---|---|---|---|---|---|---|
| Avazu[1] | N/A | N/A | 22 | 2018012 | 40428967 | 17% |
| Avito[2] | 3163597 | 28529 | 16 | 3419165 | 190107687 | 0.5% |
| Industrial | N/A | N/A | 59 | N/A | 12 Billion | 8% |

# A PROOFS OF THEORY

**Proposition 1** ***(Bias-Variance bound for knowledge distillation)*** *Pick any bounded loss $\ell$. Suppose we have a teacher model $p^t$ with corresponding distilled empirical risk $\widetilde{R}(f) = \frac{1}{N}\sum_{n\in N} p^t(x_n)\ell(f(x_n))$ and population risk $R(f) = \mathbb{E}_x\left[p^*(x)\ell(f(x))\right]$ where $p^t(x_n)$ is the teacher output confidence. For any predictor $f$: $\mathcal{X} \to \mathbb{R}^L$,*

$$\mathbb{E}\left[(\widetilde{R}(f) - R(f))^2\right] \leq \frac{1}{N}\cdot\mathbb{V}\left[p^t(x)\ell(f(x))\right] + \mathcal{O}(\mathbb{E}\left[\|p^t(x) - p^*(x)\|_2\right])^2$$

*Proof.* See Proposition 3 of (Menon et al., 2021) .

**Proposition 2** ***(Bias-Variance bound for point-wise confidence ranking)*** *Pick any convex loss $\ell$. Suppose we have a teacher model $p^t$ with corresponding empirical confidence ranking risk $\widehat{R}(f) = \frac{1}{N}\sum_{n\in N} y(x_n)\ell(f(x_n) - f_t(x_n)))$ and population risk $R(f) = \mathbb{E}_x\left[p^*(x)\ell(f(x))\right]$ where $f_t(x_n)$ is the teacher output. For any predictor $f$: $\mathcal{X} \to \mathbb{R}^L$,*

$$\mathbb{E}\left[(\widehat{R}(f) - R(f))^2\right] \leq \mathbb{E}\left[(R(f_t))^2\right]$$

*Proof.* According to the definition of (Menon et al., 2021), the confidence rank risk can be derived as $\widehat{R}(f) = \mathbb{E}_x\left[p^*(x)\ell(f(x) - f_t(x))\right]$. Since our surrogate loss function is convex, using Jensen's inequality, we have

$$\ell(f(x) - f_t(x)) \leq \ell(f(x)) - \ell(f_t(x))$$

then we have $\widehat{R}(f) - R(f) \leq \mathbb{E}_x\left[-p^*(x)\ell(f_t(x))\right]$. Thus, we further have

$$\begin{aligned}\mathbb{E}\left[(\widehat{R}(f) - R(f))^2\right] &\leq \mathbb{E}(\mathbb{E}_x\left[p^*(x)\ell(f_t(x))\right]\cdot\mathbb{E}_x\left[p^*(x)\ell(f_t(x))\right]) \\ &= \mathbb{E}\left[(R(f_t))^2\right]\end{aligned}$$

# B TRAINING DETAILS OF CTR PREDICTION

## B.1 ALGORITHM

We list our algorithm 1 and 2 for standard supervised setting and one-pass learning respectively.

## B.2 BASELINE TRAINING METHODS

: We compare the following training methods:

- ERM (Mohri et al., 2018): optimize the network by convex loss function e.g. cross-entropy loss, mean square loss function. We use binary sigmoid cross entropy loss for all experiments.
- KD (Hinton et al., 2015): optimize the retrained(online-learning) model by knowledge distillation with base model

**Algorithm 1** Algorithm for our proposed retraining strategy on standing supervised setting

**Input:** Feature extractor $f_\theta$, training and validation dataset $D_{tra}$, $D_{val}$. loss ratio $\alpha$ and $\beta$
**Training Phrase 1:**
**for** $t$=1 to $T$ **do**
  sample mini-batch $\mathcal{D}_m$ from $\mathcal{D}_{tra}$
  $\ell(\theta)$=$\frac{1}{m}\sum_{(x,y)\in\mathcal{D}_m}[\ell(y, f(x;\theta))]$
  update $\theta$ with optimization algorithm
  collect $f(x;\theta)$ as $y_{old}$
**end for**
**Training Phrase 2:**
**for** $t$= $T+1$ to $T+N$ **do**
  sample mini-batch $\mathcal{D}_m$ from $\mathcal{D}_{tra}$
  $\ell(\theta)$=$\frac{1}{m}\sum_{(x,y)\in\mathcal{D}_m}[\alpha\ell(y, f(x;\theta)) + \beta\ell_{cr}(y_{old}, f(x;\theta))]$
  update $\theta$ with optimization algorithm
  collect $f(x;\theta)$ as $y_{old}$
**end for**
**Prediction Phrase:** $y = f(x;\theta), x \in \mathcal{D}_{val}$

**Algorithm 2** Algorithm for our proposed retraining strategy on one-pass(*online-learning*) setting

**Input:** Feature extractor $f_\theta$, dataset $\mathcal{D} \triangleq \left\{\mathcal{D}^1, \mathcal{D}^2, \cdots, \mathcal{D}^{T+1}\right\}$. loss ratio $\alpha$ and $\beta$
**Pretraining Phrase: We firstly use $\hat{T}$ days data to train base model**
**for** $t$=1 to $\hat{T}$ **do**
  **while** there exists data of $\mathcal{D}^t$ is not visited **do**
    sample mini-batch $\mathcal{D}_m$ from $\mathcal{D}^t$
    $\ell(\theta)$=$\frac{1}{m}\sum_{(x,y)\in\mathcal{D}_m}[\ell(y, f(x;\theta))]$
    update $\theta$ with optimization algorithm
  **end while**
**end for**
**Online learning Phrase: We serve next-day data followed by training**
**for** $t$= $\hat{T}+1$ to $T$ **do**
  **Serve** $f$ **on** $\mathcal{D}^t$**, Collect** $y_t$ **as** $y_{old}$
  **Online training:**
  **while** there exists data of $\mathcal{D}^t$ is not visited **do**
    sample mini-batch $\mathcal{D}_m$ from $\mathcal{D}^t$
    $\ell(\theta)$=$\frac{1}{m}\sum_{(x,y)\in D_m}[\alpha\ell(y, f(x;\theta)) + \beta\ell_{cr}(y_{old}, f(x;\theta))]$
    update $\theta$ with optimization algorithm
  **end while**
**end for**
**Prediction Phrase:** $y = f(x;\theta), x \in \mathcal{D}^{T+1}$

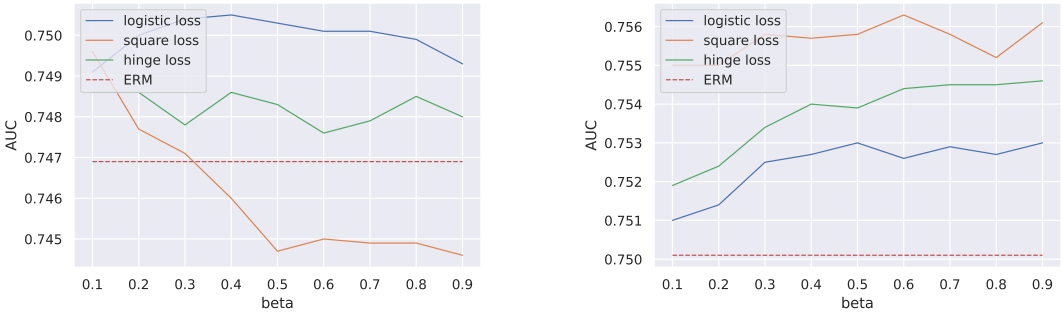
- RKD (Park et al., 2019): Original RKD propose to use feature-wise aligning strategy. However, this strategy is very expensive on one-pass setting and standard supervised learning setting for recommendation system. We follow the idea to modify it to align the base model and the retrained model by directly match the square of confidence difference of pos/neg pairs. It is defined as:

$$RKD_\ell = (d_f(x^+, x^-) - d_{f_{old}}(x^+, x^-))^2$$

- CR: we propose to rank the output of retrained model with the output of base model. It's the counterpart of KD
- RCR: we propose to rank the pos/neg difference of retrained model with the difference of base model. It's the counterpart of RKD$_\ell$

### B.3 Network Architecture

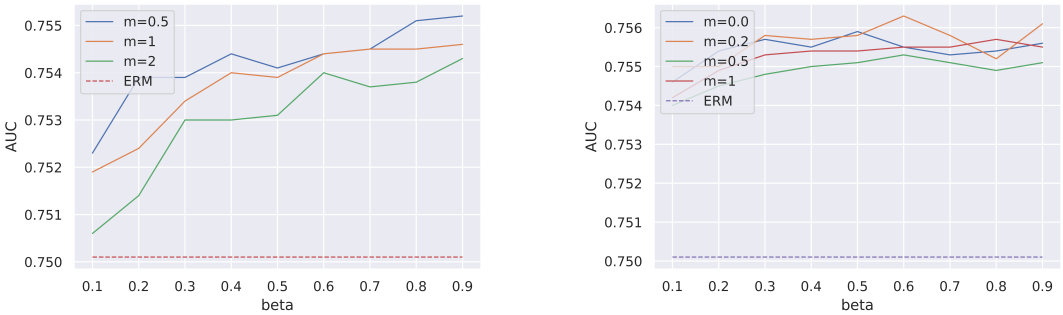
We compare the following networks :




(a) one-pass setting

(b) standard supervised setting

Figure 3: Hyperparameter selection on $\beta$ of point-wise confidence ranking. Average over 5 runs.




(a) hinge loss

(b) square loss

Figure 4: Hyperparameter selection on m for hinge loss and square loss on standard supervised setting. Average over 5 runs.

- DNN (Covington et al., 2016): An embedding + Deep neural network framework for recommendation system
- PNN (Qu et al., 2016): PNN uses product cross embedding followed by a DNN
- DCN (Wang et al., 2017): a parallel framework for combining cross network and deep neural network
- DeepFM (Guo et al., 2017): a parallel framework for combine factorization machine and deep neural network

### B.4 Implementation details

All of our experiments are evaluated with Adam(Kingma & Ba, 2014).

For avazu and avito, we initialize the learning rate as 0.001 and rate of weight decay as 0.00001. We don't use learning rate decay. For deep neural network part, we use 1024-512-256 for avazu and 512-256 for avito. We set embedding rank as 40 and 10 for avazu and avito respectively. We set batch size as 4096 and 10000 for avazu and avito respectively. Our optimization objective is defined as:

$$\ell = \alpha \ell_{ce} + \beta \ell_{cr} \tag{11}$$

We set $\alpha$ as 1 for all experiments while search the best $\beta$ for all methods.

### B.5 Ablation study

**Hyper-parameters search for $\beta$.** We provide ablation studies on the $\beta$ for various convex loss function of point-wise confidence ranking. For one-pass setting, We observe that the logistic loss function outperform all other losses. However, when we apply this loss function into model compression for image classification in Table 6 , it shows square loss outperforms other losses. For

standard supervised setting, we observe the square loss function also outperform all other losses. This interesting finding may reveal the benefit of *ad*-hoc loss function.

**Hyper-parameters search for $m$.** We provide ablation studies on the $m$ for hinge and square loss function of point-wise confidence ranking. In Table 4, we observe that the best $m$ is 0.5 and 0.2 respectively.

# C Training Details of supervised model compression

We follow the implementation of CRD(Tian et al., 2019):

## C.1 Baseline Methods

We compare the following state-of-the-art baseline KD methods:

- Knowledge Distillation (KD) (Hinton et al., 2015)
- Fitnets: Hints for thin deep nets (Romero et al., 2014)
- Attention Transfer (AT) (Zagoruyko & Komodakis, 2016a)
- Similarity-Preserving Knowledge Distillation (SP) (Tung & Mori, 2019)
- Correlation Congruence (CC) (Peng et al., 2019)
- Variational information distillation for knowledge transfer (VID) (Ahn et al., 2019)
- Relational Knowledge Distillation (RKD) (Park et al., 2019)
- Learning deep representations with probabilistic knowledge transfer (PKT) (Passalis & Tefas, 2018)
- Knowledge transfer via distillation of activation boundaries formed by hidden neurons (AB) (Heo et al., 2019)
- Paraphrasing complex network: Network compression via factor transfer (FT) (Kim et al., 2018)
- A gift from knowledge distillation: Fast optimization, network minimization and transfer learning (FSP) (Yim et al., 2017)
- Like what you like: Knowledge distill via neuron selectivity transfer (NST) (Huang & Wang, 2017)
- Contrastive representation distillation (CRD) (Tian et al., 2019)

## C.2 Network Architecture

- Wide Residual Network (WRN) (Zagoruyko & Komodakis, 2016b) WRN-d-w represnets wide resnet with depth d and width factor w.
- ResNet (He et al., 2016) We use resnet-d to represent cifar-style resnet with 3 groups of basic blocks, each with 16, 32, and 64 channels respectively. In our experiments, resnet8 x4 and resnet32 x4 indicate a 4 times wider network (namely, with 64, 128, and 256 channels for each of the block)
- MobileNetV2 (Sandler et al., 2018) In our experiments, we use a width multiplier of 0.5.
- vgg (Simonyan & Zisserman, 2014) the vgg net used in our experiments are adapted from its original ImageNet counterpart.
- ShuffleNetV1 (Zhang et al., 2018) and ShuffleNetV2 (Tan et al., 2019) .ShuffleNets are proposed for efficient training and we adapt them to input of size 32x32.

## C.3 implementation details

All methods evaluated in our experiments use SGD.

For CIFAR-100, we initialize the learning rate as 0.05, and decay it by 0.1 every 30 epochs after the first 150 epochs until the last 240 epoch. For MobileNetV2, ShuffleNetV1 and ShuffleNetV2, we use

Table 5: Abalation study of different activation function on point-wise confidence ranking. Average over 3 runs.

| Teacher<br>Student | WRN-40-2<br>WRN-16-2 | WRN-40-2<br>WRN-40-1 | resnet56<br>resnet20 | resnet110<br>resnet20 | resnet110<br>resnet32 | vgg13<br>vgg8 |
|---|---|---|---|---|---|---|
| softmax | 73.34 | 71.73 | 69.72 | 69.54 | 71.68 | 70.61 |
| sigmoid | 73.24 | 72.13 | 69.97 | 69.29 | 72.04 | 70.95 |

Table 6: Abalation study of different convex loss function on point-wise confidence ranking. Average over 3 runs.

| Teacher<br>Student | WRN-40-2<br>WRN-16-2 | WRN-40-2<br>WRN-40-1 | resnet56<br>resnet20 | resnet110<br>resnet20 | resnet110<br>resnet32 | vgg13<br>vgg8 |
|---|---|---|---|---|---|---|
| square | 75.68 | 74.36 | 70.94 | 70.95 | 73.44 | 74.16 |
| logistic | 73.24 | 72.13 | 69.97 | 69.29 | 72.04 | 70.95 |
| both | **75.79** | **74.53** | **71.59** | **71.32** | **73.44** | **73.62** |

a learning rate of 0.01 as this learning rate is optimal for these models in a grid search, while 0.05 is optimal for other models. Batch size is 64 for CIFAR-100.

The student is trained by a combination of cross-entropy classification objective and a knowledge distillation objective, shown as follows:

$$\ell = \alpha \ell_{ce} + \beta \ell_{distill} \tag{12}$$

For our proposed confidence ranking objection, we share the same combination as:

$$\ell = \alpha \ell_{ce} + \beta \ell_{cr} \tag{13}$$

where $\ell_{cr}$ is combination of point-wise confidence ranking loss and margin-based confidence ranking loss.

For the weight balance factor $\beta$, we directly use the optimal value from the original paper if it is specified, or do a grid search with teacher WRN-40-2 and student WRN-16-2. This results in the following list of $\beta$ used for different objectives:

- KD: $\alpha$ = 0.1; $\beta$ = 0.9 and T = 4
- Fitnets: $\alpha$ = 1; $\beta$ = 100
- SP: $\alpha$ = 1; $\beta$ = 3000
- CC: $\alpha$ = 1; $\beta$ = 0.02
- VID: $\alpha$ = 1; $\beta$ = 1
- RKD: $\alpha$ = 1; $\beta_1$ = 25 for distance and $\beta_2$ for angle. For this loss, we combine both term following the original paper
- PKT: $\alpha$ = 1; $\beta$ = 30000
- AB: $\alpha$ = 1; $\beta$ = 0, distillation happens in a separate pre-training stage where only distillation objective applies.
- FT: $\alpha$ = 1; $\beta$ = 500
- FSP: $\alpha$ = 1; $\beta$ = 0, distillation happens in a separate pre-training stage where only distillation objective applies.
- NST: $\alpha$ = 1; $\beta$ = 50
- CRD: $\alpha$ = 1; $\beta$ = 0.8
- CR: $\alpha$ = 1; $\beta_1$ = 1, $\beta_2$ = 0, we only use point-wise confidence ranking loss
- MRCR: $\alpha$ = 1; $\beta_1$ = 0, $\beta_2$ = 1, we only use margin-based confidence ranking loss

### C.4 ABLATION STUDY

**Sigmoid v.s. Softmax for Logistic loss** In Eq(6), the confidence ranking loss is implemented by logistic rank function. However, in multi-class setting, we can both use *sigmoid* and *softmax* as activation function. We provide ablation studies on the choice of the activation function on the supervised model compression. In Table 5, we observe *sigmoid* perform slightly better than *softmax*.

**Square loss v.s. Logistic loss** In this paper, we can use different surrogate convex loss function for confidence ranking. We provide ablation studies of square loss and logistic loss on model compression. We observe that using square loss outperforms logistic loss by a large margin. However, we find the combination of two losses can get further improvement.