Journal of Machine Learning Research 1 (2021) 1-31 Submitted 4/00; Published 10/00

# Adaptive Deep Learning for Entity Resolution by Risk Analysis

**Qun Chen** CHENBENBEN@NWPU.EDU.CN
**Zhaoqiang Chen** CHENZHAOQIANG@MAIL.NWPU.EDU.CN
**Youcef Nafa** YOUCEF.NAFA@MAIL.NWPU.EDU.CN
**Tianyi Duan** TIANYIDUAN@MAIL.NWPU.EDU.CN
**Zhanhuai Li** LIZHH@NWPU.EDU.CN
*School of Computer Science, Northwestern Polytechnical University*
*Key Laboratory of Big Data Storage and Management, Northwestern Polytechnical University, Ministry of Industry and Information Technology*
*Xi'an, China*

**Editor:** Kevin Murphy and Bernhard Schölkopf

## Abstract

The state-of-the-art performance on entity resolution (ER) has been achieved by deep learning. However, deep models are usually trained on large quantities of accurately labeled training data, and can not be easily tuned towards a target workload. Unfortunately, in real scenarios, there may not be sufficient labeled training data, and even worse, their distribution is usually more or less different from the target workload even when they come from the same domain.

To alleviate the said limitations, this paper proposes a novel risk-based approach to tune a deep model towards a target workload by its particular characteristics. Built on the recent advances on risk analysis for ER, the proposed approach first trains a deep model on labeled training data, and then fine-tunes it by minimizing its estimated misprediction risk on unlabeled target data. Our theoretical analysis shows that risk-based adaptive training can correct the label status of a mispredicted instance with a fairly good chance. We have also empirically validated the efficacy of the proposed approach on real benchmark data by a comparative study. Our extensive experiments show that it can considerably improve the performance of deep models. Furthermore, in the scenario of distribution misalignment, it can similarly outperform the state-of-the-art alternative of transfer learning by considerable margins. Using ER as a test case, we demonstrate that risk-based adaptive training is a promising approach potentially applicable to various challenging classification tasks.

**Keywords:** Risk Analysis, Deep Learning, Adaptation

## 1. Introduction

Extensively studied in the literature (Christen, 2008), ER is an important problem for data integration. Its purpose is to identify the equivalent records that refer to the same real-world entity. Considering the running example shown in Figure 1, ER needs to match the paper records between two tables, $R_1$ and $R_2$. A pair of $< r_{1i}, r_{2j} >$, in which $r_{1i}$ and $r_{2j}$ denote a record in $R_1$ and $R_2$ respectively, is called an *equivalent* pair if and only if $r_{1i}$ and $r_{2j}$ refer to the same paper; otherwise, it is called an *inequivalent* pair. In this example, $r_{11}$ and $r_{21}$ are

$R_1$

| ID | Title | Author | Venue | Year |
|---|---|---|---|---|
| $r_{11}$ | Parameter-Efficient Transfer Learning for NLP | Neil Houlsby, Andrei Giurgiu, et al. | ICML | 2019 |
| $r_{12}$ | Robust Unsupervised Feature Selection | Mingjie Qian, Chengxiang Zhai | IJCAI | 2013 |
| | ..... | | | |

$R_2$

| ID | Title | Author | Venue | Year |
|---|---|---|---|---|
| $r_{21}$ | Parameter-efficient transfer learning for NLP | Houlsby N, Giurgiu A, et al. | arXiv preprint | 2019 |
| $r_{22}$ | Partial multi-label learning | Xie, M. K., & Huang, S. J. | AAAI | 2018 |
| | ..... | | | |

Figure 1: An ER running example.

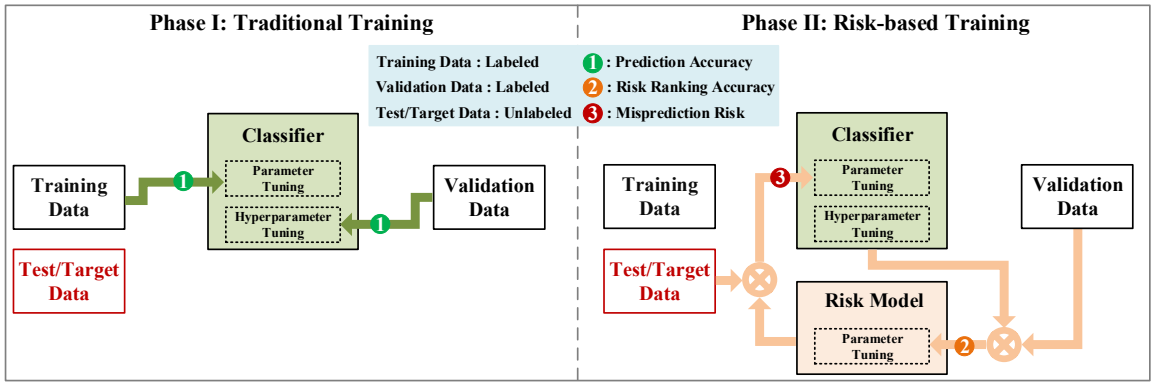



Figure 2: Risk-based Adaptive Training.

*equivalent* while $r_{11}$ and $r_{22}$ are *inequivalent*. ER can be considered as a binary classification problem tasked with labeling record pairs as *matching* or *unmatching*. Therefore, various learning models have been proposed for ER (Christen, 2008). As many other classification tasks (e.g. image and speech recognition), the state-of-the-art performance on ER has been achieved by deep learning (Ebraheem et al., 2018; Mudgal et al., 2018; Nie et al., 2019; Zhao and He, 2019; Li et al., 2021).

However, the efficacy of Deep Neural Network (DNN) models depends on large quantities of accurately labeled training data, which may not be readily available in real scenarios. Furthermore, in the typical setting of deep learning, model parameters are tuned on *labeled training data* in a way ensuring that the resulting classifier's predictions on the training instances are most consistent with their ground-truth labels. The trained classifier is then applied on a target workload. It can be observed that the typical process of model training does not involve the unlabeled data in a target workload, even though to alleviate the

over-fitting problem, labeled validation data are usually provided as the proxy workload of a target task and leveraged for hyperparameter tuning. Theoretically, the efficacy of this approach is based on the assumption that training and target data are **i**ndependently and **i**dentically **d**istributed (the **i.i.d** assumption). Unfortunately, in real scenarios, even when training and target data come from the same domain, the aforementioned assumption may not hold due to: 1) Training data are not sufficient to fully represent the statistical characteristics of a target workload; 2) Even though training data are abundant, its inherent distribution may be to some extent different from the target workload. Therefore, it is not unusual in real scenarios that a well trained deep model performs unsatisfactorily on a target workload.

Many adaptation approaches have been proposed to alleviate distribution misalignment, most notably among them *transfer learning* (Pan and Yang, 2010; Wei et al., 2018; Houlsby et al., 2019) and *adaptive representation learning* (Long et al., 2013, 2015; Zhao et al., 2019; Wu et al., 2019; Kim et al., 2019). Transfer learning aimed to adapt a model learned on the training data in a source domain to a target domain. Similarly, adaptive representation learning, which was originally proposed for image classification, mainly studied how to learn domain-invariant features shared among diversified domains. Unfortunately, distribution misalignment remains very challenging. One of the reasons is that the existing approaches focused on how to extract and leverage the common knowledge shared between a source task and a target task; however, they have limited capability to tune deep models towards a target task by its particular characteristics.

It has been well recognized that in real scenarios, with or without adaptation, a well-trained classifier may not be accurate in predicting the labels of the instances in a target workload. Even worse, it may provide high-confidence predictions which turn out to be wrong (Goodfellow et al., 2015). Such prediction uncertainty has emerged as a critical concern to AI safety (Amodei et al., 2016). Therefore, various techniques (Hendrycks and Gimpel, 2017; Chen et al., 2018; Jiang et al., 2018; Hendrycks et al., 2019; Chen et al., 2020) have been proposed for the task of risk analysis, which aims at estimating the misprediction risk of a classifier when applied to a certain workload.

Since risk analysis can measure the misprediction risk of a classifier on unlabeled data, it provides classifier training with a viable way to adapt towards a particular workload. Hence, we propose a risk-based approach to enable adaptive deep learning for ER in this paper. It is noteworthy that the recently proposed LearnRisk (Chen et al., 2020) is an interpretable and learnable risk analysis approach for ER. Compared with the simpler alternatives, LearnRisk is more interpretable and can identify mislabeled pairs with considerably higher accuracy. Therefore, we build the solution of adaptive deep training upon LearnRisk in this paper.

The proposed approach is shown in Figure 2, in which *test data* represent a target workload. It consists of two phases, the phase of *traditional training* followed by the phase of *risk-based training*. In the first phase, a deep model is trained on labeled training data in the traditional way. In the second phase, it is further tuned to minimize the misprediction risk on unlabeled target data. The main contributions of this paper are as follows:

- We propose a novel risk-based approach to enable adaptive deep learning.
- We present a solution of adaptive deep learning for ER based on the proposed approach.

- We theoretically analyze the performance of the proposed solution for ER. Our analysis shows that risk-based adaptive training can correct the label status of a mispredicted instance with a fairly good chance.
- We empirically validate the efficacy of the proposed approach on real benchmark data by a comparative study.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 presents the preliminaries. Section 4 presents the adaptive training approach and the theoretical results. Section 5 describes the empirical evaluation results. Finally, Section 6 concludes this paper.

## 2. Related Work

We review related work from three mutually orthogonal perspectives: entity resolution, model training and ensemble learning.

**Entity Resolution**. ER plays a key role in data integration and has been extensively studied in the literature (Christen, 2012; Christophides et al., 2015; Elmagarmid et al., 2007). ER can be automatically performed based on rules (Fan et al., 2009; Li et al., 2015; Singh et al., 2017), probabilistic theory (Fellegi and Sunter, 1969; Singla and Domingos, 2006) and machine learning models (Christen, 2008; Cochinwala et al., 2001; Kouki et al., 2017; Sarawagi and Bhamidipaty, 2002). The state-of-the-art performance on ER has been achieved by deep learning (Ebraheem et al., 2018; Mudgal et al., 2018; Nie et al., 2019; Zhao and He, 2019; Li et al., 2021). Specifically, a deep learning architecture template for ER has been provided in (Mudgal et al., 2018). More recently, in the following work, Li et al. (2021) presented an improved solution based on pre-trained Transformer-based language models, e.g., BERT. It is noteworthy that this paper does not attempt to propose a new DNN model for ER. It instead focuses on how to tune deep models towards a target task via risk analysis. Therefore, the existing work on deep learning for ER is orthogonal to ours. In principle, our proposed approach can work with any DNN for ER.

ER remains very challenging in real scenarios due to prevalence of dirty data. Therefore, there is a need for *risk analysis*, alternatively called *trust scoring* and *confidence ranking* in the literature. The proposed solutions ranged from those simply based on the model's output probabilities to more sophisticated interpretable and learnable ones (Zhang et al., 2014; Hendrycks and Gimpel, 2017; Jiang et al., 2018; Chen et al., 2020). Most recently, Chen et al. (2020) proposed an interpretable and learnable framework for ER, LearnRisk, which is able to construct a dynamic risk model tuned towards a specific workload. In this paper, we employ LearnRisk to adapt deep models toward a specific workload.

**Model Training**. A common challenge for model training is the *over-fitting*, which refers to the phenomenon that a model well tuned on training data performs unsatisfactorily on target data. The de-facto standard approach to alleviate *over-fitting* is by leveraging validation data for hyperparameter tuning and model selection (e.g. cross validation) (Kohavi et al., 1995). Another noteworthy complementary technique is the *regularization* (Evgeniou et al., 2000; Baldi and Sadowski, 2013; Neyshabur et al., 2017; Zhang et al., 2017), which aims to reduce the number of model parameters to a manageable level. Both hyperparameter

tuning and model selection are to a large extent orthogonal to model training considered in this paper. These works are therefore orthogonal to our work.

The classical way to alleviate the insufficiency of labeled training data, semi-supervised learning, has been extensively studied in the literature (Zhu, 2005; Iscen et al., 2019). However, semi-supervised learning investigated how to leverage unlabeled training data, which usually have a similar distribution with labeled training data. It is obvious that the techniques for semi-supervised learning can be straightforwardly incorporated into the pre-training phase of our proposed approach. They are therefore orthogonal to our work. Another way to reduce labeling cost is by active learning (Settles, 2012; Ducoffe and Precioso, 2018). While active learning focused on how to select training data for labeling, we focused on how to adapt a model towards a target workload provided with a set of training data. Therefore, active learning is also orthogonal to our work.

**Ensemble Learning.** To address the limitations of a single classifier, ensemble learning has been extensively studied in the literature (Zhou, 2009; Sagi and Rokach, 2018). Ensemble learning first trains multiple classifiers using different training data (e.g., bagging (Breiman, 1996)) or different information in the same training data (e.g., boosting (Schapire, 1990)), and then combine probably conflicting predictions to arrive at a final decision. While LearnRisk uses the ensemble of risk features to measure misprediction risk, risk-based adaptive training is fundamentally different from ensemble learning due to the following two reasons: 1) unlike the traditional labeling functions, *LearnRisk* aims to estimate an instance's misclassification risk as predicted by a classifier; 2) more importantly, the ensemble approach trains multiple models and tunes predictions based on training data; in contrast, risk-based adaptive training trains only one model, and tunes the model towards a target workload. On the other hand, since ensemble learning trains models based on training data, the work on ensemble learning is orthogonal to ours. In principle, our proposed approach can also work with an ensemble learning model. However, how to tune an ensemble model via risk measure however requires further investigation.

## 3. Preliminaries

In this section, we first define the ER classification task and then introduce the state-of-the-art risk analysis technique for ER, LearnRisk.

### 3.1 Task Statement

This paper considers ER as a binary classification problem. A classifier needs to label every unlabeled pair as *matching* or *unmatching*. As usual, we measure the quality of an ER solution by the standard metric of *F1*, which is a combination of *precision* and *recall* as follows

$$F1 = \frac{2 \times precision \times recall}{precision + recall}. \tag{1}$$

For presentation simplicity, we summarize the frequently used notations in Table 1. As usual, we suppose that an ER task $D$ consists a set of labeled training data, $D^s = \{(\mathbf{x}_i^s, y_i^s)|i\}$, where each $(\mathbf{x}_i^s, y_i^s)$ denotes a training instance with its feature representation $\mathbf{x}_i^s$ and ground-truth label $y_i^s$, a set of labeled validation data, $D^v = \{(\mathbf{x}_i^v, y_i^v)|i\}$, and a set

Table 1: The Frequently Used Notations

| Notation | Description |
|---|---|
| $D$ | an ER workload |
| $D^s, D^v, D^t$ | subsets of $D$, corresponding to training set, validation set and test set |
| $d_i$ | an instance pair in $D$ |
| $\mathbf{x}_i$ | the feature representation vector of $d_i$ |
| $y_i$ | the label of $d_i$ |
| $\mu_{d_i}$ | the expectation of equivalence probability of $d_i$ |
| $\sigma_{d_i}(resp.\sigma^2_{d_i})$ | the standard deviation (resp. variance) of equivalence probability of $d_i$ |
| $f_i$ | a risk feature |
| $w_i$ | the feature weight of $f_i$ |

of unlabeled test data, $D^t = \{(\mathbf{x}^t_i, ?)|i\}$. Note that $D^t$ denotes the target workload, and $D^v$ is provided as a proxy workload of $D^t$. Formally, we define the classification task of ER as

**Definition 1** ***[ER Classification Task]****. Given an ER workload $D$ consisting of $D^s$, $D^v$ and $D^t$, the task aims to learn an optimal classifier, $C_*$, based on $D$ such that the performance of $C_*$ on $D^t$ as measured by the metric of $F1$, or $F1(C_*, D^t)$, is maximized.*

### 3.2 Risk Analysis for ER: LearnRisk

The risk analysis pipeline of LearnRisk operates in three main steps: *risk feature generation* followed by *risk model construction* and finally *risk model training*.

#### 3.2.1 Risk feature generation

The step automatically generates risk features in the form of interpretable rules based on one-sided decision trees. The algorithm ensures that the resulting rule-set is both discriminative, i.e, each rule is highly indicative of one class label over the other; and has a high data coverage, i.e, its validity spans over a subpopulation of the workload. As opposed to classical settings where a rule is used to label pairs to be equivalent or inequivalent, a risk rule feature focuses exclusively on one single class. Consequently, risk features act as indicators of the cases where a classifier's prediction goes against the knowledge embedded in them. An example of such rules is:

$$r_i[Year] \neq r_j[Year] \rightarrow inequivalent(r_i, r_j), \tag{2}$$

where $r_i$ denotes a record and $r_i[Year]$ denotes $r_i$'s attribute value at $Year$. With this knowledge, a pair predicted as *matching* whose two records have different publication years is assumed to have a high risk of being mislabeled.

#### 3.2.2 Risk model construction

Once high-quality features have been generated, the latter are readily available for the risk model to make use of, allowing it to be able to judge a classifier's outputs backing up

its decisions with human-friendly explanations. To achieve this goal, LearnRisk, drawing inspiration from investment theory, models each pair's equivalence probability distribution (portfolio reward) as the aggregation of the distributions of its compositional features (stocks rewards).

Practically, the equivalence probability of a pair $d_i$ is modeled by a random variable $p_i$ that follows a normal distribution $\mathcal{N}(\mu_i, \sigma_i^2)$, where $\mu_i$ and $\sigma_i^2$ denote expectation and variance respectively. Given a set of $m$ risk features $f_1, f_2, ..., f_m$, let $w_1, w_2, ..., w_m$ denote their corresponding weights. Suppose that $\mu_F = [\mu_{f_1}, \mu_{f_2}, \ldots, \mu_{f_m}]^T$ and $\sigma_F^2 = [\sigma_{f_1}^2, \sigma_{f_2}^2, \ldots, \sigma_{f_m}^2]^T$ represent their corresponding expectation and variance vectors respectively, such that $\mathcal{N}(\mu_{f_j}, \sigma_{f_j}^2)$ denotes the equivalence probability distribution of the feature $f_j$. Accordingly, the distribution parameters for $d_i$ are estimated by:

$$\mu_i = \mathbf{z}_i(\mathbf{w} \circ \mu_F), \tag{3}$$

and

$$\sigma_i^2 = \mathbf{z}_i(\mathbf{w} \circ \mathbf{w} \circ \sigma_F^2), \tag{4}$$

Where $\circ$ represents the element-wise product and $\mathbf{z}_i$ is a one-hot feature vector.

*Note that besides one-sided decision rules, LearnRisk also incorporates classifier output as one of the risk features, which is referred to as the DNN risk feature in this paper.* Provided with the equivalence distribution $p_i$ for $d_i$, LearnRisk measures its risk by the metric of Value-at-Risk (VaR) (Tardivo, 2002), which can effectively capture the fluctuation risk of label status. Provided with a confidence level of $\theta$, the metric of VaR represents the maximum loss after excluding all worse outcomes whose combined probability is at most 1-$\theta$.

#### 3.2.3 Risk model training

Finally, the risk model is trained on labeled validation data to optimize a learn-to-rank objective by tuning the risk feature weight parameters ($w_i$) as well as their variances ($\sigma_i^2$). As for their expectations ($\mu_i$), they are considered as prior knowledge, and estimated from labeled training data. Once trained, the risk model can be used to assess the misclassification risk on an unseen workload labeled by a classifier.

## 4. Risk-based Adaptive Training

In this section, we propose, and then analyze the approach of risk-based adaptive training for ER. We take *DeepMatcher* (Mudgal et al., 2018), an classical DNN solution for ER, as an example to illustrate the solution. However, it is worthy to point out that in principle, the proposed approach can work with any DNN classifier; the implementation of the proposed solution on other DNNs is similar. In our empirical evaluation, we have implemented the proposed solution on both *deepmatcher* and *Ditto* (Li et al., 2021), the most recent DNN for ER based on Transformer-based language models. For comparison, we also briefly describe the traditional training approach.

### 4.1 Traditional Training

Given a workload consisting of $D^s$, $D^v$ and $D^t$, let $g(\omega)$ denote a DNN classifier with the parameters of $\omega$. The traditional approach, as shown in the left part of Figure 2, tunes $\omega$ towards the training data, $D^s$, based on a pre-specified loss function. Suppose that there are totally $n_s$ training instances in $D^s$. DeepMatcher employs the classical cross-entropy loss function to guide the process of parameter optimization as follows

$$\mathcal{L}_{train}(\omega) = \frac{1}{n_s}\sum_{i=1}^{n_s}\{ - y_i^s log(g(\mathbf{x}_i^s;\omega)) - (1-y_i^s)log(1-g(\mathbf{x}_i^s;\omega))\}, \tag{5}$$

where $y_i^s$ denotes the ground-truth label of a training instance, $(\mathbf{x}_i^s, y_i^s)$, and $g(\mathbf{x}_i^s;\omega)$ denotes its label probability as predicted by the classifier. DeepMatcher uses the Adam optimizer (Kingma and Ba, 2015) to search for the optimal parameters $\omega_*$ by gradient descent.

### 4.2 Risk-based Adaptive Training

Risk-based adaptive training, as shown in Figure 2, consists of two phases, a *traditional training* phase followed by an *risk-based training* phase. In the first phase, it tunes a deep model towards training data in the traditional way. Then, in the following *risk-based training* phase, it iteratively performs: i) using LearnRisk to learn a risk model based on a trained classifier and validation data; ii) fine-tuning the classifier by minimizing its misprediction risk upon the target workload.

Specifically, the loss function of risk-based training is defined as

$$\mathcal{L}_{test}^{risk}(\omega) = \frac{1}{n_t}\sum_{i=1}^{n_t}\{ - [1-VaR^+(d_i)]log(g(\mathbf{x}_i^t;\omega)) - [1-VaR^-(d_i)]log(1-g(\mathbf{x}_i^t;\omega))\}, \tag{6}$$

in which $n_t$ denotes the total number of instances in $D^t$, $VaR^+(d_i)$ (resp. $VaR^-(d_i)$) denotes the estimated misprediction risk of $d_i$ if it is labeled as *matching* (resp. *unmatching*).

Similar to traditional training, the *risk-based training* phase updates the parameters of the deep model by gradient descent as follows

$$\omega_{k+1} = \omega_k - \alpha * \nabla_{\omega_k}\mathcal{L}_{test}^{risk}(\omega). \tag{7}$$

Note that in each iteration, risk values are estimated based on the classifier predictions of the previous iteration. As a result, they are considered as constant while computing gradient descent .

We have sketched the process of risk-based adaptive training in Algorithm 1. The first phase pre-trains a model based on labeled training data, and selects the best one based on its performance on the validation data. Beginning with the pre-trained model, the second phase iteratively fine-tunes the parameters by minimizing the loss of $\mathcal{L}_{test}^{risk}(\omega)$.

**Algorithm 1** Risk-based Adaptive Training

**Input:** A task $D$ consisting of $D^s$, $D^v$ and $D^t$, and an ER model, $g(\omega)$;
**Output:** A learned classifier $g(\omega_*)$.
$\omega_0 \leftarrow$Initialize $\omega$ with random values;
**for** $k = 0$ **to** $m - 1$ **do**
  $\omega_{k+1} \leftarrow \omega_k - \alpha * \nabla_{\omega_k} \mathcal{L}_{train}(\omega_k)$;
**end for**
Select the best model, $g(\omega_*)$, based on $D^v$;
$\omega_m \leftarrow \omega_*$;
**for** $k = m$ **to** $m + n - 1$ **do**
  Update the risk model based on $D^v$ and $g(\omega_k)$;
  $\omega_{k+1} \leftarrow \omega_k - \alpha * \nabla_{\omega_k} \mathcal{L}^{risk}_{test}(\omega_k)$;
**end for**
Select the best model, $g(\omega_*)$, based on $D^v$.
**Return** $g(\omega_*)$

### 4.3 Theoretical Analysis

Empirically, it is widely observed that DNNs are highly expressive leading to very low training errors provided with correct information. It can also be observed that given an estimated distribution of equivalence probability, $(\mu_i,\sigma_i^2)$, for a pair $d_i$, $VaR^+(d_i) < VaR^-(d_i)$ if and only if $\mu_i > 0.5$. Hence, according the loss function defined in Eq.6, for an equivalent pair of $d_i$ in $D^t$, $\mu_i > 0.5$ would result in it being correctly classified. Similarly, if $d_i$ is an inequivalent pair, $\mu_i < 0.5$ would result in it being correctly classified.

Suppose that *LearnRisk* generates totally $m$ risk features, denoted by $\{f_1,\ldots,f_m\}$. Let $Z_i$ be a 0-1 variable indicating whether an instance has the risk feature $f_i$: $Z_i = 1$ if the instance has $f_i$, otherwise $Z_i = 0$. Let $\mathbf{Z} = (Z_1, Z_2, ..., Z_m)$ denote a risk feature distribution. We can reasonably expect that LearnRisk is generally effective: if an instance is equivalent (resp. inequivalent), its risk features (excluding its DNN output) are supposed to indicate its equivalence (resp. inequivalence) status. As shown in Eq.3, LearnRisk estimates the equivalence probability expectation of an instance by a weighted linear combination of the expectations of its DNN risk feature and rule risk features. Specifically, given an equivalent pair of $d_i$, $(\mu_i,\sigma_i^2)$, with $m$ rule risk features, we have

$$\mathbb{E}\left(\frac{\sum_{j=1}^{m} z_j \cdot w_j \cdot \mu_{f_j}}{\sum_{j=1}^{m} z_j \cdot w_j}\right) > 0.5, \tag{8}$$

in which $f_j$ denotes a rule risk feature, and $\mathbb{E}(*)$ denotes the statistical expectation. Similarly, if $d_i$ is inquivalent, it satisfies

$$\mathbb{E}\left(\frac{\sum_{j=1}^{m} z_j \cdot w_j \cdot \mu_{f_j}}{\sum_{j=1}^{m} z_j \cdot w_j}\right) < 0.5. \tag{9}$$

According to Eq. 8 and 9, once a pair is correctly labeled by a classifier, it can be expected that its label would not be flipped by risk-based fine-tuning. Our experiments on real data have confirmed that risk-based fine-tuning rarely flips the labels of true positives

and true negatives. Therefore, in the rest of this subsection, we focus on showing that given a mispredicted instance, risk-based fine-tuning can result in a value of $\mu_i$ consistent with its ground-truth label with a fairly good chance.

For theoretical analysis, since both true positives and false negatives (resp. true negatives and false postives) are equivalent (resp. inequivalent) instances, they are assumed to share the same distribution of risk feature activation. Formally, we state the assumption on risk feature distribution as follows:

**Assumption 1 Identicalness of Risk Feature Distributions.** *Given an ER workload $D^t$, the risk feature activation of each equivalent instance $d_i^+$, denoted by $\mathbf{Z}_i^+$, is supposed to follow the same distribution of $\mathbf{Z}^+$; similarly, the risk feature activation of each inequivalent instance $d_i^-$, denoted by $\mathbf{Z}_i^-$, is supposed to follow the same distribution of $\mathbf{Z}^-$.*

Based on Assumption 1, we can establish the lower bound of the estimated equivalence probability expectation of a false negative by the following theorem:

**Theorem 2** *Given a false negative $\tilde{d_j^-}$, suppose that there are totally $n$ true positives, denoted by $d_i^+$, ranked after $\tilde{d_j^-}$ by LearnRisk such that each true positive, $d_i^+$, satisfies*

$$\Delta VaR^- - \Delta C^- > \epsilon, \tag{10}$$

*in which $\Delta VaR^- = VaR^-(\tilde{d_j^-}) - VaR^+(d_i^+)$, and $\Delta C^- = \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{\tilde{d_j^-}} - 2\sigma_{\tilde{d_j^-}})$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$, its expectation of equivalence probability of $\tilde{d_j^-}$, $\mu_{\tilde{d_j^-}}$, estimated by* LearnRisk*, satisfies*

$$\mu_{\tilde{d_j^-}} \geq \frac{1}{2} + \frac{\epsilon}{2} - \sqrt{\frac{m+1}{2} ln[\frac{1}{1-(1-\delta^{\frac{1}{n}})^{\frac{1}{2}}}]}, \tag{11}$$

*in which $\mu_*$ denotes the expectation of equivalence probability and $\sigma_*$ denotes its standard deviation.*

**Proof** The proofs can be found in Appendix A.1. ■

In Theorem 2, $m$ denotes the number of rule risk features, and the value of $\Delta C^-$ corresponds to the difference of risk expectation between false negatives being labeled as *matching* and true positives being labeled as *matching*. Note that the total number of rule risk features ($m$) is usually limited (e.g., dozens or hundreds), while $n$ is usually much larger than $m$. It can be observed that in Theorem 2, by the exponential effect of $n$, the 3rd term on the right-hand side tends to become zero as the value of $n$ increases. Therefore, if $\epsilon > 0$ and there are sufficient true positives satisfying the specified condition, risk-based fine-tuning would have a fairly good chance to correctly flip the label of $\tilde{d_j^-}$ from *unmatching* to *matching*. To gain deeper insight into Theorem 2, we analyze the value of $\Delta VaR^- - \Delta C^-$. Since the optimization objective of LearnRisk is to maximize the risk difference between $VaR^-(\tilde{d_j^-})$ and $VaR^+(d_i^+)$, $\Delta VaR^-$ can be expected to large for most true positives. Therefore, we analyze the value of $\Delta C^-$. Based on Assumption 1, we have the following lemma:

**Lemma 3**

$$\begin{aligned}\Delta C^- \leq & max\big\{\mathbb{E}(w_{d_i^+}(\hat{\mu}_{d_i^+} - 2\hat{\sigma}_{d_i^+})) - \mathbb{E}(w_{\tilde{d}_j^-}(\hat{\mu}_{\tilde{d}_j^-} - 2\hat{\sigma}_{\tilde{d}_j^-})),\\ & \mathbb{E}(w_{d_i^+}\hat{\mu}_{d_i^+}) - \mathbb{E}(w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-})\big\},\end{aligned} \tag{12}$$

*where the $\hat{\mu}_*$ and $\hat{\sigma}_*$ denote the DNN output probability and its corresponding standard deviation respectively, $w_*$ denotes the learned weight of DNN risk feature.*

**Proof** The proofs can be found in Appendix A.2. ■

It is interesting to point out that as shown in Lemma 3, the value of $\Delta C^-$ only depends on the distributions of DNN outputs and their weights, but independent of the distributions of rule risk features. It has the simple upper bound of

$$\Delta C^- \leq \mathbb{E}(w_{d_i^+}). \tag{13}$$

Hence, when the learned weight of DNN output becomes smaller, which means that the DNN becomes less accurate, true positives would have a higher chance to satisfy $\Delta VaR^- - \Delta C^- > 0$. In our experiments, it is observed that the expected weight of classifier output is usually between 0.2 and 0.6, or $0.2 \leq \mathbb{E}(w_{d_i^+}) \leq 0.6$. As a result, Theorem 2 shows that a false negative has a fairly good chance to be flipped from *unmatching* to *matching*.

Based on Assumption 1, the theoretical chance of a false positive being flipped from *matching* to *unmatching* can be similarly established. The corresponding theorem and lemma are presented as follows.

**Theorem 4** *Given a false positive $\tilde{d}_j^+$, suppose that there are totally $n$ true negatives, denoted by $d_i^-$, ranked after $\tilde{d}_j^+$ by LearnRisk such that each true negative, $d_i^-$, satisfies*

$$\Delta VaR^+ - \Delta C^+ > \epsilon, \tag{14}$$

*in which $\Delta VaR^+ = VaR^+(\tilde{d}_j^+) - VaR^-(d_i^-)$, and $\Delta C^+ = \mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-})$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$, its expectation of equivalence probability of $\tilde{d}_j^+$, $\mu_{\tilde{d}_j^+}$ estimated by* LearnRisk*, satisfies*

$$\mu_{\tilde{d}_j^+} \leq \frac{1}{2} - \frac{\epsilon}{2} + \sqrt{\frac{m+1}{2} ln[\frac{1}{1-(1-\delta^{\frac{1}{n}})^{\frac{1}{2}}}]},$$

*in which $\mu_*$ denotes the mean of equivalence probability and $\sigma_*$ denotes its standard deviation.*

**Proof** The proofs can be found in Appendix A.3. ■

In Theorem 4, the value of $\Delta C^+$ corresponds to the difference of risk expectation between false positives being labeled as *unmatching* and true negatives being labeled as *unmatching*. Similar to the case of Theorem 2, it can be observed that in Theorem 4, by the exponential

effect of $n$, the 3rd term on the right-hand side tends to become zero as the value of $n$ increases. Therefore, if $\epsilon > 0$ and there are sufficient true negatives satisfying the specified condition, risk-based fine-tuning would have a fairly good chance to correctly flip the label of $\tilde{d}_j^+$ from *matching* to *unmatching*.

To gain a deeper insight into Theorem 4, we also analyze the value of $\Delta C^+$ by the following lemma:

**Lemma 5**

$$\Delta C^+ \leq max\big\{\mathbb{E}(w_{\tilde{d}_j^+}(\hat{\mu}_{\tilde{d}_j^+} + 2\hat{\sigma}_{\tilde{d}_j^+})) - \mathbb{E}(w_{d_i^-}(\hat{\mu}_{d_i^-} + 2\hat{\sigma}_{d_i^-})), \mathbb{E}(w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+}) - \mathbb{E}(w_{d_i^-}\hat{\mu}_{d_i^-})\big\}, \tag{15}$$

*where the $\hat{\mu}_*$ and $\hat{\sigma}_*$ denote the DNN output probability and its corresponding standard deviation respectively, $w_*$ denotes the learned weight of DNN risk feature.*

**Proof** The proofs can be found in Appendix A.4. ■

Similarly, as shown in Lemma 5, the value of $\Delta C^+$ has an upper bound constrained by the weights of DNN outputs. The true negatives would tend to satisfy $\Delta VaR^+ - \Delta C^+ > 0$ in the case that the trained DNN model becomes less accurate. Hence, Theorem 4 shows that a false positive has a fairly good chance to be flipped form *matching* to *unmatching*.

**Empirical Validation.** We have illustrated the efficacy of theoretical analysis on the real literature dataset of DBLP-ACM[1] The results on the first iteration of risk-based fine-tuning are presented in Table 2, in which the false negatives (resp. false positives) are clustered according to the size of true positives (resp. true negatives) that meet the specified condition. It can be observed: 1) risk-based fine-tuning rarely flips the labels of true positives and true negatives; 2) the majority of false negatives (resp. false positives) have a large number (e.g. $\geq 100$) of corresponding true positives (resp. true negatives), and most of them are correctly flipped.

## 5. Experiments

In this section, we empirically evaluate the proposed approach on real benchmark datasets by a comparative study. We first describe the experimental setting. Then, we present the comparative evaluation results. Finally, we evaluate robustness of the proposed approach w.r.t the size of validation data.

### 5.1 Experimental Setup

We have used six real datasets from three domains in our empirical study:

- **Publication**. The datasets in this domain contain bibliographic data from different sources, i.e. DBLP, Google Scholar and ACM. As in Mudgal et al. (2018), we used

1. https://github.com/anhaidgroup/deepmatcher/.

Table 2: Empirical Validation of Theoretical Analysis.

(a) on True Positives and True Negatives

| | # | # Flipped |
|---|---|---|
| True Positives | 1143 | 12 |
| True Negatives | 5987 | 27 |

(b) on False Negatives and False Positives

| $\#(\Delta VaR^- > \Delta C^-$ or $\Delta VaR^+ > \Delta C^+)$ | False Negatives | | False Positives | |
|---|---|---|---|---|
| | # | # Flipped | # | # Flipped |
| #< 100 | 1 | 0 | 0 | 0 |
| #≥ 100 | 188 | 179 | 100 | 91 |
| Total | 189 | 179 | 100 | 91 |

DBLP-Scholar[2] (denoted by DS) and DBLP-ACM [2] (denoted by DA). Additionally, we used the Cora dataset[3], which contains the citation data obtained from the Cora search engine;

- **Music**. In this domain, we used the Itunes-Amazon dataset (denoted by IA) provided by Mudgal et al. (2018). The size of IA is relatively small, containing only 539 pairs. Additionally, we used the Songs dataset[4] (denoted by SG), which contains song records; the experiments match the entries within the same table;

- **Product**. In this domain, we used a dataset containing the electronics product pairs extracted from the Abt.com and Buy.com [2]. We denote this dataset by AB.

Table 3: The statistics of datasets.

| Dataset | Size | # Matches | # Attributes |
|---|---|---|---|
| DS | 28,707 | 5,347 | 4 |
| DA | 12,363 | 2,220 | 4 |
| Cora | 12,674 | 3,268 | 12 |
| AB | 9,575 | 1,028 | 3 |
| IA | 539 | 132 | 8 |
| SG | 19,633 | 6,108 | 7 |

As usual, on all the datasets, we used the blocking technique to filter the pairs deemed unlikely to match. The datasets of DS, DA, AB and IA have been made online available at [2]. On both Cora and SG, we first filtered the pairs and then randomly selected a proportion

2. https://github.com/anhaidgroup/deepmatcher/
3. http://www.cs.utexas.edu/users/ml/riddle/data/cora.tar.gz
4. http://pages.cs.wisc.edu/ãnhai/data/falcon_data/songs/

of the resulting candidates to generate the workloads. The statistics of the test datasets are given in Table 3.

We primarily used DeepMatcher (Mudgal et al., 2018), the classical deep learning solution for ER, as the classifier. We evaluate the proposed approach both scenarios where training and test data come from the same source and they come from different sources, thus resulting in more distribution misalignment. In the scenario where training and test data come from the same source, we randomly split each dataset into three parts by a ratio (e.g. 2:2:6 in our experiments) as in Mudgal et al. (2018), which specifies the proportions of training, validation and test set respectively. We evaluate the performance of the proposed approach w.r.t different sufficiency levels of training data. Since DeepMatcher performs very well on DA and SG with only 20% of their data as training data, we randomly select 10%, 30%, 50%, 70% and 100% of the split set of training data to simulate different sufficiency levels. On AB, we fix the proportion of validation data at 20% and vary the proportions of training and test data from (60%,20%) to (20%,60%), resulting in totally 5 sufficiency levels. In this scenario, since training and target data are randomly selected from the same source, we compare the *Risk* approach with the original DeepMatcher, which is denoted by *Tradition*.

In the scenario of distribution misalignment, we used the three datasets in the domain of *publication* (i.e., DS, DA and Cora) to generate six pairwise workloads. For instance, DA2DS denotes the workload where training data come from DA while validation data and test data come from DS. On all the workloads, validation and test data are randomly selected from the original target dataset with both percentages set at 20%. In this scenario, besides *Tradition*, we also compare *Risk* with the state-of-the-art technique of transfer learning for ER proposed in Kasai et al. (2019). We denote this approach by *Transfer*. It inserted a dataset classifier into the DeepMatcher structure, which can force a deep model to focus on the parameters that are shared by both training and test data.

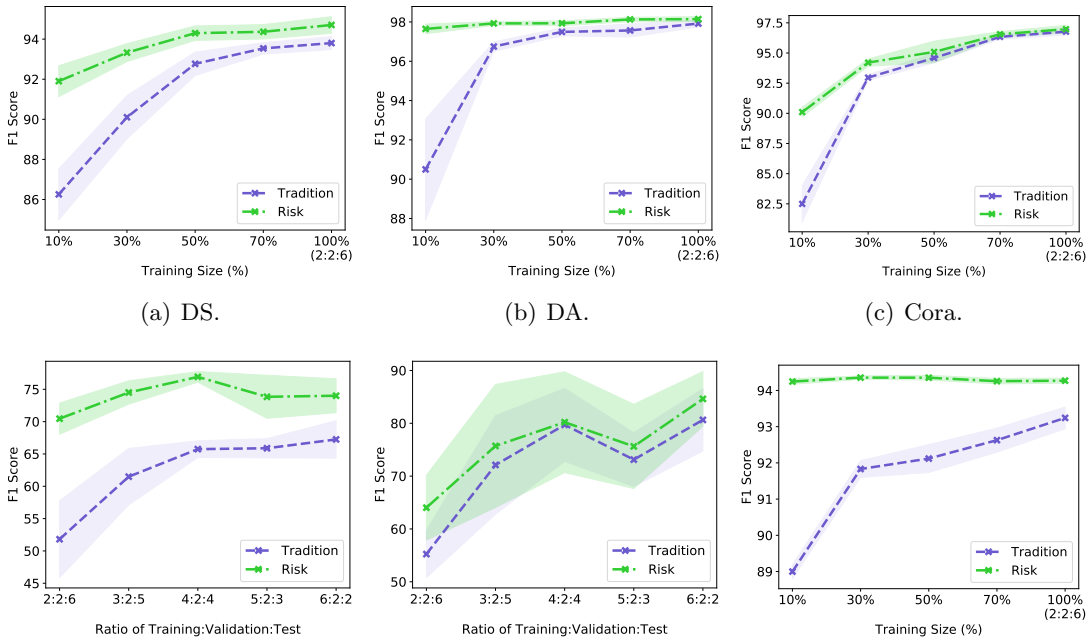
We have implemented the proposed solution on *DeepMatcher* by replacing the original loss function with the risk-based loss function. To overcome the randomness caused by model initialization and training data shuffling, on each experiment, we perform 5 training sessions and report their mean F1-score on test data. In *Tradition* and *Transfer*, each training session consists of 20 iterations; in *Risk*, the traditional training phase consists of 20 iterations and the risk-based training phase consists of 10 iterations. Our experiments showed that further increasing the number of iterations in each session had very marginal impact on performance.

Additionally, we have also implemented and evaluated the proposed solution on *Ditto* (Li et al., 2021), which is the state-of-the-art DNN for ER based on pre-trained Transformer-based language models. Note that compared with *DeepMatcher*, *Ditto* generally performs better and can perform well with less training data. Therefore, on AB and IA, our experiments begins with the ratio of training data at 10%. Similarly, we perform 5 training sessions and report their mean F1-score on test data to overcome the randomness. We use the default parameters of *Ditto* for both traditional training and adaptive training, except that the learning rate of risk-based training is set to be $3 * 10^{-6}$, instead of the default

$3 * 10^{-5}$. Our implementations on both *DeepMatcher* and *Ditto* and the test data have been made open-source at our website [5].

## 5.2 Comparative Evaluation on DeepMatcher

### 5.2.1 Same-source Scenario




(a) DS. (b) DA. (c) Cora.

(d) AB. (e) IA. (f) SG.

Figure 3: Comparative Evaluation with Deepmatcher: the Same-Source Scenario.

The comparative results are presented in Figure 3, in which we report both the mean of F1 score and its standard deviation (represented by the shadow in the figure). It can be observed that *Risk* achieves consistently better performance than *Tradition*. In the circumstances where *Tradition* performs unsatisfactorily (e.g. AB and IA), the performance margins between *Risk* and *Tradition* are very considerable. For instance, on AB, with the ratio of (2, 2, 6), *Risk* achieves a performance improvement close to 20% over *Tradition* (70% vs 51%).

In the circumstances where *Tradition* can perform well (e.g. DS, DA, Cora and SG) with the ratio of (2,2,6), it can be observed that the performance margins between *Risk* and *Tradition* are similarly considerable when training data are insufficient. For instance, on DS, with 10 percent of the training data, *Risk* outperforms *Tradition* by more than 6% and achieves the F1 score of more than 92%. In particular, on DA and SG, with only 10% of the training data, *Risk* achieves the performance very similar to what is achieved by using 100% of the training data. As the size of training data increases, the margins between *Risk* and *Tradition* tend to decrease. This trend can be expected, because when

5. https://chenbenben.org/adaptive-training.html

Table 4: Comparative Evaluation: Distribution Misalignment.

| Dataset | F1 Score (Mean ± Standard deviation) | | |
|---|---|---|---|
| | Baseline | Transfer | Risk |
| DA2DS | $19.86 \pm 5.12$ | $43.81 \pm 11.88$ | **91.67±0.56** |
| DA2Cora | $76.47 \pm 4.59$ | $74.86 \pm 3.70$ | **89.08±0.81** |
| Cora2DS | $55.81 \pm 5.90$ | $62.08 \pm 6.65$ | **86.55±1.34** |
| Cora2DA | $71.28 \pm 5.23$ | $72.92 \pm 7.57$ | **96.99±0.34** |
| DS2DA | $93.08 \pm 1.71$ | $93.50 \pm 1.49$ | **94.18±0.97** |
| DS2Cora | $83.11 \pm 1.81$ | $82.48 \pm 3.52$ | **84.88±0.29** |

training and test data are randomly selected from the same source, more training data mean less improvement potential for risk-based fine-tuning. Due to the small size of IA, the comparative results on IA have higher randomness compared with other datasets.

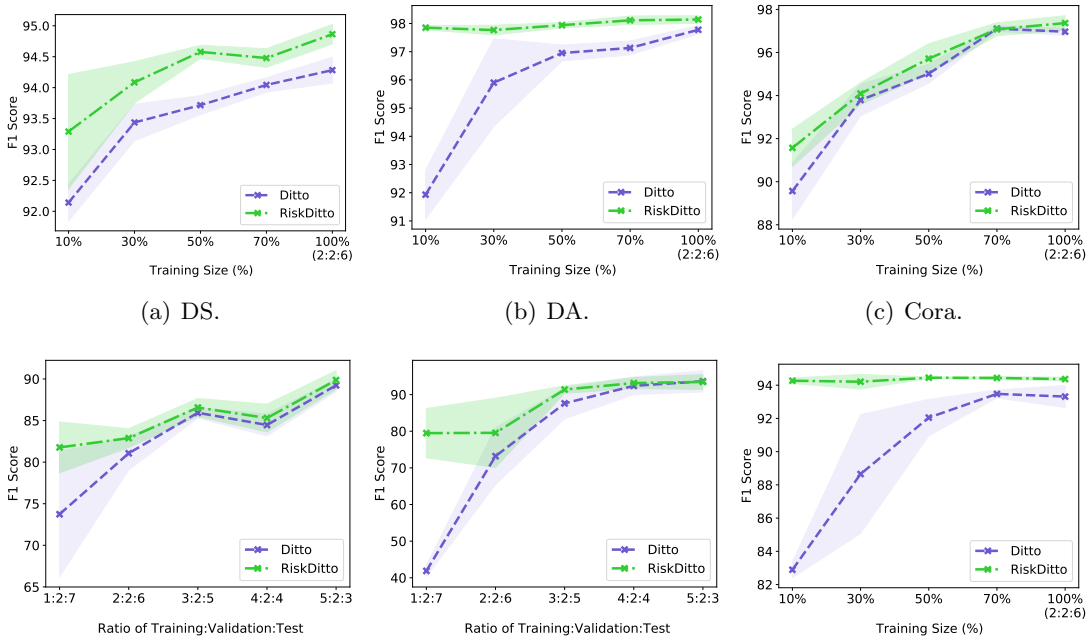
#### 5.2.2 Scenario of Distribution Misalignment

The comparative results are presented in Table 4, in which the best results have been highlighted. It can be observed that: 1) The performance of *Tradition* deteriorates significantly in most testbeds; 2) the performance of *Transfer* also fluctuates wildly across the test workloads. As shown on DA2CORA and CORA2DA, its impact becomes very marginal or even negative when *Tradition* performs decently; 3) *Risk* consistently outperforms both *Tradition* and *Transfer*, and the margins are very considerable in most cases.

We further explain the efficacy of the risk-based approach by illustrative examples. On DA2DS, we observed that the model trained on DA performs very poorly (only around 20%) on the target workload of DS. This is mainly due to the fact that DS is more challenging than DA, and the data distribution of DA to a large extent fails to reflect the more complicated distribution of DS. In contrast, LearnRisk can reliably identify the mispredictions of the pre-trained model on DS. We observed that in the first iteration of risk-based fine-tuning, it correctly identifies totally 877 mispredictions among the top 1000 risky pairs, most of which are later correctly flipped. However, on the workloads (e.g. DS2DA and DS2CORA) where *Tradition* performs well, the advantage of *Risk* over *Tradition* becomes less significant. This result should be no surprise because in this circumstance, risk analysis becomes more challenging.

### 5.3 Comparative Evaluation on Ditto

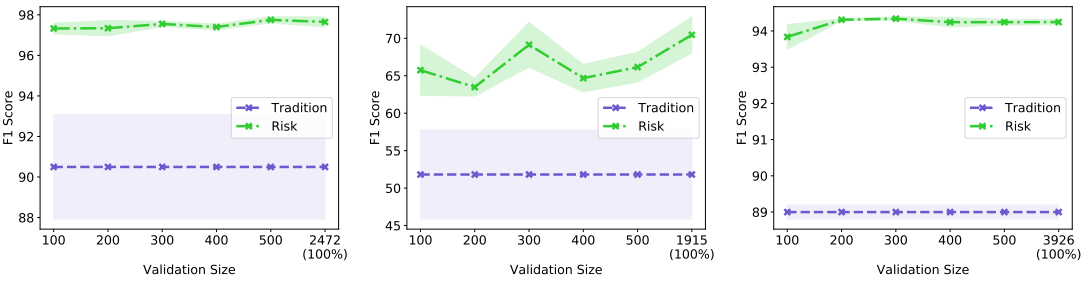
The comparative results on *Ditto* are presented in Figure 4. It can be observed that *Ditto* generally performs better than *Deepmatcher* with the most improvements on AB. On AB, with the ratio setting of (2, 2, 6), the F1 score of *Ditto* is 81% while *Deepmatcher* can only achieve 51%. Similar to what have been observed on *DeepMatcher*, risk-based fine-tuning effectively improves the performance of *Ditto* even though it is a better baseline. For instance, on SG, with the percentage of training data at 10% and 30%, the performance improvements measured by F1 are 11% and 6% respectively. The evaluation results on *Ditto*




(a) DS. (b) DA. (c) Cora.

(d) AB. (e) IA. (f) SG.

Figure 4: Comparative Evaluation with Ditto: the Same-Source Scenario.




(a) DA. (b) AB. (c) SG.

Figure 5: Robustness Evaluation.

clearly demonstrate that the proposed approach of risk-based adaptive training is generally applicable to various DNN models.

### 5.4 Robustness w.r.t Size of Validation Data

In real scenarios, validation data are necessary for hyperparameter tuning and model selection to ensure that a trained model can generalize well. However, due to labeling cost, it is usually desirable to reduce the size of validation data. Since risk analysis leverages validation data for risk model learning, we evaluate the performance robustness of the proposed approach w.r.t the size of validation data.

To this end, we fixed the sets of training and test data at 20% and 60% respectively, and varied the size of validation data by randomly selecting part of instances from the split set of validation data. The results on the DA, AB and SG workloads are presented in Figure 5, in which the performance of *Tradition* and *Risk* with the whole validation data are also included for reference. The evaluation for DA and SG is based on the setting that 10% of the split set of training data is used. It can be observed that with as few as 100 validation instances, *Risk* is able to improve classifier performance by considerable margins. Our evaluation results are consistent with those reported in Chen et al. (2020), which showed that the performance of LearnRisk is very robust w.r.t the size of validation data. These experimental results bode well for the application of the proposed approach in real scenarios.

## 6. Conclusion

In this paper, we have proposed a risk-based approach to enable adaptive deep learning for ER. It can effectively tune deep models towards a target workload by its particular characteristics. Both theoretical analysis and empirical study have validated its efficacy. For future work, it is worthy to point out that the proposed approach is generally applicable to other classification tasks; their technical solutions however need further investigations.

## Appendix A. Theoretical Analysis

In theoretical analysis, for simplicity of presentation, without loss of generality, we suppose that *LearnRisk* sets the confidence value at $\theta = 0.975$. Hence, given a pair $d_i$ with the equivalence probability distribution of $\mathcal{N}(\mu_i, \sigma_i^2)$, its VaR risk is equal to $1-(\mu_i - 2\sigma_i)$ if it is labeled as *matching* by a classifier, and its VaR risk is equal to $\mu_i + 2\sigma_i$ if it is labeled as *unmatching*.

### A.1 Proof of Theorem 2

**Theorem 2** *Given a false negative $\tilde{d_j^-}$, suppose that there are totally n true positives, denoted by $d_i^+$, ranked after $\tilde{d_j^-}$ by LearnRisk such that each true positive, $d_i^+$, satisfies*

$$\Delta VaR^- - \Delta C^- > \epsilon, \tag{16}$$

*in which $\Delta VaR^- = VaR^-(\tilde{d_j^-}) - VaR^+(d_i^+)$, and $\Delta C^- = \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{\tilde{d_j^-}} - 2\sigma_{\tilde{d_j^-}})$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$, its expectation of equivalence probability of $\tilde{d_j^-}$, $\mu_{\tilde{d_j^-}}$, estimated by* LearnRisk, *satisfies*

$$\mu_{\tilde{d_j^-}} \geq \frac{1}{2} + \frac{\epsilon}{2} - \sqrt{\frac{m+1}{2} ln[\frac{1}{1-(1-\delta^{\frac{1}{n}})^{\frac{1}{2}}}]},$$

*in which $\mu_*$ denotes the mean of equivalence probability and $\sigma_*$ denotes its standard deviation.*

Note that in Theorem 2, $m$ denotes the number of rule risk features and $\Delta C^-$ denotes the difference of risk expectation between false negatives being labeled as *matching* and true positives being labeled as *matching*. In the rest of this section, we first prove a lemma that states the concentration inequalities of VaR risk functions, and then prove Theorem 2 based on the lemma.

**Lemma 6** *Given a randomly selected pair $d_i^+$ from true positives, whose equivalence probability distribution estimated by* LearnRisk *is denoted by $\mathcal{N}(\mu_{d_i^+}, \sigma_{d_i^+})$, for any $\delta \in (0,1)$, with probability at least $(1-\delta)$, the following inequality holds*

$$(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) \leq \varepsilon,$$

*where $\varepsilon = \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})}$. Similarly, for a randomly selected false negative $\tilde{d_j^-}$ with equivalence probability distribution of $\mathcal{N}(\mu_{\tilde{d_j^-}}, \sigma_{\tilde{d_j^-}})$, with probability at least $(1-\delta)$, the following inequality holds*

$$\mathbb{E}(\mu_{\tilde{d_j^-}} - 2\sigma_{\tilde{d_j^-}}) - (\mu_{\tilde{d_j^-}} - 2\sigma_{\tilde{d_j^-}}) \leq \varepsilon.$$

**Proof** Consider the randomly selected pair $d_i^+$ from true positives. The mean of its equivalence probability can be represented by

$$\mu_{d_i^+} = \frac{\sum_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{d_i^+} \hat{\mu}_{d_i^+}}{\sum_{k=1}^{m} w_k Z_k + w_{d_i^+}},$$

where $m$ is the number of rule risk features, $w_k$ is the learned weight of a risk feature $f_k$, $\mu_{f_k}$ is the probability mean of the feature, $Z_k$ is a random variable indicates if a selected pair has this feature, and $\hat{\mu}_{d_i^+}$ is the output probability by a classifier with its weight $w_{d_i^+}$. Note that for a randomly selected true positive, the values of $w_k$ and $\mu_{f_k}$ for each rule risk feature are fixed, while $Z_k$, $\hat{\mu}_{d_i^+}$ and $w_{d_i^+}$ are random variables. Note that according to LearnRisk, the value of $w_{d_i^+}$ totally depends on $\hat{\mu}_{d_i^+}$.

The standard deviation of its equivalence probability can also be represented by

$$\sigma_{d_i^+} = \frac{1}{\sum_{k=1}^{m} w_k Z_k + w_{d_i^+}} \sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{d_i^+}^2 \hat{\sigma}_{d_i^+}^2},$$

where $\hat{\sigma}_{d_i^+}^2$ denotes the corresponding variance of a classifier's output $\hat{\mu}_{d_i^+}$.

Recall that a function $f : X^n \to \mathbb{R}$ has the *bounded differences property* if for some non-negative constants $c_1, c_2, ..., c_n$,

$$\sup_{x_1,...,x_n,x_k' \in X} |f(x_1, ..., x_{k-1}, x_k, x_{k+1}, ..., x_n) - f(x_1, ..., x_{k-1}, x_k', x_{k+1}, ..., x_n)| \leq c_k, 1 \leq k \leq n$$

The bounded differences property shows that if the $i$th variable is changed while all the others being fixed, the value of $f$ will not change by more than $c_k$.

Let $f(Z_1, ..., Z_m, \hat{\mu}_{d_i^+}) = \mu_{d_i^+} - 2\sigma_{d_i^+}$. Now we proceed to consider the bounded differences property of $f$. Note that a valid equivalence probability should be between 0 and 1. Hence, for all $\mu \geq 0, \sigma \geq 0$, we have $0 \leq \mu \pm 2\sigma \leq 1$. As a result, by changing the value of $Z_k$, we have

$$sup|f(z_1, ..., z_k, ..., z_m, \hat{\mu}_{d_i^+}) - f(z_1, ..., z_k', ..., z_m, \hat{\mu}_{d_i^+})| \leq 1$$

Similarly, the upper bound of $f$ by changing the value of $\hat{\mu}_{d_i^+}$ is

$$sup|f(z_1, ..., z_m, \hat{\mu}_{d_i^+}) - f(z_1, ..., z_m, \hat{\mu}_{d_i^+}')| \leq 1$$

At this point, we have obtained the upper bounds of the function $f(Z_1, ..., Z_m, \hat{\mu}_{d_i^+})$ by changing any one of the variables. Denoting these bounds as $c_1, ..., c_m, c_{m+1}$, where $c_k = 1, 1 \leq k \leq m+1$, we have

$$\sum_{k=1}^{m+1} c_k^2 = m + 1$$

Recall that the McDiarmid's inequality (McDiarmid, 1989) states that if a function $f$ satisfies the bounded differences property with constants $c_1, ..., c_n$. Let $Y = f(X_1, ..., X_n)$, where the $X_k$s are independent random variables. Then, for all $\varepsilon > 0$,

$$\mathbb{P}(Y - \mathbb{E}Y \geq \varepsilon) \leq exp(-\frac{2\varepsilon^2}{\sum_{k=1}^{n} c_k^2});$$

$$\mathbb{P}(\mathbb{E}Y - Y \geq \varepsilon) \leq exp(-\frac{2\varepsilon^2}{\sum_{k=1}^{n} c_k^2}).$$

Based on the McDiarmid's inequality, for all $\varepsilon > 0$, we have

$$\mathbb{P}(\mu_{d_i^+} - 2\sigma_{d_i^+} - \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) \geq \varepsilon) \leq exp(-\frac{2\varepsilon^2}{\sum_{k=1}^{m+1} c_k^2}) = exp(-\frac{2\varepsilon^2}{m+1}).$$

Let $\delta = exp(-\frac{2\varepsilon^2}{m+1})$, we can get that $\varepsilon = \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})}$. Hence, with the probability at least $1-\delta$, the inequality $\mu_{d_i^+} - 2\sigma_{d_i^+} - \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) \leq \varepsilon$ holds.

Similarly, with the probability at least $1-\delta$, we have $\mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) - (\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) \leq \varepsilon$. ■

In the following, we present the proof of Theorem 2.

**Proof [Theorem 2]** According to Lemma 6, with the probability at least $(1-\delta)^2$, the following inequalities hold

$$(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) + \mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) - (\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) \leq 2\varepsilon;$$
$$(\mu_{d_i^+} - 2\sigma_{d_i^+} - \mu_{\tilde{d}_j^-} + 2\sigma_{\tilde{d}_j^-}) + \mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) - \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) \leq 2\varepsilon;$$

Hence, we have

$$\mu_{d_i^+} - 2\sigma_{d_i^+} - \mu_{\tilde{d}_j^-} + 2\sigma_{\tilde{d}_j^-} \leq 2\varepsilon + \big[\mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-})\big], \tag{17}$$

where $\varepsilon = \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})}$. We denote

$$\Delta C^- = \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) = \mathbb{E}(1 - (\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-})) - \mathbb{E}(1 - (\mu_{d_i^+} - 2\sigma_{d_i^+})), \tag{18}$$

where $\mathbb{E}(1-(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}))$ is the risk expectation of false negatives being labeled as *matching* and $\mathbb{E}(1 - (\mu_{d_i^+} - 2\sigma_{d_i^+}))$ is the risk expectation of true positives being labeled as *matching*. Based on the definition of VaR, we have $VaR^-(\tilde{d}_j^-) = \mu_{\tilde{d}_j^-} + 2\sigma_{\tilde{d}_j^-}$, and $VaR^+(d_i^+) = 1 - (\mu_{d_i^+} - 2\sigma_{d_i^+})$. Denoting $\Delta VaR^- = VaR^-(\tilde{d}_j^-) - VaR^+(d_i^+)$, we have,

$$\begin{aligned} 1 + \Delta VaR^- =& \mu_{\tilde{d}_j^-} + 2\sigma_{\tilde{d}_j^-} + \mu_{d_i^+} - 2\sigma_{d_i^+} \\ \leq& \mu_{\tilde{d}_j^-} + \mu_{\tilde{d}_j^-} + 2\varepsilon + \big[\mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-})\big] \\ =& 2\mu_{\tilde{d}_j^-} + 2\varepsilon + \Delta C^-. \end{aligned} \tag{19}$$

Hence, for a randomly selected false negative and a randomly selected true positive, with probability at least $(1-\delta)^2$, we have

$$\mu_{\tilde{d}_j^-} \geq \frac{1}{2} + \frac{\Delta VaR^-}{2} - \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})} - \frac{\Delta C^-}{2}. \tag{20}$$

Note that the probability of the above inequality does not hold is $[1-(1-\delta)^2]$. Suppose that there are totally $n$ true positives $d_i^+$ ranked after $\tilde{d}_j^-$ by LearnRisk such that each true positive, $d_i^+$, satisfies $\Delta VaR^- - \Delta C^- > \epsilon$. Then the probability of Inequality 20 fails can be approximated by $[1-(1-\delta)^2]^n$. That is, the probability of at least one of the true positives

can support the Inequality 20 is $\{1-[1-(1-\delta)^2]^n\}$. Let $1-[1-(1-\delta)^2]^n = 1-\delta'$, we can get $\delta = 1-\sqrt{1-\delta'^{\frac{1}{n}}}$. Therefore, with probability at least $(1-\delta)$,

$$\mu_{\tilde{d}_j^-} \geq \frac{1}{2} + \frac{\epsilon}{2} - \sqrt{\frac{m+1}{2} ln[\frac{1}{1-(1-\delta^{\frac{1}{n}})^{\frac{1}{2}}}]}. \tag{21}$$

■

Note that the total number of rule risk features ($m$) is usually limited (e.g., dozens or hundreds), while $n$ is usually much larger than $m$. By the exponential effect of $n$, the 3rd term on the right-hand side tends to become zero as the value of $n$ increases.

### A.2 Proof of Lemma 3

**Lemma 3**

$$\Delta C^- \leq max\big\{\mathbb{E}(w_{d_i^+}(\hat{\mu}_{d_i^+} - 2\hat{\sigma}_{d_i^+})) - \mathbb{E}(w_{\tilde{d}_j^-}(\hat{\mu}_{\tilde{d}_j^-} - 2\hat{\sigma}_{\tilde{d}_j^-})), \mathbb{E}(w_{d_i^+}\hat{\mu}_{d_i^+}) - \mathbb{E}(w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-})\big\}, \tag{22}$$

*where the $\hat{\mu}_*$ and $\hat{\sigma}_*$ denote the DNN output probability and its corresponding standard deviation respectively, $w_*$ denotes the learned weight of DNN risk feature.*

**Proof** For simplicity of presentation, let $N_{d_i^+}$ denote the weight normalization factor of $d_i^+$, or $N_{d_i^+} = \sum_{k=1}^{m} w_k Z_k + w_{d_i^+}$. Similarly, let $N_{\tilde{d}_j^-}$ denote the weight normalization factor of $\tilde{d}_j^-$, or $N_{\tilde{d}_j^-} = \sum_{k=1}^{m} w_k Z_k + w_{\tilde{d}_j^-}$. According to the weight function defined by *LearnRisk* (Chen et al., 2020), without loss of generality, we suppose that $w_{d_i^+} = w_{\tilde{d}_j^-}$. As a result, $N_{d_i^+} = N_{\tilde{d}_j^-}$.

Based on Assumption 1, we have,

$$
\begin{aligned}
&\mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) \\
=&\mathbb{E}((\mu_{d_i^+} - 2\sigma_{d_i^+}) - (\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-})) \\
=&\mathbb{E}\Bigg(\frac{1}{N_{d_i^+}}\Bigg(\sum_{k=1}^{m} w_k\mu_{f_k}Z_k + w_{d_i^+}\hat{\mu}_{d_i^+} - 2\sqrt{\sum_{k=1}^{m} w_k^2\sigma_{f_k}^2 Z_k + w_{d_i^+}^2\hat{\sigma}_{d_i^+}^2}\Bigg) - \\
&\frac{1}{N_{\tilde{d}_j^-}}\Bigg(\sum_{k=1}^{m} w_k\mu_{f_k}Z_k + w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-} - 2\sqrt{\sum_{k=1}^{m} w_k^2\sigma_{f_k}^2 Z_k + w_{\tilde{d}_j^-}^2\hat{\sigma}_{\tilde{d}_j^-}^2}\Bigg)\Bigg) \\
=&\mathbb{E}\Bigg(\frac{1}{N_{d_i^+}}\Bigg(\sum_{k=1}^{m} w_k\mu_{f_k}Z_k + w_{d_i^+}\hat{\mu}_{d_i^+} - 2\sqrt{\sum_{k=1}^{m} w_k^2\sigma_{f_k}^2 Z_k + w_{d_i^+}^2\hat{\sigma}_{d_i^+}^2} - \\
&\sum_{k=1}^{m} w_k\mu_{f_k}Z_k - w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-} + 2\sqrt{\sum_{k=1}^{m} w_k^2\sigma_{f_k}^2 Z_k + w_{\tilde{d}_j^-}^2\hat{\sigma}_{\tilde{d}_j^-}^2}\Bigg)\Bigg) \\
=&\mathbb{E}\Bigg(\frac{1}{N_{d_i^+}}\Bigg(w_{d_i^+}\hat{\mu}_{d_i^+} - w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-} + 2\Bigg[\sqrt{\sum_{k=1}^{m} w_k^2\sigma_{f_k}^2 Z_k + w_{\tilde{d}_j^-}^2\hat{\sigma}_{\tilde{d}_j^-}^2} - \sqrt{\sum_{k=1}^{m} w_k^2\sigma_{f_k}^2 Z_k + w_{d_i^+}^2\hat{\sigma}_{d_i^+}^2}\Bigg]\Bigg)\Bigg) \qquad \cdots (S_1)
\end{aligned}
$$

If $w_{\tilde{d}_j^-}\hat{\sigma}_{\tilde{d}_j^-} < w_{d_i^+}\hat{\sigma}_{d_i^+}$, then

$$S_1 \leq \mathbb{E}\big(\frac{1}{N_{d_i^+}}\big(w_{d_i^+}\hat{\mu}_{d_i^+} - w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-}\big)\big).$$

If $w_{\tilde{d}_j^-}\hat{\sigma}_{\tilde{d}_j^-} \geq w_{d_i^+}\hat{\sigma}_{d_i^+}$, then

$$
\begin{aligned}
S_1 &\leq \mathbb{E}\Bigg(\frac{1}{N_{d_i^+}}\Bigg(w_{d_i^+}\hat{\mu}_{d_i^+} - w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-} + 2\Bigg[\sqrt{w_{\tilde{d}_j^-}^2\hat{\sigma}_{\tilde{d}_j^-}^2} - \sqrt{w_{d_i^+}^2\hat{\sigma}_{d_i^+}^2}\Bigg]\Bigg)\Bigg) \qquad \cdots (S_2) \\
&= \mathbb{E}\big(\frac{w_{d_i^+}}{N_{d_i^+}}\big(\hat{\mu}_{d_i^+} - 2\hat{\sigma}_{d_i^+}\big)\big) - \mathbb{E}\big(\frac{w_{\tilde{d}_j^-}}{N_{d_i^+}}\big(\hat{\mu}_{\tilde{d}_j^-} - 2\hat{\sigma}_{\tilde{d}_j^-}\big)\big)
\end{aligned}
$$

From step $S_1$ to step $S_2$, we apply the rule that if $a \geq 0, b \geq 0, c \geq 0$ and $b \geq c$, then $\sqrt{a+b} - \sqrt{a+c} \leq \sqrt{b} - \sqrt{c}$. For simplicity of presentation, we denote the normalization of $\frac{w_{d_i^+}}{N_{d_i^+}}$ by $w_{d_i^+}$, and similarly, the normalized $w_{\tilde{d}_j^-}$.

Hence, we have

$$\Delta C^- \leq max\big\{\mathbb{E}(w_{d_i^+}(\hat{\mu}_{d_i^+} - 2\hat{\sigma}_{d_i^+})) - \mathbb{E}(w_{\tilde{d}_j^-}(\hat{\mu}_{\tilde{d}_j^-} - 2\hat{\sigma}_{\tilde{d}_j^-})), \mathbb{E}(w_{d_i^+}\hat{\mu}_{d_i^+}) - \mathbb{E}(w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-})\big\},$$

where the $\hat{\mu}_*$ and $\hat{\sigma}_*$ denote the DNN output probability and its corresponding standard deviation respectively, $w_*$ denotes the learned weight of DNN risk feature. ■

### A.3 Proof of Theorem 4

Similarly, based on Assumption 1, we theoretically analyze the chance of a false positive being flipped from *matching* to *unmatching*. We first prove a lemma, and then prove Theorem 4 based on the lemma.

**Lemma 7** *For a randomly selected pair $d_i^-$ from true negatives, we denote the mean of its equivalence probability by $\mu_{d_i^-}$, and the corresponding standard deviation by $\sigma_{d_i^-}$. For any $\delta \in (0,1)$, with probability at least $(1-\delta)$, the following inequality holds*

$$\mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-}) - (\mu_{d_i^-} + 2\sigma_{d_i^-}) \leq \varepsilon,$$

*where $\varepsilon = \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})}$, $m$ denotes the total number of rule risk features. Similarly, for a randomly selected false positive $\tilde{d}_j^+$ with the equivalence probability mean of $\mu_{\tilde{d}_j^+}$ and the standard deviation of $\sigma_{\tilde{d}_j^+}$, with probability at least $(1-\delta)$, the following inequality holds*

$$(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) \leq \varepsilon.$$

**Proof** Consider a randomly selected pair $d_i^-$ from true negatives. The mean of its equivalence probability can be represented by

$$\mu_{d_i^-} = \frac{\sum_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{d_i^-} \hat{\mu}_{d_i^-}}{\sum_{k=1}^{m} w_k Z_k + w_{d_i^-}},$$

where $m$ denotes the number of rule risk features, $w_k$ denotes the weight of a risk feature $f_k$, $\mu_{f_k}$ is the equivalence probability mean of the feature $f_k$, $Z_k$ is a random variable indicates if a selected pair has this feature, and $\hat{\mu}_{d_i^-}$ is the output probability by a classifier with its weight $w_{d_i^-}$. Note that for a randomly selected true negative, the values of $w_k$ and $\mu_{f_k}$ for each risk feature are fixed, while $Z_k$, $\hat{\mu}_{d_i^-}$ are random variables. Note that the value of $w_{d_i^-}$ totally depends $\hat{\mu}_{d_i^-}$.

The standard deviation of its equivalence probability can also be represented by

$$\sigma_{d_i^-} = \frac{1}{\sum_{k=1}^{m} w_k Z_k + w_{d_i^-}} \sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{d_i^-}^2 \hat{\sigma}_{d_i^-}^2},$$

where $\hat{\sigma}_{d_i^-}^2$ denotes the corresponding variance of a classifier's output $\hat{\mu}_{d_i^-}$.

Let $f(Z_1, ..., Z_m, \hat{\mu}_{d_i^-}) = \mu_{d_i^-} + 2\sigma_{d_i^-}$. Now we proceed to consider the bounded differences property of $f$. As in the proof of lemma 1, for all $\mu \geq 0, \sigma \geq 0$, we have $0 \leq \mu \pm 2\sigma \leq 1$. Hence, by changing the value of $Z_k$, we have

$$sup|f(z_1, ..., z_k, ..., z_m, \hat{\mu}_{d_i^-}) - f(z_1, ..., z_k', ..., z_m, \hat{\mu}_{d_i^-})| \leq 1$$

Similarly, the upper bound of $f$ by changing the value of $\hat{\mu}_{d_i^-}$ is,

$$sup|f(z_1,...,z_m,\hat{\mu}_{d_i^-}) - f(z_1,...,z_m,\hat{\mu}'_{d_i^-})| \leq 1$$

At this point, we have obtained the upper bounds of function $f(Z_1,...,Z_m,\hat{\mu}_{d_i^-})$ by changing any one of the variables. Denoting these bounds by $c_1,...,c_m,c_{m+1}$, where $c_k = 1, 1 \leq k \leq m+1$, we have

$$\sum_{k=1}^{m+1} c_k^2 = m+1.$$

By applying the McDiarmid's inequality, for all $\varepsilon > 0$, we have

$$\mathbb{P}(\mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-}) - (\mu_{d_i^-} + 2\sigma_{d_i^-}) \geq \varepsilon) \leq exp(-\frac{2\varepsilon^2}{\sum_{k=1}^{m+1} c_k^2}) = exp(-\frac{2\varepsilon^2}{m+1}).$$

Let $\delta = exp(-\frac{2\varepsilon^2}{m+1})$, we can get that $\varepsilon = \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})}$. Hence, with the probability at least $1-\delta$, the inequality $\mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-}) - (\mu_{d_i^-} + 2\sigma_{d_i^-}) \leq \varepsilon$ holds.

Similarly, with the probability at least $1-\delta$, we have $(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) \leq \varepsilon$. ■

**Theorem 4** *Given a false positive $\tilde{d}_j^+$, suppose that there are totally $n$ true negatives, denoted by $d_i^-$, ranked after $\tilde{d}_j^+$ by LearnRisk such that each true negative, $d_i^-$, satisfies*

$$\Delta VaR^+ - \Delta C^+ > \epsilon, \tag{23}$$

*in which $\Delta VaR^+ = VaR^+(\tilde{d}_j^+) - VaR^-(d_i^-)$, and $\Delta C^+ = \mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-})$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$, its expectation of equivalence probability of $\tilde{d}_j^+$, $\mu_{\tilde{d}_j^+}$ estimated by* LearnRisk*, satisfies*

$$\mu_{\tilde{d}_j^+} \leq \frac{1}{2} - \frac{\epsilon}{2} + \sqrt{\frac{m+1}{2} ln[\frac{1}{1-(1-\delta^{\frac{1}{n}})^{\frac{1}{2}}}]},$$

*in which $\mu_*$ denotes the mean of equivalence probability and $\sigma_*$ denotes its standard deviation.*

**Proof** With Lemma 7, with probability at least $(1-\delta)^2$, the following inequalities hold

$$(\mu_{d_i^-} + 2\sigma_{d_i^-}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-}) + \mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - (\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) \geq -2\varepsilon;$$
$$(\mu_{d_i^-} + 2\sigma_{d_i^-} - \mu_{\tilde{d}_j^+} - 2\sigma_{\tilde{d}_j^+}) + \mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-}) \geq -2\varepsilon;$$

Hence, we have

$$\mu_{d_i^-} + 2\sigma_{d_i^-} - \mu_{\tilde{d}_j^+} - 2\sigma_{\tilde{d}_j^+} \geq -2\varepsilon - \big[\mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-})\big], \tag{24}$$

where $\varepsilon = \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})}$. We denote

$$\Delta C^+ = \mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-}), \tag{25}$$

where $\mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+})$ is the risk expectation of false positives being labeled as *unmatching* and $\mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-})$ is the risk expectation of true negatives being labeled as *unmatching*. Based on the definition of VaR, we have $VaR^+(\tilde{d}_j^+) = 1 - (\mu_{\tilde{d}_j^+} - 2\sigma_{\tilde{d}_j^+})$, and $VaR^-(d_i^-) = \mu_{d_i^-} + 2\sigma_{d_i^-}$. Denoting $\Delta VaR^+ = VaR^+(\tilde{d}_j^+) - VaR^-(d_i^-)$, we have

$$\begin{aligned} 1 - \Delta VaR^+ =& \mu_{d_i^-} + 2\sigma_{d_i^-} + \mu_{\tilde{d}_j^+} - 2\sigma_{\tilde{d}_j^+} \\ \geq& \mu_{\tilde{d}_j^+} + \mu_{\tilde{d}_j^+} - 2\varepsilon - \left[\mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-})\right] \\ =& 2\mu_{\tilde{d}_j^+} - 2\varepsilon - \Delta C^+. \end{aligned} \tag{26}$$

In Equation 26, the inequality is obtained by applying the Inequality 24. Hence, for a randomly selected false positive and a randomly selected true negative, with probability at least $(1-\delta)^2$, the following inequality holds

$$\mu_{\tilde{d}_j^+} \leq \frac{1}{2} - \frac{\Delta VaR^+}{2} + \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})} + \frac{\Delta C^+}{2}. \tag{27}$$

Note that the probability of the above inequality does not hold is $[1-(1-\delta)^2]$. Suppose that there are totally $n$ true negatives, denoted by $d_i^-$, ranked after $\tilde{d}_j^+$ by LearnRisk such that each true negative, $d_i^-$, satisfies $\Delta VaR^+ - \Delta C^+ > \epsilon$. Then the probability of Inequality 27 fails can be approximated by $[1-(1-\delta)^2]^n$. That is, the probability of at least one of the true negatives can support the Inequality 27 is $\{1-[1-(1-\delta)^2]^n\}$. Let $1-[1-(1-\delta)^2]^n = 1-\delta'$, we can get $\delta = 1 - \sqrt{1-\delta'^{\frac{1}{n}}}$. Therefore, with probability at least $(1-\delta)$, we have

$$\mu_{\tilde{d}_j^+} \leq \frac{1}{2} - \frac{\epsilon}{2} + \sqrt{\frac{m+1}{2} ln[\frac{1}{1-(1-\delta^{\frac{1}{n}})^{\frac{1}{2}}}]}, \tag{28}$$

■

### A.4 Proof of Lemma 5

**Lemma 5**

$$\Delta C^+ \leq max\big\{\mathbb{E}(w_{\tilde{d}_j^+}(\hat{\mu}_{\tilde{d}_j^+} + 2\hat{\sigma}_{\tilde{d}_j^+})) - \mathbb{E}(w_{d_i^-}(\hat{\mu}_{d_i^-} + 2\hat{\sigma}_{d_i^-})), \mathbb{E}(w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+}) - \mathbb{E}(w_{d_i^-}\hat{\mu}_{d_i^-})\big\}, \tag{29}$$

*where the $\hat{\mu}_*$ and $\hat{\sigma}_*$ denote the DNN output probability and its corresponding standard deviation respectively, $w_*$ denotes the learned weight of DNN risk feature.*

**Proof** For simplicity of presentation, let $N_{\tilde{d}_j^+}$ denote the weight normalization factor of $\tilde{d}_j^+$, $N_{\tilde{d}_j^+} = \sum_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{\tilde{d}_j^+}$. Similarly, $N_{d_i^-}$ denote the weight normalization factor of

$d_i^-$, $N_{d_i^-} = \sum_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{d_i^-}$. As in the proof of Lemma 1, we suppose that $N_{\tilde{d}_j^+} = N_{d_i^-}$. Based on Assumption 1, we have,

$$
\begin{aligned}
&\mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-}) \\
=&\mathbb{E}((\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - (\mu_{d_i^-} + 2\sigma_{d_i^-})) \\
=&\mathbb{E}\Bigg(\frac{1}{N_{\tilde{d}_j^+}}\Bigg(\sum_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+} + 2\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{\tilde{d}_j^+}^2 \hat{\sigma}_{\tilde{d}_j^+}^2}\Bigg) - \\
&\frac{1}{N_{d_i^-}}\Bigg(\sum_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{d_i^-}\hat{\mu}_{d_i^-} + 2\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{d_i^-}^2 \hat{\sigma}_{d_i^-}^2}\Bigg)\Bigg) \\
=&\mathbb{E}\Bigg(\frac{1}{N_{\tilde{d}_j^+}}\Bigg(\sum_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+} + 2\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{\tilde{d}_j^+}^2 \hat{\sigma}_{\tilde{d}_j^+}^2} - \\
&\sum_{k=1}^{m} w_k \mu_{f_k} Z_k - w_{d_i^-}\hat{\mu}_{d_i^-} - 2\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{d_i^-}^2 \hat{\sigma}_{d_i^-}^2}\Bigg)\Bigg) \\
=&\mathbb{E}\Bigg(\frac{1}{N_{\tilde{d}_j^+}}\Bigg(w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+} - w_{d_i^-}\hat{\mu}_{d_i^-} + 2\Bigg[\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{\tilde{d}_j^+}^2 \hat{\sigma}_{\tilde{d}_j^+}^2} - \sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{d_i^-}^2 \hat{\sigma}_{d_i^-}^2}\Bigg]\Bigg)\Bigg) \qquad \cdots (S_3)
\end{aligned}
$$

If $w_{\tilde{d}_j^+}\hat{\sigma}_{\tilde{d}_j^+} < w_{d_i^-}\hat{\sigma}_{d_i^-}$, then

$$S_3 \leq \mathbb{E}\big(\frac{1}{N_{\tilde{d}_j^+}}\big(w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+} - w_{d_i^-}\hat{\mu}_{d_i^-}\big)\big).$$

If $w_{\tilde{d}_j^+}\hat{\sigma}_{\tilde{d}_j^+} \geq w_{d_i^-}\hat{\sigma}_{d_i^-}$, then

$$
\begin{aligned}
S_3 &\leq \mathbb{E}\Bigg(\frac{1}{N_{\tilde{d}_j^+}}\Bigg(w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+} - w_{d_i^-}\hat{\mu}_{d_i^-} + 2\Bigg[\sqrt{w_{\tilde{d}_j^+}^2 \hat{\sigma}_{\tilde{d}_j^+}^2} - \sqrt{w_{d_i^-}^2 \hat{\sigma}_{d_i^-}^2}\Bigg]\Bigg)\Bigg) \qquad \cdots (S_4) \\
&= \mathbb{E}\big(\frac{w_{\tilde{d}_j^+}}{N_{\tilde{d}_j^+}}\big(\hat{\mu}_{\tilde{d}_j^+} + 2\hat{\sigma}_{\tilde{d}_j^+}\big)\big) - \mathbb{E}\big(\frac{w_{d_i^-}}{N_{\tilde{d}_j^+}}\big(\hat{\mu}_{d_i^-} + 2\hat{\sigma}_{d_i^-}\big)\big)
\end{aligned}
$$

From step $S_3$ to step $S_4$, we apply the rule that if $a \geq 0, b \geq 0, c \geq 0$ and $b \geq c$, then $\sqrt{a+b} - \sqrt{a+c} \leq \sqrt{b} - \sqrt{c}$. For simplicity of presentation, we denote the normalization of $\frac{w_{\tilde{d}_j^+}}{N_{\tilde{d}_j^+}}$ by $w_{\tilde{d}_j^+}$, and similarly, the normalized $w_{d_i^-}$. Hence, we have

$$\Delta C^+ \leq max\big\{\mathbb{E}(w_{\tilde{d}_j^+}(\hat{\mu}_{\tilde{d}_j^+} + 2\hat{\sigma}_{\tilde{d}_j^+})) - \mathbb{E}(w_{d_i^-}(\hat{\mu}_{d_i^-} + 2\hat{\sigma}_{d_i^-})), \mathbb{E}(w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+}) - \mathbb{E}(w_{d_i^-}\hat{\mu}_{d_i^-})\big\},$$

where the $\hat{\mu}_*$ and $\hat{\sigma}_*$ denote the DNN output probability and its corresponding standard deviation respectively, $w_*$ denotes the learned weight of DNN risk feature. ■

## References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. In *arXiv:1606.06565*, pages 1–29, 2016.

Pierre Baldi and Peter J. Sadowski. Understanding dropout. In *Advances in Neural Information Processing Systems*, pages 2814–2822, 2013.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Zhaoqiang Chen, Qun Chen, Boyi Hou, Murtadha Ahmed, and Zhanhuai Li. Improving machine-based entity resolution with limited human effort: A risk perspective. In *Proceedings of the International Workshop on Real-Time Business Intelligence and Analytics*, pages 1–5, 2018.

Zhaoqiang Chen, Qun Chen, Boyi Hou, Zhanhuai Li, and Guoliang Li. Towards interpretable and learnable risk analysis for entity resolution. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 1165–1180, 2020.

Peter Christen. Automatic record linkage using seeded nearest neighbour and support vector machine classification. In *Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 151–159, 2008.

Peter Christen. *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*, chapter 2, pages 32–34. Springer Science & Business Media, 2012.

Vassilis Christophides, Vasilis Efthymiou, and Kostas Stefanidis. Entity resolution in the web of data. *Synthesis Lectures on the Semantic Web*, 5(3):1–122, 2015.

Munir Cochinwala, Verghese Kurien, Gail Lalk, and Dennis Shasha. Efficient data reconciliation. *Information Sciences*, 137(1-4):1–15, 2001.

Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. In *arXiv:1802.09841*, pages 1–10, 2018.

Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq R. Joty, Mourad Ouzzani, and Nan Tang. Distributed representations of tuples for entity resolution. *PVLDB*, 11 (11):1454–1467, 2018.

Ahmed K Elmagarmid, Panagiotis G Ipeirotis, and Vassilios S Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(1):1–16, 2007.

Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization networks and support vector machines. *Advances in computational mathematics*, 13(1):1–50, 2000.

Wenfei Fan, Xibei Jia, Jianzhong Li, and Shuai Ma. Reasoning about record matching rules. *PVLDB*, 2(1):407–418, 2009.

Ivan P Fellegi and Alan B Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, pages 1–10, 2015.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, pages 1–12, 2017.

Dan Hendrycks, Mantas Mazeika, and Thomas G Dietterich. Deep anomaly detection with outlier exposure. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, pages 1–18, 2019.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzkebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 2790–2799, 2019.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5070–5079, 2019.

Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems*, pages 5546–5557, 2018.

Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. Low-resource deep entity resolution with transfer and active learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, pages 5851–5861, 2019.

Taekyung Kim, Minki Jeong, Seunghyeon Kim, Seokeon Choi, and Changick Kim. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12456–12465, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, pages 1–11, 2015.

Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2, pages 1137–1145, 1995.

Pigi Kouki, Jay Pujara, Christopher Marcum, Laura Koehly, and Lise Getoor. Collective entity resolution in familial networks. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 227–236, 2017.

Lingli Li, Jianzhong Li, and Hong Gao. Rule-based method for entity resolution. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 27(1):250–263, 2015.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. Deep entity matching with pre-trained language models. *PVLDB*, 14(1):50–60, 2021.

Mingsheng Long, Guiguang Ding, Jianmin Wang, Jiaguang Sun, Yuchen Guo, and Philip S. Yu. Transfer sparse coding for robust image representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 407–414, 2013.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37, pages 97–105, 2015.

Colin McDiarmid. *On the method of bounded differences*, page 148–188. London Mathematical Society Lecture Note Series. Cambridge University Press, 1989.

Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. Deep learning for entity matching: A design space exploration. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 19–34, 2018.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

Hao Nie, Xianpei Han, Ben He, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. Deep sequence-to-sequence entity matching for heterogeneous entity resolution. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 629–638, 2019.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 22(10):1345–1359, 2010.

Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249, 2018.

Sunita Sarawagi and Anuradha Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 269–278, 2002.

Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

Rohit Singh, Vamsi Meduri, Ahmed Elmagarmid, Samuel Madden, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Armando Solar-Lezama, and Nan Tang. Generating concise entity matching rules. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 1635–1638, 2017.

Parag Singla and Pedro Domingos. Entity resolution with markov logic. In *Proceedings of the IEEE 6th International Conference on Data Mining (ICDM)*, pages 572–582, 2006.

Giuseppe Tardivo. Value at risk (var): The new benchmark for managing market risk. *Journal of Financial Management & Analysis*, 15(1):16–26, 2002.

Ying Wei, Yu Zhang, Junzhou Huang, and Qiang Yang. Transfer learning via learning to transfer. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pages 5072–5081, 2018.

Zuxuan Wu, Xin Wang, Joseph E Gonzalez, Tom Goldstein, and Larry S Davis. Ace: Adapting to changing environments for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 2121–2130, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, pages 1–11, 2017.

Peng Zhang, Jiuling Wang, Ali Farhadi, Martial Hebert, and Devi Parikh. Predicting failures of vision systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3566–3573, 2014.

Chen Zhao and Yeye He. Auto-em: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning. In *The World Wide Web Conference*, pages 2413–2424, 2019.

Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J. Gordon. On learning invariant representations for domain adaptation. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 7523–7532, 2019.

Zhi-Hua Zhou. Ensemble learning. *Encyclopedia of biometrics*, 1:270–273, 2009.

Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.