2024 the 9th International Conference on Big Data Analytics

Adaptive Cognitive Diagnosis under Distribution Shift with a Double-level Adversarial Framework

Yuhong Zhang
School of Computer and
Information Technology
Hefei University of Technology
Hefei, China
zhangyh@hfut.edu.cn

Chenyang Si
School of Computer and
Information Technology
Hefei University of Technology
Hefei, China
2021171172@mail.hfut.edu.cn

***Abstract*—Cognitive diagnosis is an important task in intelligent education, and it aims to diagnose the mastery level of students for knowledge concepts by modeling the interaction among student proficiency, exercise difficulty and exercise discrimination. Existing methods assume the training and test data follow identical distribution. However, there does exist the distribution shift between training and test data. Moreover, the manifestation of this distribution shift is multidimensional. To this end, we propose the adaptive cognitive diagnosis framework and design a double-level adversarial framework to address the multidimensional distribution shifts in cognitive diagnosis. Specifically, the multidimensional distribution shifts are divided into two levels, the difference on the representation level and the difference on the cognitive level. The adversarial at the representation level is designed for the adaptive learning of student proficiency, exercise difficulty and discrimination. And the adversarial at the cognitive level is for adaptive training of the diagnostic framework. Extensive experimental results on four datasets demonstrate the effectiveness of our method.**

***Keywords*—Intelligent education, Cognitive diagnosis, Domain adaptation, Adversarial framework**

I. Introduction

Massive data has emerged in educational teaching and campus activities. And mining the education big data is of great value in driving educational development. Cognitive diagnosis (CD) is an important task in data mining of education, which models knowledge acquisition of students, i.e. their cognitive state, based on their response logs of exercises and the association of exercises with knowledge concepts.

However, the existing cognitive diagnosis methods (CDMs) require the training and test sets are identically distributed, which is usually difficult to hold in applications. Referring to the notion of distribution shift in domain adaptation (DA) [15] [16], we denote the difference in data and distribution between the training and test in CD as distribution shift, which poses a great challenge for CDMs.

Firstly, the distribution shift challenges the established CDMs. In applications, the records of students doing exercises are easy to obtain, but we cannot get the scores of students doing exercises. It is obvious that the distributions in two scenarios are different. Thus, the model trained from one scenario performs unsatisfactory for another.

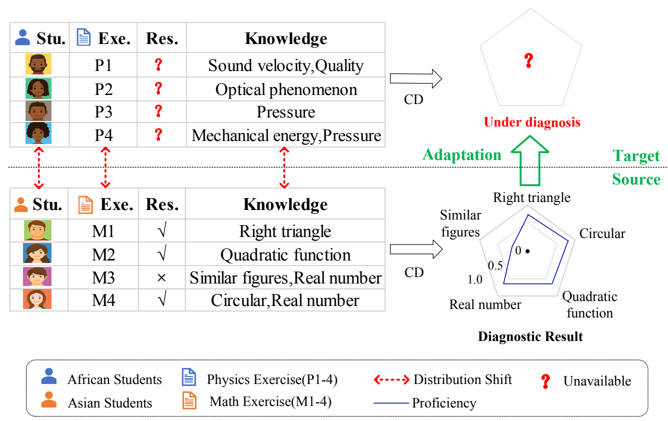


Fig. 1. An example of adaptive cognitive diagnosis under distribution shift. The red dashed lines indicate the different forms of distribution shift and the red question marks indicate the unavailable data or model.

Secondly, the multidimensional distribution shifts in CD task challenges both CD methods and DA methods. The distribution shift in CD is different from that in other tasks, such as text classification [15] and image classification [16]. The distribution shift in CD is caused by the co-existence and interaction of multiple forms of differences. In Fig. 1, there may be different students, different exercises, different knowledge concepts and different learning strategies. The multiple forms of shifts co-exist and interact. Although DA has been researched widely for distribution shift in other applications, it cannot be used for CD task directly and cannot address the multidimensional shifts.

To this end, we propose an adaptive cognitive diagnosis with double-level adversarial framework (ACDDA) to address the co-existence and interaction of multiple distribution shifts in form of students, exercises, knowledge concepts and the cognitive states. One adversarial level is used to learn the representations of student proficiency, exercise difficulty and exercise discrimination adaptively and respectively, and the other adversarial level is used for the adaptive training of cognitive states. Our main contributions are as follows:

- We propose the adaptive cognitive diagnosis to address the distribution shift in cognitive diagnosis.

2024 9th International Conference on Big Data Analytics (ICBDA) | 979-8-3503-5296-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICBDA61153.2024.10607264

- We propose a double-level adversarial framework for ACD, which can learn the representations and model the diagnosis adaptively. Moreover, our framework has good generality and can be implemented based on multiple CDMs.
- Extensive experiments results on four real-world data sets validate the effectiveness of our method.

II. Related work

A. Domain adaptation

Domain adaptation has been proposed to solve the problem of retraining due to label lacking or distribution shift. Maximum mean discrepancy(MMD) [9], Correlation alignment(CORAL) [10] minimize the distance between source and target. Deep domain confusion(DDC) [11] combines MMD-based domain confusion loss to learn domain-invariant feature representations. Joint maximum mean discrepancy(JMMD) [19] computes the distance in view of joint distribution to make the instances with the same label closer. Generative adversarial networks(GAN) [12] learn domain-invariant representations with generative adversarial networks. Domain-adversarial neural network(DANN) [13] first introduces the domain discriminator to identify features are from source or target domain. Multi-adversarial domain adaptation(MADA) [14] uses multiple domain discriminators to achieve fine-grained alignment.

B. Cognitive diagnosis

In the past few decades, many CDMs have been proposed in the field of educational psychology. MIRT [2] and DINA [3] are two of the most classic CDMs, they use hand-designed linear function. Recently, NN-based CDMs have received widespread attention due to their outstanding performance. NCD [4] is the first model to use deep NN, and it provides a new way of representing and modeling the interaction between students and exercises. ECD [5] introduces the student's contexts to enrich the proficiency representation. RCD [6] learns the representations with a graph about students, exercises and concepts with GAT model. KSCD [8] can infer students' mastery of non-interactive knowledge concepts. SCD [21] constructs a self-supervised model to learn student and exercise features representations to address the long-tailed problem. Although there are a few attempts have been made to introduce domain adaptation to other tasks, such as Knowledge Tracing (KT) [7] and Question Difficulty Prediction (QDP) task [18], there is no work about the adaptive cognitive diagnosis.

III. The proposed framework

In this section, our proposed ACDDA is described in detail.

A. Adaptive representation level

In this subsection, we first describe the representation learning for student proficiency, exercise difficulty and exercise discrimination in the source scenario, and then show the adaptive learning of the target scenario using the source representations.

a) The representations in source scenario: We learn the representations in source scenario as NCD did. Specifically, the student one-hot vectors, denoted as $\boldsymbol{x}^s_{stu}$, are encoded as the student's proficiency $\boldsymbol{h}^s_{prof}$ using a trainable matrix $\boldsymbol{A}^s$. And difficulty and discrimination, denoted as $\boldsymbol{h}^s_{diff}$ and $\boldsymbol{h}^s_{disc}$, are learned in a similar way. The processes are shown as:

$$\boldsymbol{h}^s_{prof} = sigmoid(\boldsymbol{x}^s_{stu} \times \boldsymbol{A}^s), \tag{1}$$

$$\boldsymbol{h}^s_{diff} = sigmoid(\boldsymbol{x}^s_{exe} \times \boldsymbol{B}^s), \tag{2}$$

$$\boldsymbol{h}^s_{disc} = sigmoid(\boldsymbol{x}^s_{exe} \times \boldsymbol{D}^s), \tag{3}$$

where $\boldsymbol{h}^s_{prof} \in (0,1)^{1\times K^s}$, $\boldsymbol{h}^s_{diff} \in (0,1)^{1\times K^s}$, $\boldsymbol{h}^s_{disc} \in (0,1)$, denote the representations of proficiency, difficulty and discrimination, respectively. $\boldsymbol{x}^s_{stu} \in \{0,1\}^{1\times N^s}$, $\boldsymbol{x}^s_{exe} \in \{0,1\}^{1\times M^s}$ denote the one-hot vectors of students and exercises, respectively. $\boldsymbol{A}^s \in R^{N^s\times K^s}$, $\boldsymbol{B}^s \in R^{M^s\times K^s}$, $\boldsymbol{D}^s \in R^{M^s\times 1}$ denote the three trainable matrices, respectively.

b) Adaptive representation in target scenario: Owing to the distribution shift and the absence of response logs in target scenario, it is impossible to learn the representations in the end-to-end training way, as did in source scenario. Therefore, we introduce three adversarial modules to learn the proficiency, difficulty and discrimination adaptively for target scenario respectively. We illustrate the adaptive representation in the following, taking the proficiency as an example. In the adversarial process, a generator and a discriminator play against each other to learn a reasonable representation for the target.

First, two generators are used to encode the proficiency for both scenarios respectively, as the following equation:

$$\begin{aligned} \boldsymbol{G}^s_{prof} &= sigmoid(u^T \times \boldsymbol{h}^s_{prof} + d), \\ \boldsymbol{G}^t_{prof} &= sigmoid(u^T \times \boldsymbol{h}^{\mathrm{t}}_{prof} + d), \end{aligned} \tag{4}$$

where $\boldsymbol{G}^s_{prof}$ and $\boldsymbol{G}^t_{prof}$ are the generated proficiency in source and target scenarios, respectively. u and d denote the trainable parameters of the hidden layer and u^T are transpositions of u. $\boldsymbol{h}^s_{prof}$ and $\boldsymbol{h}^t_{prof}$ are the proficiency in source and target scenarios.

Secondly, a discriminator is used to identify whether the current representation is from the source or target, which can be computed with the following equation:

$$\begin{aligned} L^D_{prof}(\boldsymbol{G}^s_{prof}, \boldsymbol{G}^t_{prof}) = &-\frac{1}{N^s}\sum_{i=1}^{N^s}\log(1-\boldsymbol{G}^s_{prof,i}) \\ &-\frac{1}{N^t}\sum_{j=1}^{N^t}\log\boldsymbol{G}^t_{prof,j}, \end{aligned} \tag{5}$$

where L^D_{prof} denotes the loss for the discriminator in view of the proficiency vectors, N^s and N^t represent the number of students in the source and target scenario, respectively. L^D_{diff} and L^D_{disc} are learned in a similar way. During the training process, the Q-matrix vector is not updated, so no adaptive module for the Q-matrix is designed.

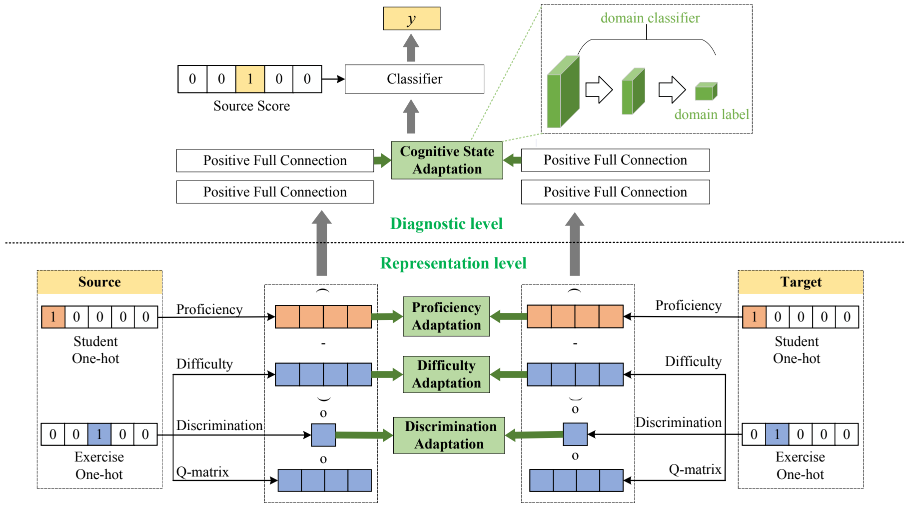


Fig. 2. The framework of our ACDDA.

Based on the three adaptive modules mentioned above, the overall loss of the representation level is obtained as follows:

$$Loss_{embe} = \lambda_{prof} L^{D}_{prof} + \lambda_{diff} L^{D}_{diff} + \lambda_{disc} L^{D}_{disc}, \quad (6)$$

where λ_{prof} , λ_{diff} , λ_{disc} are the weight parameters.

B. Adaptive diagnosis level

Based on the representations of proficiency, difficulty and discrimination, existing methods train diagnostic models by interacting these representations. And one common interaction is performed with a neural network with fully connected layer according to the following equation:

$$\boldsymbol{x}^s = \boldsymbol{Q}^s_e \times (\boldsymbol{h}^s_{prof} - \boldsymbol{h}^s_{diff}) \times \boldsymbol{h}^s_{disc}. \quad (7)$$

Firstly, the cognitive state is modeled with the neural network with two fully connected layers [4].

$$\boldsymbol{f}^s_{\ell+1} = \phi(\boldsymbol{W}^s_\ell \times \boldsymbol{f}^s_\ell + b^s_\ell), \quad (8)$$

$$\boldsymbol{f}^t_{\ell+1} = \phi(\boldsymbol{W}^t_\ell \times \boldsymbol{f}^t_\ell + b^t_\ell), \quad (9)$$

where ϕ is the Sigmoid function. $\boldsymbol{f}^s_{\ell+1}$ and $\boldsymbol{f}^t_{\ell+1}$ are the output of the fully connected ℓ-layer in the source and target scenarios, and the initial input $\boldsymbol{f}^s_0 = x^s$ and $\boldsymbol{f}^t_0 = x^t$. $\boldsymbol{W}^s_\ell$ and b^s_ℓ are the neural network parameters for source scenario.

Next, the cognitive state is trained adaptively. Specifically, using generators $\boldsymbol{G}^s$ and $\boldsymbol{G}^t$ to encode $\boldsymbol{f}^s_\ell$ and $\boldsymbol{f}^t_\ell$, respectively, and then a discriminator L^D is used to discriminate whether the output of the generator is from the source or target.

The loss of the diagnosis level is obtained as follows:

$$Loss_{diag} = L^D(\boldsymbol{G}^s(\boldsymbol{f}^s_\ell, u, d), \boldsymbol{G}^t(\boldsymbol{f}^t_\ell, u, d)). \quad (10)$$

And the final diagnostic output:

$$y = \phi(\boldsymbol{W}^s_\ell \times \boldsymbol{f}^s_\ell + b^s_\ell), \quad (11)$$

$y \in (0, 1)$ is the predicted probability of the student s answering exercise e correctly, $\boldsymbol{f}^s_\ell$ is the output representation of the penultimate fully connected layer in the source scenario.

In summary, the overall loss of the adaptive cognitive diagnosis is expressed as:

$$\begin{aligned} Loss &= -\sum_i (r_i \log y_i + (1 - r_i)\log(1 - y_i)) \\ &+ \lambda_{diag} Loss_{diag} + \lambda_{embe} Loss_{embe}, \end{aligned} \quad (12)$$

where the first item is the cross-entropy of the predicted results of student doing exercises y_i and the real response r_i, λ_{diag} and λ_{embe} are two trad-off parameters.

IV. EXPERIMENT

In this section, we conduct extensive experiments on four data sets to demonstrate the effectiveness of our ACDDA framework.

A. Experimental setup

a) Datasets: Four popular datasets from the real-world are used in our experiments, namely Assistment09, Mooper, Math1, and NIPS2020.

b) Data preparation for distribution shift : To conduct ACD task, we divide the students, exercises and knowledge concepts into two subsets, representing the source and target scenarios respectively. In our datasets, there is no context for the students. Thus, we divided the subsets randomly. Specially, the student set is marked as S, and we select a subset S_{in} as the shared set of students, and the other students are divided into two subsets S_{d1} and S_{d2}, where $S_{d1} \cap S_{d2} = \emptyset$. Then $S_{in} \cup S_{d1}$ and $S_{in} \cup S_{d2}$ form the student group in source and target scenario respectively. This setup makes the students in two scenarios with certain similarity as well as certain difference, fulfilling the conditions for adaptation. In addition, the ratio

TABLE I
THE OVERALL PERFORMANCE OF ALL CDMS FOR ACD WITH DIFFERENT FORMS OF DISTRIBUTION SHIFT.

(a) Source and target scenarios with shift of students

CDMs		Mooper		Assistment09		Math1		NIPS2020	
		S→T	T→S	S→T	T→S	S→T	T→S	S→T	T→S
Statistical	DINA [3]	0.7533	0.7485	0.6248	0.6302	0.7621	0.7635	0.5756	0.5812
	MIRT [2]	0.7821	0.7765	0.6651	0.6681	0.7894	0.7951	0.5843	0.5926
NN-based	NCD [4]	0.8514	0.8527	0.7645	0.7721	0.8826	0.8829	0.7821	0.7852
	NCD-A	**0.8782**	**0.8765**	**0.7719**	**0.7790**	**0.8855**	**0.8848**	**0.7917**	**0.7882**
	RCD [6]	0.8133	0.8111	0.7523	0.7586	0.8576	0.8567	0.7348	0.7374
	RCD-A	**0.8282**	**0.8256**	**0.7553**	**0.7622**	**0.8606**	**0.8596**	0.7224	0.7286
	KSCD [8]	0.8701	0.8698	0.7442	0.7478	0.8319	0.8355	0.7298	0.7343
	KSCD-A	**0.8760**	**0.8746**	**0.7607**	**0.7688**	**0.8526**	**0.8471**	**0.7368**	**0.7376**
	SCD [21]	0.8201	0.8221	0.7581	0.7605	0.8588	0.8595	0.7388	0.7411
	SCD-A	**0.8281**	**0.8256**	**0.7611**	**0.7687**	**0.8606**	**0.8596**	**0.7390**	**0.7413**

(b) Source and target scenarios with shift of exercises

CDMs		Mooper		Assistment09		Math1		NIPS2020	
		S→T	T→S	S→T	T→S	S→T	T→S	S→T	T→S
Statistical	DINA [3]	0.7215	0.7236	0.6958	0.6933	0.6011	0.5932	0.6584	0.6525
	MIRT [2]	0.7625	0.7566	0.7269	0.7311	0.6280	0.6019	0.6765	0.6723
NN-based	NCD [4]	0.8214	0.8001	0.7670	0.7674	0.6309	0.6535	0.7125	0.7147
	NCD-A	**0.8912**	**0.8867**	**0.7887**	**0.7893**	**0.7513**	**0.7171**	**0.7594**	**0.7581**
	RCD [6]	0.9049	0.8991	0.7712	0.7740	0.7135	0.6551	0.7455	0.7449
	RCD-A	**0.9128**	**0.9068**	**0.7784**	**0.7789**	**0.7722**	**0.6742**	0.7224	0.7221
	KSCD [8]	0.7853	0.7710	0.7531	0.7517	0.6359	0.6588	0.6894	0.6935
	KSCD-A	**0.8834**	**0.8485**	**0.7812**	**0.7663**	**0.7266**	**0.7878**	**0.7197**	**0.7181**
	SCD [21]	0.9088	0.9022	0.7786	0.7773	0.7155	0.6568	0.7489	0.7476
	SCD-A	**0.9113**	**0.9068**	**0.7810**	**0.7798**	**0.7622**	**0.6642**	0.7342	0.7336

(c) Source and target scenarios with shift of knowledge concepts

CDMs		Mooper		Assistment09		Math1		NIPS2020	
		S→T	T→S	S→T	T→S	S→T	T→S	S→T	T→S
Statistical	DINA [3]	0.7033	0.7017	0.6830	0.6810	0.8264	0.8272	0.6638	0.6621
	MIRT [2]	0.7486	0.7312	0.7155	0.7231	0.8724	0.8716	0.7044	0.6903
NN-based	NCD [4]	0.7883	0.7673	0.7405	0.7665	0.9445	0.9363	0.7379	0.7105
	NCD-A	**0.8392**	**0.8354**	**0.7562**	**0.7812**	**0.9642**	**0.9743**	**0.7631**	**0.7281**
	RCD [6]	0.9115	0.8994	0.7561	0.7705	0.9268	0.9307	0.7740	0.7746
	RCD-A	**0.9227**	**0.9081**	**0.7591**	**0.7771**	**0.9344**	**0.9350**	**0.7748**	**0.7779**
	KSCD [8]	0.7122	0.7309	0.7519	0.7656	0.8988	0.8980	0.7166	0.6780
	KSCD-A	**0.8026**	**0.8355**	**0.7723**	**0.7742**	**0.9069**	**0.9050**	**0.7596**	**0.7354**
	SCD [21]	0.9165	0.9123	0.7571	0.7765	0.9288	0.9306	0.7749	0.7756
	SCD-A	**0.9246**	**0.9181**	**0.7598**	**0.7788**	**0.9336**	**0.9349**	**0.7755**	**0.7786**

of S_{in} to S can be used to control the difference degree of distribution shift in two scenarios. In this way, the exercises and knowledge concepts can also be treated to prepare the data for ACD.

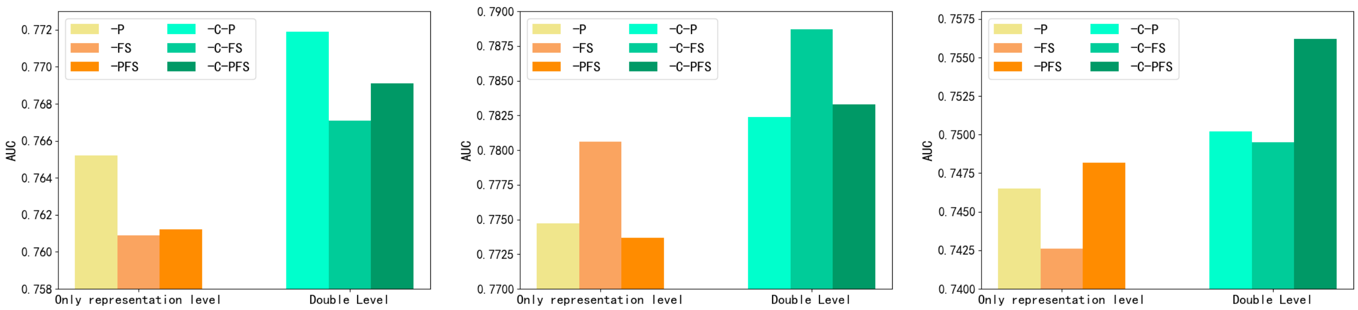
c) Baselines: The baselines include the statistical-based methods, (such as DINA [3] and MIRT [2]), and the NN-based methods (such as NCD [4], RCD [6], KSCD [8] and SCD [21]). The training method for the baseline is to directly apply the network parameters trained in the source scenario to the target scenario. It is noted that the established methods for domain adaptation cannot be used for CD task. Thus, the DA methods are not listed as baselines.

d) Experiment settings: There are several neural networks with parameters in our method. In the training of NN, we use Xavier initialization to initialize the parameters, and each parameter is sampled from $N(0, \mathrm{std}^2)$, where $std = \sqrt{\frac{2}{n_{in}+n_{out}}}$. n_{in} , n_{out} denote the number of input and output neurons, respectively. The network in the diagnosis level is a 3-layer fully-connected NN, and Sigmoid is used as the activation function for all layers. The best values of relevant parameters are set with the experimental results. AUC is used to evaluate the performance of all CDMs, as other works did [2]–[4]. All models are implemented by PyTorch using Python and all experiments are conducted with Intel(R) Core(TM) i9-10900F CPU @ 2.80GHz and NVIDIA GeForce RTX 3060 GPU.

B. RQ1. How does our proposed ACDDA framework perform compared with existing CD models? And how ACDDA performs compared with other baselines?

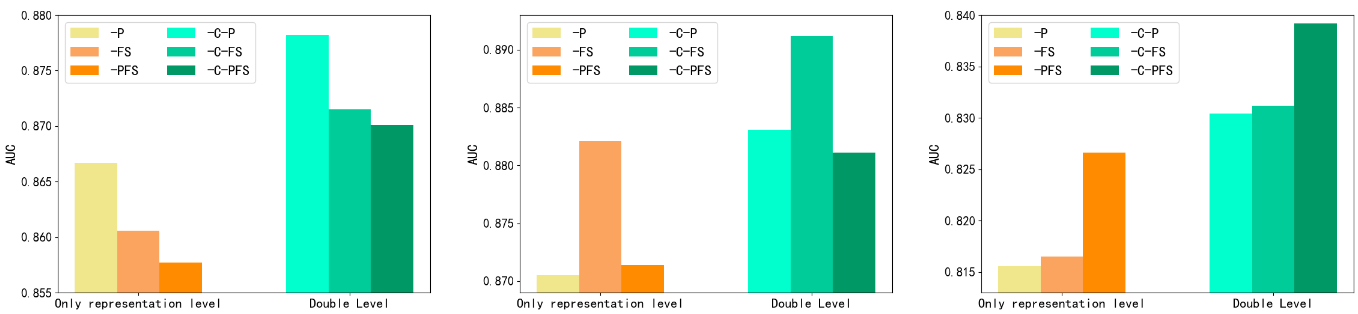
Table I shows the cognitive diagnosis results for ACDDA and all baselines. In Table I, difference ratio is 40% (as mentioned in Data preparation for distribution shift) , S→T denotes training CDM for T scenario without any log with the assist of S with logs. -A represents our proposed method.

(1) Introducing our ACDDA framework can improve the performance of CDMs on the target scenario. NCD-A outperforms NCD by 0.71%, 4.92%, and 3.12% on average, and KSCD-A outperforms KSCD by 1.14%, 6.08%, and 4.24%



(a) Different students (b) Different exercises (c) Different concepts

Fig. 3. Ablation of NCD-A for three adaptive modules in the representation level on Assistment09 dataset



(a) Different students (b) Different exercises (c) Different concepts

Fig. 4. Ablation of NCD-A for three adaptive modules in the representation level on Mooper dataset

on average. The improvement of RCD-A and SCD-A is insignificant, because the representations of proficiency and difficulty contain partial concepts in the target domain. The original baselines perform unsatisfactory because they use CDMs trained from the source scenario directly to predict the mastery of students in the target, not considering the distribution shift between two scenarios. In contrast, our framework achieves better results by representing and training the model adaptively to address the distribution shift.

(2) Our ACDDA has good compatibility with other baselines and the adaptive modules can be implemented based on multiple baselines. Four popular baselines, NCD, RCD, KSCD and SCD all can be integrated with our ACDDA, and the performances of NCD-A, RCD-A, KSCD-A and SCD-A all achieve obvious improvement.

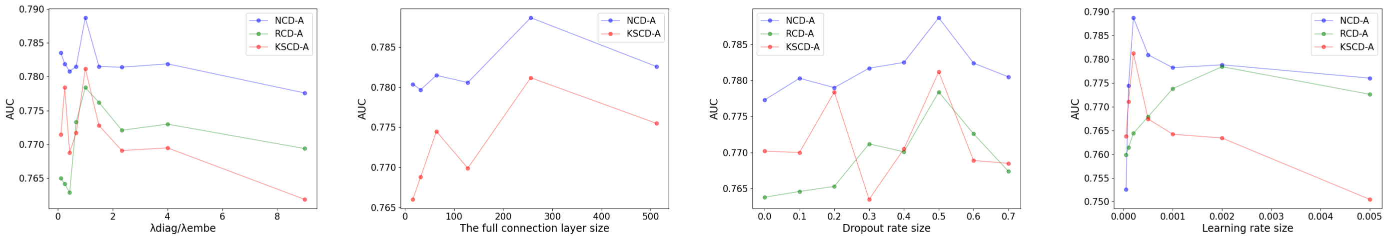
C. *RQ2: Are all three adaptive models in the representation level and diagnosis level necessary?*

To further explore the necessity of the three adaptive modules under different applications, we conduct the ablation about the adaptive modules with NCD-A on Assistment09 and Mooper, as shown in Fig. 3-4. The adaptive modules include the adaptive proficiency (denoted as -P), adaptive difficulty (denoted as -F) and adaptive discrimination (denoted as -S). Moreover, The adaptive cognitive state is denoted as -C.

From Fig. 3 and Fig. 4, we can obtain the three adaptive modules in the representation level are selective according to the scenarios. When the students in two scenarios are different and the exercises are the same, the adaptive module for proficiency can significantly improve the diagnostic performance, while the adaptive modules for discrimination and difficulty are not necessary. In Fig. 3-(a) and Fig. 4-(a), only using the proficiency adaptive module (-P) performs better than using the difficulty and discrimination modules (-FS) and using three modules. It can also be concluded from Fig. 3-(b) and Fig. 4-(b) that the adaptive modules for difficulty and discrimination are necessary when the exercises are different in two scenarios. When there are differences in both students and exercises distributions, all three adaptive modules of the representation level are needed simultaneously. The above conclusion is valid under a certain ratio of difference degree.

D. *RQ3: Is our method sensitive to hyperparameters?*

In Eq.12, λ_{diag} and λ_{embe} are two trad-off parameters. We show the performance of our model varying with different values in Fig 5-(a)). It can be seen that the best values of $\lambda_{diag}/\lambda_{embe}$ is 1 . In addition, we also perform parameter



(a) Loss weight (b) The full connection layer size (c) Dropout rate (d) Learning rate

Fig. 5. Impact of hyperparameters on the performance of NCD-A, KSCD-A and RCD-A in terms of AUC on Assistment09.

sensitivity analysis for the size of fully connected layer, dropout rates, and learning rate, as shown in Fig. 5-(b), -(c) and -(d) respectively. The best results are obtained when the size of fully connected layer is set to 256, the dropout rate is 0.5. And the learning rate of NCD-A and KSCD-A are 0.0002, the learning rate of RCD-A is 0.002 respectively. In addition, the size in RCD is fixed to the number of knowledge concepts, thus we do not list the performance of RCD-A in Fig. 5 -(b).

V. Conclusion and Future Work

In this paper, we propose an adaptive cognitive diagnosis problem to address the distribution shift in CD task. And considering that the co-existence of multiple forms of distribution shift in CD task, we design a double-level adversarial adaptive framework for cognitive diagnosis. Extensive experimental results demonstrate the superiority of our proposed framework. On the other hand, our method is still a rather primitive one, and there are still many problems in adaptive cognitive diagnosis for further research. Firstly, cognitive modeling is a complex process, and the distribution shifts are not only reflected in the representation and diagnostic steps, but also in other more hidden and higher-order factors. Therefore, to find other causes of distribution shifts and to learn the transferable information need future research. Secondly, the negative transfer is an unavoidable problem in ACD. Not only the transferable information be addressed, but also the negative transfer should be avoided.

References

[1] Embretson, S. E. and Reise, S. P., ”Item response theory,” Psychology Press, 2013.
[2] Reckase, M. D., ”Multidimensional item response theory models,” in Multidimensional Item Response Theory, Springer, 2009, pp. 79-112.
[3] De La Torre, J., ”DINA model and parameter estimation: A didactic,” Journal of Educational and Behavioral Statistics, vol. 34, no. 1, pp. 115-130, 2009.
[4] F. Wang, Q. Liu, E. Chen, Z. Huang, Y. Chen, Y. Yin, et al., “Neural cognitive diagnosis for intelligent education systems,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 4, 2020, pp. 6153-6161.
[5] Zhou, Y., Liu, Q., Wu, J., Wang, F., Huang, Z., Tong, W., ..., and Ma, J., ”Modeling context-aware features for cognitive diagnosis in student learning,” in Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2420-2428.
[6] Gao, W., Liu, Q., Huang, Z., Yin, Y., Bi, H., Wang, M. C., . . . , and Su, Y., “Rcd: Relation map driven cognitive diagnosis for intelligent education systems,” in Proceedings of the 44th International ACM Conference on Research and Development in Information Retrieval, 2021, pp. 501-510.
[7] Cheng, S., Liu, Q., Chen, E., Zhang, K., Huang, Z., Yin, Y., . . . , and Su, Y., “Adaptkt: A domain adaptable method for knowledge tracing,” in Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, pp. 123-131.
[8] Ma, H., Li, M., Wu, L., Zhang, H., Cao, Y., Zhang, X., and Zhao, X., “Knowledge-Sensed Cognitive Diagnosis for Intelligent Education Platforms,” in Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022
[9] K. M. Borgwardt, A. Gretton, M. J. Rasch, H. P. Kriegel, B. Schölkopf, and A. J. Smola, “Integrating structured biological data by kernel maximum mean discrepancy,” Bioinformatics, vol. 22, no. 14, pp. 49-57, 2006.
[10] B. Sun, J. Feng, and K. Saenko, “Return of frustratingly easy domain adaptation,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, no. 1, 2016.
[11] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep domain confusion: Maximizing for domain invariance,” arXiv preprint arXiv:1412.3474, 2014.
[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, ... and Y. Bengio, “Generative adversarial networks,” Communications of the ACM, vol. 63, no. 11, pp. 139-144, 2020.
[13] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, ... and V. Lempitsky, “Domain-adversarial training of neural networks,” The Journal of Machine Learning Research, vol. 17, no. 1, pp. 2096-2030, 2016.
[14] Z. Pei, Z. Cao, M. Long, and J. Wang, “Multi-adversarial domain adaptation,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018.
[15] V. Behbood, J. Lu, and G. Zhang, “Text categorization by fuzzy domain adaptation,” in 2013 IEEE International Conference on Fuzzy Systems, 2013, pp. 1-7.
[16] C. Deng, X. Liu, C. Li, and D. Tao, “Active multi-kernel domain adaptation for hyperspectral image classification,” Pattern Recognition, vol. 77, pp. 306-315, 2018.
[17] Y. Kao, B. J. Matlen, and D. Weintrop, “From One Language to the Next: Applications of Analogical Transfer for Programming Education,” ACM Transactions on Computing Education (TOCE), vol. 42, no. 1, pp. 1-21, 2021.
[18] Y. Huang, W. Huang, S. Tong, Z. Huang, Q. Liu, E. Chen, ... and S. Wang, “STAN: Adversarial Network for Cross-domain Question Difficulty Prediction,” in Proceedings of the IEEE International Conference on Data Mining (ICDM), 2021, pp. 220-229.
[19] M. Long, H. Zhu, J. Wang, and M. I. Jordan, “Deep transfer learning with joint adaptation networks,” in International Conference on Machine Learning (ICML), 2017, pp. 2208-2217.
[20] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, “Deep reconstruction-classification networks for unsupervised domain adaptation,” in European Conference on Computer Vision (ECCV), 2016.
[21] Wang, S., Zeng, Z., Yang, X., and Zhang, X, “Self-supervised graph learning for long-tailed cognitive diagnosis,” in Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 110-118.