

A Learner Ability Profiling Model Based on Dynamic Knowledge Graph and Graph Neural Network

Chen Lei *

School of Electronic and Information Engineering, Southwest Minzu University, Chengdu 610041, China

* Corresponding author

Abstract: To address the insufficient exploitation of structural information, the inadequate modelling of temporal dynamics, and the limited interpretability of ability dimensions in learner ability profiling under open education, this paper proposes a learner ability profiling model based on a dynamic knowledge graph and a graph neural network, termed DKG-LPA. The model first constructs a dynamic heterogeneous knowledge graph comprising student, course and ability nodes, and employs a gated recurrent unit to characterise the cross-period evolution of student states. On this basis, a spatio-temporal graph attention network fuses structural information with temporal information, and a graph-aware ability context mechanism is designed to inject course-ability structural relations into the generation of ability profiles. A memory-augmented knowledge tracing module is further introduced to simulate the accumulation and forgetting of knowledge mastery states. On the Open University Learning Analytics Dataset, the model achieves an AUC of 0.9825, an F1 score of 0.9382 and a mean absolute error of 0.0215, significantly outperforming the baseline methods LightGBM, DKT, DKVMN, AKT and GKT. Ablation studies show that the dynamic graph evolution module contributes most to the overall performance, and that the graph neural network reduces the ability profiling error by 55.6 percent through the ability context mechanism, which validates the effectiveness of each design.

Keywords: Learner profiling; Knowledge graph; Graph neural network; Learning analytics; Multi-task learning.

1. Introduction

The widespread adoption of large-scale online education, including open universities and massive open online course platforms, has led to the retention of learning process data for tens of thousands of learners. The Open University Learning Analytics Dataset (OULAD) released by the Open University of the United Kingdom records demographic information, virtual learning environment interaction behaviour and academic results of more than 30 000 students across 22 course presentations, and provides a public benchmark for learning analytics research [1]. How to construct accurate learner ability profiles from such data, and thereby support academic early warning and personalised intervention, is one of the core problems in the field of educational data mining.

A learner profile is an abstraction and characterisation of multi-dimensional features of a learner, including knowledge level, ability structure and learning behaviour. Early work on profile construction relied mainly on statistical methods and conventional machine learning models. Zhao Chun et al. [2] took learning ability and learning behaviour preference as two dimensions, and constructed a stability label model of learning ability by means of a linearly weighted coefficient of variation algorithm. Liu Fang et al. [3] integrated static and dynamic learner features into a collaborative filtering framework and improved the effect of online learning resource recommendation. The limitation of such methods lies in the fact that feature engineering depends on manual design, and it is difficult to capture the deep structural relations among students, courses and abilities.

In recent years, the introduction of knowledge tracing and deep learning methods has advanced learner modelling. DKT [4] was the first to apply a recurrent neural network to the modelling of student response sequences; DKVMN [5] introduced a key-value memory structure to characterise knowledge concept states; and AKT [6] exploited an attention

mechanism to model the contextual relations among concepts. These methods, however, all treat courses or knowledge points as mutually independent sequence elements and neglect the information contained in the knowledge structure itself. GKT [7] modelled the relations among knowledge points as a graph structure and aggregated knowledge states with a graph neural network, thereby opening up research on graph-based methods in this field. Subsequent work, such as that of Zheng Haodong et al. [8], who incorporated forgetting factors and a memory gate structure into GKT, and Zhang Kai et al. [9], who extracted multi-order concept representations through a graph neural network pooling operation, has verified the value of structural graph information for learner modelling. With respect to profile construction, Bai Jinghao et al. [10] proposed a multi-channel Transformer prediction framework to address the temporal dynamics of profile labels.

Nevertheless, three deficiencies remain in existing work. First, knowledge graphs are mostly constructed statically, without considering the evolution of student states and course offerings over time, so that course presentations from different semesters are treated identically in the graph structure. Second, graph neural networks are mostly used for knowledge state tracing, and research on their mechanism of action with respect to the more abstract goal of ability profiling is still insufficient. Third, most models take a single task, either score prediction or response correctness prediction, as the optimisation objective, so that the output profile is one-dimensional and cannot support practical academic early warning scenarios.

To address the above problems, this paper proposes a learner ability profiling model based on a dynamic knowledge graph and a graph neural network, named DKG-LPA. The main contributions of the model are as follows.

1) A dynamic heterogeneous knowledge graph oriented towards learner profiling is constructed. It contains three types of nodes, namely students, courses and abilities, and

eight types of relation edges, including enrolment, prerequisite, same-domain, resource-similarity, cultivation and ability-association edges. Five temporal window snapshots of student behaviour are taken as the temporal input of graph evolution.

2) A dynamic graph evolution module is proposed, which employs a gated recurrent unit to encode the cross-period behavioural trajectories of students, so that the graph representation is updated as the semester proceeds.

3) A graph-aware ability context mechanism is designed. Through an attention mechanism, a student extracts a personalised ability context from the ability node representations that have been propagated along the course-ability structure. This context is injected directly into the generation of the ability profile, which assigns an explicit modelling role to the graph neural network.

4) Memory-augmented knowledge tracing is combined with multi-task profile generation, producing four types of profile output within a unified framework, namely score prediction, failure risk, fine-grained scores and ability levels.

Experiments on the public OULAD dataset show that DKG-LPA attains an AUC of 0.9825, an F1 score of 0.9382 and a mean absolute error of 0.0215, significantly outperforming five comparison methods. Ablation experiments verify the effectiveness of each module, among which the dynamic graph evolution module and the graph-aware ability context mechanism make particularly prominent contributions.

2. Related Work

2.1. Learner profile construction

The construction of learner profiles usually follows the route of indicator definition, feature engineering and label generation. Early studies adopted unsupervised methods such as clustering and association analysis to group learners. The label model proposed by Zhao Chun et al. [2] takes learning ability and behavioural preference as typical dimensions and solves the problem of dimensional and sample-size discrepancy in label construction. Lai Yingxu et al. [11] combined the stimulus-organism-response (SOR) theory to build a personalised label system and verified the gain of that label system for dropout prediction. These works established the basic framework of multi-dimensional profiling, but the label generation process is tightly coupled with feature engineering and the generalisation ability of the model is limited. Bai Jinghao et al. [10] formulated profile label prediction as a time series forecasting problem and exploited a multi-channel Transformer to mine cross-period features, achieving better results than static methods on campus card data and online course records, which indicates that temporal information plays an important role in profile construction.

2.2. Knowledge tracing and graph neural networks

Deep knowledge tracing methods take a student response sequence as input and predict the mastery state of knowledge concepts. DKT [4] uses an LSTM to model the sequence but suffers from the inability to represent concept states explicitly. DKVMN [5] stores the knowledge state of each concept explicitly with a key-value memory network and achieves a stable improvement. AKT [6] further exploits an attention mechanism to weight historical interactions adaptively and has become a strong baseline in the field of knowledge tracing.

With respect to graph methods, GKT [7] models concept relations as a graph and aggregates neighbouring knowledge states through a graph neural network. Zheng Haodong et al. [8] computed knowledge point correlations to build a knowledge graph and combined seven forgetting features with a memory gate structure. Zhang Wei et al. [12] constructed an exercise-concept heterogeneous graph and introduced a relational graph convolutional network, using a dual gating mechanism to characterise changes in knowledge states. All of the above works take response-level prediction as their objective; their graph structure is confined to the level of knowledge points or exercises, they do not involve the heterogeneous relations between the curriculum system and the student population, and they rarely discuss high-level semantics such as ability dimensions.

2.3. Distinction of this work from existing work

Compared with existing work, the distinction of this paper is reflected in three aspects. First, the modelling granularity is different. This paper models on a three-layer heterogeneous graph of students, courses and abilities, and the dynamics of the graph originate from the period-by-period evolution of student behaviour snapshots rather than from the static relations of knowledge concepts. Second, the output objective is different. This paper simultaneously outputs scores, risk, fine-grained scores and ability levels in a multi-task manner to form a complete profile, rather than a single response prediction. Third, the role assigned to the graph neural network is different. Through the graph-aware ability context mechanism, this paper directly associates the graph-propagated ability representations with the ability prediction task, providing a testable mechanism for the value of graph neural networks in profiling scenarios. This point is verified by the ablation experiment in Section 4.4.

3. The DKG-LPA Model

3.1. Problem definition

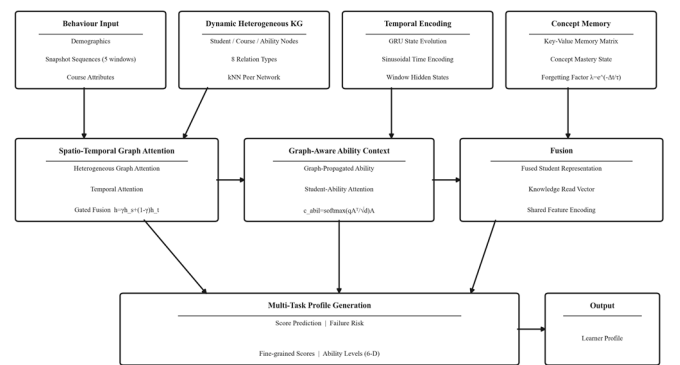


Fig. 1 Architecture of the DKG-LPA model

Given a set of course presentations C and a set of enrolled students S , each student s has demographic features x_s and a sequence of behaviour snapshots over five observation windows $\{b_1, b_2, \dots, b_T\}$ with $T = 5$. The target course presentation is denoted c . The learning outcomes of the student in that presentation comprise the final examination score y_{reg} in $[0, 1]$, where absence is recorded as a missing value, the binary pass or fail label y_{cls} in $\{0, 1\}$, and the ability pseudo-labels y_{abil} in $[0, 1]^6$ constructed from behavioural data. The learning analytics task requires the use

of all observation window information to predict y_{reg} , y_{cls} and y_{abil} , and to provide a failure risk probability.

The overall architecture of the proposed DKG-LPA model is shown in Fig. 1. It consists of four cooperating modules, namely the dynamic graph evolution module, the spatio-temporal graph attention module, which contains the graph-aware ability context, the memory-augmented knowledge tracing module, and the multi-task profile generation module.

3.2. Construction of the dynamic heterogeneous knowledge graph

The node set of the graph contains three types: student nodes, numbering 32 593 and corresponding one-to-one with student-course enrolment records, course nodes, numbering 22 course presentations, and ability nodes, numbering six and corresponding to the ability dimensions of literature reading, teamwork, logical thinking, system design, experimental operation and engineering practice. The edge set contains eight types of relations:

- 1) Enrolment edges (student to course): records of a student registering for a course presentation;
- 2) Reverse enrolment edges (course to student): the inverse relation of the enrolment edges, used to pass aggregated course information back to student nodes;
- 3) Similar student edges (student to student): k nearest neighbours based on the cosine similarity of student features, with $k = 8$, characterising the learner peer network;
- 4) Prerequisite edges and same-domain edges (course to course): prerequisite relations and disciplinary domain membership defined by the curriculum system;
- 5) Resource similarity edges (course to course): cosine similarity computed from the distribution of activity types of the virtual learning environment resources of a course, with the top three connections retained;
- 6) Cultivation edges (course to ability): the cultivation intensity of a course with respect to an ability dimension;
- 7) Ability association edges (ability to ability): correlational relations among ability dimensions.

Student node features are composed of demographic attributes and behavioural statistics. Course node features include attributes such as credits, duration and assessment weight. Ability nodes are initialised with one-hot encoding. The construction of edges is based entirely on the definition of the curriculum system and on student behaviour statistics, and does not involve prediction labels, which avoids label leakage.

3.3. Dynamic graph evolution module

The student behaviour snapshot bt contains three features, namely the logarithmic transform of cumulative clicks, the cumulative number of active days and the within-window score proxy. The module first maps the snapshot into a latent space through a projection layer, and then updates the student state window by window along the temporal dimension with a gated recurrent unit (GRU), as given in Eq. (1):

$$ht = \text{GRU}(ht-1, Wb \cdot bt) \quad (1)$$

where ht is the hidden state of the student in the t -th window. To introduce time interval information, a time encoding vector is used to apply sinusoidal encoding to the window index, and the result is concatenated with the snapshot features. The module outputs the hidden state sequence of all windows as the temporal input of the graph attention module. Unlike static concatenation, the update gate and reset gate of the GRU allow the model to adaptively determine the fusion

ratio between the historical state and the current behaviour, which is consistent with the empirical observation in learning behaviour that recent performance reflects the current state more accurately.

3.4. Spatio-temporal graph attention module

In the spatial dimension, a two-layer heterogeneous graph attention network (HeteroGAT) is used to propagate information over the graph structure. For each edge $e = (u, v)$, the attention coefficient is computed as in Eq. (2):

$$\alpha_{uv} = \text{softmax}_v(\text{LeakyReLU}(aT[W \cdot hu \parallel W \cdot hv])) \quad (2)$$

where hu and hv are the hidden states of the endpoints, W is a linear transformation, a is the attention parameter vector and $\parallel$ denotes concatenation. Message passing employs multi-head attention with four heads for aggregation, and a Pre-LN together with a stochastic depth strategy is used to stabilise deep training. After graph propagation, student nodes receive information from their peer network and from course nodes, course nodes aggregate the group state of enrolled students, and ability nodes receive course information along cultivation edges, thus forming a hierarchical information flow from students to courses to abilities.

In the temporal dimension, the window hidden state sequence output by the module in Section 3.3 is fed into multi-head temporal attention, which performs a weighted sum over the importance of different windows to obtain a temporally aggregated representation of the student state.

The spatial and temporal representations are fused through a gating mechanism, as shown in Eq. (3):

$$hfuse = \gamma \odot h_{spatial} + (1 - \gamma) \odot h_{temporal} \quad (3)$$

where the gating coefficient $\gamma = \sigma(\text{MLP}([h_{spatial} \parallel h_{temporal}]))$ and σ is the sigmoid function. This design allows the model to adaptively select the information source for different students: students with rich behavioural data may rely more on the temporal branch, whereas students with sparse behaviour may obtain peer information through graph propagation.

3.5. Graph-aware ability context mechanism

After graph propagation, the representation of an ability node aggregates the group learning state of the courses that cultivate it. To enable the student profile to make full use of this structural information, this paper designs a graph-aware ability context mechanism. Taking the fused student representation $hfuse$ as the query, attention is computed over the ability node representations A in $R6 \times d$, as expressed in Eq. (4):

$$cabil = \text{softmax}(hfuse \cdot A^T / \sqrt{d}) \cdot A \quad (4)$$

The context vector $cabil$ summarises the course-ability structural information most relevant to the current state of the student, and is concatenated with the output of the shared encoder before being fed into the ability prediction head. The motivation for this mechanism is that students are at different stages of ability development, and the degree to which their ability assessment depends on structural information therefore differs; attention weighting enables the model to extract the ability context as required. When the spatial GNN is removed, the ability nodes degenerate into static one-hot projections and the above equation degenerates into a weighted combination of static ability vectors, so that the prediction accuracy of the ability profile decreases. This contrast constitutes the core verification of the ablation experiment in this paper.

3.6. Memory-augmented knowledge tracing module

Following the key-value memory design of DKVMN, this module maintains a key matrix K and a value matrix V , both of size $22 \times dm$, for 22 course concepts. The student representation interacts with the key matrix to obtain concept relevance, and the knowledge state of the corresponding concept in the value matrix is then read and written in a weighted manner. To characterise the forgetting phenomenon, an Ebbinghaus-style decay factor is applied on writing, as given in Eq. (5):

$$\lambda = e^{-\Delta t/\tau} \quad (5)$$

where Δt is the time interval from the target course and τ is the forgetting rate parameter. The module outputs a knowledge mastery distribution and a read vector, the latter serving as one of the inputs to profile generation.

3.7. Multi-task profile generation and loss function

The shared encoder concatenates the fused representation with the knowledge tracing read vector and encodes them into shared features. Four task heads predict in parallel: the score regression head outputs a predicted score, the failure classification head outputs a risk probability, the fine-grained head outputs four types of scores for quizzes, assignments, discussions and laboratory reports, and the ability head outputs six-dimensional ability levels. The total loss is defined in Eq. (6):

$$L = w_1 L_{mae} + w_2 L_{focal} + w_3 L_{fine} + w_4 L_{abil} + w_5 L_{kt} \quad (6)$$

where Focal Loss is adopted for the classification task to handle class imbalance, with $\gamma = 2.0$ and $\alpha = 0.75$, MAE is adopted for the regression task, MSE is adopted for the ability and fine-grained tasks, and L_{kt} is the supervision term of the knowledge mastery distribution. The weight of each task is set according to validation set performance. The total number of model parameters is approximately 3.9 million.

4. Experiments

4.1. Dataset and preprocessing

The experiments use the public OULAD dataset [1]. This dataset records the learning records of 32 593 students of the Open University of the United Kingdom in the 2013-2014 academic year across seven course modules and four presentation periods, amounting to 22 course presentations. It contains demographic information such as gender, region, age and education level, virtual learning environment interaction behaviour with 10 655 280 click records, and assessment results. The final results of students are divided into four categories, namely Pass, Distinction, Fail and Withdrawn, among which the Pass and Distinction samples account for approximately 47.2 percent. The basic statistics of the data are given in Table 1.

Preprocessing includes the following steps: student nodes are constructed at the granularity of the module-presentation-student triple; VLE clicks are aggregated into behaviour snapshots over five observation windows, comprising the logarithm of cumulative clicks, the cumulative number of active days and the within-window score proxy; and numerical features are standardised. The dataset is split by student into a training set of 70 percent, a validation set of 15 percent and a test set of 15 percent, and the experiments are

repeated with five fixed random seeds.

Table 1. Statistics of the OULAD dataset

Item	Value
Number of students	32 593
Course modules / presentation periods	7 / 4
Number of course presentation nodes	22
VLE interaction records	10 655 280
Pass and Distinction: Fail and Withdrawn	15 385: 17 208
Students with a final examination score	4 959

4.2. Experimental settings and evaluation metrics

The hidden dimension of the model is 256, the number of GNN layers is two, the number of attention heads is four and the memory dimension is 64. The model is trained for 200 epochs with the AdamW optimiser, using a learning rate of 0.001 and a weight decay of 0.0001, and the gradient clipping threshold within a batch is 1.0. Focal Loss is adopted for the classification task with $\gamma = 2.0$ and $\alpha = 0.75$, the MAE loss is adopted for the regression task, and MSE is adopted for the ability and fine-grained tasks. All experiments are completed on an ordinary personal computer with 16 GB of memory using the CPU.

The evaluation metrics include AUC, the area under the ROC curve, the F1 score, the accuracy ACC and the mean absolute error MAE. The classification threshold is selected on the validation set by maximising Youden's J statistic and is then applied to the test set. All metrics are reported as the mean and standard deviation over five random seeds, and the Welch t-test for independent samples with unequal variances is used to test the statistical significance of the differences between models.

The comparison methods include LightGBM, a gradient boosting model taking demographic features and course attributes as input, DKT [4], DKVMN [5] and AKT [6], which take behaviour snapshot sequences as input, and GKT [7], a graph-based method taking the course graph structure as input.

4.3. Comparison experiment

The results of each method on the test set are given in Table 2 and visualised in Fig. 2. The following observations can be made.

1) DKG-LPA outperforms all comparison methods on all metrics. Its AUC reaches 0.9825, which is 0.55 percentage points higher than the strongest baseline AKT, 3.04 percentage points higher than GKT, which is also a graph-based method, and 2.29 percentage points higher than the conventional method LightGBM. Its MAE is 0.0215, which is only about one third of that of the best baseline, AKT, at 0.0593.

2) DKG-LPA has the best stability. Its AUC standard deviation is 0.0018, whereas that of DKT reaches 0.0399, the latter being clearly sensitive to the random seed. This stability originates from the smoothing effect of multi-task joint training on single-task fluctuation.

3) The Welch t-test shows that the differences in AUC between DKG-LPA and all baselines are statistically significant, with $p < 0.05$: $p = 0.032$ against DKT, $p = 0.0009$ against AKT, and $p < 0.001$ against LightGBM, DKVMN and GKT. The effect size Cohen's d ranges from 2.05 against DKT

to 20.43 against DKVMN, indicating that the differences are not only significant but also considerable in magnitude.

Table 2. Performance of each method on the test set (mean plus or minus standard deviation over five seeds)

Model	AUC	F1	ACC	MAE
LightG BM	0.9596±0. 0037	0.9100±0. 0034	0.9102±0. 0027	0.1523±0. 0014
DKT	0.9247±0. 0399	0.8418±0. 0619	0.8409±0. 0591	0.1480±0. 0515
DKVM N	0.8980±0. 0056	0.8084±0. 0022	0.8057±0. 0053	0.1102±0. 0239
AKT	0.9770±0. 0009	0.9265±0. 0060	0.9246±0. 0056	0.0593±0. 0055
GKT	0.9521±0. 0028	0.9035±0. 0046	0.9040±0. 0040	0.1634±0. 0038
DKG- LPA	0.9825±0. 0018	0.9382±0. 0041	0.9364±0. 0039	0.0215±0. 0007

Note: the values in the table are the mean plus or minus the sample standard deviation over five random seeds on the test set. The classification threshold is selected on the validation set by maximising Youden's index and is then fixed for the test set. The bold row is the model proposed in this paper.

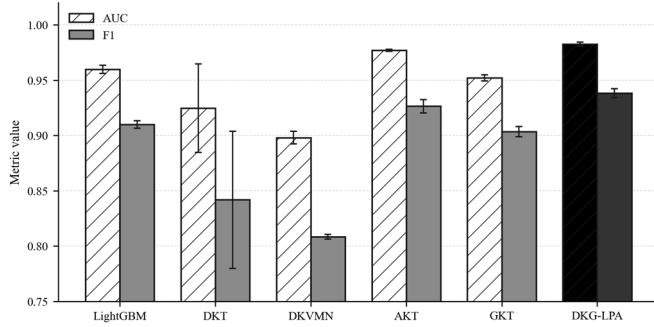


Fig. 2 Performance comparison between DKG-LPA and the baseline methods

4.4. Ablation experiment

Table 3. Results of the ablation experiments (mean plus or minus standard deviation over three seeds)

Model configuration	AUC	F1	MAE	Ability MAE
Full model	0.9846±0. 0019	0.9422±0. 0034	0.0182±0. 0047	0.0727±0. 0064
w/o dynamic graph evolution	0.9599±0. 0058	0.9096±0. 0061	0.1575±0. 0011	0.1016±0. 0155
w/o spatial GNN	0.9841±0. 0018	0.9430±0. 0040	0.0160±0. 0016	0.1131±0. 0078
w/o temporal attention	0.9833±0. 0016	0.9428±0. 0055	0.0197±0. 0014	0.0701±0. 0037
w/o memory KT	0.9840±0. 0017	0.9425±0. 0038	0.0203±0. 0035	0.0761±0. 0125

Note: Table 3 is an independent ablation training process, and its full model row should serve as the reference of the same batch for the ablation variants, so as to measure the relative contribution of each module. Its absolute values differ from the five-seed results in Table 2 by about 0.002, which originates from the random initialisation of the independent training process and from the different seed sets, since Table 3 uses the three seeds 42, 123 and 456. This does not affect the relative conclusions among the variants.

To verify the effectiveness of each module, the model is ablated as follows: the variant without dynamic graph evolution removes it so that the behaviour snapshot degenerates into a statically repeated input; the variant without the spatial GNN removes that module; the variant without temporal attention removes it so that snapshots are directly mean-pooled; and the variant without memory KT removes memory-augmented knowledge tracing. Each ablation variant is trained with three random seeds, and the results are given in Table 3 and Fig. 3.

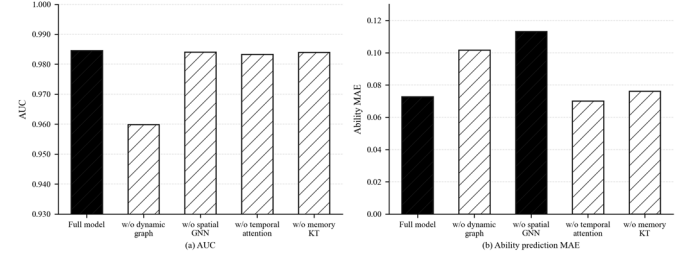


Fig. 3 Ablation study on AUC and ability prediction MAE

The ablation results give three conclusions. First, the dynamic graph evolution module contributes most to the overall performance. After its removal, the AUC decreases by 2.5 percentage points and the MAE deteriorates by nearly an order of magnitude, from 0.0182 to 0.1575, which indicates that window-by-window state evolution modelling is the most critical source of information in this scenario. Second, the value of the spatial GNN is concentrated in the ability profiling task. After its removal, the ability prediction MAE rises from 0.0727 to 0.1131, a deterioration of 55.6 percent, while the AUC of the main task remains essentially unchanged. This shows that the role of the graph neural network is not to replace temporal information but to generate more accurate ability representations through course-ability structural propagation, which is consistent with the design motivation in Section 3.5. Third, the memory-augmented knowledge tracing and the temporal attention module provide stable marginal gains: the former mainly improves the regression metrics, with the MAE rising by 11.5 percent after its removal, while the latter contributes about 0.13 percentage points to the AUC. Taken together, each module exhibits a positive effect on the metric dimension to which it corresponds, and the overall design of the model is therefore effective.

4.5. Analysis of ability profiling results

To examine the association between the ability profile output and the actual academic results, the students in the test set are divided into four quartile groups from low to high according to their predicted level on each ability dimension, and the actual failure rate of each group is counted, as shown in Table 4.

It can be seen that, apart from engineering practice, the predicted level of the remaining five ability dimensions is clearly negatively correlated with the failure rate: the failure rate of students in the lowest quartile exceeds 98 percent in all cases, while that of the highest quartile is below 20 percent in all cases. Taking the logical thinking dimension as an example, the difference in failure rate between the highest and the lowest predicted-level groups is 92.7 percentage points. This indicates that the ability levels output by the model are not merely a numerical fit to the supervisory signal, but are highly consistent with the actual academic results and

therefore possess genuine value as a reference for academic early warning. The discriminative power of the engineering practice dimension is relatively weak, with a failure rate of 42.2 percent in the highest quartile. The reason may be that the behavioural features on which the pseudo-label construction of this dimension depends are themselves weakly correlated with the final assessment results, and this may be improved in the future by drawing on engineering-oriented assessment data in real course scenarios.

Table 4. Quartiles of predicted ability level and the actual failure rate (percent)

Ability dimension	Q1 (lowest)	Q2	Q3	Q4 (highest)
Logical thinking	99.8	85.7	20.5	7.1
System design	98.9	82.1	24.5	7.6
Experimental operation	99.9	92.6	17.7	2.9
Teamwork	99.8	68.6	38.6	6.2
Literature reading	98.5	37.4	59.9	17.4
Engineering practice	89.5	56.5	24.9	42.2

5. Conclusion

Aiming at the problem of learner ability profile construction in open education scenarios, this paper proposes DKG-LPA, a multi-task profiling model based on a dynamic knowledge graph and a graph neural network. Through the cooperation of four modules, namely dynamic graph evolution, spatio-temporal graph attention, graph-aware ability context and memory-augmented knowledge tracing, the model achieves prediction performance superior to five comparison methods on the public OULAD dataset, with an AUC of 0.9825, an F1 score of 0.9382 and an MAE of 0.0215. Ablation experiments verify the effectiveness of each module, among which the dynamic graph evolution module contributes most to the overall performance and the graph neural network improves the accuracy of ability profiling particularly markedly.

Future work will proceed in two directions. First, the model will be extended to more public datasets, such as Junyi and ASSISTments, in order to examine its cross-platform generalisation. Second, the ability dimensions will be aligned with the formalised knowledge of course cultivation programmes so as to further strengthen the interpretability of the profile, and an attempt will be made to feed the profiling results back into practical applications of personalised learning path recommendation.

References

- [1] Kuzilek, J., Hlosta, M., & Zdrahal, Z. (2017). Open University learning analytics dataset. *Scientific Data*, 4, Article 170171. <https://doi.org/10.1038/sdata.2017.171>
- [2] Zhao, C., & Li, X. (2023). Research on the construction of label model for learning portrait in online learning space. *Computer Technology and Development*, 33(10), 176-182.
- [3] Liu, F., Tian, F., Li, X., Wang, H., & Zhang, Y. (2021). Collaborative filtering recommendation method for online learning resources incorporating learner model. *CAAI Transactions on Intelligent Systems*, 16(6), 1117-1125.
- [4] Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. In *Advances in Neural Information Processing Systems 28 (NeurIPS)* (pp. 505-513).
- [5] Zhang, J., Shi, X., King, I., & Yeung, D.-Y. (2017). Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web (WWW)* (pp. 765-774). ACM. <https://doi.org/10.1145/3038912.3052580>
- [6] Ghosh, A., Heffernan, N., & Lan, A. S. (2020). Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (pp. 2330-2339). ACM. <https://doi.org/10.1145/3394486.3403282>
- [7] Nakagawa, H., Iwasawa, Y., & Matsuo, Y. (2019). Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI)* (pp. 156-163). IEEE. <https://doi.org/10.1145/3350546.3352513>
- [8] Zheng, H. D., Ma, H., Xie, Y. C., Wang, L., & Chen, Y. (2023). Graph neural network knowledge tracing model integrating forgetting factors and memory gate. *Journal of Computer Applications*, 43(9), 2747-2752.
- [9] Zhang, K., Liu, Y., Qin, Z. C., Li, H., & Wang, J. (2022). Knowledge tracing model with enhanced concept representation. *Application Research of Computers*, 39(11), 3309-3314.
- [10] Bai, J. H., Zhuang, J. X., & Lai, Y. X. (2026). Temporal dynamic label construction and prediction method for learner portraits. *Computer Science*, 53(5), 79-89.
- [11] Lai, Y. X., Zhang, Y. W., & Zhuang, J. X. (2026). Construction and generation of SOR label system for personalised learner portraits. *Computer Science*, 53(8), 9-19.
- [12] Zhang, W., Li, Z. X., Gong, Z. W., Liu, Y., & Chen, H. (2024). Knowledge tracing model enhanced by exercise embedding and learning forgetting features. *Application Research of Computers*, 41(11), 3265-3271.