ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

A Study on the Integration of Task-Driven Models and Multi-Dimensional Interactive Strategies in Digital Training Systems for Higher Education Faculty

Caihong Feng[1,*]

[1] Department of Development and Planning of Yulin University, Yulin, 719000, China

SUMMARY: *This paper proposes a task-driven framework for digital training of university teachers, focusing on task parsing, multi-dimensional interaction modeling and adaptive resource scheduling. Relying on the actual deployment platform, a structured data set containing 214 teachers, 638 training sessions, 96420 behavior records and 12860 feedback records is constructed. The system generates a task graph according to teaching objectives, participation trajectories and evaluation logs, and combines time series aggregation, relation-aware state coding and interaction strategy fusion to complete the unified representation. On this basis, the system further completes the feedback timing calculation, resource matching and training path generation. Experimental results show that the task recognition accuracy of the framework reaches 92.8%, the interactive decision accuracy reaches 90.6%, the path matching consistency reaches 88.9%, the average response delay is 74 ms, and the content adaptation rate reaches 93.4%. It maintains a stable running state and process control ability in different training scenarios. The framework has good computing stability, deployment adaptation ability and scalability in the digital training environment of university teachers.*

KEYWORDS: *Task graph modeling; Interaction state representation; Adaptive feedback computing; Digital training system*

1 Introduction

With the transformation of university teacher training from offline teaching to platform-based, data-based and interactive organization, digital training system has evolved from resource carrying tool to an integrated environment of task orchestration, behavior perception and feedback calculation. Teachers continuously generate heterogeneous data such as log, click, stay, submit, voice and text in course design, teaching demonstration, case study, peer collaboration and platform reflection. This kind of data has the characteristics of parallel task chain, fine interaction granularity, fast state switching and strong context dependence. Common learning analysis panels, chat agents and rule recommendation mechanisms can support the visualization and instant assistance of training activities. However, the digital training process for university teachers involves not only content push, but also calculation steps such as task decomposition, state estimation, collaborative feedback and resource scheduling. From the perspective of computing implementation, teacher training is not a single learning session, but a dynamic process composed of goal release, task execution, evidence recovery, feedback update and peer collaboration. Without the joint modeling of task

*fch4856@126.com
https://doi.org/10.65102/is2026711

Published: 30 April, 2026

nodes, interaction intensity and resource occupancy, the system is prone to feedback timing mismatch, content call dispersion and training path jump, and the computing benefits on the platform side are difficult to release stably. This shows that system-level modeling is more necessary.

Recent studies have gradually integrated teachers 'digital competence, AI literacy and platform usage behaviors into a unified analysis framework. Basilotta-Gomez-Pablos et al. [1] sorted out the core dimensions of higher education teachers 'digital competence. Susnjak et al. [2] built a learning analytics dashboard that could output action suggestions. Essel et al. [3] verified the support effect of virtual teaching assistant in the university scene. Laupichler et al. [4] summarized the structure of artificial intelligence literacy in higher and adult education. Cukurova et al. [5] used learning analysis to monitor the quality of online one-to-one tutoring. Celik[6] proposed Intelligence-TPACK to characterize teachers' expertise in integrating AI tools. Crompton and Burke[7] summarized the main technical paths of artificial intelligence application in higher education. Chan[8] established an AI policy education framework for university teaching. Chiu et al. [9] reviewed the implementation opportunities and research trends of the application of artificial intelligence in education. Pereira et al. [10] sorted out the design types and application boundaries of conversational agents in colleges and universities. Rets et al. [11] give suggestions for ethical implementation of predictive learning analysis; Zhou and Kang[12] enhanced the joint attention characterization in collaborative tasks by temporizing multimodal data.

Howard and Tondeur[13] discussed the composition of university teachers' digital ability under the background of hybrid future. Trevisan et al. [14] analyzed the key factors affecting the formation of teachers 'online teaching ability. Chiu et al. [15] illustrate the relationship between AI chat agents and learning motivation from the perspective of support mechanisms. Delcker et al. [16] investigated the prediction effect of AI ability on tool usage intention and real usage behavior. Sperling et al. [17] reviewed the research on AI literacy in teacher education. Chiu and Sanusi[18] proposed the idea of defining and evaluating the AI ability of students and teachers. Mah and Groß[19] identified the differentiated portraits and development needs of university teachers using AI. Henriquez et al. [20] verified the application value of learning analytics consulting tools based on three years of practice. Tammets and Ley[21] constructed the integration model of AI tools in teacher professional learning. Azanza et al. [22] revealed the organizational and individual factors that affect the use of digital technology by university teachers.

Table 1: Comparison of technical concerns and scope of application of existing studies

Reference Range	Main Research Content	Computational Focus	Applicable Features
[1]–[4]	Teachers' digital competence and AI literacy structure	Competency framework modeling and literacy dimension classification	Suitable for competence assessment, but does not directly support task scheduling
[5]–[8]	Learning analytics and policy support environment	Dashboard computation, behavior monitoring, and institutional framework design	Suitable for platform governance, but weak in interactive linkage
[9]–[12]	AI educational applications and multimodal collaboration	Conversational agents, joint attention, and temporal representation	Suitable for interaction recognition, but limited in task structure representation
[13]–[16]	University teachers' digital competence and tool use	Behavior prediction and usage intention analysis	Suitable for competence measurement, but insufficient for training path generation
[17]–[19]	Teachers' AI literacy and development profiling	Profile recognition, competence definition, and group classification	Suitable for difference analysis, but less connected to resource scheduling
[20]–[22]	Learning analytics tools and integration with professional learning	Advisory support, integration models, and influencing factor analysis	Suitable for application deployment, but system-level collaborative computing still needs further development

As shown in Table 1, the existing research has covered the direction of capability framework, analysis panel, dialogue agent, AI literacy and support tools. However, most of the work mainly focuses on functional verification or factor explanation, and less on the structural disassembly of training tasks, continuous representation of interaction states, and linkage calculation of resource allocation. The digital environment for teacher training has the characteristics of multi-task concurrency, heterogeneous content types, frequent role interactions and sensitive feedback timeliness. It is difficult to support the whole process of training choreography by only relying on static rules or single recommendation. Based on this, this paper puts the task-driven model and multi-dimensional interaction strategy into a unified digital training system, and constructs a deployable computing framework around task analysis, state representation, feedback calculation and path generation. Its recognition accuracy, response efficiency, scene stability and content adaptation performance are verified on real university teacher training data.

2 Task-driven modeling and multidimensional interaction mechanism design

2.1 University teacher training task analysis and digital behavior data modeling

In the digital training scenario of university teachers, the starting point of task analysis and behavior modeling is the structured arrangement of the original platform log. The system organizes events such as course research, case browsing, resource invocation, peer collaboration, assessment submission and feedback review into a unified sequence of teacher training events:

$$\mathcal{T} = \{e_1, e_2, \dots, e_N\} \tag{1}$$

Here, $\mathcal{T}$ represents the complete set of events in a single training cycle, e_t represents the t event, and N represents the total number of events in that cycle. This definition merges the originally scattered click, stay and commit records into a unified timeline, which is convenient for subsequent task chain identification and status tracking. In order to preserve the task semantics, resource context and interaction role of the event, each event is represented as a five-tuple $e_t = (u_t, q_t, r_t, s_t, \tau_t)$. Here, u_t represents the teacher identification, q_t represents the task category, r_t represents the resource unit number, s_t represents the interaction state, and τ_t represents the timestamp. Combining task dictionary, resource label table and state mapping rules, the system projects heterogeneous records from different schools, different courses and different training rounds into a unified coding space, so as to form comparable pairs of task behavior samples.

On this basis, a heterogeneous directed graph $G = (V, E)$ is used to model the training process. The node set V contains task nodes, resource nodes, interaction nodes and feedback nodes, and the edge set E describes the relations such as execution, access, cooperation and return. Considering that the training activities have the characteristics of stage advancement and concurrent collaboration, the system uses a sliding window with length L and step size Δ to segment the event sequence, and retains the temporal connection and cross-task association within the window. The association strength of any two nodes i and j is defined as follows.

$$A_{ij} = \alpha_1 \cos(q_i, q_j) + \alpha_2 e^{-\lambda|\tau_i - \tau_j|} + \alpha_3 \rho(s_i, s_j) \tag{2}$$

Here, A_{ij} represents the edge weight between node i and node j, q_i and q_j represent the task type embedding vector, $\cos(\cdot)$ represents the cosine similarity, $\rho(s_i, s_j)$ represents the interaction state coupling function, α_1, α_2 and α_3 represent the weight coefficients of the three types of relationships, and λ represents the time decay factor. This formula simultaneously describes the semantic proximity, temporal proximity and state cooperativity of tasks, so that the graph structure can reflect the real association in the training process.

To avoid over-connected edges caused by high-frequency clicks and repeated browsing, the system sets a threshold θ and a maximum out-degree d_{max} for structure screening. If $Aij < \theta$, the corresponding edge is deleted. If the out-degree of a node exceeds d_{max}, only the first d_{max} edges with the highest edge weight are kept. After filtering, the initial representation of the node is generated by task embedding, resource embedding, state embedding and time and location encoding:

$$h_i^{(0)} = \sigma\left(W_q q_i + W_r r_i + W_s s_i + W_p p_i + b\right) \tag{3}$$

where $h_i^{(0)}$ represents the initial feature vector of node i, q_i, r_i and s_i represent the embedding results of tasks, resources and states respectively, p_i represents the normalized temporal position vector, W_q, W_r, W_s and W_p represent the trainable projection matrix, b represents the bias term, and $\sigma(\cdot)$ represents the nonlinear activation function. This representation unifies the task content, resource dependency, interaction phase and time advance into the same feature space, which provides stable input for subsequent state recognition.

After the node encoding is completed, the local graph in the window needs to be compressed into a task fragment representation to support the subsequent training path generation and interactive decision calculation. To this end, attention aggregation is used to obtain the subgraph-level representation:

$$g_k = \sum_{i \in V_k} \beta_i h_i^{(0)}, \quad \beta_i = \frac{\exp\left(a^\top \tanh(W_h h_i^{(0)})\right)}{\sum_{j \in V_k} \exp\left(a^\top \tanh(W_h h_j^{(0)})\right)} \tag{4}$$

Here, g_k represents the task subgraph representation corresponding to the k time window, V_k represents the set of nodes within this window, β_i represents the contribution weight of node i to the subgraph representation, and W_h and a represent the aggregation parameters. In this way, the system can compress scattered events into structured fragments with task-directed representation, while retaining the coupling relationship between training goals, resource calls, interaction states and time evolution.

After the above processing, the teacher training process is transformed into a computable, comparable, and extensible task graph representation. The proposed representation can not only support the subsequent state representation module recognition training promotion stage, but also provide a unified data basis for multi-dimensional interaction strategy fusion, adaptive feedback calculation and resource path generation.

2.2 Representation mechanism of teacher training status under task-driven model

Under the task-driven model, teacher training status is not a single behavior label, but a dynamic representation composed of task advancement degree, resource absorption intensity, interaction and collaboration level and feedback response rhythm. To enable the system to recognize phase changes during continuous training, we construct a relation-aware encoding structure for state representation. The structure not only preserves the topological dependencies in the task graph, but also explicitly models the migration direction and stay differences between different task nodes, so as to form a state vector suitable for subsequent interaction decision calls. Fig. 1 shows the state representation mechanism of teacher training under the task-driven model. The input side is the task subgraph and the initial embedding of nodes, the middle part is completed in turn by relation attention aggregation, stage memory update, state gated fusion and segment-level output, and the end part generates a unified state representation and sends it to the subsequent strategy module.

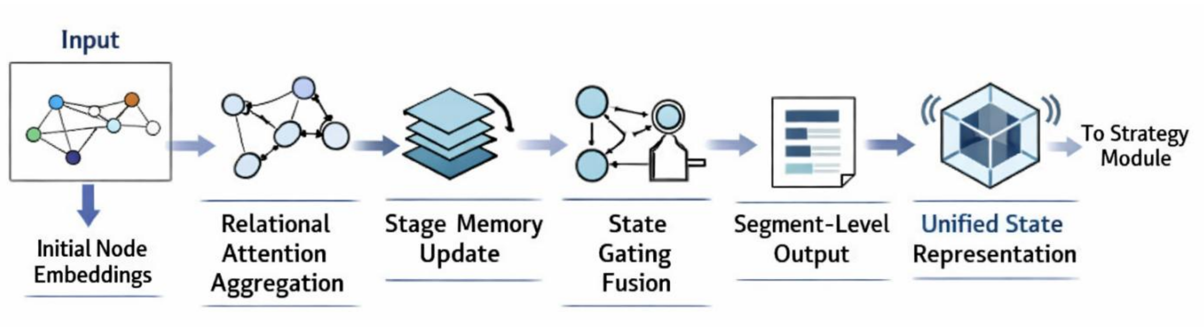


Figure 1: Teacher training state representation mechanism under task-driven model

In the encoding phase, node representations are no longer aggregated by fixed means, but adaptively propagated according to task type, edge relationship and temporal location. The message of node i receiving neighborhood information in layer l is expressed as follows.

$$m_i^{(l)} = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} W_{\psi(i,j)} h_j^{(l-1)} \tag{5}$$

Here, $m_i^{(l)}$ represents the neighborhood message received by node i at layer l, $\mathcal{N}(i$ represents the neighborhood set of node i, $W_{\psi(i,j)}$ represents the projection matrix with relation type $\psi(i,j)$,$h_j^{(l-1)}$ represents the feature vector of node j at the previous layer, and $\alpha_{ij}^{(l)}$ represents the attention weight. This formula makes three kinds of relationships, task dependency, resource call and feedback transmission, have different contributions in the propagation process. The relationship between nodes attention weight is defined as follows.

$$\alpha_{ij}^{(l)} = \frac{\exp\left(\mathrm{LeakyReLU}\left(a^{\top}\left[W_q h_i^{(l-1)} \| W_k h_j^{(l-1)} \| W_e e_{ij}\right]\right)\right)}{\sum_{p \in \mathcal{N}(i)} \exp\left(\mathrm{LeakyReLU}\left(a^{\top}\left[W_q h_i^{(l-1)} \| W_k h_p^{(l-1)} \| W_e e_{ip}\right]\right)\right)} \tag{6}$$

Here, $\alpha_{ij}^{(l)}$ represents the normalized influence coefficient of node i on node j, e_{ij} represents the relational embedding of edges, W_q and W_k are the query projection and key projection matrices respectively, W_e represents the relational mapping matrix, a represents the learnable parameter vector, and $\|$ represents the splicing operation. By introducing relation embedding and node alignment information, the system can distinguish the state effect of different training segments such as case study, collaborative discussion and evaluation submission.

After completing the local propagation, the system organizes the subgraph states in window order and uses gated memory to suppress transient fluctuations. The stage memory update for the KTH window is as follows.

$$c_k = f_k \odot c_{k-1} + u_k \odot \tilde{c}_k \tag{7}$$

Here, c_k represents the stage memory of the current window, c_{k-1} represents the historical memory of the previous window, $\tilde{c}_k$ represents the candidate state, f_k represents the forget gate, u_k represents the update gate, and the symbol $\odot$ represents element-wise multiplication. This update method makes short-term high-frequency clicks not directly cover the real training progress, and retains the transition characteristics during task switching.

In order to obtain the state vector that can be directly invoked by the subsequent multidimensional interaction strategy, the system further fuses the current structure representation with the phase memory, which is calculated as

$$s_k = \eta_k \odot g_k + (1 - \eta_k) \odot c_k, \quad \eta_k = \sigma\left(W_d d_k + W_r r_k + W_t t_k + b_\eta\right) \tag{8}$$

Here, s_k represents the final state vector of the k window, g_k represents the structure encoding result, η_k represents the fusion gate, d_k, r_k and t_k represent the current task density, feedback delay and resource occupancy characteristics, respectively, W_d, W_r and W_t represent the corresponding projection matrix. The state representation output by this step simultaneously describes the task completion trend, interaction activity and feedback stability, and can be used as a unified input for resource matching, path generation and adaptive feedback calculation.

In addition, the state representation module also performs identity alignment on the cross-round segments of the same teacher before output, and the stable features in the historical training are cached into the teacher-level index, which is used to weaken the offset caused by the length difference of a single session. The obtained state vector can not only reflect the current task chain position, but also retain individual participation style and resource preference, so that the system can maintain a consistent coding scale under different schools, different topics and different training densities. Thus, the state representation no longer stays at the log statistical layer, but is transformed into a computable, comparable, and traceable phase semantic representation.

2.3 Multi-dimensional interaction strategy fusion and adaptive feedback calculation method

After the state representation is completed, the system needs to translate the task status, resource provision, collaboration relationship and feedback timing into executable interactive decisions. The multi-dimensional interactive strategy fusion module takes the window-level state vector, the teacher's historical preference vector and the resource availability matrix as the joint input, and outputs the strategy distribution and feedback strength of the current stage. The interaction here is not limited to message push, but covers case recommendation, exercise trigger, peer collaboration connection, expert comment insertion and review prompt generation at the same time, so the calculation results must take into account task matching, timing rationality and platform load constraints. In order to avoid the single dimension dominant decision-making, the system takes task progress, resource relevance, behavior stability and feedback delay into the fusion calculation. Table 2 shows the computational role of each core dimension in policy fusion.

Table 2: Main computational dimensions for the fusion of multidimensional interaction strategies

Dimension	Data Basis	Functional Description
Task Progression	Training stage records, task completion evidence	Identifies the current stage position within the training chain
Resource Adaptation	Resource labels, access records, call frequency	Evaluates the consistency between content supply and task objectives
Collaboration Intensity	Collaboration logs, discussion records, expert feedback	Characterizes the intervention demand of collaborative support at the current stage
Feedback Timing	Submission intervals, response rhythm, review trajectories	Constrains the time window and output density of feedback generation

For each time window, the system first constructs the interaction context representation, and maps the current state, candidate resources and historical response records to the unified decision space. In order to enhance the coupling expression between the inputs of each dimension, this paper uses the bilinear scoring mechanism to model the m interaction, which is calculated as follows.

$$\zeta_{k,m} = v_m^{\top} \tanh(U_s s_k + U_r \bar{x}_k + U_u u_k + b_m) + s_k^{\top} B_m \bar{x}_k \tag{9}$$

where $\zeta_{k,m}$ represents the original score of the m type of interaction action under window k, s_k represents the current training state representation, $\bar{x}_k$ represents the candidate resource aggregation representation,u_k represents the teacher's historical preference representation, U_s, U_r and U_u are linear projection matrices, v_m and b_m are the parameters corresponding to the m type of action, and B_m is the bilinear relationship matrix. This formula not only retains the linear mapping results of three types of inputs, state, resource and preference, but also explicitly describes the cross matching strength between state and resource, so that the action ordering does not depend on the sum of a single vector.

After obtaining the original score of the action, the system continues to combine the platform load and the teacher response inertia to generate the executable policy strength. Instead of simple gated mixing, we introduce normalized temperature regulation and load penalty terms to obtain the final action distribution:

$$\pi_{k,m} = \frac{\exp\big((\zeta_{k,m} - \mu\ell_k - \nu\delta_k)/\tau_k\big)}{\sum_{n=1}^{M} \exp\big((\zeta_{k,n} - \mu\ell_k - \nu\delta_k)/\tau_k\big)} \tag{10}$$

Here, $\pi_{k,m}$ represents the final execution probability of the m action under window k, ℓ_k represents the current platform load level, δ_k represents the response inertia between the last two interactions of the teacher, μ and ν are penalty coefficients, and τ_kk represents the dynamic temperature parameter. This formula makes the output of the policy more gentle under the condition of high load, and also suppresses the probability of triggering the same kind of feedback repeatedly in a short time, so as to ensure that the interaction rhythm is consistent with the system load.

After the action distribution is determined, the system generates a comprehensive feedback value according to the task target template, resource consumption cost and recent feedback density, which is used as the basis for subsequent resource matching and path updating. The adaptive feedback function is defined as follows.

$$\mathcal{F}_k = \sum_{m=1}^{M} \pi_{k,m}\ \Omega_m(s_k, \bar{x}_k) + \lambda_1\ \kappa(s_k, y_k) - \lambda_2\ C_k - \lambda_3\ R_k \tag{11}$$

where $\mathcal{F}_k$ represents the comprehensive feedback value of window k, $\Omega_m(s_k, \bar{x}_k)$ represents the action function of the m action under the current state and resource conditions, y_k represents the target task template, $\kappa(s_k, y_k)$ represents the consistency function between the current state and the target template, C_k represents the resource consumption cost corresponding to the current feedback. R_k denotes the feedback density in the most recent time period, and λ_1, λ_2, and λ_3 are the balance coefficients. This formula makes the feedback generation consider the degree of task alignment, resource overhead and output density at the same time, avoiding high-frequency intervention and inefficient allocation in the training process.

After the above calculation, the system is able to complete policy ranking, intensity assignment and feedback output simultaneously within each task window. The interaction result has two functions. One is to control when and what support is pushed to the teacher's current task. On the other hand, the strategy weight and resource allocation order are continuously modified for the whole system. Therefore, the multi-dimensional interaction is no longer a static recommendation set, but a computational process that is continuously updated with the change of state, and provides direct input for resource matching and task path generation in the next section. At the same time, the system saves the acceptance result, completion evidence and look-back trajectory after each interaction, which are used to update the teacher preference representation online, and keep the strategy scale consistent with the feedback diameter in the subsequent rounds, so that the calculation link is more stable.

2.4 Resource matching and task path generation for training process

In the stage of resource matching and task path generation oriented to the training process, the system receives the teacher training state representation, interaction strategy distribution and feedback strength results output by the preorder module, and organizes them as the unified input of the resource scheduling layer. Different from static recommendation, the core of this stage is not a single content ranking, but under the condition of continuous progress of the training chain, it calculates the adaptation relationship between the current teacher state and the candidate resources, and generates the execution path that satisfies the task constraints, feedback rhythm and resource load constraints. To this end, the system retains stage semantics, task goals and collaboration requirements in the state representation in the resource matching side, and introduces transfer benefits, execution costs and remaining task distances in the path generation side to ensure that the output results not only have the ability of local content adaptation, but also maintain the consistency of the whole process of training progress. Fig. 2 shows the overall process of resource matching and task path generation module. The left input includes state representation, candidate resource pool and task template, and the middle one completes resource scoring, path expansion, cost correction and node selection in turn. The right output is the execution resource set and the next stage task node.

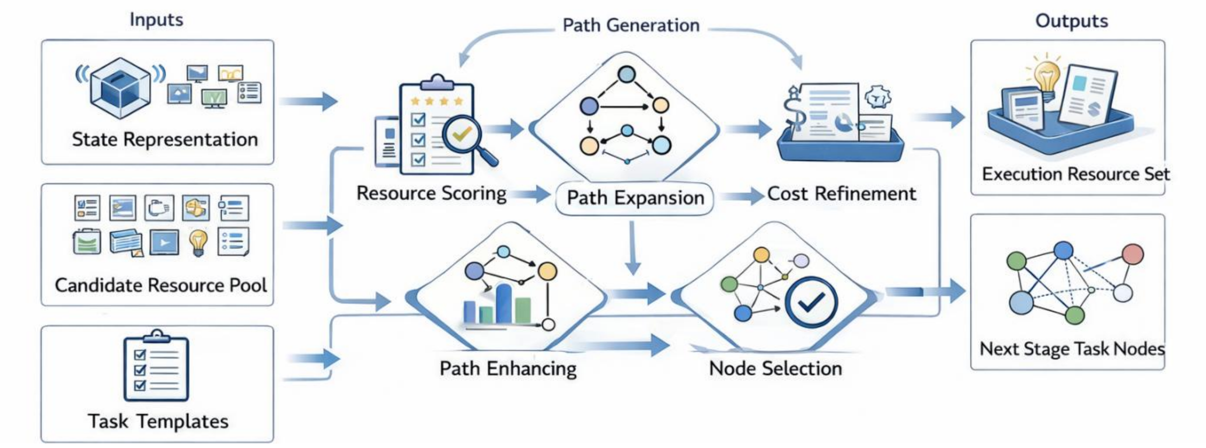


Figure 2: Resource matching and task path generation process for training process

In the calculation of resource matching, the system first represents the set of candidate resources as a matrix $R = \{r_1, r_2, \dots, r_J\}$, where each resource vector simultaneously contains attributes such as content topic, interaction type, time cost, and applicable phase. In order to enhance the cross expression between training state and resource features, the matching score between the state and the j resource in the k window is defined as follows.

$$M_{k,j} = \frac{s_k^{\top} Q r_j}{\|s_k\|_2 \|r_j\|_2} + \alpha\, \chi(y_k, r_j) - \beta\, \omega_j \tag{12}$$

where $M_{k,j}$ represents the comprehensive matching score of state s_k and resource r_j, Q represents the cross-space mapping matrix, $\chi(y_k, r_j)$ represents the consistency function between the target task template and the resource label, ω_j represents the resource execution overhead, α and β are the balance coefficients. This formula makes the resource selection not only depend on the semantic proximity, but also consider the task goal and execution cost at the same time, so as to form a more stable candidate ranking.

After obtaining the resource matching results, the system abstracted the training process into a directed path graph $P = (\mathcal{V}, \mathcal{E})$, where the node set $\mathcal{V}$ represents different task stages and the edge set $\mathcal{E}$ represents the stage transition relationship. For the transition from the current node v_t to the candidate successor node v_{t+1}, the path revenue function is defined as follows.

$$G(v_t, v_{t+1}) = \max_{j \in \mathcal{C}(v_{t+1})} \left[M_{k,j} + \xi_1 \psi(v_t, v_{t+1}) - \xi_2 c(v_t, v_{t+1}) \right] \tag{13}$$

Here, $\mathcal{C}(v_{t+1})$ represents the set of resources that can be allocated to node v_{t+1}, $\psi(v_t, v_{t+1})$ represents the reasonable degree of connection between adjacent stages, $c(v_t, v_{t+1})$ represents the execution cost of migrating from the current node to the next node, and ξ_1 and ξ_2 are the weight coefficients. This formula combines the optimal resource matching result with the stage transfer profit, so that the path expansion is constrained by both resource adaptation and process continuity.

In the path selection phase, the system further solves the subsequent optimal execution chain in a recursive manner. Let the function $V(v_t)$ denote the optimal residual path value from node v_t, then:

$$V(v_t) = \max_{v_{t+1} \in \mathcal{N}(v_t)} [G(v_t, v_{t+1}) + \gamma V(v_{t+1}) - \xi_3 \upsilon_t] \tag{14}$$

Here, $\mathcal{N}(v_t)$ represents the successor set of node v_t, γ represents the path discount coefficient, υ_t represents the resource occupancy risk in the current stage, and ξ_3 represents the risk penalty coefficient. Through this formula, the system can achieve a balance between local optimal matching and global path revenue, avoid excessive consumption of high-value resources in the early stage, and avoid content breakage and feedback imbalance in the later stage of training.

After solving the path, the control layer issues resource invocation instructions, feedback insertion instructions and cooperative connection instructions according to the optimal node sequence, and writes the execution results back to the state buffer and resource index area for the next round of matching calculation. The resource matching and path generation mechanism formed in this way can not only support the accurate advancement of the current training task, but also maintain the coordination relationship between strategy scale, content intensity and resource consumption in multiple rounds of training, which provides a traceable execution basis for subsequent system testing and effect evaluation.

3 Experimental setup and system implementation

3.1 Digital training system architecture and platform deployment environment

In the design of digital training system architecture and platform deployment environment, the system needs to meet multiple operational requirements such as task modeling and calculation, interactive service distribution, resource call control and training process record at the same time. Therefore, the platform structure adopts a hierarchical organization method of front-end presentation layer, business scheduling layer, model calculation layer and data support layer. The front-end presentation layer is responsible for course task display, feedback information push, collaboration entry call and training progress visualization, and maintains low latency communication with the back-end through asynchronous request mechanism. The business scheduling layer is responsible for task parsing, resource access verification, behavior log merging and feedback policy issuing, which is the core middle layer connecting interface services and model reasoning services. The computing layer of the model deploies core modules such as state representation, interactive decision-making, dynamic training path generation, and adaptive resource scheduling. The reasoning interface is encapsulated in a container-based way to ensure rapid migration and independent expansion under different training topics. The data support layer saves teacher portraits, task records, resource indexes, interaction logs and feedback caches, and provides retrieval and update capabilities through a unified interface. Table 3 summarizes the main deployment environments of digital training systems.

Table 3: Overview of the digital training system platform deployment environment

Deployment Module	Hardware/Software Environment	Specification	Functional Description
Server Host	Ubuntu Server 22.04	16-core CPU, 64 GB memory	Hosts backend services and model inference tasks
Front-end Interface	Vue3 + Element Plus	Adapted to 1920×1080	Implements task display, progress visualization, and interactive control
Model Service	Python + PyTorch + FastAPI	CUDA 11.8	Deploys state representation, strategy fusion, and path generation modules
Data Cache	Redis	In-memory key-value storage	Supports high-frequency state reading and rapid feedback writing
Data Management	PostgreSQL	Structured transaction processing	Stores teacher profiles, task logs, and resource indexes
Resource Storage	MinIO	Object storage service	Uniformly manages videos, case documents, and assessment attachments

The system server is deployed in Linux environment, Python, PyTorch and FastAPI are used to build model service, the front-end interface is based on Vue3 and Element Plus to achieve interactive display, Redis is responsible for high-frequency state cache, PostgreSQL is responsible for structured data management. The object storage module is responsible for the unified access of videos, case documents and evaluation attachments. This deployment mode ensures that model reasoning, resource scheduling and interface response can run collaboratively in the same architecture, and provides a stable and reproducible system foundation for subsequent experimental verification. At the same time, the peak shaving mechanism of message queue and unified routing control are introduced to ensure that the platform maintains stable throughput and service continuity in high concurrent training scenarios, and reduces the risk of interface jitter.

3.2 Construction and preprocessing process of university teacher training dataset

The training data of university teachers comes from the real operating environment of the deployed digital training system. The platform covered pre-job training, curriculum design research, case discussion, collaborative evaluation and feedback review, and ran for 16 weeks. The system terminal is composed of teacher operation end, management and audit end, resource server end and log collection end. All events are written to the unified log library through the interface gateway, and high-frequency behavior compensation synchronization is completed by the cache queue. Finally, the training data of 214 teachers were collected, and 96420 effective behavior records and 12860 feedback records were formed. The fields include task type identification, event trigger time, resource number, interaction status, feedback result, page stay time, collaborative object index, and teacher identity number. Each record was organized into a uniform structure sample, and constituted a task sequence set according to teacher identity and time order. In the preprocessing stage, it cleaned according to time legitimacy, task integrity and resource resolvability, deleted missing numbers, repeated writing, abnormal stay and invalid return records. When the consecutive interval exceeds the set threshold, the breakpoints are executed, and the valid sequences of length at least 10 are retained. Then, the discrete coding of task categories, the hash mapping of resource labels, the merging of status codes and the normalization of time and position were completed. The samples were non-overlapping divided according to the teacher's identity, and the ratio of training set to test set was set to 8:2. Finally, the samples were saved in three types of structured files: state sequence, task subgraph and resource index for subsequent model training and system evaluation. At the same time, a field check table and a resource index table are established to ensure the consistency and traceability of training samples across rounds.

3.3 Model training parameter configuration and evaluation index setting

In order to verify the training stability and deployment adaptability of the task-driven model in the digital training scenario of university teachers, this paper sets up a unified training scheme around the three links of state representation, interaction strategy fusion and path generation, and compares and configures the key parameter combinations. AdamW optimizer was used in the training process, and cosine annealing learning rate scheduling was used to control the gradient update amplitude to reduce the parameter oscillation in different training rounds. The input samples were grouped with teacher identity as the basic unit, and the task sequence length and resource call intensity were kept comparable within the same batch. Table 4 presents the main parameter Settings in the model training phase.

Table 4: Model training parameter configuration table

Parameter Item	Configuration 1	Configuration 2	Configuration 3	Configuration 4
Initial Learning Rate	0.001	0.0005	0.0003	0.0001
Batch Size	64	128	128	256
Number of State Encoding Layers	2	3	3	4
Hidden Dimension	128	256	192	256
Number of Training Epochs	80	100	120	100
Weight Decay	0.0001	0.0001	0.0005	0.0005

The parameter combinations in Table 4 are used to observe the convergence speed, state representation completeness and resource decision sensitivity of the model under different training intensities. The unified validation set performs one evaluation after each round of training, and four metrics are recorded: task recognition accuracy, interaction decision accuracy, path matching consistency, and average response delay. The task recognition accuracy is used to measure the system's ability to judge the training phase and behavior intention, the interaction decision accuracy reflects the consistency between the policy output and the manually labeled actions, the path matching consistency is used to describe the degree of fit between the system generation process and the target task template, and the average response time is used to measure the real-time processing ability in the deployment state. Table 5 lists the overall performance for different parameter combinations.

Table 5: Performance comparison for different combinations of parameters

Configuration	Task Recognition Accuracy/%	Interactive Decision Accuracy/%	Path Matching Consistency/%	Average Response Latency/ms
Configuration 1	90.8	88.9	87.6	82
Configuration 2	92.8	90.6	88.9	74
Configuration 3	91.9	89.7	88.1	79
Configuration 4	91.1	89.2	87.4	86

According to the comprehensive results, configuration 2 achieves a more balanced performance between accuracy, stability and response efficiency. When the batch size is increased to 128, the distribution of training samples is smoother, the three-layer state encoding structure enhances the extraction ability of stage semantics and resource dependence, and the 256-dimensional hidden representation retains enough expression space for the calculation of interactive strategies. When the initial learning rate is set to 0.0005, the parameter update process is more stable, and the validation curve in the later training period fluctuates less, so this paper uses configuration 2 as the default training scheme in the subsequent experiments. Therefore, the overall training is more stable.

4 Analysis of results

4.1 Comparison of task identification and interactive decision performance

In order to evaluate the applicability of the proposed method in the digital training scenario of university teachers, a comparative experimental group consisting of MLP, BiGRU, Transformer, GraphPolicy and the proposed model is constructed. MLP is used to provide feature mapping baselines, BiGRU is used to depict task sequence dependencies, Transformer is used to model long-range interaction relationships, GraphPolicy focuses on the structure propagation between resource nodes, and the proposed model combines state representation, multi-dimensional interaction fusion and path constraint calculation. Four types of training behaviors, including resource browsing, task submission, collaborative discussion and feedback review, are selected for evaluation, and compared from four dimensions: task recognition accuracy, interactive decision-making accuracy, path matching consistency and cross-scene transfer stability. Fig. 3 shows the test results of different models.

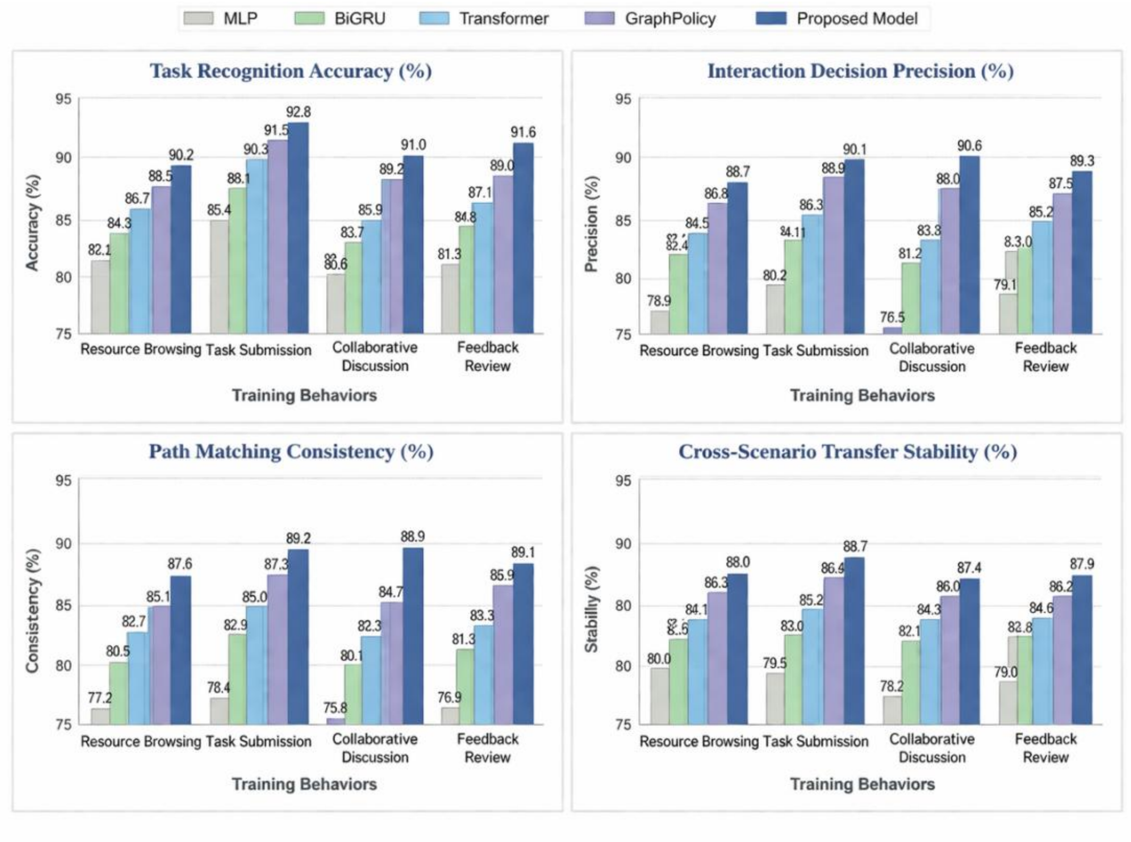
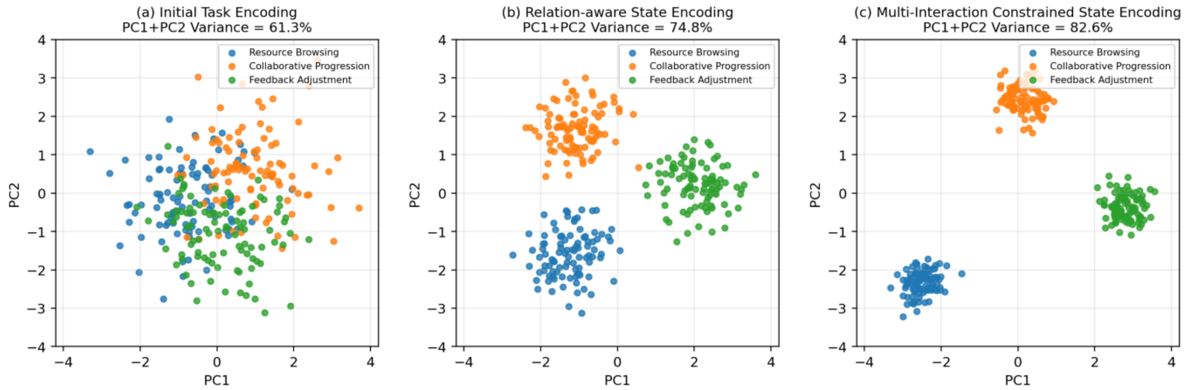


Figure 3: Performance comparison of different models in task identification versus interaction decision making

It can be seen from Fig. 3 that the proposed model performs best in all four indicators, among which the recognition accuracy of task submission behavior reaches 92.8%, and the accuracy of interactive decision-making in collaborative discussion context reaches 90.6%. Compared with GraphPolicy, the consistency of path matching of the proposed model is improved from 84.7% to 88.9%, indicating that task chain constraints and feedback timing control enhance the consistency of process output. In the cross-scenario transfer test, the proposed model maintains 87.4% stability, which indicates that the proposed method has the ability to adapt to the switching of training topics and the change of resource density. The overall results show that the joint design of task-driven modeling and multi-dimensional interaction strategy can improve the comprehensive performance of system in recognition, decision making and flow control.

4.2 Spatial distribution characteristics of training state representation

In order to test the influence of state representation mechanism on the spatial distribution of training states, this paper sets up three comparison stages, corresponding to the initial task encoding, relation-aware state encoding and state encoding after fusing multi-dimensional interaction constraints, and observe the change of representation space on a unified sample set. In the experiment, all state vectors are mapped to the same dimension, and then compressed to a two-dimensional plane by principal component analysis. The cumulative variance explained by the first two principal components is 61.3%, 74.8% and 82.6%, respectively. The samples are labeled according to three training states: resource browsing type, collaborative promotion type and feedback adjustment type. Fig. 4 shows the distribution results of the three state representations in the two-dimensional space.



(a) Initial task encoding distribution; Figure 4 (b) Distribution of relation-aware state encoding; Figure 4 (c) Distribution of state codes after fusing multi-dimensional interaction constraints

Figure 4: Comparison of spatial distribution characteristics of training state representation

In Fig. 4 (a), the initial task coding overlaps obviously, and the average contour coefficient of the three types of samples is only 0.31, the average center distance between classes is 1.84, and the average dispersion radius within class is 1.27, indicating that it is difficult to stably describe the differences in the training stage by only relying on the original task features. In Fig.4 (b), after relation-aware state encoding, the average contour coefficient is increased to 0.47, the center distance between classes is expanded to 2.63, and the overlap ratio of collaborative promotion and feedback adjustment samples is reduced from 22.4% to 13.1%, indicating that task dependence and time order information enhance the structural constraint ability of state representation. In Fig.4 (c), after the introduction of multidimensional interaction constraints, the average contour coefficient is further increased to 0.58, the center distance between classes is increased to 3.41, the dispersion radius within class is reduced to 0.79, the resource browsing samples and the other two types of states form a stable separation in the direction of the second principal component, and the proportion of samples in the boundary area is controlled at 6.8%. The results show that the state representation continuously compresates redundant representations and strengthens decision boundaries by jointly modeling task chain position, interaction strength and feedback rhythm, which provides a more stable input basis for subsequent resource matching and path generation.

4.3 Results of multi-dimensional interaction path matching

In order to test the path generation ability of multi-dimensional interaction strategy in teacher digital training, this paper constructs a node matching experiment between the standard training process and the system recommendation process. The standard path contains six nodes, which are entering the training home page, receiving task description, browsing case resources, participating in collaborative discussion, viewing feedback reports and submitting completion confirmation, corresponding to task initiation, content absorption, interactive promotion, result review and end marking in the training chain. The system recommendation path is dynamically generated by the model according to the state representation, resource adaptation score and feedback strength, and a total of six strategy nodes are output, corresponding to the home page entry, task resolution, case call, collaboration trigger, feedback presentation and result recovery. After the two types of paths are semantically encoded, the node similarity heat map is constructed, and the results are shown in Fig. 5.

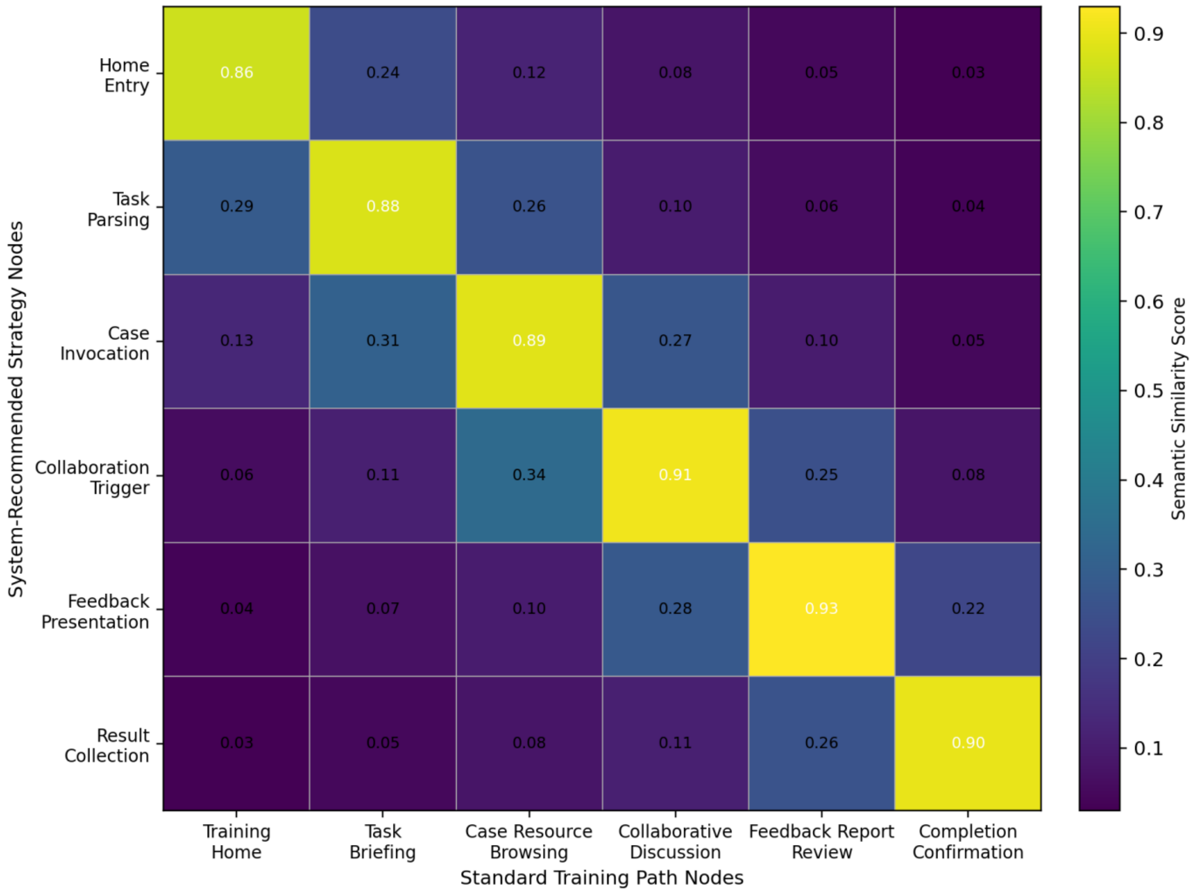


Figure 5: Heat map of node similarity between multidimensional interaction paths and standard training paths

The heat map shows that the fourth node and the fifth node form a high-value area, and the matching degree reaches 0.91 and 0.93 respectively, indicating that the system can stably recover the target process in the stage of collaborative promotion and feedback presentation. The matching degree of the third node is 0.89, indicating a high consistency between the case resource call and the standard task chain. The matching degrees of the first node and the second node are 0.86 and 0.88, respectively, which are slightly lower than those of the middle and late segments, but the hot area distribution still remains concentrated, indicating that the system has the path alignment ability at the beginning stage of the task. The sixth node has a matching degree of 0.90, which indicates that the mapping relationship between result recovery and completion confirmation is relatively stable. On the whole, the average matching degree between the system generated path and the standard training path reaches 0.895, and the path deviation is mainly concentrated in the early task explanation link, and the fitting is more stable in the middle and later stages, which indicates that task-driven modeling and interactive constraint calculation can effectively improve the structural consistency and semantic fit of the training process generation.

4.4 System response delay and resource scheduling efficiency test

In order to test the response ability and scheduling efficiency of the digital training system under different operating loads, this paper constructs five typical training scenarios, which correspond to single teacher login, group training concurrent, cross-topic collaborative switching, feedback centralized review and platform-level high load operation. The first scenario focuses on initial task loading. Scenario two corresponds to multiple teachers accessing case resources at the same time. Scenario 3 simulates the continuous jump of task chain and collaboration discussion in parallel. Scenario 4 emphasizes the cross execution of report review, resource callback and feedback writing. Scenario five is then used to observe the system resilience under a high density of requests. Fig. 6 illustrates the average response delay, resource scheduling success rate, and computing resource occupancy variation in each scenario.

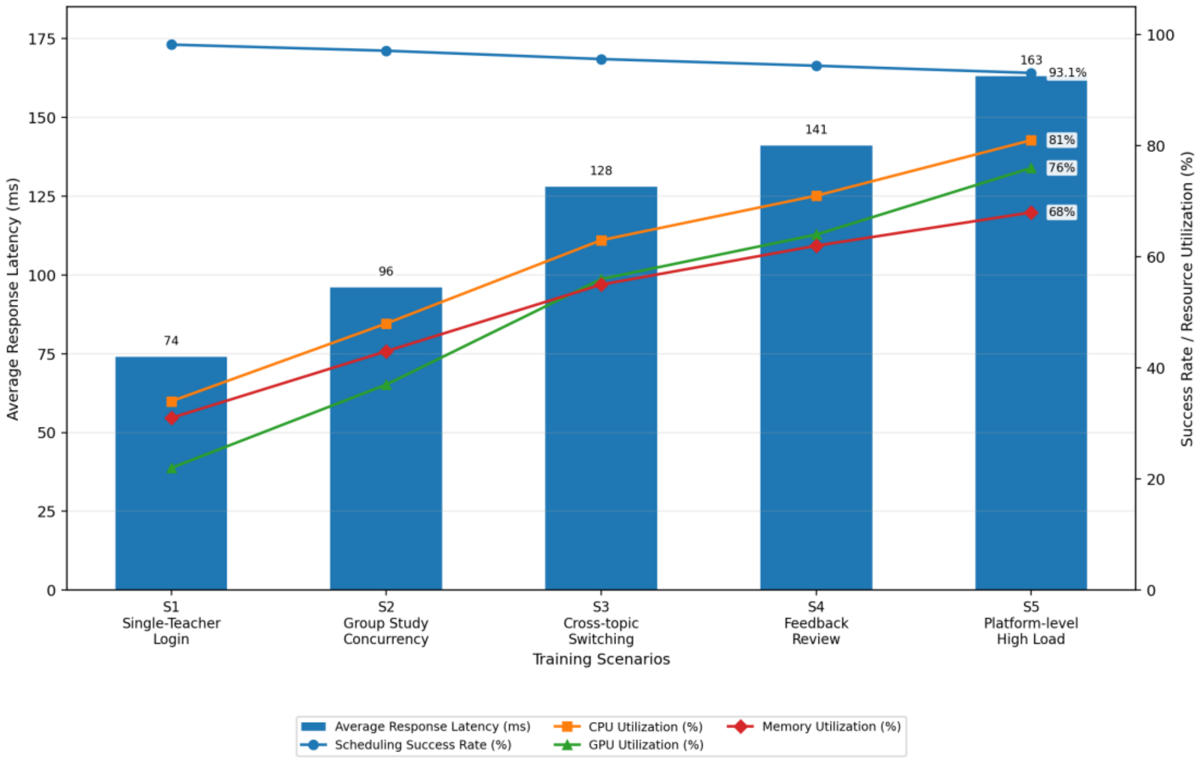


Figure 6: Comparison of system response delay and resource scheduling efficiency under different training scenarios

As can be seen from Fig. 6, the average response delay increases from 74 ms in scenario I to 163 ms in scenario V, and the success rate of scheduling remains above 93.1%, indicating that task distribution and resource orchestration still have good stability under high concurrency conditions. The delay growth is more obvious in scenarios three and four, reaching 128 ms and 141 ms, respectively, indicating that cross-module calls and feedback writeback will increase the middle layer scheduling overhead. The CPU occupancy rate reaches 81% in scenario five, and the GPU occupancy rate reaches 76%, while the memory fluctuation is controlled within 68%, indicating that the cache reuse and asynchronous queue mechanism effectively weaken the peak pressure. Under the condition of repeated requests, the resource hitting delay of scenario 2 and scenario 3 is reduced to 52 ms and 67 ms, respectively, and the first loading phase is shortened by 31.6% and 28.4%, respectively, indicating that the prefetching strategy and state caching can support the promotion of the training process.

4.5 System stability verification under different training scenarios

In order to further verify the operation stability of the system in different training scenarios, this paper uses the system level continuous monitoring method to repeat the test of five scenarios: single teacher task execution, group collaborative training, cross-topic resource switching, feedback centralized review and high load concurrent platform. In the experiment, each type of scenario was run continuously for 30 rounds, and four indicators including average response delay, resource utilization, task completion rate and state drift amplitude were recorded to observe the overall stable performance of model reasoning, resource distribution and feedback writeback in the process of task chain advancement. Fig. 7 shows the system stability results for different scenarios.

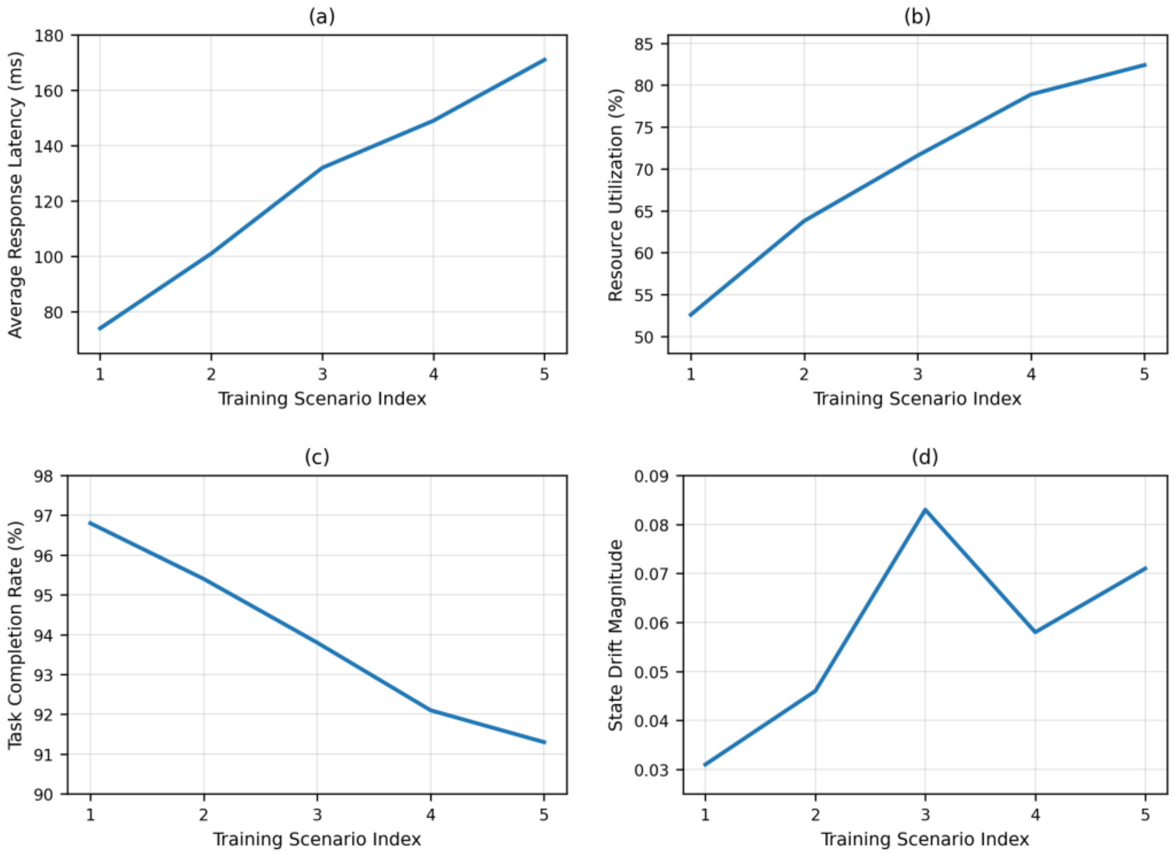


Figure 7: System stability validation results for different training scenarios

As can be seen from Fig. 7, (a) shows that the average response delay continues to rise with the increase of scene complexity, from 74 ms in the single-teacher task scenario to 171 ms in the high-load scenario of the platform. (b) shows that the resource utilization rate increases from 52.6% to 82.4%, indicating that the system resource occupation is significantly improved after multiple modules are continuously called. (c) indicates that although the task completion rate has decreased, it has always maintained above 91.3%, and the overall process maintenance ability is relatively stable. (d) shows that the state drift amplitude reaches 0.083 in the cross-topic resource switching scenario, which is the highest in the five scenarios, indicating that frequent switching is more likely to cause intermediate representation fluctuations. Combined with the changes of the four indicators, it can be seen that the system still maintains stable operation under the condition of high concurrency, and the standard deviation of each scene index is less than 4.7%, and the dispersion of repeated test results is small. Looking back, the resource utilization rate is 78.9%, and the response delay is controlled within 149 ms, indicating that the writeback mechanism is stable.

4.6 Evaluation of training content adaptation effect and task completion quality

In order to test the content adaptation ability and completion quality of the system in different training tasks, this paper selects six representative training tasks of university teachers as evaluation objects, which are platform guidance training, curriculum design research and training, case diagnosis analysis, collaborative teaching and research discussion, feedback report interpretation and comprehensive ability assessment. The six types of tasks cover typical processes such as task initiation, resource absorption, interactive promotion, result review and phase wrapping, which can completely reflect the adaptation performance of the digital training system in the real operating environment. The system calculates the content adaptation rate and task completion quality scores under various tasks, and draws the results as a comparison chart. As shown in Fig. 8, different tasks show obvious differences in content adaptation effect and completion quality.

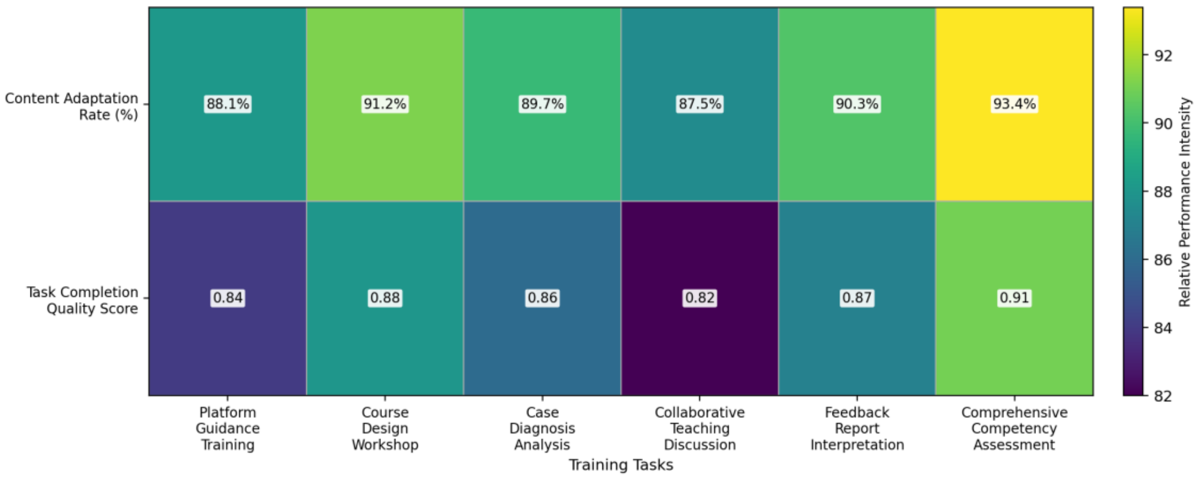


Figure 8: Comparison of content adaptation rate and task completion quality under different training tasks

Fig. 8 shows that the comprehensive ability assessment achieves the highest overall performance, the content adaptation rate reaches 93.4%, and the task completion quality score is 0.91, indicating that the system can stably output matching resources and feedback strategies in tasks with clear goals and relatively complete paths. Course design research and case diagnosis analysis reached 91.2%, 89.7% and 0.88, 0.86, respectively, indicating that state representation and path generation had a strong supporting effect on medium and high complexity training tasks. The content adaptation rate of collaborative teaching and research discussion is 87.5%, and the completion quality is 0.82. Although the overall performance is stable, the parallel interaction of multiple roles will increase the fluctuation of the task chain. The adaptation rate and completion quality of platform guidance training are 88.1% and 0.84, respectively, and the interpretation of feedback reports is 90.3% and 0.87, respectively, which shows that the output of the system in the clear structure task is relatively stable.

In order to further analyze the contribution of key modules to the final results, this paper conducts ablation experiments on the basis of the complete model, removing task-driven coding, multi-dimensional interaction fusion and path constraint modules respectively, and statistics the content adaptation rate and task completion quality change. As shown in Table 6, the full model remains optimal in both metrics.

Table 6: Results of ablation experiments for key modules

Model Configuration	Content Adaptation Rate/%	Task Completion Quality
Full Model	93.4	0.91
Without Task-driven Encoding	88.6	0.84
Without Multidimensional Interaction Fusion	87.9	0.82
Without Path Constraints	88.1	0.80

Table 6 shows that after removing task-driven encoding, the content adaptation rate drops to 88.6% and the completion quality drops to 0.84. After removing the multi-dimensional interaction fusion, the two indexes further decreased to 87.9% and 0.82. After removing the path constraints, the quality of completion decreases the most significantly, to only 0.80, indicating that this module has a direct effect on the process bundle and result consistency. Combining the results in Fig. 8 and Table 6, it can be seen that the system performance not only depends on the content recall ability, but also is closely related to the interaction chain stability, path continuity and feedback timing control. The overall results show that task-driven modeling, multi-dimensional interactive fusion and path generation constraints jointly determine the content adaptation accuracy and task execution quality in university teachers 'digital training.

5 Discussion

The experimental results show that the joint design of task-driven modeling, multi-dimensional interaction fusion and path generation constraints makes the system show stable advantages in task identification, interactive decision-making, path matching, response delay and content adaptation. Compared with the baseline methods that only rely on sequence modeling or single structure propagation, the proposed framework can simultaneously preserve the stage relationship, resource dependence and feedback rhythm in the training process, thus forming clearer state boundaries and more stable decision outputs in complex training chains. This result shows that the digital training of university teachers is not suitable to be processed by only static recommendation or simple classification, and the structural computing framework oriented to task chain should be adopted.

From the mechanism level, the state representation module strengthens the continuous association between training stages, the interaction strategy fusion module improves the degree of collaboration between resource scheduling and feedback timing, and the path generation module ensures the process integrity in the process of task advancement. With the combined effect of the three, the system can not only maintain low fluctuation in high-frequency interaction scenes, but also maintain high matching degree in multi-topic switching conditions. At the same time, the completion quality in the collaborative discussion and cross-topic switching scenarios is still lower than that in the comprehensive ability assessment task, indicating that the concurrent and asynchronous feedback of multiple roles will increase the uncertainty of the intermediate representation. The current data still comes from a single platform environment, and the deployment conditions also depend on the support of stable computing power. Therefore, the cross-college sample can be further expanded in the future, and the lightweight reasoning and incremental update mechanism can be introduced to enhance the portability and deployment flexibility of the system in a wider teacher training environment.

6 Conclusions

Facing the digital training scenario of university teachers, this paper constructs a system framework of the synergy between task-driven model and multi-dimensional interaction strategy, and forms a complete calculation chain around task analysis, state representation, feedback calculation, resource matching and path generation. Experimental results show that the framework has achieved stable performance in task recognition accuracy, interactive decision accuracy, path matching consistency and system response efficiency. The task recognition accuracy reaches 92.8%, interactive decision accuracy reaches 90.6%, path matching consistency reaches 88.9%, and the average response delay is controlled at 74 ms. The content adaptation rate reaches 93.4%, which shows that the task chain modeling and interaction constraint calculation can effectively enhance the structural expression ability and execution stability of the digital training system. The experimental results show that the framework has shown good stability, but there is still room for further improvement in terms of sample coverage, complex collaboration scene processing, and deployment adaptation under resource-constrained conditions. The existing samples mainly come from a single platform environment, and the differences of cross-college and interdisciplinary training scenarios have not been fully included. High concurrent requests in collaborative discussion and cross-topic switching tasks put additional pressure on path generation and feedback timing. The system operation also depends on the caching mechanism and computing power conditions to a certain extent, and the service elasticity in resource-constrained environments needs to be further verified. The follow-up research can be carried out from three directions. The first is to expand the multi-institution teacher training data and enhance the cross-scene transfer ability of the model. Secondly, it introduced lightweight reasoning, incremental update and graph structure compression mechanism to improve the efficiency of real-time service. Thirdly, richer behavioral signals and task semantic information are integrated to further improve the content adaptation and process control ability in the complex training chain, and improve the generalization level and performance.

Funding

The 2024 Higher Education Science Research Planning Project of China Association of Higher Education

Project Name: "Task - Driven, Multi - Dimensional Interaction, and Evaluation - Promoted" - The Construction of Digital Teacher Training Model and System

Project Number: 24PX0405

References

[1] Basilotta-Gómez-Pablos V, Matarranz M, Casado-Aranda L A, et al. Teachers' digital competencies in higher education: a systematic literature review[J]. International journal of educational technology in higher education, 2022, 19(1): 8.

[2] Susnjak T, Ramaswami G S, Mathrani A. Learning analytics dashboard: a tool for providing actionable insights to learners[J]. International Journal of Educational Technology in Higher Education, 2022, 19(1): 12.

[3] Essel H B, Vlachopoulos D, Tachie-Menson A, et al. The impact of a virtual teaching

assistant (chatbot) on students' learning in Ghanaian higher education[J]. International Journal of Educational Technology in Higher Education, 2022, 19(1): 57.

[4] Laupichler M C, Aster A, Schirch J, et al. Artificial intelligence literacy in higher and adult education: A scoping literature review[J]. Computers and Education: Artificial Intelligence, 2022, 3: 100101.

[5] Cukurova M, Khan-Galaria M, Millán E, et al. A learning analytics approach to monitoring the quality of online one-to-one tutoring[J]. Journal of Learning Analytics, 2022, 9(2): 105-120.

[6] Celik I. Towards Intelligent-TPACK: An empirical study on teachers’ professional knowledge to ethically integrate artificial intelligence (AI)-based tools into education[J]. Computers in human behavior, 2023, 138: 107468.

[7] Crompton H, Burke D. Artificial intelligence in higher education: the state of the field[J]. International journal of educational technology in higher education, 2023, 20(1): 1-22.

[8] Chan C K Y. A comprehensive AI policy education framework for university teaching and learning[J]. International journal of educational technology in higher education, 2023, 20(1): 38.

[9] Chiu T K F, Xia Q, Zhou X, et al. Systematic literature review on opportunities, challenges, and future research recommendations of artificial intelligence in education[J]. Computers and Education: Artificial Intelligence, 2023, 4: 100118.

[10] Pereira D S M, Falcão F, Costa L, et al. Here's to the future: Conversational agents in higher education-a scoping review[J]. International Journal of Educational Research, 2023, 122: 102233.

[11] Rets I, Herodotou C, Gillespie A. Six Practical Recommendations Enabling Ethical Use of Predictive Learning Analytics in Distance Education[J]. Journal of Learning Analytics, 2023, 10(1): 149-167.

[12] Zhou Y, Kang J. Enriching Multimodal Data: A Temporal Approach to Contextualize Joint Attention in Collaborative Problem-Solving[J]. Journal of Learning Analytics, 2023, 10(3): 87-101.

[13] Howard S K, Tondeur J. Higher education teachers' digital competencies for a blended future[J]. Educational technology research and development, 2023, 71(1): 1-6.

[14] Trevisan O, De Rossi M, Christensen R, et al. Factors shaping faculty online teaching competencies during the Covid-19 pandemic[J]. Educational technology research and development, 2023, 71(1): 79-98.

[15] Chiu T K F, Moorhouse B L, Chai C S, et al. Teacher support and student motivation to learn with Artificial Intelligence (AI) based chatbot[J]. Interactive Learning Environments, 2024, 32(7): 3240-3256.

[16] Delcker J, Heil J, Ifenthaler D, et al. First-year students AI-competence as a predictor

for intended and de facto use of AI-tools for supporting learning processes in higher education[J]. International Journal of Educational Technology in Higher Education, 2024, 21(1): 18.

[17] Sperling K, Stenberg C J, McGrath C, et al. In search of artificial intelligence (AI) literacy in teacher education: A scoping review[J]. Computers and Education Open, 2024, 6: 100169.

[18] Chiu T K F, Sanusi I T. Define, foster, and assess student and teacher AI literacy and competency for all: Current status and future research direction[J]. Computers and Education Open, 2024, 7: 100182.

[19] Mah D K, Groß N. Artificial intelligence in higher education: exploring faculty use, self-efficacy, distinct profiles, and professional development needs[J]. International Journal of Educational Technology in Higher Education, 2024, 21(1): 58.

[20] Henríquez V, Guerra J, Scheihing E. The impact of an academic counselling learning analytics tool: Evidence from 3 years of use[J]. British Journal of Educational Technology, 2024, 55(5): 1884-1899.

[21] Tammets K, Ley T. Integrating AI tools in teacher professional learning: a conceptual model and illustrative case[J]. Frontiers in artificial intelligence, 2023, 6: 1255089.

[22] Azanza G, Korres O, Paños-Castro J, et al. Empowering university lecturers in the digital age: Exploring the factors influencing the use of digital technologies in higher education[J]. Education Sciences, 2024, 14(7): 728.