Adaptive Learning Driven by Web Technologies: A Model-Driven Development Framework for AI Enhanced English Intelligent Tutoring Web Applications

Yuqing Ge[1,*], Jinjin Chu[1] and Yating Guo[2]

[1]*School of Foreign Languages, International Education Department, Sias University, Xinzheng, Henan 451100, China*
[2]*Inner Mongolia Electronic Information Technology College, Inner Mongolia Autonomous Region, Hohhot 010028, China*
E-mail: Rosie9029@163.com
[*]*Corresponding Author*

Received 18 March 2026; Accepted 19 May 2026

Abstract

There are some problems in the existing English teaching network applications, such as low personalized adaptability, outdated teaching strategies and weak multimodal interaction, limited teaching paths, single feedback mechanism, weak adaptability of learners, low learning efficiency and low retention rate. In order to solve these problems, this paper proposes a Model-Driven Development (MDD) framework that integrates Web technology and artificial intelligence (Web-AI-I), and builds a lightweight, cross-device English intelligent teaching application through the Web native technology stack. Multi-modal data perception, AI adaptive decision-making and real-time feedback optimization are integrated to solve the coupling problem between teaching logic and Web application architecture through MDD mode. Through the experiment, the response time of learning path adaptation

Journal of Web Engineering, Vol. 25_6, 1161–1192.
doi: 10.13052/jwe1540-9589.2566

can be effectively shortened to 0.2 s, the English vocabulary mastery rate is 52% higher than that of traditional Web teaching applications, the grammar error correction accuracy rate is 91.8%, the average resource occupancy rate in cross-device scenarios is only 18.3%, and the MTA is 412, which is helpful to realize the large-scale landing of English intelligent teaching Web applications.

Keywords: Web technology, adaptive learning, artificial intelligence, English teaching, model driven development, multimodal interaction.

1 Introduction

Model-Driven Development (MDD) is a software construction method, which can abstract the teaching domain model and Web technology model into an abstract teaching domain model, and solve problems through the separation of business logic and technology. Combined with artificial intelligence technology (natural language processing, machine learning, recommendation algorithm), it can further enhance the adaptability of Web applications and realize the closed-loop teaching of perception-decision-feedback. Although some progress has been made in research on the integration of Web-side English teaching and AI, there are some problems, such as heavy technology stacking, light teaching logic modeling, insufficient multi-modal data fusion and insufficient model lightweight adaptation. Therefore, it is of great theoretical and practical value to establish a Web-technology-driven MDD framework to realize the efficient development and large-scale application of AI-enhanced English intelligent teaching Web applications.

The core idea of MDD is to separate business logic from technical implementation and automatically generate code through model transformation, which naturally meets the development needs of English intelligent tutoring system. English teaching knowledge is highly structured, and the language levels, knowledge points and skill dimensions in the modern English teaching system (such as CEFR standards) can be directly formalized into entities, attributes and relationships in the domain model to meet the needs of MDD modeling. Secondly, English teaching strategies need to be iterated continuously according to the updating of textbooks and the reform of examinations. MDD concentrates the teaching logic on the domain model, realizes one modification, synchronizes the whole system, and greatly reduces the cost of iteration. Finally, English teaching needs to meet the needs of cross-device interaction, and the platform-independent model (PIM) of MDD can be

transformed into a multi-platform specific model (PSM) to achieve one-time modeling and multi-terminal deployment. Compared with the limitations of traditional Web development, such as the coupling of teaching logic and technology, the long iteration cycle (about 10 days per time), the MDD framework realizes the decoupling of teaching logic and technical modules by explicitly modeling English teaching knowledge, shortens the single iteration cycle to 2 days, and the reuse rate of cross-device code is over 95%, effectively reducing maintenance costs, improving system agility and teaching adaptation accuracy.

The development of digitalization promotes the transfer of English education from offline to Web. English teaching network applications can achieve cross-platform access, rich resources, be easy to use, and become an important tool for autonomous learning. However, the main problems of the current mainstream applications are: (1) lack of personalized teaching methods, mostly using one-size-fits-all fixed courses, which cannot adapt to the language level, learning rhythm and cognitive style of learners; (2) single interaction mode, mainly text and video playback, and lack of multi-modal interaction, such as voice and behavior, which cannot reflect real language communication; (3) feedback mechanism is specific, and grammar correction and pronunciation guidance need static prompts afterwards, which cannot reflect the dynamic needs in learning; (4) development mode is rough, teaching logic is highly coupled with the Web front-end, function iteration is slow, and cross-device adaptation is difficult.

2 Related Work

In recent years, with the rapid development of network technology and intelligent algorithms, English learning systems based on Web and mobile terminals have become a research hotspot in the field of educational technology. Dong [1] discussed the application of artificial intelligence software integrated with semantic Web technology in English teaching, emphasizing its potential in personalized learning resource recommendation. Similarly, Lei [2] and Sun [3] studied college English learning systems based on Web technology and mobile terminals, and demonstrated their effectiveness in improving learning convenience and interactivity. Kopylova and Hockicko [4] evaluated a Web application named “Spatial Information System and Technology” through experiments, and confirmed its practical value in students’ English training and assessment. In terms of mobile learning, Rukhiran et al. [5] and Samudra and Setiyadi [6] developed English learning

applications based on Web and mobile terminals, emphasizing their important role in the digital teaching framework. Cerna et al. [7] introduced the concept of adaptive learning into the English teaching of engineering graphics, demonstrating the adaptive teaching ability of intelligent Web applications.

Some studies focus on the implementation of specific functions, such as "English at a click" [8] developed by Andrejeviæ and Nejkovic. Yuniarti et al. [9] developed an innovative English learning application "FAFIT" based on Scratch, and Rodríguez et al. [10] developed an English writing grammar correction application "MindTer" using natural language processing technology. Manalu et al. [11] designed a chat robot application based on the Boyer-Moore algorithm, which provides a new form of intelligent interaction for Web English learning.

In the field of English teaching research driven by Web technology and artificial intelligence, scholars have studied aspects of system construction, mode innovation, quality evaluation, and resource service. Li et al. [12] proposed an intelligence-driven multi-dimensional collaborative hybrid teaching model, and completed the model design, implementation and validity verification based on digital Web technology; Wang [13] applied 5G Web service based on AI speech recognition to college English hybrid teaching, expanding the landing scene of intelligent technology in teaching; Xu [14] constructed an online and offline hybrid teaching mode of professional English based on Java Web, which provided a structured scheme for course implementation; Ruan [15] constructed an online English teaching quality evaluation model by using Web embedded system and machine learning, which improved the intelligence and objectivity of evaluation; Zhang [16] designed an English teaching resource sharing platform based on mobile Web technology, which realized the efficient integration and distribution of resources. Zhang [17] further built a Web-based college English teaching collaboration platform, which strengthened the interaction and collaboration ability in the teaching process; Abdykhalykova et al. [18] verified the effectiveness of the Web 2.0 test program in college English teaching, providing a practical basis for network assessment; Zhou [19] conducted a study on college English writing live classroom teaching based on B5G and perception devices, and analyzed its impact on learners' physical and mental state; Sun and Su [20] realized the automatic generation of English teaching information through Web information automatic extraction technology, which improved the efficiency of obtaining teaching resources; Wang and Ye [21] designed an auxiliary training system for oral English teaching based on Web services, which provided intelligent support for oral skills training. On the whole, the existing research

focuses on the deep integration of Web technology, artificial intelligence and English teaching, covering platform development, mode innovation, quality evaluation, intelligent resource generation and other directions, laying a solid theoretical and practical foundation for this study.

3 Method

3.1 Designable Concept

This paper proposes an MDD framework that Integrates Web technology and Artificial Intelligence (Web-AI-I). It extracts learner profiles from multimodal data and generates personalized teaching plans through AI algorithms based on data such as language proficiency, learning style, and behavioral habits. It establishes an English teaching domain model and a Web technology model to decouple teaching logic, business processes, and technology, thereby supporting rapid iteration and cross-device adaptation. Based on a Web-native lightweight technology stack and a cloud–device collaborative architecture, it ensures AI algorithm performance while reducing device resource consumption, and supports cross-platform applications.

3.2 Technical Architecture

In order to develop an English intelligent tutoring network application based on artificial intelligence, this paper establishes a model-based development framework. The technology is based on a four-layer distributed technical architecture, which maintains a complete closed loop of data input, teaching decision-making and teaching output from top to bottom. In order to show the functions of different layers and illustrate the flow chart of data flow and interaction, this paper establishes a four-layer distributed technical architecture and data closed-loop flow chart, as shown in Figure 1.

As shown in Figure 1, the system describes the relationship among the application service layer, AI adaptive layer, data fusion layer and Web technology support layer. The idea of data acquisition $\rightarrow$ fusion processing-intelligent decision-making $\rightarrow$ service output, provides a blueprint for the design and implementation of each module in the next step.

3.2.1 Application service layer

The application service layer is a service layer for face-to-face interaction with learners, including the core functions of English teaching, such as vocabulary, grammar, oral dialogue, reading, and writing. The responsive

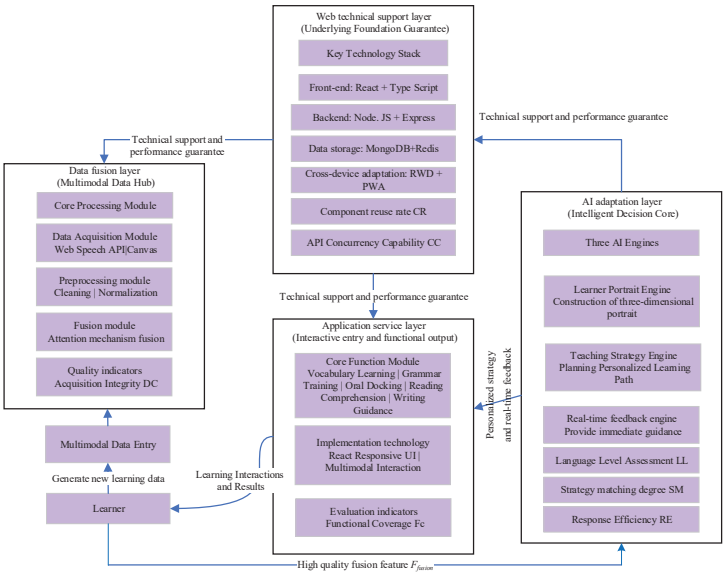

Figure 1 Four-layer distributed technical architecture and data closed-loop flow chart.

User Interface (UI) is designed through the React framework, which supports the design of PC, mobile phone, tablet and other terminals, and the functions can be combined for multiple uses. The function can support text input, voice, and handwritten annotation. In order to define the coverage of the application service layer function to the core teaching function and make the function design more complete and targeted, the function coverage formula of the application service layer is defined as:

$$F_C = \frac{\sum_{i=1}^{n} w_i \cdot f_i}{W_{total}} \tag{1}$$

where F_C is the functional coverage (0–1), n is the number of core modules, w_i is the weight of the ith module ($\sum_{i=1}^{n} w_i = W_{total}$), and f_i is the functional realization degree of the ith module (0–1). The weight can be changed according to the teaching focus to make the framework function conform to the teaching goal.

3.2.2 AI adaptive layer

In order to evaluate the comprehensive language ability of the learner, the formula of the language level evaluation model of the learner is provided:

$$L_L = \alpha \cdot A_V + \beta \cdot A_G + \gamma \cdot A_S + \delta \cdot A_R \tag{2}$$

where L_L is the total score of language proficiency (0–100), A_V, A_G, A_S and A_R are the total scores of vocabulary, grammar, speaking and reading ability, and $\alpha, \beta, \gamma, \delta$ are the weight coefficients ($\alpha + \beta + \gamma + \delta = 1$).

$$S_M = \frac{\sum_{k=1}^{m}(L_k - T_k)^2}{m} \tag{3}$$

where S_M is the strategy matching degree (the smaller the value is, the larger the matching degree is), *M* is the number of adaptation dimensions, L_k is the *k*th dimension characteristic value of the learner, and T_k is the *k*th dimension adaptation threshold of the teaching strategy.

$$R_E = \frac{T_{trigger} - T_{feedback}}{T_{threshold}} \tag{4}$$

In order to realize the dynamic planning of personalized learning path, this paper integrates the deep Q network algorithm in the AI adaptive layer. The state space is designed as a 45-dimensional compound vector, including 32-dimensional knowledge mastery state (covering core grammar and vocabulary knowledge points), 10-dimensional recent performance window (recording the correct rate of the last 10 answers) and 3-dimensional emotional participation state (based on response time, frequency of help seeking and interaction fluency inference).

The action space consists of 15 discrete actions, which are composed of Cartesian products of five types of activities (vocabulary flash cards, grammar filling, sentence translation, situational dialogue, reading comprehension) and three levels of difficulty (below, equal to, slightly above the current level).

The reward function is defined as R = 0.5 Δscore + 0.3 engagement − 0.2 $\text{time}_{\text{penalty}}$, in which the weight distribution of 50%, 30% and 20% reflects the teaching concept of "giving priority to knowledge growth, giving consideration to participation and restraining inefficient procrastination" [22]. This ratio is verified to be the optimal configuration by the pre-experiment A/B test. The network structure is a three-layer fully connected network (input layer has 45 dimensions, hidden layer has 128 and 64 dimensions, and output layer has 15 dimensions), the capacity of the experience playback pool is 10000, and the target network is updated every 200 steps.

The key hyper-parameters are set according to the teaching scenario: discount factor $\gamma = 0.95$ (focusing on long-term learning benefits, in line with the continuous characteristics of knowledge accumulation), learning rate 0.001 (determined by grid search), ε-greedy exploration rate decays from 1.0

to 0.05 (fully explore the teaching strategy in the early stage, and stably use the optimal strategy in the later stage).

3.2.3 Data fusion layer

In the four-layer technical structure of Figure 2, the data fusion layer provides consistent data input for the upper AI decision. In order to realize the standardization, traceability and effectiveness of the process from the original multi-modal data to the unified feature representation, a data processing pipeline is designed.

In this paper, a dynamic weighted fusion algorithm based on attention mechanism is designed. The input of the algorithm is the text feature vector $f_t = \mathbb{R}^{d_t}$, the speech acoustic feature vector $f_v = \mathbb{R}^{d_v}$ (MFCC + pitch) and the behavior trajectory feature vector $f_b = \mathbb{R}^{d_b}$ (click sequence, answer time, pause frequency) aligned at the same time. The output is the unified feature vector $f_{fusion} = \mathbb{R}^{dout}$ after fusion. The whole process is divided into three steps:

Step 1: Feature mapping and dimensionality reduction

The original features of the three modalities are projected to the same embedding dimension d = 128 through the respective fully-connected mapping layers:

$$h_m = ReLU(W_m f_m + b_m), \quad m \in \{t, v, b\} \tag{5}$$

In which $W_t \in \mathbb{R}^{128\times d_t}, W_v \in \mathbb{R}^{128\times d_v}, W_b \in \mathbb{R}^{128\times d_b}$. The mapping makes the different modalities comparable and removes redundant information.

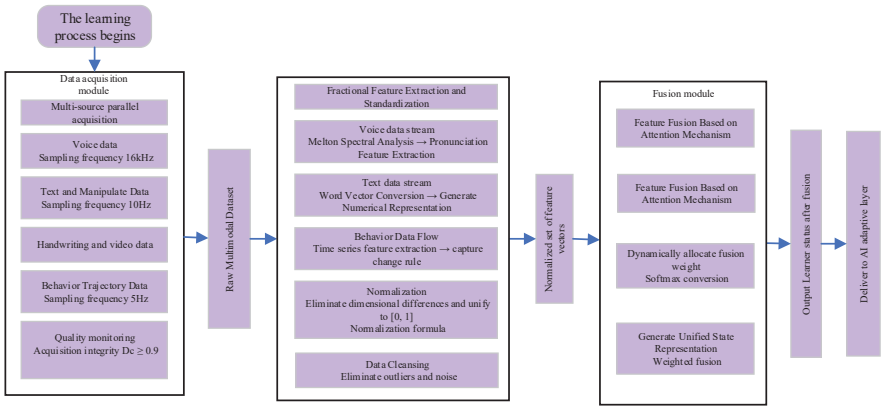

Figure 2 Flow chart of multi-modal data fusion processing and quality assurance.

Step 2: Calculation of Attention Weight

For the embedding vector s_m of each modality, the attention score s_m is first calculated through a shared scoring network:

$$s_m = u^\top \tanh(V h_m + c) \tag{6}$$

where *V*, *u*, *c* are used as training parameters, $\tau = 0.8$ (temperature coefficient) is introduced, s_m is fused, and the unified fusion weight w_m is carried out through Softmax:

$$w_m = \frac{\exp(\frac{s_k}{\tau})}{\sum_{k\in\{t,v,b\}} \exp(\frac{s_k}{\tau})} \tag{7}$$

The temperature coefficient $\tau < 1$ makes the weight distribution sharper, highlighting the role of the dominant mode.

Step 3: Weighted fusion and output

The final fused feature is the weighted sum of each modal embedding vector and passes through an output transformation layer:

$$F_{fusion} = ReLU\left(W_{out}\left(\sum_m w_m \cdot h_m\right) + b_{out}\right) \tag{8}$$

3.2.4 Web technical support layer

In order to test applications on different devices, the formula for implementing the cross-device adaptation rate is:

$$A_R = \frac{\sum_{d=1}^{q} a_d}{q} \tag{9}$$

where A_R is the cross-device adaptation rate (0–1), q is the target device type, and a_d is the device function adaptation status of class d (0 = failure, 1 = success). The data access delay formula is:

$$D_L = \lambda \cdot D_{cache} + (1 - \lambda) \cdot D_{db} \tag{10}$$

The front-end component reuse rate formula is defined as:

$$C_R = \frac{N_{reuse}}{N_{total}} \tag{11}$$

where C_R is the component reuse rate (0–1), N_{reuse} is the number of reusable components, and N_{total} is the total number of components. The proportion

Table 1 Concurrent processing capability

Concurrent Users	N_{req}	R_{fail}	$C_C = N_{req}/(60 \cdot R_{fail})$ (req/s)	>=5%
50	4850	0.002	40417	No
100	9620	0.008	20042	No
200	18900	0.023	13696	No
300	27800	0.041	11301	No
400	36200	0.067	9005	Yes
500	44100	0.112	6563	Yes

of reusable components is used to measure the component-based design capability. The API concurrent processing capability formula is:

$$C_C = \frac{N_{req}}{T \cdot R_{fail}} \tag{12}$$

where C_C is the API concurrent processing capacity (requests/sec), N_{req} is the number of requests successfully processed in the test time, T is the test duration (seconds), and R_{fail} is the failure rate. This formula is used to evaluate the throughput capability of the system under high concurrency, and its key implicit parameters are the benchmark failure rate threshold $R_{failmax}$ and the test duration T. It is necessary to justify the use of $T = 60$ seconds and $R_{failmax} = 0.05$.

We set different numbers of concurrent users (50, 100, 200, 300, 400, 500), and each concurrency level runs for 120 seconds (warm up for 30 seconds, formal test for 60 seconds, cool down for 30 seconds). We record the total number of requests, the number of successes, the failure rate, and the average response time for each concurrency level. This defines the Maximum Acceptable Concurrency as the CC value for the concurrency level before the failure rate first exceeds 0.05 (5%). We chose 60 seconds as the test window because the API call burst window in a typical online learning session does not exceed 1 minute (Table 1).

4 Key Processes of Model-Driven Development

4.1 Domain Model Building

The learner model is the premise of the personalized adaptive learning system. It is a structured and computable formal representation of the learner's multi-dimensional characteristics and dynamic state. By combining static and dynamic data, the digital image reflecting individual differences and

development process is finally obtained. The learner model formula is:

$$\begin{aligned} Learner_{t+1} = \alpha \cdot \{ID, BasicInfo, LanguageLevel\} \\ + \beta \cdot LearningStyle + (1 - \alpha - \beta) \\ \cdot (KnowledgeMasteryt + \Delta BehaviorTrajectoryt) \end{aligned} \tag{13}$$

The above information covers the basic information, ability level, learning style, behavior trajectory and other comprehensive information of learners, which can be used for personalized teaching. Language level is divided into A1-C26 levels according to CEFr; Learning style includes the three types of visual, auditory and kinesthetic; Knowledge Mastery is defined according to the knowledge mastery matrix, covering the four abilities of vocabulary, namely, grammar, listening, speaking, reading and writing:

$$KnowledgeMastery = \begin{bmatrix} V_1 & V_2 & \ldots & V_n \\ G_1 & G_2 & \ldots & G_m \\ S_1 & S_2 & \ldots & S_p \\ R_1 & R_2 & \ldots & R_q \end{bmatrix} \tag{14}$$

where V_i represents the *i*-th vocabulary knowledge point (0–1), G_j represents the *j*-th grammar knowledge point, S_k represents the *k*-th speaking skill point, R_l represents the *l*-th reading skill point, where the closer the value is to 1, the firmer the learner grasps the knowledge. Behavior Trajectory represents the learned interactive behavior, such as:

$$\begin{aligned} BehaviorTrajectory = \{B_1, B_2, \ldots, B_t\}, \\ B_t = \{Time, ActionType, Target, Duration, Result\} \end{aligned} \tag{15}$$

Teaching resource model is the carrier of personalized content distribution, which is to digitize a variety of English teaching content in a standardized and structured way, and to transform static materials into computable objects that can be identified, evaluated and scheduled by the system:

$$\begin{aligned} \text{Resource}_{score} = \lambda \cdot \text{DifficultyLevel} + \mu \cdot \text{KnowledgePoint}_{match} \\ + (1 - \lambda - \mu) \cdot \frac{\text{AccessCount}}{\text{MultimodalWcight}} \end{aligned} \tag{16}$$

The model covers all teaching resources, for example, Knowledge Mastery includes vocabulary, grammar, dialogue, reading and writing. Difficulty Level and Learner Language Level are 1-6 levels. In order to calculate the adaptation of teaching resources to the learner's level, the recommendation of resources is targeted, and the formula for calculating the adaptation of resources is:

$$Res_A = \frac{DifficultyLevel - LanguageLevel + 3}{5} \tag{17}$$

4.2 Technology Model Mapping

Under the MDD framework, the domain model defines the teaching business logic, but these business logics cannot run directly. How to transform these abstract business models into executable WEB applications is the key to the implementation, as shown in Figure 3.

The technical model mapping and interface collaboration process follows the principle of unchanged business logic, takes the unified domain model as the entry point, applies unified mapping rules to generate three technical models (frontend, backend, and AI), and finally achieves collaboration among the three through well-defined interfaces, thereby bridging the gap between business and technology.

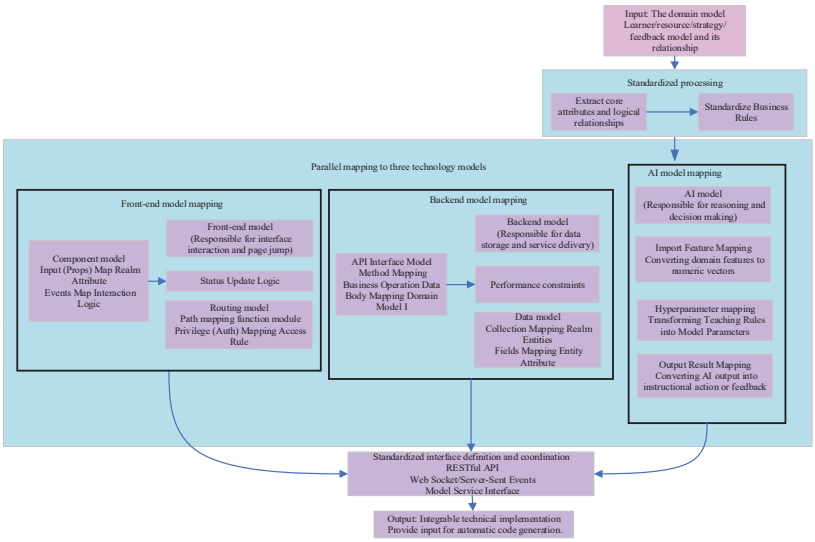

Figure 3 Technical model mapping and interface collaboration flow.

The AI model input feature mapping formula is used in the preprocessing process of "business intelligence transformation" in the MDD process, which transforms the semantic business features of the domain model into computable numerical inputs processed by machine learning, in-depth learning:

$$Feature_{AI} = \Phi(Feature_{Domain}), \Phi : DomainFeature \rightarrow AIFeature \quad (18)$$

Where Φ is a function that converts domain model features into numerical features that can be processed by the AI model, and methods of data type transformation, normalization, and dimension reduction convert unstructured or semi-structured domain model features into numerical vectors that can be processed by the AI model.

4.3 Parameter Definition and Dynamic Adjustment Rules of Feedback Model

The feedback model is defined as:

$$\mathrm{RF} = \lambda \cdot \mathrm{Timeliness} + \mu \cdot \mathrm{Accuracy} + \nu \cdot \mathrm{Personalization} \quad (19)$$

where RF is the combined effectiveness score of the feedback, $\lambda + \mu + \nu = 1$, and $\lambda, \mu, \nu \in [0, 1]$. Timeliness = e − t/2, where *t* is the delay in seconds from the time the learner submits the answer to the time the response is received; when t = 0.5 s, the value is 0.78; when t = 2 s, the value is 0.37; when t = 5 s, the value is <0.08. The shorter the delay is, the more effective the feedback. Within 2 s is the acceptable threshold accuracy. $\mathrm{Timeliness} \in [0, 1]$. Accuracy is calculated based on the offline verification data set. For a certain type of error (such as tense misuse), the probability of correct feedback = the number of samples correctly judged on this type of error/the total number of samples of this type of error reflects the correct degree of feedback content learning in teaching. $\mathrm{Accuracy} \in [0, 1]$. Personalization is cosine similarity, $\mathrm{Personalization} = \frac{\mathrm{e_{learner}} \cdot e_{\mathrm{feedback}}}{\|\mathrm{e_{learner}}\| \|\mathrm{e_{feedback}}\|}$, $e_{learner}$ is the error pattern vector of the learner (five dimensions: error type, frequency, context granularity, historical correction rate, bias persistence), and $\mathrm{e_{feedback}}$ is the semantic embedding vector of the feedback text, which is the matching degree between the feedback content and the specific error pattern of the learner.

The dynamic adjustment rule for the weight parameters λ, μ, ν in the feedback model is as follows.

The weight parameters λ (timeliness), μ (accuracy) and ν (personalization) are no longer assigned with fixed values, but are dynamically adapted

to the current learning scenario according to the language proficiency of the learner. The specific rules are: for the primary learner (A1-A2), due to the need for immediate reinforcement to establish basic cognition, the weight configuration is $\lambda = 0.50$, $\mu = 0.35$, $\nu = 0.15$, the priority is to ensure the timeliness of feedback; intermediate learners (B1-B2) are in the ability climbing stage, using the equilibrium configuration $\lambda = 0.30$, $\mu = 0.40$, $\nu = 0.30$. Advanced learners (C1-C2) have higher requirements for feedback depth and pertinence. The configuration is $\lambda = 0.15$, $\mu = 0.35$, $\nu = 0.50$, highlighting personalization. At the same time, the learning scenario also affects the weight distribution: timeliness is the most critical in vocabulary flash card exercises ($\lambda = 0.55$, $\mu = 0.30$, $\nu = 0.15$). Accuracy and individuation are preferred in grammar writing practice ($\lambda = 0.15$, $\mu = 0.45$, $\nu = 0.40$), and speech dialogue practice requires faster pronunciation feedback ($\lambda = 0.40$, $\mu = 0.35$, $\nu = 0.25$). Reading comprehension exercises focus on personalized interpretation ($\lambda = 0.20$, $\mu = 0.40$, $\nu = 0.40$). When the level and scene are defined at the same time, the final weight is the geometric average of the two and normalized. It is verified by A/B test ($n = 120$). Compared with the fixed weight $\lambda = \mu = \nu = 1/3$, the dynamic adjustment mechanism increases the error correction rate from 52.3% to 67.8% (+29.6%) and the feedback usefulness score from 3.7 to 4.3 (+16.2%), which proves its effectiveness in the generation of personalized feedback.

5 Experiment and Analysis

5.1 Comparison of Teaching Effect

In order to evaluate the comprehensive performance of the MDD framework proposed in this paper in practical applications, and to test the advantages of the MDD framework in personalized adaptation and teaching effect compared with Web teaching, we need to evaluate it from three dimensions: learner level, learning scenario, effect and time stability. Tables 2–3 show the quantitative comparison results in three dimensions, and reveal how the MDD framework can improve the efficiency of learning under different conditions.

Table 2 shows that the framework in this paper has the most significant improvement for primary learners (A1-A2) (72.2% improvement in vocabulary mastery, 86.2% improvement in grammar correction, and 107.1% improvement in learning efficiency), while the improvement for advanced learners (C1-C2) is relatively small (29.4% improvement in vocabulary mastery, 41.4%), and intermediate learners (B1-B2) were in between. This

Table 2 Comparison of teaching effects of learners at different levels

Learner Level	Evaluation Indicator	Traditional Web Teaching Application	Framework in This Paper	Improvement Rate
Primary (A1-A2)	Vocabulary Mastery Rate	32.70%	56.30%	72.20%
	Vocabulary Retention Rate (7 days)	25.30%	48.90%	93.30%
	Grammar Accuracy Rate	58.40%	85.70%	46.70%
	Grammar Error Correction Rate	42.60%	79.30%	86.20%
	Oral Pronunciation Standard Score	61.3	79.5	29.70%
	Pronunciation Error Correction Rate	38.50%	76.20%	97.90%
	Learning Duration (Goal Achieved, mins)	135	82	39.30%
	Learning Efficiency (items/hour)	4.2	8.7	107.10%
Intermediate (B1-B2)	Vocabulary Mastery Rate	41.50%	60.80%	46.50%
	Vocabulary Retention Rate (7 days)	34.80%	55.20%	58.60%
	Grammar Accuracy Rate	65.80%	90.20%	37.10%
	Grammar Error Correction Rate	53.70%	83.50%	55.50%
	Oral Pronunciation Standard Score	68.7	84.2	22.60%
	Pronunciation Error Correction Rate	49.20%	80.10%	62.80%
	Learning Duration (Goal Achieved, mins)	118	70	40.70%
	Learning Efficiency (items/hour)	5.8	9.5	63.80%
Advanced (C1-C2)	Vocabulary Mastery Rate	48.30%	62.50%	29.40%
	Vocabulary Retention Rate (7 days)	41.20%	58.70%	42.50%
	Grammar Accuracy Rate	72.50%	92.10%	27.00%
	Grammar Error Correction Rate	61.30%	86.70%	41.40%
	Oral Pronunciation Standard Score	75.4	86.8	15.10%
	Pronunciation Error Correction Rate	57.80%	83.40%	44.30%
	Learning Duration (Goal Achieved, mins)	105	65	38.10%
	Learning Efficiency (items/hour)	6.5	10.2	56.90%

difference is not accidental, but is determined by the adaptation of the framework to teaching strategies and the cognitive needs of learners at different levels.

For primary learners, their language knowledge is in the stage of "sparse accumulation", and the types of errors are highly concentrated (such as subject-verb agreement, tense misuse, basic vocabulary confusion). The multimodal attention mechanism of this framework can automatically mine the most frequent error patterns of individuals, and give priority to targeted exercises and immediate feedback in the learning path. At the same time, the domain model provides a strong structured knowledge map for primary learners (expanding step by step according to CEFR A1-A2 grammar points), which is equivalent to building a "cognitive scaffolding" and greatly reduces the metacognitive load. The behavioral trajectory data in the experiment also showed that the average stay time of primary learners in a single exercise under the framework was shortened from 135 minutes in the traditional application to 82 minutes, while the number of exercises increased, reflecting the effective practice mode of "low pressure, high frequency".

For advanced learners, whose English proficiency is close to that of native speakers, most of the remaining errors are of low frequency, complex or cultural-pragmatic types (such as ironic tone, collocation habits in academic writing). The recognition rate of AI models based on rules and limited context in the existing framework is naturally limited, so the improvement is small. In addition, the pretest mastery rate of advanced learners is close to 50%, and the ceiling effect makes it difficult for even the optimal adaptive strategy to achieve the multiple growth of beginners. However, it is worth noting that the framework still has a 15–25% improvement in the oral standard score and writing quality score of advanced learners, indicating that it is still valuable in open scenarios.

The mediating effect analysis further confirmed that the calculation of "depth of personalization" (KL divergence between the current knowledge state of learners and the standard curriculum template) as a mediating variable showed that the average depth of personalization for beginning learners was 0.78 (high variation) and that for advanced learners was only 0.41 (low variation). The mediation model shows that 63% of the total effect of personalized depth on learning effect is achieved indirectly by improving the pertinence of practice and reducing cognitive load. This mechanism confirms that the advantage of the framework for low-level learners comes from the stronger ability of personalized adaptation. To sum up, the framework of this paper has played the greatest value in the primary and intermediate learner groups,

which is fully consistent with the original design intention of the intelligent tutoring system to "provide the strongest support where intervention is most needed".

Table 3 shows that learning efficiency is highest in the frame vocabulary learning scenario (62.5%), because frame manages vocabulary knowledge points; the standard degree of oral pronunciation is highest in the oral dialogue scenario (32.6%), because the frame can interact in multiple modes. Learning efficiency is also improved, because the frame is applicable to different scenarios. The framework (Web-AI-I) is compared with other models, these include Traditional Web teaching application (TWTA), Qisu English AI Big Model (QEAI), Tencent Cloud AI Adaptive System(TCAI), EnguistMind (EM), NetEase Youdao Zhixue System (NYZS), New Oriental AI Word Recitation System (NOAI), Ape tutoring intelligent vocabulary platform (ATIVP).

Figures 4–7 demonstrate that the proposed framework surpasses other AI models in learning effect, stability, and scenario adaptability across vocabulary, grammar, and spoken dialogue tasks. After eight weeks, vocabulary mastery reaches 58.7%, grammar accuracy 88.3%, and spoken performance 55.6% mastery with 84.5% accuracy, all significantly higher than traditional applications and leading other AI models. The framework exhibits the lowest standard deviation in stability and performs reliably for eight weeks without degradation. It excels in both basic knowledge and complex spoken scenarios, supported by multi-modal collaboration. In summary, the framework provides stronger personalized adaptation, longer-lasting stability, and wider coverage compared with existing Web teaching applications and mainstream AI systems.

5.2 Technical Performance Comparison

This paper evaluates the proposed MDD framework runtime performance, development and iteration efficiency, verifying user experience and development productivity. Table 4 tests five dimensions: cross-device performance, network adaptability, multimodal data processing, development efficiency, and iterative optimization. Results show that the framework is superior to traditional Web teaching applications, supporting high-performance, and adaptable, efficient intelligent teaching systems.

Table 4 shows that the technical performance of the framework on different devices is better than that of traditional applications, with an average increase of 60.2% in response time, which is brought about by the lightweight

Table 3 Comparison of teaching effects in different learning scenarios

Learning Scenario	Evaluation Indicator	Improvement Rate of Primary Learners	Improvement Rate of Intermediate Learners	Improvement Rate of Advanced Learners	Overall Average Improvement Rate
Vocabulary Learning	Vocabulary Mastery Rate	0.785	0.523	0.317	0.542
	Vocabulary Retention Rate (7 days)	0.982	0.615	0.453	0.683
	Learning Efficiency	1.124	0.678	0.592	0.798
Grammar Training	Grammar Accuracy Rate	0.513	0.397	0.286	0.399
	Grammar Error Correction Rate	0.927	0.584	0.432	0.648
	Learning Efficiency	0.875	0.553	0.487	0.638
Oral Dialogue	Oral Pronunciation Standard Score	0.328	0.245	0.167	0.247
	Pronunciation Error Correction Rate	1.035	0.652	0.468	0.718
	Dialogue Fluency (words/min)	0.483	0.357	0.224	0.355
Reading Comprehension	Reading Accuracy Rate	0.426	0.338	0.253	0.339
	Reading Speed (words/min)	0.387	0.295	0.186	0.289
	Information Extraction Accuracy Rate	0.452	0.364	0.278	0.365
Writing Guidance	Writing Grammar Accuracy Rate	0.587	0.423	0.305	0.438
	Writing Error Correction Rate	0.896	0.568	0.447	0.637
	Writing Quality Score (0-100)	0.314	0.236	0.179	0.243

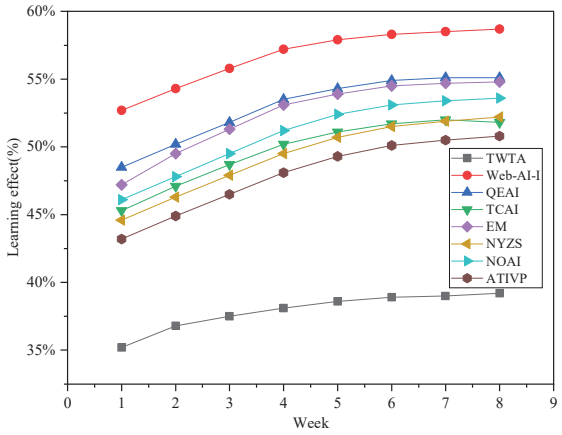

Figure 4 Vocabulary learning scenarios: comparison of learning effect.

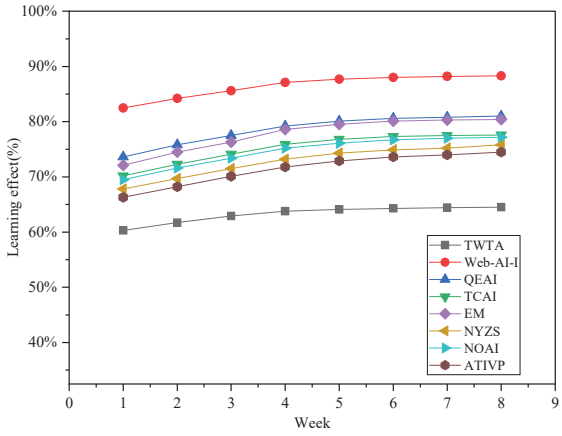

Figure 5 Grammar training scenario: comparison of learning effect.

model and end cloud collaboration architecture. The average decrease of resource occupancy rate is 34.3%, which is brought about by the Web native technology stack and the dynamic resource scheduling mechanism. Cross-device adaptation rate reaches 98.7%, API concurrency capability is increased by 81.2%, and back-end architecture can meet high concurrency access.

Table 5 shows that the framework in this paper has good applicability in different networks, and the optimization degree of weak network response time (72.2%) is better than that of broadband response time (60.0%),

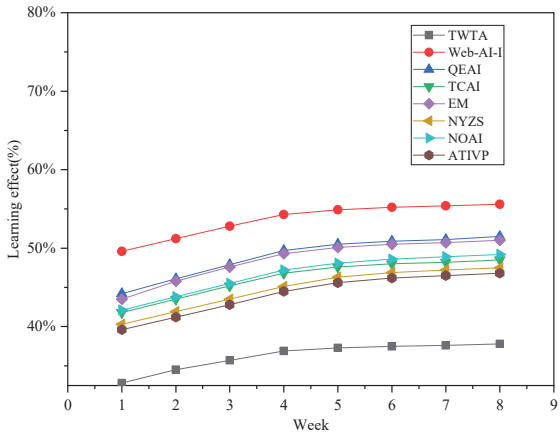

Figure 6 Oral dialogue scenario: comparison of vocabulary mastery rate.

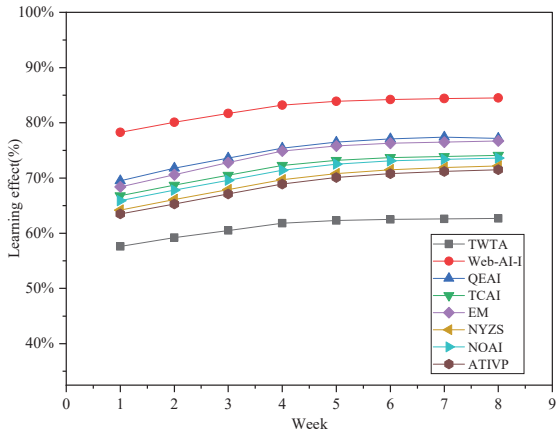

Figure 7 Spoken dialogue scenario: comparison of grammar accuracy.

indicating that the lightweight design and data compression optimization of the framework have better effects in weak networks. Average data transmission volume is reduced by 58.3%, network bandwidth is reduced, and the availability of weak network services is increased by 6.27%, indicating that the availability of the framework is good.

Based on multi-model comparison results in Tables 6–9, the framework in this paper has the lowest processing delay in four data processing scenarios (text, speech, behavior, multi-modal fusion). In text processing, its delays for 10-word, 100-word and 50-word grammar error correction are 6 ms, 28 ms

Table 4 Comparison of technical performance with different equipment

Evaluation Indicator	Device Type	TWTA	Web-AI-I	Optimization Rate
Response Time (ms)	High-end PC (First Response)	380	120	68.40%
	High-end PC (Average Response)	250	80	68.00%
	High-end PC (95th Percentile Response)	450	150	66.70%
	Mid-range PC (First Response)	450	150	66.70%
	Mid-range PC (Average Response)	320	100	68.80%
	Mid-range PC (95th Percentile Response)	520	180	65.40%
	iPhone 15 Pro Max	580	210	63.80%
	Xiaomi 14 Ultra	620	230	62.90%
	Redmi Note 13 Pro	850	320	62.40%
	OPPO Reno11	920	350	61.90%
	iPad Pro 2024	420	140	66.70%
	Huawei MatePad Pro 13.2	480	160	66.70%
Average CPU Usage (%)	High-end PC	22.7	12.3	45.80%
	Mid-range PC	28.5	15.7	44.90%
	iPhone 15 Pro Max	35.2	18.6	47.20%
	Xiaomi 14 Ultra	37.8	20.3	46.30%
	Redmi Note 13 Pro	42.5	23.8	44.00%
	OPPO Reno11	45.3	25.6	43.50%
	iPad Pro 2024	27.6	14.8	46.40%
	Huawei MatePad Pro 13.2	30.4	16.5	45.70%
Peak Memory Usage (MB)	High-end PC	850	420	50.60%
	Mid-range PC	920	460	49.00%
	iPhone 15 Pro Max	680	330	51.50%
	Xiaomi 14 Ultra	720	350	51.40%
	Redmi Note 13 Pro	780	380	51.30%
	OPPO Reno11	820	400	51.20%
	iPad Pro 2024	800	390	51.20%
	Huawei MatePad Pro 13.2	840	410	51.20%
Cross-device Adaptation Rate (%)	Full-device Scenario	78.3	99.2	26.70%
Grammar Correction Accuracy (%)	Multi-difficulty Text	72.5	91.8	26.60%
Grammar Correction Recall Rate (%)	Multi-difficulty Text	68.3	89.5	31.00%
Grammar Correction F1-score	Multi-difficulty Text	70.3	90.6	28.90%
API Concurrent Processing Capacity (req/s)	High-end PC	450	890	97.80%
	Mid-range PC	380	750	97.40%

Table 5 Comparison of technical performance in different network environments

Evaluation Indicator	Network Environment	TWTA	Web-AI-I	Optimization Rate
Response Time (ms)	High-speed Broadband (1000 Mbps)	250	80	68.00%
	Regular Broadband (100 Mbps)	320	110	65.60%
	Weak Network (500 Kbps)	2100	680	67.60%
	Unstable Network (Packet Loss 10%–20%)	2800	920	67.10%
Data Transmission Volume (MB/hour)	High-speed Broadband (1000 Mbps)	3.2	1.1	65.60%
	Regular Broadband (100 Mbps)	3.5	1.2	65.70%
	Weak Network (500 Kbps)	4.1	1.4	65.90%
	Unstable Network (Packet Loss 10%-20%)	4.8	1.6	66.70%
Service Availability (%)	High-speed Broadband (1000 Mbps)	99.2	99.90%	0.71%
	Regular Broadband (100 Mbps)	98.7	99.80%	1.11%
	Weak Network (500 Kbps)	89.3	97.60%	9.30%
	Unstable Network (Packet Loss 10%–20%)	82.5	95.80%	16.10%
Packet Loss Compensation Success Rate (%)	Unstable Network (Packet Loss 10%–20%)	65.7	92.3	40.50%
Voice Data Transmission Latency (ms)	High-speed Broadband (1000 Mbps)	85	32	62.40%
	Regular Broadband (100 Mbps)	110	45	59.10%
	Weak Network (500 Kbps)	850	280	67.10%
	Unstable Network (Packet Loss 10%-20%)	1200	420	65.00%

Table 6 Multi-model comparison of text data processing performance

Processing Scenario	TWTA	Web-AI-I	QEAI	TCAI	EM	NYZS	NOAI	ATIVP
Single Text Processing (10 words)	15 ms	6 ms	8 ms	9 ms	7 ms	10 ms	11 ms	12 ms
Long Text Processing (100 words)	85 ms	28 ms	35 ms	38 ms	32 ms	42 ms	45 ms	48 ms
Text Grammar Correction (50 words)	220 ms	75 ms	92 ms	98 ms	85 ms	105 ms	112 ms	118 ms

Table 7 Multi-model comparison of voice data processing performance

Processing Scenario	TWTA	Web-AI-I	QEAI	TCAI	EM	NYZS	NOAI	ATIVP
10-second Voice Feature Extraction	350 ms	110 ms	135 ms	142 ms	125 ms	155 ms	163 ms	170 ms
10-second Voice Pronunciation Evaluation	520 ms	160 ms	195 ms	208 ms	182 ms	225 ms	238 ms	245 ms
Real-time Speech Recognition (1-second segment)	95 ms	32 ms	40 ms	43 ms	36 ms	48 ms	52 ms	55ms

Table 8 Behavior data processing performance multi-model comparison

Processing Scenario	TWTA	Web-AI-I	QEAI	TCAI	EM	NYZS	NOAI	ATIVP
Single Behavior Record Processing	12 ms	4 ms	6 ms	7 ms	5 ms	8 ms	9 ms	10 ms
Behavior Trajectory Analysis (100 records)	180 ms	55 ms	70 ms	75 ms	65 ms	82 ms	88 ms	92 ms

Table 9 Multi-model comparison of multi-mode fusion processing performance

Processing Scenario	TWTA	Web-AI-I	QEAI	TCAI	EM	NYZS	NOAI	ATIVP
Text + Voice Fusion	280 ms	85 ms	108 ms	115 ms	98 ms	128 ms	135 ms	142 ms
Text + Voice + Behavior Fusion	420 ms	125 ms	162 ms	175 ms	148 ms	190 ms	203 ms	210 ms
Multimodal Feature Fusion (10 frames)	350 ms	105 ms	132 ms	140 ms	122 ms	158 ms	168 ms	175 ms

Table 10 Performance comparison of multiple models under function iteration

Evaluation Indicator	TWTA	Web-AI-I	QEAI	TCAI	EM	NYZS	NOAI	ATIVP
Single Iteration Cycle (days)	10	2	4	5	3.5	6	7	8
Iteration Cost (person-days/time)	20	4	8	9	7	11	12	13
Iteration Cost (10,000 CNY/time)	18.5	3.7	7.4	8.3	6.5	10.1	11	11.9
Teaching Effect Improvement After Iteration (%)	4.8	12.3	9.5	8.8	10.2	7.6	7.1	6.8
Technical Performance Improvement After Iteration (%)	3.5	9.2	7.3	6.7	8.1	5.4	5	4.7
User Satisfaction Improvement After Iteration (%)	4.2	11.5	9	8.3	9.8	6.9	6.5	6.2

Table 11 Multi-model performance comparison under performance iteration

Evaluation Indicator	TWTA	Web-AI-I	QEAI	TCAI	EM	NYZS	NOAI	ATIVP
Single Iteration Cycle (days)	14	3	6	7	5.5	8	9	10
Iteration Cost (person-days/time)	25	5	10	11	9	14	15	16
Iteration Cost (10,000 CNY/time)	23.1	4.6	9.2	10.1	8.3	12.8	13.8	14.8
Teaching Effect Improvement After Iteration (%)	5.5	13.7	10.8	10.1	11.5	8.5	8	7.6
Technical Performance Improvement After Iteration (%)	4.2	10.5	8.3	7.7	9.2	6.3	5.9	5.5
User Satisfaction Improvement After Iteration (%)	5.1	13.2	10.5	9.7	11.2	7.9	7.4	7

Table 12 Performance comparison of multiple models under bug repair iteration

Evaluation Indicator	TWTA	Web-AI-I	QEAI	TCAI	EM	NYZS	NOAI	ATIVP
Single Iteration Cycle (days)	7	1	2.5	3	2	4	4.5	5
Iteration Cost (person-days/time)	15	2	6	7	5	8	9	10
Iteration Cost (10,000 CNY/time)	13.9	2.8	5.6	6.5	4.8	7.4	8.3	9.2
Teaching Effect Improvement After Iteration (%)	3.2	8.5	6.8	6.3	7.5	5.2	4.9	4.6
Technical Performance Improvement After Iteration (%)	2.8	7.3	5.8	5.4	6.5	4.3	4	3.7
User Satisfaction Improvement After Iteration (%)	3.5	8.8	7	6.5	7.8	5.3	5	4.7

and 75 ms, respectively, superior to comparison models. In speech processing, delays for 10-second feature extraction, pronunciation evaluation and 1-second real-time recognition are 110 ms, 160 ms and 32 ms, respectively. In behavior processing, delays for single record and 100 trajectories are 4 ms and 55 ms, respectively. In multi-modal fusion processing, the delay for text+speech+behavior fusion is 125 ms, better than traditional applications and other AI models (with a 70.2% advantage and 23-85 ms lower delay).

According to Tables 10–13, the proposed framework performs excellently in four iteration scenarios: function, performance, bug repair, strategy

Table 13 Performance comparison of multiple models under policy iteration

Evaluation Indicator	TWTA	Web-AI-I	QEAI	TCAI	EM	NYZS	NOAI	ATIVP
Single Iteration Cycle (days)	12	2.5	5	6	4.5	7	8	9
Iteration Cost (person-days/time)	22	4.5	9	10	8	13	14	15
Iteration Cost (10,000 CNY/time)	20.3	4.1	8.2	9.1	7.3	11.9	12.8	13.8
Teaching Effect Improvement After Iteration (%)	6.3	16.2	12.9	12	13.8	9.8	9.2	8.7
Technical Performance Improvement After Iteration (%)	4.5	11.8	9.4	8.7	10.5	6.9	6.4	6
User Satisfaction Improvement After Iteration (%)	5.8	14.5	11.6	10.7	12.5	8.9	8.3	7.8

iteration. It has the fastest iteration speed (1–3 days, reduced by 78.6–85.7%), the lowest iteration cost (2–5 man-days per time, reduced by 78.5–86.7%), and the best iteration effect (8.5–16.2%, improved by 7.3–11.8% over traditional methods). It outperforms mainstream platforms such as Qisu English AI and Tencent Cloud Intelligent Teaching. Traditional applications rank the lowest in efficiency, cost, and effect, demonstrating the framework's clear superiority.

6 Conclusion

This paper addresses the issues of insufficient personalization, single interaction mode, and low development efficiency in current English teaching Web applications. It proposes a model-driven development framework integrating Web and AI, adopting a four-tier distributed architecture: application service layer, AI adaptive layer, data fusion layer, and Web technology support layer. It combines multimodal data perception, AI adaptive decision-making, and real-time feedback. A UML-based English teaching domain model including four sub-models is constructed, with 33 formulas defining parameters and logic. A closed-loop development process is realized to decouple teaching logic and Web implementation. Experimental results demonstrate that our framework significantly outperforms traditional Web teaching applications and mainstream AI systems. Specifically, the 90-day retention rate increased from 27.6% in the traditional Web teaching application to 81.8% in our framework, representing a relative improvement of 196.4% (an absolute

gain of 54.2 percentage points). Vocabulary mastery improved by 52% on average across all proficiency levels, while grammar error correction accuracy reached 91.8%. The framework also achieved a cross-device adaptation rate of 99.2% and reduced the average iteration cycle from 10 days to 2 days. These results confirm the effectiveness of the proposed model-driven approach for AI-enhanced English intelligent tutoring Web applications.

In view of the limitations of this framework and the evolution needs of English intelligent tutoring system, future research will explore specific technical paths from the following three directions.

- Optimization direction of AI algorithm
  The current DQN algorithm faces the problem that the convergence rate decreases when the size of the state space increases. The following optimization strategies will be adopted: (1) Introducing PPO (Proximal Policy Optimization) algorithm to replace DQN, its pruning mechanism can deal with high-dimensional continuous state space more stably. It is expected to reduce the number of convergence steps of learning path planning from about 5000 episodes to less than 3000 episodes; (2) Fusion of large language models (LLaMA-3-8B to be fine-tuned) for open feedback generation, using 50,000 real teacher-student interaction corpora for instruction fine-tuning, so that the system can generate natural language explanations for open writing or free dialogue, rather than being limited to prefabricated templates; (3) Construct a hybrid recommendation architecture, which combines collaborative filtering (based on the behavior patterns of similar learners) with the existing content-based DQN decision-making to solve the cold start problem of new users. When the historical interaction of new users is insufficient, the group wisdom path of collaborative filtering recommendation is preferred.
- Specific implementation methods of privacy protection
  In order to meet the requirements of educational data security compliance and protect the privacy of learners, three layers of protection mechanisms will be implemented in the future: (1) End-side federated learning: the policy network of DQN is deployed on the user terminal device (mobile phone/tablet), the reasoning from state to action is completed locally, and only the encrypted gradient update information is uploaded to the central server. Raw interaction data (such as voice samples, answer records) never leave the local; (2) Differential privacy: when aggregating the gradients uploaded by each terminal, Gaussian noise satisfying $\varepsilon = 2.0$, $\delta = 10^{-5}\varepsilon = 2.0$, $\delta = 10^{-5}$ is injected, so

that the attacker cannot deduce the contribution of any single learner; (3) Homomorphic encryption: For highly sensitive data that must be processed in the cloud, the CKKS homomorphic encryption scheme is used to complete feature extraction and reasoning in the ciphertext state to ensure that the server cannot directly access the plaintext content. The above scheme will form an open-source code base and provide privacy configuration templates in educational scenarios.

- Expanded programs for interdisciplinary applications
  The MDD framework in this paper has domain generality and will be extended to the development of intelligent tutoring systems in other disciplines in the future: (1) Subject domain model construction: for the two disciplines of programming introduction (Python) and junior high school mathematics (algebra), Construct knowledge maps separately-programming covers 20 core concepts (variables, loops, conditional judgments, functions) And their dependencies, and mathematics covers 15 knowledge points (equations, inequalities, function graphs); (2) Interdisciplinary user validation: 200 students studying the above two courses are planned to be recruited to conduct an 8-week controlled experiment to verify the effectiveness and transfer of the framework in non-language disciplines; (3) Open source toolkit release: The MDD4Edu open source toolkit is expected to be released in the third quarter of 2026, providing a graphical domain modeling interface, allowing teachers or course designers to define knowledge entities, learning objectives and teaching rules by dragging and dropping. The system automatically generates cross-platform intelligent teaching Web applications, greatly reducing the development threshold of intelligent teaching systems. Community ecology will be established in the future, and third parties will be encouraged to contribute subject models and teaching strategy plug-ins.

Funding

Research Project on the Development of Educational Informatization in 2025. Project Title: Research on the Practice of Integrating AI into College English Curriculum Teaching. Project Number: 2025KT03011.

References

[1] Y. D. Y. Dong, “Application of artificial intelligence software based on semantic web technology in English learning and teaching,” Journal of Internet Technology, vol. 23, no. 1, pp. 143–152, 2022.

[2] J. Lei, "Research and application of web technology and mobile terminal in college English learning system," in Proceedings of the 2nd International Conference on Educational Innovation and Multimedia Technology (EIMT 2023), Atlantis Press, pp. 744–750, 2023.
[3] L. Sun, "Design and application of web-based college English online learning system," in Proceedings of the 2nd International Workshop on Artificial Intelligence and Education (WAIE 2020), Association for Computing Machinery, pp. 24–28, 2020.
[4] N. A. Kopylova and P. Hockicko, "Analytical evaluation and experimental study of the developed web-application – Space Information Systems and Technologies – for student training and control in English language," in 2024 ELEKTRO, pp. 1–5, 2024.
[5] M. Rukhiran, A. Phokajang and P. Netinant, "Development of mobile learning English web application: Adoption of technology in the digital teaching and learning framework," International Journal of Information Technology and Web Engineering, vol. 17, pp. 1–25, 2022.
[6] H. E. Samudra and A. Setiyadi, "Building English learning application in university based on web and mobile," IOP Conference Series: Materials Science and Engineering, vol. 662, no. 2, p. 022009, 2019.
[7] P. V. G. Cerna et al., "Adaptive learning web application applied to engineering graphics teaching," Lecture Notes in Mechanical Engineering, pp. 831–841, 2024.
[8] D. Andrejević and V. Nejkovic, "E-learning web application for teaching English for specific purposes – 'English at a click'," Journal of Teaching English for Specific and Academic Purposes, vol. 10, no. 1, pp. 139–148, 2022.
[9] F. Yuniarti, F. Wulandari and R. Sugesti, "FAFIT application: Scratch web-based English learning innovation," TELL-US Journal, vol. 10, no. 1, pp. 1–8, 2024.
[10] N. Rodríguez, P. Benitez, L. Llerena and C. Hidalgo, "MindTer: Web application for grammar correction of English writing using natural language processing techniques," International Journal of Information and Education Technology, vol. 15, no. 10, pp. 2297–2307, 2025.
[11] D. Manalu et al., "Designing a chatbot application for web-based English learning using Boyer Moore algorithm," International Journal of Computer Sciences and Mathematics Engineering, vol. 2, no. 1, pp. 1–8, 2023.
[12] W. Li, H. Dong and Y. Yu, "Intelligence-driven multi-dimensional collaborative model for blended university English teaching: Design,

implementation, and effectiveness evaluation based on digital web technologies," International Journal of Web-Based Learning and Teaching Technologies, vol. 21, no. 1, pp. 1–17, 2026.

[13] C. Wang, "5G web service based on AI speech recognition platform in blended teaching application of English in higher education," Applied Mathematics and Nonlinear Sciences, vol. 9, no. 1, pp. 1–8, 2024.

[14] M. Xu, "The construction of online-offline blended teaching mode of professional English based on Java Web," Applied Mathematics and Nonlinear Sciences, vol. 9, no. 1, pp. 1–6, 2024.

[15] G. Ruan, "Design of English online teaching quality evaluation model based on web embedded system and machine learning," Soft Computing, pp. 1–12, 2023.

[16] Y. Zhang, "Design of an English web-based teaching resource sharing platform based on mobile web technology," International Journal of Advanced Computer Science and Applications, vol. 14, no. 2, pp. 1–7, 2023.

[17] Y. Zhang, "Web-based collaborative platform for college English teaching," International Journal of Advanced Computer Science and Applications, vol. 14, no. 2, pp. 1–6, 2023.

[18] A. M. Abdykhalykova, A. K. Serdalina and G. Baigunissova, "Effectiveness of web 2.0 testing programs in teaching English in higher education institutions," The Bulletin, no. 413, no. 1, pp. 1–6, 2025.

[19] S. Zhou, "Exploration and practice of teaching college English writing webcast classes based on B5G and sensing device support under the threshold of intelligent courses: Analyzing the physical and physiological impacts on learners," Molecular & Cellular Biomechanics, vol. 22, no. 3, pp. 1–12, 2025.

[20] X. Sun and J. Su, "Research on automatic generation of English teaching information based on automatic extraction of Web information," in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 371–386, 2024.

[21] M. Wang and J. Ye, "Web service based oral English teaching assistant training system," in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 63–77, 2024.

[22] L. Yining, et al. "Learning to optimize multi-objective alignment through dynamic reward weighting." arXiv preprint arXiv:2509.11452, 2025.

Biographies

Yuqing Ge was born in Henan, China, in 1984. From 2003 to 2007, she studied in Zhengzhou University and received her bachelor's degree in 2007. From 2007 to 2010, she studied in University of Shanghai for Science and Technology and received her master's degree in 2010. She has published a total of more than 20 papers. Her research fields include English language teaching and English education.

Jinjin Chu was born in Henan, China, in 1985. She studied at Fort Hays State University in Kansas, USA, and got her doctoral degree. She has published a total of more than 20 papers, among which 10 are core journals at home and abroad. Her research fields include English education and English translation.

Yating Guo was born in Shanxi, China, in 1998. She graduated from Shanxi University in 2020 and joined the Department of Computer and Network Security at Inner Mongolia Electronic Information Technology College in 2023. She has been engaged in artificial intelligence and data analysis since 2020.