STAN: Adversarial Network for Cross-domain Question Difficulty Prediction

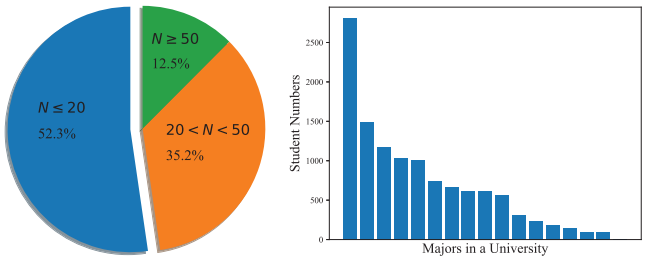
Ye Huang[1], Wei Huang[1], Shiwei Tong[1], Zhenya Huang[1,*], Qi Liu[1], Enhong Chen[1], Jianhui Ma[1], Liang Wan[1], and Shijin Wang[2]
[1]Anhui Province Key Laboratory of Big Data Analysis and Application, School of Data Science & School of Computer Science and Technology, University of Science and Technology of China
[2]iFLYTEK Research, iFLYTEK CO., LTD.
{huangyehy, ustc0411, tongsw}@mail.ustc.edu.cn, {huangzhy, qiliuql, cheneh, jianhui}@ustc.edu.cn
wanl2016@mail.ustc.edu.cn, sjwang3@iflytek.com

***Abstract*—In intelligent education systems, question difficulty prediction (QDP) is a fundamental task of many applications, such as personalized question recommendation and test paper analysis. Previous work mainly focus on data-driven QDP methods, which are heavily relied on the large-scale labeled dataset of courses. To alleviate the labor intensity, an intuitive method is to introduce domain adaptation into QDP and consider each course as a domain. In educational psychology, there are two factors influencing difficulty common to different courses: the obstacles of comprehending the question and generating a response, namely stimulus and task difficulty. To this end, we propose a novel Stimulus and Task difficulty-based Adversarial Network (STAN) that models question difficulty from the views of stimulus and task. Then, in order to align the difficulty distribution of the source domain and the target domain, we utilize the conditional adversarial learning with readability-enhanced pseudo-labels. Meanwhile, we proposed a sampling method based on density estimation to implicit alignment. Finally, we conduct experiments on the real questions datasets to evaluate the effectiveness of our QDP model and domain adaptation method. Our method significantly improves accuracy over state-of-the-art methods on real-world question data of multiple courses.**

***Index Terms*—domain adaptation, question difficulty prediction, text readability**

I. Introduction

In recent years, intelligent education systems have received widespread attention due to efficiency and convenience. These systems can not only help tutors to design high-quality papers, but also provide students with personalized questions recommendation to improve their study efficiency [1]. Among these applications, the difficulty is the most useful property of test questions analysis. The difficulty of a question is defined as an estimate of the skill level needed to pass it. There are two ways to measure the difficulty. In classical test theory, it is measured by calculating the proportion of students who answer a question correctly [2]. In practice, the experts are able to use their own experience to estimate the difficulty index. However, neither of these two measurements can be applied to the intelligent education system. The former requires test logs, which are unavailable before the question is answered by many students. Although test logs are unnecessary in the latter, this human-based way is labor-intensive, subjective and powerless in the face of massive data.



(a) The proportion of courses by student number. (b) The distribution of the student numbers in different majors.

Fig. 1. Statistics of the student number in the course.

Recently, data-driven solutions have emerged [3], [4], which combine educational psychology and NLP methods. By building question representation and prediction algorithms, data-driven methods learn from a large amount of data. However, the dependence on large-scale difficulty-labeled datasets hinders its application. A statistics from a university [5] shows that more than 50% of the courses are only taken by a small number of students, while the courses with more students are concentrated in a few popular majors, as shown in Figure 1. For these courses without lots of students, although the teacher can make many questions, the difficulty label is hard to obtain. Moreover, few schools share their difficulty-labeled question data, which are private and commercial. Therefore, the data-driven methods suffer from the lack of labeled question data.

Fortunately, there are some popular public courses that are resource-rich, such as calculus and probability theory. It's meaningful to reuse data and model from resource-rich courses to resource-poor courses. In order to implement this, we pay attention to the common factors influencing question difficulty. According to educational assessment theory [6], the difficulty of a question is affected by the following two factors: One is the obstacle that students try to understand the words and phrases in the question, named stimulus difficulty, which is related to semantic and readability of question stem. For example, considering the math-physics course pair, there are

* Corresponding Author.

2374-8486/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDM51629.2021.00032

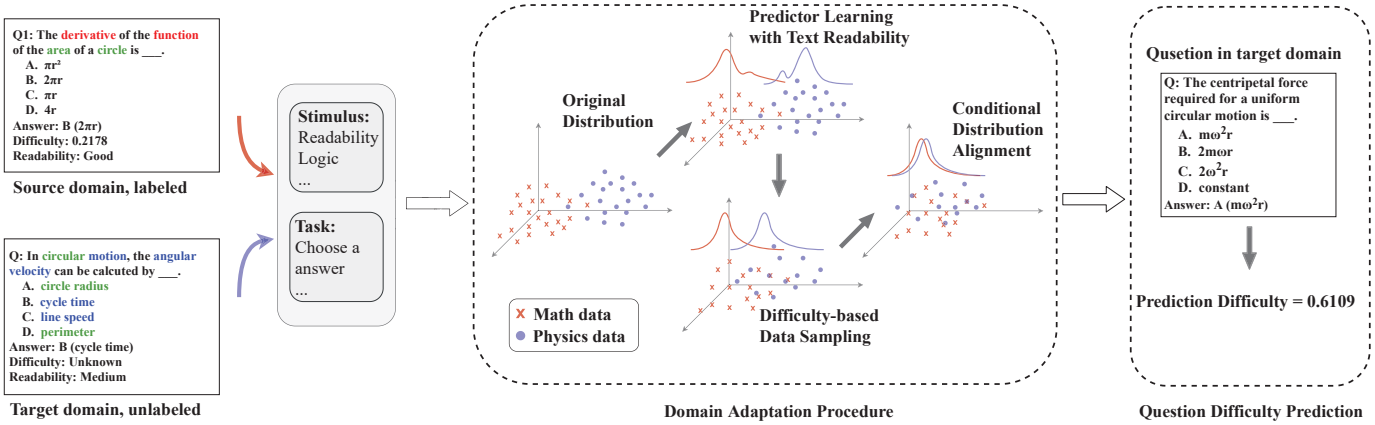


Fig. 2. An illustration of the cross-domain question difficulty prediction. In the *Domain Adaptation Procedure* box, points represent the distribution of features, and the curves represent the distribution of question difficulty. *left*: original distribution; *top*: learning the predictor with readability and generating the pseudo-label; *bottom*: difficulty-based sampling to pre-align the conditional distributions. *right*: explicitly alignment to obtain final adaptation result.

both some complex formulas and sentences, which are hard to understand. The other one is the barrier that students attempt to generate a response, named task difficulty. For multiple choice questions, it is about the difficulty of thinking over options and choosing the correct answer, and the difficulty depends on the similarity between the answer and distractors. Therefore, we extract common features by modeling stimulus and task difficulty. However, due to the distribution shift between different courses, we introduce domain adaptation to learn more transferable representations, where each course is considered as a domain.

There are two major challenges along this line. First, the professional terms of various courses are quite different. As shown in the left of Figure 2, red and blue words are domain-specific words that only appear in specific courses. Although there are some common words in green color, a large number of specific vocabulary is detrimental to the representation. Second, only aligning margin distributions to learn domain-invariant representations may cause difficulty mismatch between domains. Although some prior works use pseudo-labels to learn discriminative representations, the bias caused by falsely-pseudo-labeled samples will make it be vulnerable [7].

To address the challenges above, we propose a new framework called Stimulus and Task difficulty-based Adversarial Network (STAN). Firstly, we extract the stimulus and task difficulty representation according to semantic and readability features. Text readability is natural transferable since questions with worse readability are tend to be more difficult, thus we combine semantics and readability to bridge two courses. Secondly, we propose readability-enhanced pseudo-label (REPL), which utilize a pre-trained readability-based predictor to obtain more accurate pseudo-labels, especially in the beginning stage of learning. Thirdly, we adopt difficulty-based data sampling. After obtaining the REPL of the target domain, we choose data in the same difficulty distribution from two domains to pre-alignment. Finally, we train the network by a mean-squared-error loss and a discriminator which will align the distribution of different domains to learn the more transferable representation as shown in Figure 2. In summary, the key contributions of our work can be summarized as follows:

1) We propose a new QDP method with modeling educational concepts to improve both accuracy and domain-invariant when compared with prior work.
2) We develop a domain adaptation strategy to align the difficulty-conditioned distribution between two courses by both data sampling and explicit alignment in a continuous label space.
3) We conduct comprehensive experiments on diverse real-world questions of multiple course pairs to validate the effectiveness of STAN framework.

II. Related Work

Generally, the related work can be classified into the following two categories, i.e., question difficulty studies both in education and NLP field and domain adaptation.

A. Question Difficulty Prediction

Traditional Educational Psychology Method. How to measure question difficulty has been studied for a long time in the education field. Some psychological theories used students' feedback in the test to evaluate the difficulty of the question, such as Classical Test Theory (CTT) and Item Response Theory (IRT) [8]. Besides, some works focus on the relations between several factors of questions and the corresponding difficulty. For example, Cheng et al. [6] proposed a question difficulty framework comprising concepts such as content difficulty, stimulus difficulty, task difficulty and expected response difficulty. Sim et al. [9] found that the difficulty is related to item discrimination. Wang et al. [10] found that the setting of options affects the difficulty of the multiple-choice questions. However, the common limitation of these works is that they are subjective and labor-intensive. Therefore, all these works are not suitable for intelligent education systems.

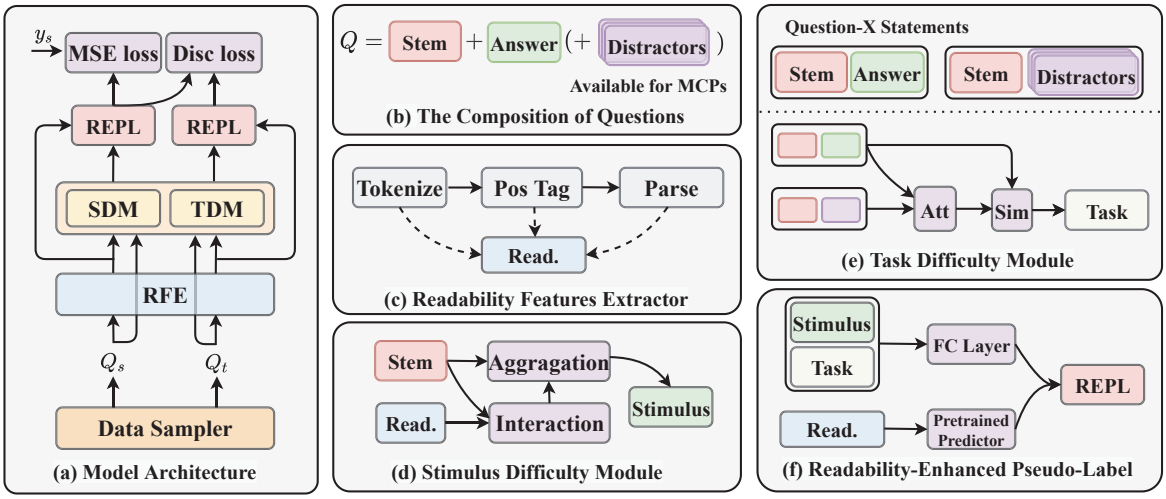


Fig. 3. The overview of the proposed STAN model.

Natural Language Processing Method. The QDP methods based on NLP and Representation Learning have recently attracted a lot of attention [11]–[13]. Relying on hand-craft features, Mothe et al. [14] studied that text linguistic features are closely related to question difficulty, such as word frequency, word diversity, average word and sentence length. There are many end-to-end framework studies on CNN / RNN attention mechanism. Ran et al. [15] proposed an option comparison network for multiple-choice problems (MCP) in READING problems, which compares options at word-level to identify their correlations. Qiu et al. [4] proposed a document enhanced attention-based network (DAN) to predict the difficulty of MCP in medical exams. Huang et al. [3] proposed a test-aware attention-based CNN framework to predict the difficulty of Reading questions. However, all these data-driven methods rely on a large scale question data with difficulty labels. And existed QDP models are designed for a specific course, which limits their application. Unlike the above solutions, our method utilizes the common features without any assumptions that are only valid in a specific course.

B. Domain Adaptation

There are lots of domain adaptation methods that learn the domain-invariant representation by statistical discrepancy [16]–[19] or discriminator-based method [20]–[23]. Most of the work did not pay attention to the labels distribution, which may cause label mismatch and poor transferable. In recent years, there are some works about this shortage [24]–[27].

Long et al. [21] first proposed to align the conditional distribution by the multilinear map and a joint domain discriminator. Since the ground-truth in the target domain is unavailable, the pseudo-label is used for explicit alignment. After that, there are many pseudo-label-based works. Cicek et al. [24] proposed an adversarial regularization method to improve the performance in image classification. Luo et al. [25] proposed self-adaptive adversarial loss for the image semantics task. Chen et al. [26] found that inaccurate pseudo-label will be damaging, and designed a novel training strategy with the confidence of pseudo-label. The above works all utilized explicit alignment, and Jiang et al. [27] proposed a implicit alignment method from an sampling perspective. Through sampling, the label distribution is aligned without false-pseudo-label risk.

A common assume of these works is that the label space is discrete in the classification setting. But in the QDP problem, the label space is continuous in interval $[0, 1]$, which will make the false-pseudo-label more harmful. And implicit alignment can't guarantee the label distribution is the same. Therefore our method is proposed to alleviate this disadvantage.

III. STAN Framework

In this section, we first formally define the problem of cross-domain question difficulty prediction. Then, we present the technical details of our QDP model. Finally, we present our domain adaptation method and training strategy.

A. Problem Setup

In this paper, we assume that there are two domains, D_s is the source domain which represents resource-rich course, and D_t is the target domain which refers to resource-poor course. We further assume that we are given a set of labeled training data $\mathbf{X}_s^l = \{(Q_s^i, y_s^i)\}_{i=1}^{n_s^l}$ and unlabeled training data $\mathbf{X}_s^u = \{Q_s^i\}_{i=1}^{n_s^u}$ from the source domain, where n_s^l and n_s^u are the number of labeled question data and unlabeled question data, respectively. Besides, we have a set of unlabeled data $\mathbf{X}_t = \{Q_t^i\}_{i=1}^{n_t}$ from the target domain, where n_t is the number of all question data which is unlabeled.

Each question Q_i has a question stem S, a correct answer A ,and three distractors $\{C_1, C_2, C_3\}$ if it's a multiple-choice problem. We define the question-answer statement S_a as the sentence formed by combining the question stem and its answer. For multiple-choice problems, there are question-distractors statements S_i. If the question is labeled, it has a difficulty attribute $y_i \in \mathbb{R}$ obtained from students' test logs.

Domain adaptation reuses labeled data from one domain to train model for a different but related domain. The goal of cross-domain question difficulty prediction is to train a robust model based on all labeled and unlabeled data and adopt it to predict the unlabeled data in the target domain.

B. Question Difficult Prediction Model

Our QDP model is designed not only to predict the difficulty of an unseen question, but also to extract more robust features of questions, which is important for improving performance in different domains. As shown in Figure 3(a), for a question, we firstly extract deep semantic features by Bi-LSTM, and readability features of question text by linguistic processing in **Readability Feature Extractor (RFE)**. Then, we model the stimulus and task difficulty representation and designed **Stimulus Difficulty Module (SDM)** and **Task Difficulty Module (TDM)** to get the representation of a question, which is utilized to predict an estimated value. Meanwhile, we pretrain a difficulty predictor only with readability features. After that, we combine there two prediction results to obtain **Readability-Enhanced Pseudo-Label (REPL)** as final estimation of difficulty. The details are showed as follows:

Question Text Encode. We use Bi-LSTM to encode each text sequence since it can learn not only low-level linguistic features (e.g., relation and structure), but also high-level semantic. For question stem $S = \{w_0, w_1, ..., w_{l_a}\}$. Then we take word w in S to its d-dimensional embedding $\mathbf{e}$ with an embedding matrix $\mathbf{E}$. This matrix is initialized by the Word2Vec tool [28], and $\mathbf{E} \in \mathbb{R}^{|V| \times d}$, where $|V|$ is the size of vocabulary. Finally, we extract the semantic features for sentence with Bi-LSTM. Specifically, we set $\overrightarrow{\mathbf{h}}^{(0)} = \overleftarrow{\mathbf{h}}^{(0)} = \{e_0, e_1, ...e_{L_s}\}$. At each position t, bidirectional hidden states are updated with input from previous layer:

$$\overrightarrow{h}_t^{(l)} = \mathrm{LSTM}(\overrightarrow{h}_{\mathrm{t}}^{(\mathrm{l}-1)}, \overrightarrow{h}_{\mathrm{t}-1}^{(\mathrm{l})}; \overrightarrow{\theta}_{\mathrm{LSTM}}), \quad (1)$$

$$\overleftarrow{h}_t^{(l)} = \mathrm{LSTM}(\overleftarrow{h}_{\mathrm{t}}^{(\mathrm{l}-1)}, \overleftarrow{h}_{\mathrm{t}-1}^{(\mathrm{l})}; \overleftarrow{\theta}_{\mathrm{LSTM}}). \quad (2)$$

Because the hidden states can capture the context and linguistic information, the encode result of the sentence is $\mathbf{s} \in \mathbb{R}^{L \times 2d}$ where $\mathbf{s}_t \in \mathbb{R}^{2d}$:

$$\mathbf{s} = \mathrm{concat}(\overrightarrow{\mathbf{h}}^{(l)}, \overleftarrow{\mathbf{h}}^{(l)}). \quad (3)$$

And for other text, such as question-distractor statement S_a and question-distractor statement S_i, we also can obtain Bi-LSTM outputs $\mathbf{s}_a \in \mathbb{R}^{L_a \times 2d}, \mathbf{s}_i \in \mathbb{R}^{L_i \times 2d}$ via the same encoding process. L_s, L_a and L_i are the lengths of the question stem and question-answer and question-distractor statements respectively.

Readability Features. In order to obtain the readability features of a given question stem, we utilize the StanfordCoreNLP toolkit [29] to finish tokenization, part-of-speech tagging and syntax parsing as shown in Figure 3. Then, from the levels of vocabulary, sentence and syntax, We construct a multi-level linguistic feature set with 12 features to measure the readability of the question text:

TABLE I
QUESTION TEXT READABILITY FEATURES.

Level	Feature
Word	Average word length Frequency weighted average word length
Part of speech	Proportions of verb, noun, adjective
Sentence	The sum of word lengths Number of words TTR
Syntax	Syntax tree height Dependence distance
Comprehensive index	Flesch reading ease Gunning-Fog index

- Words are the most basic unit of sentence formation in language and word recognition is an important process in question reading [30]. Word recognition in reading is affected by many factors, such as the the average length of a word, and its frequency weighted average.
- In part of speech, we count the proportions of verbs, nouns, adjectives and symbols in the question text.
- Syntax analysis is divided into two aspects: syntax analysis based on phrase structure and syntactic analysis based on dependence. The former analyzes the structural relationship between the phrases in the sentence in a tree manner [31]. We adopt the height of syntax tree as a feature; the latter reveals the syntax dependence between the words in the sentence on the basis of identifying the main predicate in the sentence, such as the modification relationship and the dominance relationship [32], and we adopt the average dependence distance.
- Moreover, we take two classic and popular readability scores: Flesch reading ease and Gunning-Fog index [33], which measure the complexity of the questions comprehensively.

In summary, the linguistic features is shown in Table I. For question Q, the readability feature of its stem is $\mathbf{f}_r \in \mathbb{R}^{N_r}$, where N_r is the number of readability features.

Stimulus Difficulty Module. Stimulus difficulty refers to the difficulty about comprehending the words and phrases in a question and the information that accompanies the question i.e. note and table. For example, questions with simple easy-to-understand descriptions are usually easier than those that need careful comprehension. Consider that stimulus difficulty is determined by question complexity, knowledge depth and text readability, we combine the average of semantic features of stem $\bar{\mathbf{s}}$ and readability features $\mathbf{f}_r$ as the representation of stimulus difficulty as shown in Figure 3(d). First, we take the weighted tensor product of $\bar{\mathbf{s}}$ and $\mathbf{f}_r$ as interaction matrix $\mathbf{Ir}$. Then we take the attention mechanism to aggregate the matrix to stimulus representation $\mathbf{f}_s$:

$$\mathbf{Ir} = \mathbf{W_s} \circ (\bar{\mathbf{s}} \otimes \mathbf{f}_r), \quad (4)$$

$$\mathbf{f}_s = \mathrm{Attention}(\bar{\mathbf{s}}, \mathbf{Ir}) \times \mathbf{Ir}, \quad (5)$$

$$\text{Attention}(\bar{\mathbf{s}}, \mathbf{Ir}) = \text{Softmax}(\frac{\bar{\mathbf{s}} \times \mathbf{Ir}^T}{\sqrt{d_k}}), \tag{6}$$

where $\circ$ and $\otimes$ denote the Hadamard product and the Kronecker product respectively, d_k is the dimension of text encoder, and $\mathbf{W_s}$ is a trainable parameter.

Task Difficulty Module. Task difficulty refers to difficulty students face when they finish the task. For multiple-choice problems (MCP), students need to choose from multiple options. Hence the task difficulty can be represented by confusion of distractors. For fill-in-blank problems (FBP), students need to write an answer to the blank, thus the difficulty can be represented by the similarity between its stem and question-answer statement. In detail, for MCP, we use the word-level matching [4] to measure the similarity between distractors and answer. Specifically, the question-answer statement is compared with all question-distractors one by one to collect the information describing the misleading information. Then, the distraction information gathered from the question-distractor statement $\mathbf{s_i}$ is computed as:

$$\mathbf{A}_i = \text{Attention}(\mathbf{s}_a, \mathbf{s}_i) \times \mathbf{s}_i, \tag{7}$$

$$\mathbf{M}_i = \text{concat}(\mathbf{s}_a - \mathbf{A}_i, \mathbf{s}_a \circ \mathbf{A}_i), \tag{8}$$

$$\hat{\mathbf{s}}_\mathbf{i} = \text{ReLU}(\mathbf{W}_m \mathbf{M}_i + \mathbf{b}_m), \tag{9}$$

where $\mathbf{W_m}$ and $\mathbf{b_m}$ are trainable parameters of the predictor. $\hat{\mathbf{s}}_\mathbf{i}$ represents the misleading information from the i-th distractor. Then we gather all the misleading information collected from distractors together as:

$$\mathbf{f}_t = \text{concat}(\hat{\mathbf{s}}_\mathbf{1}, \hat{\mathbf{s}}_\mathbf{2}, \hat{\mathbf{s}}_\mathbf{3}). \tag{10}$$

For FBP, we measure the similarity between question stem and question-answer statement. Hence in above procedure, the distractors $\mathbf{s}_i$ is replaced with stem $\mathbf{s}$ and we obtain its task representation $\mathbf{f}_t$.

Readability-Enhanced Difficulty Prediction. It's effective that utilize a pre-trained readability-based predictor to enhance our model in the training stage. First, we predict readability difficulty y_r. Then, based on the concatenated stimulus and task representation $\mathbf{f}_q$, we use a fully connected layer to predict the model difficulty:

$$y_d = \mathbf{W_p} \mathbf{f}_q + b_p, \tag{11}$$

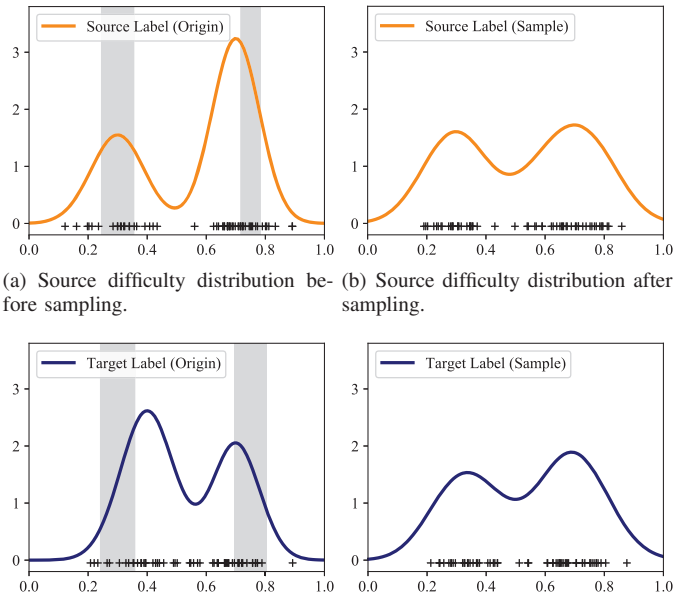
where $\mathbf{W_p}$ and $\mathbf{b_p}$ are trainable parameters. Finally, we use a progress variable α from $1 \to 0$ to adjust the weight of these two predictions:

$$\hat{y} = \alpha \times y_r + (1 - \alpha) \times y_d, \tag{12}$$

where $\hat{y}$ is the REPL that is more accurate than simple pseudo-label. Each training 5000 steps, the α is reduced by 1/2. Note that in the test stage, we set $\alpha = 0$.

C. Adaptation Method

As mentioned above, the STAN framework is trained on labeled question data from the source course, and unlabeled data from the target course. To prevent prediction mismatch, we adopt a simple but effective difficulty-based sampler to align two domains implicitly. Then we utilize all the data X_s and X_t to train a domain classifier which learns transferable features, and minimize the prediction loss with labeled data (X_s, y_s) simultaneously.



(a) Source difficulty distribution before sampling. (b) Source difficulty distribution after sampling.

(c) Target difficulty distribution before sampling. (d) Target difficulty distribution after sampling.

Fig. 4. An illustration of difficulty-based sampling method.

Difficulty-based Implicit Alignment. We use a sampling method to align the two domains implicitly. First, we take the common interval (y_{min}, y_{max}) of the difficulty value of the two domains. Then, we uniformly sample N values $\{y_i\}_{i=1}^{i=N}$ from this interval. After that, for each sample value y_i, we randomly choose data (X_s, y_s) satisfying that $y_s \in [y_i - \delta, y_i + \delta]$ as shown in Figure 4(a) and Figure 4(c). The radius of the neighbourhood δ is determined by the estimated density function. With Epanechnikov kernel, the density function can be estimated by:

$$\widehat{f}_h(y) = \frac{1}{N_s} \sum_{i=1}^{N_s} [1 - \frac{(y - y_{si})^2}{h^2}], \tag{13}$$

$$\delta(y_i) = w / \widehat{f}_h(y_i), \tag{14}$$

where $\widehat{f}_h(x)$ is the final estimated density function, h is the bandwidth parameter, and w determines the sampling range. The function $[1 - \frac{x^2}{h^2}]$ is Epanechnikov kernel, which is effective in scoped data distribution with less border effect. In our experiments, we set $h = 0.07$ and $\delta = 0.2$.

For unlabeled target domain data, we predict pseudo-labels $(X_t, \hat{y}_t)$ mentioned above to replace its absence of ground-truth difficulty. Finally, we obtained the difficulty-aligned batch $(X_s, y_s, X_t, \hat{y}_t)$ where y_s and $\hat{y}_t$ is closed enough as shown in Figure 4(b) and Figure 4(d), we use it to train STAN framework and repeat this process until the model converges.

Training Strategy. In adversarial network training, there is a domain classifier to discriminate which domain the data come from, and a difficulty predictor to estimate a difficulty. Since the above two optimization goals are opposite, the adversarial training is used to train the overall network as a minimax game. With the gradient reverse layer, the feature extractor receives gradients from both the discriminator and the predictor, and gradients of the discriminator are negative. Conditional domain discriminator takes features $\mathbf{f_q}$ and its pseudo label $\hat{y}$ as input. Formally, given a conditional discriminator $D(Q;\theta_d)$ with parameters θ_d, a feature extractor $F(Q;\theta_f)$ with parameters θ_f, a difficulty predictor $G(Q;\theta_g)$ with parameters θ_g. Then, distribution alignment will be applied to reduce the shift between the two domains is implemented by:

1) Training Conditional Domain Discriminator. To align the conditional distribution $P(F(Q_s),\hat{y}_s)$ and $P(F(Q_t),\hat{y}_s)$, we train the domain discriminator with parameter θ_d. The loss function of domain discriminator can be written as:

$$\begin{aligned}\mathcal{L}_{\text{CDA}} = E_{Q_s\sim X_s}[\log(1-D(F(x_s),\hat{y}_s))] \\ +E_{Q_t\sim X_t}[\log(D(F(Q_t),\hat{y}_t))].\end{aligned} \tag{15}$$

2) Minimize Prediction Loss and Confusing Discriminator. To make the prediction $\hat{y}_s$ more closed to the true difficulty y_s, we train the predictor with parameters θ_f and θ_g, while confusing the domain discriminator with parameter θ_d. The loss function of train feature extractor and predictor can be written as:

$$\begin{aligned}\mathcal{L}_{\text{MSE}} = E_{(Q_s,y_s)}[(G(F(Q_s))-y_s)^2 \\ -\log(1-D(F(x_s),\hat{y}_s))] - E_{Q_t}[\log(D(F(Q_t),\hat{y}_t))].\end{aligned} \tag{16}$$

In summary, using the gradient reversal layer, we integrate the aforementioned two terms into a whole objective function as below:

$$\begin{aligned}\mathcal{L}_{\text{overall}} = E_{(Q_s,y_s)}[(G(F(Q_s))-y_s)^2 \\ +\lambda\log(1-D(F(Q_s),\hat{y}_s))] + E_{Q_t}[\lambda\log(D(F(Q_t),\hat{y}_t))],\end{aligned} \tag{17}$$

where λ is trade-off hyper-parameter. In GRL, this parameter is usually changed with a specified schedule during training. In our network, we set $\lambda = 1$ fixed for all experiments. In summary, the labeled source samples and unlabeled target samples are used to train our model. Then we adopt the learned predictor to estimate the difficulty of the unlabeled target questions.

D. Theoretical Understanding

We give theoretical motivations of the STAN with the similar formalism of the domain adaptation theory. Formally, given source domain S and target domain T, let $\mathcal{H}$ be the hypothesis space, the error $\epsilon_T(h)$ of hypothesis $h \in \mathcal{H}$ on the target domain is bounded by three terms [34]: the error $\epsilon_S(h)$ on the target domain, the distribution discrepancy $|\epsilon_S(h,h^*) - \epsilon_T(h,h^*)|$, and the optimal shared error $\lambda = \epsilon_S(h^*) + \epsilon_T(h^*)$.

The shared error λ is considered to be a constant, but it is not guaranteed that λ will be small even $\epsilon_S(h)$ and $|\epsilon_S(h,h^*) - \epsilon_T(h,h^*)|$ are minimal [26]. To address this problem, We give the bound of $\epsilon_T(h)$ from the perspective of pseudo-labeling function $\hat{h}$:

$$\begin{aligned}\epsilon_T(h) &\le \epsilon_T(\hat{h}) + \epsilon_T(h,\hat{h}) \\ &\le \epsilon_T(\hat{h}) + \epsilon_S(h,\hat{h}) + |\epsilon_S(h,\hat{h}) - \epsilon_T(h,\hat{h})| \\ &\le \epsilon_S(h) + [\epsilon_S(\hat{h}) + \epsilon_T(\hat{h})] + |\epsilon_S(h,\hat{h}) - \epsilon_T(h,\hat{h})|.\end{aligned} \tag{18}$$

The bound can be divided into three parts which are correspond to our different motivations:

- Error in source domain. The discriminator confusion will disturb the optimization of the error in source domain. The proposed adaptive method alleviates this disadvantage by difficulty-aligned sampling and conditional adversarial learning.
- Error of pseudo-label. The proposed REPL aims to improve the accuracy of pseudo-label, especially at the beginning of learning, which can be shown in experiments. Therefore, this part can be reduced compared to simple pseudo-label.
- Domain discrepancy. This term is optimized in the process of training the conditional discriminator. Long et al. [21] have given the theory of domain discrepancy optimization. Because the QDP problem doesn't meet the setting of the classification, we give a similar result by redefining the error of a hypothesis as follow.

Define a loss hypothesis space $\Psi = \{\psi = \mathbb{1}[|h(f) - \hat{y}| \le \delta] : h \in \mathcal{H}\}$, where ψ is a function: $(f,\hat{y}) \mapsto \{0,1\}$. Then, we found that even if the loss hypothesis is different, it can still give the same result as in [21]. First, by considering Ψ distance, the distribution discrepancy can be bounded as that:

$$d_\Psi(S,T) \ge |\epsilon_S(h,\hat{h}) - \epsilon_T(h,\hat{h})|. \tag{19}$$

Then, due to the function ψ and D are both $(f,\hat{y}) \mapsto \{0,1\}$, by the fact that $\Psi \subset \mathcal{D}$, the Ψ distance can be bound by training conditional domain discriminator:

$$\begin{aligned}d_\Psi(S,T) &\le \sup_{D\in\mathcal{D}} |\mathbb{E}_S[D(f,\hat{y}) \ne 0] - \mathbb{E}_T[D(f,\hat{y}) \ne 0]| \\ &\le \sup_{D\in\mathcal{D}} [\mathbb{E}_S[D(f,\hat{y}) = 1] + \mathbb{E}_T[D(f,\hat{y}) = 0]].\end{aligned} \tag{20}$$

According to Eq.(19) and Eq.(20), the domain discrepancy $|\epsilon_S(h,\hat{h}) - \epsilon_T(h,\hat{h})|$ can be optimized with discriminator even in the QDP task setting.

IV. Experiments

In order to verify the effectiveness of our method, we conduct comprehensive experiments on multiple courses. First, we compare with baseline approaches to show that our framework STAN achieves the best performance. Then, we conduct an ablation study to show the effectiveness of each module, especially our domain adaptation method. After that, we show our REPL is more effective than using simple pseudo-label

TABLE II
THE STATISTICS OF THE MATHEMATICS DATASET.

Statistics	Raw	Processed
# of test logs	651,944,494	27,169,290
# of questions	688,598	20,000
# of students	9,670,747	4,130,308
Average test logs per question	947	1,358
Average test logs per student	67	7

TABLE III
THE QUESTIONS NUMBER OF FOUR DATASETS.

	Multi-Choice	Fill-in-Blank
Math (**M**)	227,899	148,684
Physics (**P**)	154,442	37,719
History (**H**)	309,179	29,814
Moral Edu (**ME**)	115,547	21,507

by comparing the accuracy. Finally, we show the relationship between the performance and the similarity between datasets.

A. Dataset Preparation

Our dataset includes four courses: Mathematics, Physics, History and Moral Education. We collect a large number of questions with difficulty labels from Zhixue[1] and LUNA[2] online education system. The Mathematics dataset is collected from Zhixue, and the Physics, History and Moral Education datasets are collected from LUNA. The statistics on the Mathematics dataset is shown in Table II, and the question numbers of all the courses are shown in Table III.

Data Preprocess. For data with test logs, we firstly drop duplicate logs. In the online system, some students solve the same question multiple times. We delete all the duplicate test logs and only keep the first attempt to make the difficulty more truthful. After dropping duplicates, there is only one test log for a student and a specific question. Then, we drop the questions that are related to few test logs. If only a small number of students try to solve the problem, the difficulty of the question will be very unstable. To avoid this, we filter the questions having no more than 80 test logs.

Question Difficulty. After removing duplicate records and filtering questions with fewer records, we calculate the real difficulty of questions. Because the question difficulty cannot be directly observed, we obtain the real difficulty of each question from the test logs. Instead of calculating the percentage of students who answer the question wrongly, we divide the student's average score of a question by the total score of this question to get the proportion of incorrect answers. This approach can handle partially correct situations. The real difficulty R_i of Q_i computed as follows:

$$R_i = \frac{G_i}{N \times g_i}, \tag{21}$$

where G_i is the sum of all students' scores for the question, N is the number of answers and g_i is the full score of this question.

B. Experimental Setup

Network Architecture. Firstly, all word embeddings are 128-d vectors pre-trained by the Word2vec tool [28]. Secondly, for the question text encoder, we adopt a Bi-LSTM with 2 layers, and the sizes of hidden states in these modules are set to 128. The parameters of the Bi-LSTM are shared between the processing of stem, question-answer statements and question-distractor statements. For the predictor and discriminators, we use 3 layers of fully-connection with batch normalization [35]. Finally, all weight matrices are randomly initialized by Kaiming uniform distribution [36]. All biases are set to zeros except for prediction layer, and bias of prediction layer is randomly initialized by uniform distribution in the range between $-\sqrt{\text{fan\_in}}$ and $\sqrt{\text{fan\_in}}$, where $\sqrt{\text{fan\_in}}$ are the numbers of input features.

All models are implemented by PyTorch and all experiments are run on a Linux server with two 2.20GHz Intel(R) Xeon(R) Processor E5-2699 v4 CPUs and a Tesla P100 PCIe GPU. Our codes are available in https://github.com/bigdata-ustc/STAN.

Training Settings. We use Adam optimizer and the learning rate is set as 0.001. The dropout [37] is introduced between layers with probability 0.2 to prevent overfitting. Our model is trained with a batch size of 32. For all datasets, we split 75% of the data as the training set and the rest as the test set.

Evaluation Metrics. To measure the performance of our method, we use the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). RMSE and MAE are widely used in regression tasks, which are used to measure the distance between the predicted difficulty and the ground truth difficulty in the target domain. Ranges of both RMSE and MAE are $(0, +\infty)$, and the smaller they are, the better performance the results have.

Baseline Approaches Before comparing model performance, we select a comprehensive baseline approaches. Firstly, we choose source-only baselines LSTM and CNN with attention (ACNN), which are used in QuesNet [12] and TACNN [3], respectively. In addition, pre-training methods are widely used in the NLP field, of which BERT [38] is the most representative method. Secondly, three basic domain adaptation methods are adopted as baselines, includes the pioneering work deep domain adaptation MK-MDD based DAN [17], discriminator-based DANN [39] and ADDA [40]. Then, we selected two adaptation methods JAN [18] and CDAN [21] which consider joint feature alignment. Finally, we compared with a recent work MDD [19], which is. Note that for all DA baselines, the feature extractors are the same as our STAN.

C. Performance Comparison

To observe how our STAN perform in different courses, we choose four course pairs: M→P (MCP), H→ME (MCP), M→P (FBP) and H→ME (FBP). Although data in the target domain is unlabeled, in order to ensure that there are no

[1]https://www.zhixue.com/
[2]https://luna.bdaa.pro/

TABLE IV
THE PERFORMANCE RESULTS.

Domain	M→P (MCP)		H→ME (MCP)		M→P (FBP)		H→ME (FBP)		Avg.	
Method	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
LSTM	0.2963	0.3615	0.2845	0.3432	0.3011	0.3695	0.2844	0.3781	0.2916	0.3673
ACNN	0.2941	0.3372	0.2902	0.3517	0.2934	0.3661	0.2865	0.3576	0.2911	0.3531
BERT	0.2915	0.3497	0.2812	0.3202	0.2987	0.3524	0.2738	0.3359	0.2863	0.3396
DANN	0.2895	0.3415	0.2738	0.3128	0.2950	0.3481	0.2816	0.3471	0.2850	0.3374
ADDA	0.2784	0.3342	0.2849	0.3160	0.2843	0.3263	0.2794	0.3310	0.2818	0.3269
DAN	0.2961	0.3546	0.2593	0.3211	0.2901	0.3407	0.2896	0.3409	0.2838	0.3393
JAN	0.2710	0.3238	0.2546	0.3093	0.2765	0.3328	0.2690	0.3282	0.2678	0.3235
CDAN	0.2707	0.3175	0.2672	0.3116	0.2628	0.3289	0.2325	0.2786	0.2570	0.3208
MDD	0.2611	0.3072	0.2369	0.2982	0.2429	0.3173	0.2443	0.2947	0.2463	0.3044
STAN	**0.2483**	**0.2945**	**0.2364**	**0.2857**	**0.2407**	**0.2853**	**0.2281**	**0.2709**	**0.2384**	**0.2841**

TABLE V
THE RESULTS OF THE ABLATION STUDY OF OUR METHOD.

	QDP Model		Domain Adaptation		M→P (MCP)		H→ME (MCP)	
	Stimulus Difficulty	Task Difficulty	REPL	Sample	MAE	RMSE	MAE	RMSE
1	✓				0.2994	0.3358	0.2696	0.3029
2		✓			0.3016	0.3372	0.2715	0.3177
3	✓	✓			0.2912	0.3409	0.2602	0.3104
4	✓		✓	✓	0.2518	0.3022	0.2459	0.3065
5	✓	✓	✓		0.2547	0.3105	0.2471	0.2982
6	✓	✓		✓	0.2571	0.3084	0.2402	0.2921
7	✓	✓	✓	✓	**0.2483**	**0.2945**	**0.2372**	**0.2858**

overlaps between the questions in training sets and testing sets, we remove the questions in training sets with the same documents which exist in testing sets. Thus, the questions in testing sets are all new questions in target domain. Table IV shows the overall QDP results of all models. The accuracy values reported here are the average of five-times test. From the results, we can get several observations:

1) In summary, we can see that STAN performs best, source-only methods e.g. LSTM and ACNN perform worst, which means that when the model trained from a course directly applied to another course, its performance will decrease obviously.
2) With domain adaptation, the performance of DAN, ADDA and DANN is improved comparing to source-only methods. But they're beaten by STAN because of training examples mismatch. Meanwhile, BERT does not perform as well as STAN, which indicates that the pre-trained BERT which aims for the general NLP task is not the best model for the QDP task, which contains a large number of symbols and special expressions.
3) We can see that he MDD shows a good performance on our datasets, and the models with joint alignment (STAN, JAN, CDAN) perform better than those with marginal alignment (DAN, ADDA, DANN). And the STAN performs best because of the REPL and difficulty-based implicit alignment. This observation suggests that alignment of conditional distribution between question features and difficulties are effective.

D. Ablation Study

The ablation study is conducted to highlight the individual contribution of each components in STAN, and the result is shown in Table V. From the results, we can find that:

1) Overall, the more terms are added, the better the performance is. Our complete model (Row 7) shows the better performance than all the others ablation methods.
2) The first two rows only keep stimulus and task difficulty representation, respectively. By comparing Row 1 and Row 2, we find that stimulus difficulty outperforms the task difficulty. This observation suggests that the stimulus module can reflect the difficulty better in different courses than the task module. The reason is that the stimulus representation contains the readability features of the question text, which is more common in different courses, especially in M-P and H-ME course pairs which have similar question compositions.
3) After DA is deleted in Row 3, the STAN degenerates into a pre-training method, which is comparable to pre-training methods such as LSTM and ACNN in Table IV. We find that STAN without DA performs better than other methods. This observation shows that STAN captures common difficulty-related features better.
4) Performance in Rows 4-7 is better than Rows 1-3, so the domain adaptation can help reduce the domain discrepancy and improve performance. Besides, Rows 5 and Rows 6 show that the addition of the REPL and sampling alignment bring a performance improvement.

TABLE VI
IMPROVEMENT OF REPL COMPARED WITH PL.

Domains	M→P (MCP)				
Steps	1000	2000	3000	4000	Avg.
PL	0.4337	0.3889	0.3196	0.2820	0.3561
REPL	0.3281	0.3015	0.2965	0.2774	0.3009
Improve	24.35%	22.47%	7.23%	1.63%	15.50%
Domains	H→ME (MCP)				
Steps	1000	2000	3000	4000	Avg.
PL	0.4417	0.3644	0.3358	0.2960	0.3595
REPL	0.3560	0.3221	0.3017	0.2841	0.3160
Improve	19.40%	11.61%	10.15%	4.02%	12.10%

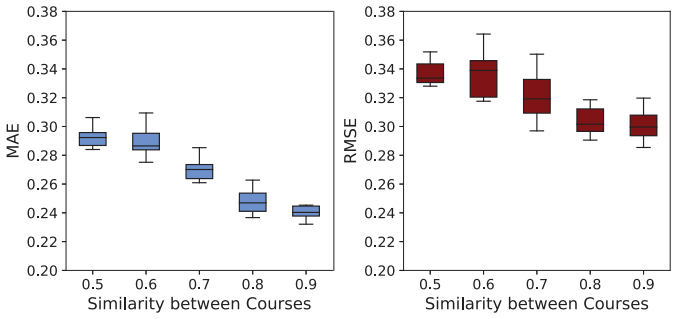


Fig. 5. The relationship between similarity between courses and performance.

E. Impact of Readability-Enhanced Pseudo-Label

In this section, we analyze the effect of REPL in domain adaptation. we explore the MAE error of REPL and simple PL at four training steps to confirm the REPL does improve the pseudo-label accuracy in target domain, which can guarantee the generalization bound. The results are shown in Figure VI.

Experimental results show that the readability information improves the accuracy of pseudo-labels during the training stage. At the beginning of the training stage, the exists of course terms and domain shift make deep semantic features almost unavailable. Due to the influence of the initialization parameters, it is not even guaranteed that the value of simple PL is between [0, 1]. In this stage, it's obvious that pseudo-label without readability will cause serious mismatch and error accumulation because of its inaccuracy. The pre-trained readability model can alleviate this shortage, thus the REPL improves performance by about 20%. As the training continues, the semantic features are gradually aligned, and we reduce the proportion of the readability part. Therefore when semantic features are better than readability, the former is almost ignored.

F. Impact of Courses Similarity

In domain adaptation, our model should extract domain-invariant features that represent semantic information in question. Since each question is associated with domain-specific terminologies, so the words similarity between different courses will affect the transfer performance. In order to measure this impact, we randomly sample 10,000 questions from two different courses in our datasets, and calculate the similarity between representations of words from these two courses. Specifically, for a course C_i, we first collect n_i words $\{w_1, w_2, ..., w_{n_i}\}$ from sample questions. Note that we remove the stopwords to capture course-specific words. Then we utilize pre-trained word embedding to output its features $\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{n_i}\}$. Finally, the representation vector of course C_i and cosine similarity between course C_i and course C_j are defined as:

$$\mathbf{C}_i = \sum_{i=1}^{n_i} \frac{\mathbf{v}_i}{n_i}, \tag{22}$$

$$\text{sim}(\mathbf{C}_i, \mathbf{C}_j) = \frac{\mathbf{C}_i \cdot \mathbf{C}_j}{||\mathbf{C}_i|| \cdot ||\mathbf{C}_j||}. \tag{23}$$

As illustrated in the Figure 5, for each similarity from 0.5 to 0.8, we sample 10 times and make boxplots of MAE in blue and RMSE in red. Even these results are better than source-only training without DA, we can observe that the more similar source and target domain, the better transfer performance. This observation shows that if the two courses are similar, such as math and physical, we can obtain a good performance. And for these not similar, such as math and history, using only text readability and semantics is not enough. In this situation, the textual expressions of the two courses are different. For example, there are lots of formulas in math, which rarely appear in history. Therefore additional domain information is needed, which we will explore in the future.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a framework for cross-domain question difficulty prediction. Specifically, we first designed feature extractors based on stimulus and task difficulty representation that are common in different courses. Then, we proposed readability-enhanced pseudo-label and difficulty-based sampling methods to implicit alignment. Finally, we utilized an adversarial learning method to learn more transferable features. From the experimental results, we showed that our STAN framework outperforms other prior domain adaptation and QDP work in accuracy. And we conducted the ablation study to show the improvement of each module. Besides, we explored the impact of pseudo-label and course similarity.

In the future, there are still some directions for further studies. First, besides the text semantic and readability, we will consider the other factors affecting the stimulus difficulty, e.g. images and tables attached to the questions. Second, as our STAN is a general framework, we will test its performance on other kinds of domains (e.g. learning period) and meanwhile, on the education applications in other tasks, such as the similarity measurement of questions and problem solver [41].

Acknowledgement. This research was partially supported by grants from the National Natural Science Foundation of China (Grants No. 61922073, U20A20229 and 62106244) and the Foundation of State Key Laboratory of Cognitive Intelligence, iFLYTEK, P.R.China (No. CI0S-2020SC05).

REFERENCES

[1] Q. Liu, Z. Huang, Y. Yin, E. Chen, H. Xiong, Y. Su, and G. Hu, "Ekt: Exercise-aware knowledge tracing for student performance prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 1, pp. 100–115, 2019.

[2] Y. Susanti, T. Tokunaga, H. Nishikawa, and H. Obari, "Controlling item difficulty for automatic vocabulary question generation," *Research and practice in technology enhanced learning*, vol. 12, no. 1, pp. 1–16, 2017.

[3] Z. Huang, Q. Liu, E. Chen, H. Zhao, M. Gao, S. Wei, Y. Su, and G. Hu, "Question difficulty prediction for reading problems in standard tests," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[4] Z. Qiu, X. Wu, and W. Fan, "Question difficulty prediction for multiple choice problems in medical exams," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 139–148.

[5] "Statistical analysis of the grade data of postgraduate courses," http://yjsy.uibe.edu.cn/cms/infoSingleArticle.do?articleId=3866& columnId=2154, accessed: 2015-10-26.

[6] L. S. Cheng, "On varying the difficulty of test items," in *The 32nd Annual Conference of the International Association for Educational Assessment*, 2006.

[7] C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, and J. Huang, "Progressive feature alignment for unsupervised domain adaptation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[8] A. F. De Champlain, "A primer on classical test theory and item response theory for assessments in medical education," *Medical education*, vol. 44, no. 1, pp. 109–117, 2010.

[9] S.-M. Sim and R. I. Rasiah, "Relationship between item difficulty and discrimination indices in true/false-type multiple choice questions of a para-clinical multidisciplinary paper," *Annals-Academy of Medicine Singapore*, vol. 35, no. 2, p. 67, 2006.

[10] L. Wang, "Does rearranging multiple-choice item response options affect item and test performance?" *ETS Research Report Series*, vol. 2019, no. 1, pp. 1–14, 2019.

[11] S. A. Crossley, S. Skalicky, M. Dascalu, D. S. McNamara, and K. Kyle, "Predicting text comprehension, processing, and familiarity in adult readers: New approaches to readability formulas," *Discourse Processes*, vol. 54, no. 5-6, pp. 340–359, 2017.

[12] Y. Yin, Q. Liu, Z. Huang, E. Chen, W. Tong, S. Wang, and Y. Su, "Quesnet: A unified representation for heterogeneous test questions," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1328–1336.

[13] Z. Huang, X. Lin, H. Wang, Q. Liu, E. Chen, J. Ma, Y. Su, and W. Tong, "Disenqnet: Disentangled representation learning for educational questions," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 696–704.

[14] J. Mothe and L. Tanguy, "Linguistic features to predict query difficulty," in *ACM Conference on research and Development in Information Retrieval, SIGIR, Predicting query difficulty-methods and applications workshop*, 2005, pp. 7–10.

[15] Q. Ran, P. Li, W. Hu, and J. Zhou, "Option comparison network for multiple-choice reading comprehension," *arXiv preprint arXiv:1903.03033*, 2019.

[16] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2010.

[17] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*. PMLR, 2015, pp. 97–105.

[18] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *International conference on machine learning*. PMLR, 2017, pp. 2208–2217.

[19] Y. Zhang, T. Liu, M. Long, and M. Jordan, "Bridging theory and algorithm for domain adaptation," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7404–7413.

[20] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of The 32nd International Conference on Machine Learning*, 2015, pp. 1180–1189.

[21] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," *arXiv preprint arXiv:1705.10667*, 2017.

[22] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8503–8512.

[23] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4893–4902.

[24] S. Cicek and S. Soatto, "Unsupervised domain adaptation via regularized conditional alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1416–1425.

[25] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2507–2516.

[26] C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, and J. Huang, "Progressive feature alignment for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 627–636.

[27] X. Jiang, Q. Lao, S. Matwin, and M. Havaei, "Implicit class-conditioned domain alignment for unsupervised domain adaptation," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4816–4827.

[28] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[29] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 2014, pp. 55–60.

[30] A. J. Stenner, "Measuring reading comprehension with the lexile framework." in *The Fourth North American Conference on Adolescent/Adult Literacy*. ERIC, 1996.

[31] S. E. Schwarm and M. Ostendorf, "Reading level assessment using support vector machines and statistical language models," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 2005, pp. 523–530.

[32] E. Gibson, "The dependency locality theory: A distance-based theory of linguistic complexity," *Image, language, brain*, vol. 2000, pp. 95–126, 2000.

[33] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel," Naval Technical Training Command Millington TN Research Branch, Tech. Rep., 1975.

[34] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine Learning*, vol. 79, no. 1-2, pp. 151–175, 2010.

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[39] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[40] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.

[41] X. Lin, Z. Huang, H. Zhao, E. Chen, Q. Liu, H. Wang, and S. Wang, "Hms: A hierarchical solver with dependency-enhanced understanding for math word problem," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4232–4240.