Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 20 No. 21 (2026) | OPEN ACCESS

https://doi.org/10.3991/ijim.v20i21.63623

PAPER

Design of a Personalized Learning System for Higher Education Using Artificial Intelligence and Interactive Mobile Technologies

Weijing Li, **Tingting Fu** (✉)

Cangzhou Jiaotong College, Huanghua, China

futingting198403@163.com

ABSTRACT

Mobile learning in higher education is shifting from ubiquitous access toward context-sensitive personalization. However, cloud-based recommendation systems are constrained by unpredictable latency, insufficient contextual awareness, and limited capacity for continuous adaptation. To address these limitations, an on-device context-adaptive personalization loop (OCAPL) was proposed, in which personalized inference, path generation, and online adaptation were performed directly on mobile devices. The proposed framework was built on three technical pillars: on-device multimodal state inference, context-constrained path generation, and implicit-feedback-driven parameter-efficient updating. Learning behaviors, touch interactions, and device context were jointly integrated. Learning actions were subsequently generated under constraints on latency, energy consumption, memory, and path coherence. Continuous adaptation was further enabled by natural interaction signals and low-rank fine-tuning. Experimental results demonstrate that OCAPL maintained on-device inference latency below 200 ms, peak memory usage below 50 MB, and additional energy consumption below 5% per hour of learning. Compared with cloud-based recommendation, edge-server-based, and offline on-device baselines, OCAPL achieved superior learning gains, path coherence, and long-term adaptation performance. A low-latency, low-energy, and privacy-friendly unified closed-loop mechanism is thereby provided for context-sensitive personalized learning in higher education.

KEYWORDS

mobile learning, on-device intelligence, context awareness, personalized path generation, implicit feedback, parameter-efficient fine-tuning, higher education

1 INTRODUCTION

Mobile learning in higher education is gradually shifting from ubiquitous access toward context-sensitive personalization, where learning content and interaction

Li, W., Fu, T. (2026). Design of a Personalized Learning System for Higher Education Using Artificial Intelligence and Interactive Mobile Technologies. *International Journal of Interactive Mobile Technologies (iJIM)*, 20(21), pp. 120–134. https://doi.org/10.3991/ijim.v20i21.63623

Article submitted 2026-08-22. Revision uploaded 2026-10-03. Final acceptance 2026-10-09.

© 2026 by the authors of this article. Published under CC-BY.

strategies are adapted to learner characteristics, device capabilities, and contextual conditions [1]. Mobile devices are used by learners in heterogeneous contexts such as commuting, dormitories, self-study rooms, weak-network conditions, and low-battery conditions, and the learning process is characterized by fragmentation, high-frequency interaction, and rapid contextual change [2]. Problems of anytime access have mainly been addressed by conventional mobile learning systems [3], whereas real-time personalization under resource constraints and rapid contextual change has not been resolved. Higher education learners are characterized by heterogeneous knowledge backgrounds, strong autonomy in learning goals, and complex interaction behaviors; therefore, personalization cannot be based solely on response outcomes [4], and touch interaction, device state, and the network environment must also be sensed [5]. If the primary locus of personalized inference is migrated from the cloud to mobile terminals, context-adaptive learning can be achieved under low-latency, low-energy, and privacy-friendly conditions [6]. Thereby, mobile learning is advanced from ubiquitous access toward context sensitivity, while recent smart educational platforms have also demonstrated the value of integrating adaptive digital technologies with deployable learning environments [7].

However, a unified closed loop has not yet been formed by existing research. In the cloud-to-mobile master–slave architecture, recommendations are generated in the cloud and displayed on mobile terminals, so recommendation latency is affected by network fluctuations; under weak-network conditions, delays of several seconds can be reached [8], and the continuity of mobile learning interaction is disrupted. Related edge-computing studies have shown that moving time-sensitive processing closer to end devices can reduce transmission latency and data-transfer overhead compared with cloud-centered processing [9]. Existing adaptive learning is largely dependent on knowledge mastery or response records [10], and mobile interaction features such as touch trajectories, swipe gestures, and dwell-and-review behaviors are rarely integrated [11]; consequently, learning-state inference is one-sided, and fragmented learning cannot be easily distinguished from focused learning. Continual personalization is usually achieved through cloud retraining or federated synchronization [12], but communication overhead, energy consumption, and synchronization latency cannot be easily borne by personal mobile devices [13]. If only offline model deployment is adopted, rapid deviation from the learner's true state is induced [14]. Thus, a research gap is formed: whether learning states can be inferred in real time on mobile terminals under resource constraints, context-adapted paths can be generated immediately, and continual evolution can be achieved on-device [15]. Latency, memory, energy consumption, communication, and deployment feasibility are emphasized by mobile technology journals [16], but a unified closed loop has not yet been formed by existing work [17].

An on-device context-adaptive personalization loop (OCAPL) is proposed, in which personalized inference, path decision-making, and online adaptation are offloaded to each learner's personal mobile device, whereas global knowledge graph maintenance and teacher model updating are undertaken only by the cloud. Three technical pillars are established. In on-device multimodal state inference, learning behavior, touch interaction, and device context are fused, and real-time state estimation is achieved through cross-modal attention and distillation-based quantization. In context-constrained path generation, resource type,

difficulty adjustment, and interaction modality are output by lightweight reinforcement learning under constraints on latency, energy consumption, memory, and path coherence. In parameter-efficient updating from implicit feedback, continual on-device adaptation is achieved through natural interaction signals, such as review, skip, and dwell, together with low-rank fine-tuning. A deployable system is constructed, and multidimensional experiments are conducted to validate on-device inference performance, state inference accuracy, context-constrained path generation effectiveness, and long-term online adaptation capability. The remainder is organized as follows: Problem modeling and the OCAPL methodology are presented in Section 2. Experiments are designed and results are analyzed in Section 3. Theoretical implications, practical implications, and limitations are discussed in Section 4.

2 ON-DEVICE CONTEXT-ADAPTIVE PERSONALIZATION LOOP

2.1 Closed-loop logic and unified optimization objective

The system architecture and data flow of OCAPL are illustrated in Figure 1. As shown in the figure, observation, state, decision, feedback, and updating are offloaded to the mobile terminal so that personalization capability is closed on-device. On-device priority is given to ensuring that real-time decision-making remains independent of the network, whereas only teacher model training, global knowledge graph maintenance, and periodic parameter aggregation are undertaken by the cloud. Under context-as-state, network latency, bandwidth, battery level, usage period, motion state, and touch mode are incorporated into decision conditions. Under interaction-as-evidence, swipe speed, direction switching, click interval, review, skip, and dwell are treated as higher-frequency process signals than response outcomes. Under implicit-feedback-as-supervision, rewards are constructed from natural interactions. Under constraints-as-optimization, latency, memory, energy consumption, and path coherence are incorporated into the loss and decision constraints. At the time t, the multimodal observation is denoted as $o_t = [x_t^b; x_t^g; x_t^c]$, where x_t^b denotes learning behaviors such as answering, dwelling, and reviewing; x_t^g denotes touch interactions such as swiping, clicking, and direction switching; and x_t^c denotes device context such as network, battery, time period, and motion state. The closed loop is jointly defined by state inference, path decision-making, and parameter updating:

$$\hat{s}_t = f_{\theta_t}(o_t), \quad a_t = \pi_{\phi_t}(\hat{s}_t, q_t), \quad \theta_{t+1}, \phi_{t+1} = Update(\theta_t, \phi_t, r_t, o_t) \tag{1}$$

where, $\hat{s}_t$ is the learning state estimate, q_t is the device context embedding, a_t is the learning action, r_t is the total reward, and θ_t and ϕ_t are the personalized parameters for state inference and path decision-making, respectively. From an engineering perspective, the loop is driven by event streams: learning behaviors are collected at second-level granularity, touch events are aggregated into fixed time windows, and context data are updated when state changes occur. A short-term replay buffer is maintained on-device. After each interaction, the state estimate is updated before an action is generated. Online updating is triggered at fixed intervals or during charging-idle states so that energy consumption and storage pressure are controlled.

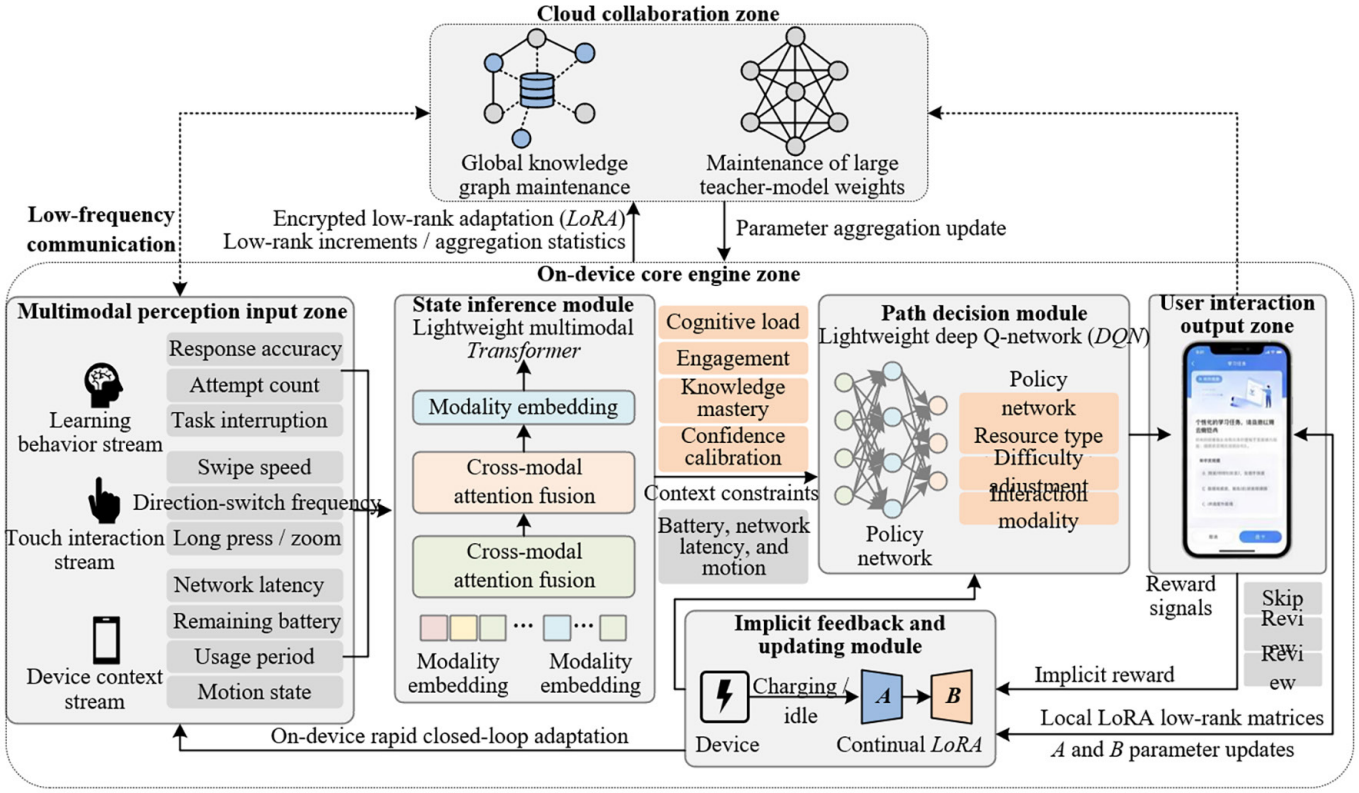


Fig. 1. System architecture and data flow of the on-device context-adaptive personalization loop (OCAPL)

Closed-loop optimization is required to satisfy mobile resource constraints, and its unified objective is formulated as:

$$\begin{aligned} \max_{\pi} \quad & E\left[\sum_{t=0}^{T} \gamma^t r_t\right] \\ \text{s.t.} \quad & T_{infer} \le T_{max}, M_{model} \le M_{max}, E_{bat} \le E_{max}, B_{comm} \le B_{max} \end{aligned} \tag{2}$$

where, π is the learning policy, γ is the discount factor, T_{infer} is the on-device inference latency, M_{model} is the model memory, E_{bat} is the energy consumption per unit time, and B_{comm} is the communication volume. During training, the above constraints are incorporated into the loss function so that on-device latency, memory, energy consumption, and communication overhead are directly controlled while long-term learning returns are optimized, rather than recommendation accuracy being pursued first and model compression being performed afterward. State inference, path generation, and online updating are unified into the same optimization problem by this objective, thereby forming the mathematical foundation of the on-device closed loop.

2.2 On-device multimodal learning state inference

Observations are divided into three streams. The learning behavior stream includes response accuracy, attempt count, hint use, review count, dwell duration, task completion, and task interruption. The touch interaction stream includes swipe speed, acceleration, direction-switch frequency, click interval, long press, zoom, and false-touch rate. The device context stream includes network type, round-trip latency, bandwidth, remaining battery, charging state, usage period, motion state, and notification interference. Touch features are considered to possess explicit semantics.

High direction-switch frequency and high swipe acceleration are often associated with high cognitive load or attentional distraction; low click frequency accompanied by prolonged dwell without review may indicate attentional disengagement; frequent review and repeated practice often indicate relatively high engagement but insufficient mastery; and rapid guessing and consecutive skipping often indicate frustration or content mismatch. Device context is likewise incorporated into state inference: under weak-network and low-battery conditions, lightweight text or offline practice tends to be favored; under fragmented periods, short tasks and immediate feedback tend to be favored; under focused periods, deep tasks and long paths tend to be favored.

On-device, a lightweight multimodal Transformer is adopted, by which the three modalities are mapped into unified embeddings and dynamically fused. Modality embeddings, cross-modal attention, state outputs, and confidence can be uniformly expressed as:

$$h_t^m = E_m x_t^m + p_t^m,\ \alpha_t^m = softmax\left(\frac{(W_q \bar{h}_t)^\top (W_k h_t^m)}{\sqrt{d}}\right),\ z_t = \sum_{m\in\{b,g,c\}} \alpha_t^m W_v h_t^m \tag{3}$$

$$\begin{aligned} \widehat{cog}_t &= \sigma(W_{cog} z_t + b_{cog}),\ \widehat{eng}_t = \sigma(W_{eng} z_t + b_{eng}) \\ \hat{\boldsymbol{k}}_t &= softmax(W_k z_t + b_k),\ \boldsymbol{\rho}_t = \sigma(W_\rho z_t + b_\rho) \end{aligned} \tag{4}$$

where, $m\hat{I}\{b,g,c\}$ is the modality index, E_m is the modality embedding matrix, p_t^m is the temporal positional encoding, d is the embedding dimension, $\bar{h}_t$ is the global context, α_t^m is the modality attention weight, and z_t is the fused representation. The state estimate is denoted as $\hat{s}_t = \left[\widehat{cog}_t, \widehat{eng}_t, \hat{\boldsymbol{k}}_t\right]$, where normalized cognitive load and engagement estimates are denoted by $\widehat{cog}_t$ and $\widehat{eng}_t$, class probabilities of the knowledge mastery state are denoted by $\hat{\boldsymbol{k}}_t$, and confidence is denoted by ρ_t. When $\rho_t < \rho_0$, a conservative strategy is adopted by the system, difficulty is reduced, or review content is pushed. The cloud teacher model is trained on large-scale multimodal logs, and the on-device student model is obtained through knowledge distillation, structured pruning, and mixed-precision quantization. Random modality dropout and masked reconstruction are incorporated during training so that inference can still be performed under weak-network, low-battery, or missing-touch-data conditions. The state inference loss is composed of supervised state loss, distillation loss, masked reconstruction loss, and confidence calibration loss, where the mean absolute error is used for continuous states and cross-entropy is used for knowledge mastery classes. Touch interaction and device context are used as inputs for state inference by this module so that a state estimate usable for decision-making is obtained on-device within 200-ms-level latency.

2.3 Context-aware dynamic path generation

In the path generator, the state estimate, context embedding, and path memory are concatenated into a decision state, and a triplet action space is defined as:

$$\tilde{s}_t = [\hat{s}_t; q_t; m_t],\ a_t = (u_t, d_t, f_t)\hat{I}A \tag{5}$$

where, u_t is the resource type, including explanation, worked example, interactive exercise, short video, code fill-in-the-blank, and review card; d_t is the difficulty adjustment amount; and f_t is the interaction modality, including click selection,

drag-and-drop matching, code input, voice response, and swipe sorting. A lightweight deep Q-network is adopted on-device, and its loss is given by:

$$L_Q = E_{(\tilde{s},a,r,\tilde{s}')\sim D}\left[(r+\gamma \max_{a'} Q_{target}(\tilde{s}',a';\phi^-)-Q(\tilde{s},a;\phi))^2\right] \quad (6)$$

where, D denotes the experience replay buffer and ϕ^- denotes the target network parameters. Low-dimensional state embeddings, a small number of hidden layers, and action branch decomposition are used by the Q-network. The resource type is selected first, followed by the difficulty and interaction modality, so that combinatorial explosion of the action space is avoided. Only the most recent interactions are retained in the experience replay buffer, and prioritized sampling is adopted so that low-probability but high-information failure interactions are learned more frequently. In the exploration strategy, ϵ-greedy is combined with context constraints. Under high cognitive load or low battery, the exploration rate is reduced and safe actions are prioritized. During focused periods and favorable network conditions, the exploration rate is moderately increased.

The reward function is formulated to jointly encode learning gains and mobile costs:

$$r_t = w_1\Delta k_t + w_2\widehat{eng}_t - w_3\widehat{cog}_t - w_4 T_{lat} - w_5 E_{bat} - w_6 D_{path} + w_7 r_t^{imp} \quad (7)$$

where, Δk_t is the knowledge gain estimated from changes in knowledge quiz or response performance between adjacent learning stages, T_{lat} is the total latency of on-device inference and network transmission, E_{bat} is the battery consumption per unit time, D_{path} is the path jump penalty, and r_t^{imp} is the implicit feedback reward. The path jump penalty is defined as the Kullback–Leibler divergence between adjacent policy distributions, by which severe recommendation jumps caused by context switching are suppressed. To satisfy mobile-side constraints, latency, energy consumption, and memory constraints are incorporated into a Lagrangian loss, and dual variables are adaptively updated during training according to the degree of constraint violation. Safety rules are added during actual deployment: high-energy video is prohibited under low battery, offline caching is prioritized under weak network, and difficulty reduction is forced with encouraging feedback pushed when consecutive frustration is detected. Path memory is updated as $m_{t+1} = \mathrm{GRU}(m_t, a_t, \hat{s}_t)$. Context variables, path memory, and resource constraints are simultaneously incorporated into the decision state and reward by this module so that personalization is upgraded from content recommendation to interaction strategy generation under context constraints.

2.4 On-device continual adaptation driven by implicit feedback

Explicit ratings are scarce in mobile learning, whereas implicit signals such as review, repetition, dwell, skip, rapid guessing, and interruption are abundant. These signals are encoded into an implicit reward by the system:

$$r_t^{imp} = \beta_1 replay_t + \beta_2 dwell_t^{norm} - \beta_3 skip_t - \beta_4 rapid_t - \beta_5 abandon_t + \beta_6 streak_t \quad (8)$$

where, $replay_t$ denotes review, $dwell_t^{norm}$ denotes normalized dwell duration, $skip_t$ denotes skipping, $rapid_t$ denotes rapid guessing, $abandon_t$ denotes interruption and exit, and $streak_t$ denotes consecutive learning days. Review and repeated practice are usually regarded as positive signals; however, active review must be distinguished from confused repetition. Skipping and rapid guessing are usually

regarded as negative signals. Abandonment is to be assessed in combination with task completion. For improved robustness, a moving average and outlier clipping are applied to the implicit reward.

On-device continual adaptation is implemented through parameter-efficient updating with low-rank adaptation:

$$W = W_0 + BA,\ B\hat{I}R^{d\times r},\ A\hat{I}R^{r\times k},\ r \ll \min(d,k) \tag{9}$$

$$L_{online} = \lambda_{imp} L_{imp} + \lambda_{reg}\left(\| A \|_F^2 + \| B \|_F^2\right) + \lambda_{dist} L_{dist} \tag{10}$$

where, W_0 is the frozen original weight, B and A are low-rank incremental matrices, r is the rank, L_{imp} is the implicit reward prediction loss, and L_{dist} is the distillation regularization used to prevent catastrophic forgetting. Only A and B are updated. Updating is triggered by interaction count, charging-idle status, or state drift. Gradient clipping and anomaly detection are performed before updating, and validation replay is performed after updating. If performance degradation exceeds a threshold, rollback is performed. To address forgetting during long-term adaptation, a small episodic memory bank is maintained by the system, in which data are stored hierarchically by knowledge point, context type, and error pattern, and mixed sampling is performed during online updating. Personalized parameters are stored in a versioned manner, and rollback, merging, and encrypted cloud backup are supported. Raw touch data are not received by the cloud; only encrypted low-rank increments or aggregate statistics are received. On-device continual adaptation is achieved by this module with extremely low parameter overhead, so that personalization is transformed from one-time model adaptation into continual evolution.

2.5 Unified training and theoretical highlights

The three modules are coupled through a unified loss rather than being trained separately and concatenated. The unified training objective is expressed as:

$$L_{OCAPL} = L_{state} + \lambda_{RL} L_{RL} + \lambda_{online} L_{online} \tag{11}$$

where, L_{state} is the state inference loss, L_{RL} is the path generation loss, L_{online} is the online adaptation loss, and λ_{RL} and λ_{online} are balancing coefficients. On-device state inference complexity is primarily determined by self-attention and cross-modal attention. The number of trainable parameters under low-rank adaptation is far smaller than that under full fine-tuning, and the communication reduction ratio is related to the rank and original dimensions. Compared with cloud recommendation, edge servers, federated learning, and offline on-device models, personal mobile terminals are adopted as the primary inference locus in OCAPL, context constraints are adopted as decision conditions, and implicit feedback is adopted as the continual adaptation signal, thereby forming a unified method.

3 EXPERIMENTAL DESIGN AND RESULTS ANALYSIS

3.1 Experimental setup

A public mobile learning behavior dataset and self-collected higher education mobile learning logs were adopted. Learner responses, dwell, review, swiping,

clicking, network, battery, time period, and course information were covered by the dataset. The baseline methods included cloud-only recommendation, edge-server (edge education scheme), knowledge-only recommendation, no-context deep Q-network, no-update offline model, and full fine-tuning in the cloud. On-device performance, state inference, path generation, online adaptation, ablation robustness, and user study were covered by the evaluation metrics. High-end iPhone, mid-range Android, and low-end Android devices were covered. The statistical significance level was set at $p < 0.05$.

Table 1. Statistics of dataset and device configurations

Item	Number of Learners	Number of Interaction Events	Number of Courses	Mid-Range Devices	Low-End Devices	Weak-Network Scenarios	Low-Battery Scenarios	Commuting Fragmented Periods	Study-Room Focused Periods
Value/ Proportion	1,200	2,340,000	12	47%	25%	12%	16%	31%	38%

Continuous labels for cognitive load and engagement were constructed using task-level standardized ratings, and discrete labels for knowledge mastery states were formed based on knowledge-point quiz results. Learning gain was calculated using the normalized gain $G = (S_{post}-S_{pre})/(S_{max}-S_{pre})$, where S_{pre} and S_{post} are the pre-learning and post-learning knowledge test scores, respectively, and S_{max} is the maximum test score. To characterize the coverage of the experimental data and the distribution of mobile learning contexts, sample sizes, device tiers, and typical network and learning scenarios were first statistically summarized; the specific configurations are listed in Table 1.

Multiple devices, networks, and contexts are covered by the data. Mid-range devices account for 47%, low-end devices account for 25%, and weak-network and low-battery scenarios together account for 28%, so robustness under resource-constrained conditions can be examined. Commuting fragmented periods and study-room focused periods account for 31% and 38%, respectively, and two typical learning scenarios are thereby covered.

3.2 On-device inference performance and resource consumption

Table 2. On-device inference performance and resource consumption

Model Configuration	Model Size	Peak Memory	Low-End Device Latency	Hourly Energy Consumption	Knowledge Mastery F1
Teacher model	1.2 GB	420 MB	1860 ms	11.8%	0.842
Student model	120 MB	58 MB	410 ms	7.6%	0.821
After pruning	80 MB	42 MB	318 ms	6.2%	0.812
After quantization	22 MB	28 MB	196 ms	4.8%	0.803
Quantization+low-rank adaptation	24 MB	30 MB	198 ms	4.9%	0.807

To verify the feasibility of on-device deployment across different compression and adaptation stages, comparisons of model size, peak memory, low-end device

latency, hourly energy consumption, and knowledge mastery F1 were performed, and the results are listed in Table 2. On-device deployment is primarily constrained by latency and memory. A knowledge mastery F1 of 0.842 was achieved by the teacher model, but a low-end device latency of 1860 ms and an hourly energy consumption of 11.8% were observed. Therefore, real-time personalization conditions were not satisfied. After pruning and quantization of the student model, model size was reduced from 120 MB to 22 MB, peak memory was reduced from 58 MB to 28 MB, low-end latency was reduced from 410 ms to 196 ms, energy consumption was reduced to 4.8%, and knowledge mastery F1 decreased only from 0.821 to 0.803. After quantization plus low-rank adaptation, an increase of 2 MB in model size and 2 ms in low-end latency was observed, hourly energy consumption was 4.9%, and knowledge mastery F1 recovered to 0.807. Thus, on-device real-time inference is feasible within the resource budget, and additional overhead caused by online adaptation is within an acceptable range.

3.3 Accuracy of multimodal learning state inference

To evaluate the contributions of different information modalities and fusion mechanisms to learning state inference, input and structural ablations were conducted under identical experimental conditions, and cognitive load mean absolute error, engagement mean absolute error, and knowledge mastery F1 were compared; the results are listed in Table 3. The accuracy of state inference is determined by whether multimodal fusion is sufficient. For full OCAPL, a cognitive load mean absolute error of 0.118, an engagement mean absolute error of 0.103, and a knowledge mastery F1 of 0.842 were achieved, which approached the values of 0.112, 0.098, and 0.851 obtained by the cloud teacher. After touch interaction was removed, the cognitive load mean absolute error increased to 0.156, and the engagement mean absolute error increased to 0.138, indicating that the largest contribution to process-oriented state judgment is made by touch features. After device context was removed, knowledge mastery F1 decreased to 0.812, indicating that contextual information is helpful for distinguishing fragmented learning from focused learning. After cross-modal attention was replaced by simple concatenation, knowledge mastery F1 decreased to 0.819. The weakest performance was observed for the behavior-only sequence, with a cognitive load mean absolute error of 0.172 and a knowledge mastery F1 of only 0.784. The three types of input—behavior, touch, and context—are not mutually substitutable in state inference.

Table 3. Ablation of state inference accuracy

Input Configuration	Cognitive Load Mean Absolute Error	Engagement Mean Absolute Error	Knowledge Mastery F1
Full On-device Context-Adaptive Personalization Loop (OCAPL)	0.118	0.103	0.842
Without touch interaction	0.156	0.138	0.801
Without device context	0.149	0.131	0.812
Without cross-modal attention	0.141	0.126	0.819
Behavior-only sequence	0.172	0.152	0.784
Cloud teacher	0.112	0.098	0.851

3.4 Context-adaptive path generation performance

To examine the adaptability of the path generation strategy under heterogeneous mobile contexts, four typical scenarios—commuting, fragmented periods, weak-network, low-battery, and study-room-focused periods—were selected, and ranking quality, learning gain, task completion rate, and response latency were compared across different methods; the results are listed in Table 4. Marked differences in path generation performance were observed across heterogeneous contexts. In the commuting fragmented scenario, NDCG@5, learning gain, and completion rate were 0.812, 0.183, and 0.91 for OCAPL, respectively, whereas values of 0.756, 0.149, and 0.74 were obtained by cloud-only, and an increase in average latency from 0.42 s to 2.10 s was observed. Under weak-network conditions, the advantage of on-device inference was further amplified; a latency of 0.48 s and a completion rate of 0.89 were achieved by OCAPL, whereas a latency of 3.40 s and a completion rate of only 0.61 were observed for cloud-only. In the low-battery scenario, a completion rate of 0.86 was achieved by OCAPL, which exceeded the 0.76 obtained by the no-context deep Q-network. In the study-room-focused scenario, the highest learning gain of 0.214 and a completion rate of 0.94 were achieved by OCAPL. On-device context-aware path generation was demonstrated to be effective in fragmented, weak-network, low-battery, and focused scenarios.

Table 4. Path generation performance under different contexts

Scenario	Method	NDCG@5	Learning Gain	Completion Rate	Average Latency
Commuting fragmented periods	On-device Context-Adaptive Personalization Loop (OCAPL)	0.812	0.183	0.91	0.42 s
	Cloud-only	0.756	0.149	0.74	2.10 s
Weak-network	OCAPL	0.803	0.176	0.89	0.48 s
	Cloud-only	0.721	0.132	0.61	3.40 s
Low-battery	OCAPL	0.795	0.168	0.86	0.45 s
	No-context deep Q-network	0.758	0.149	0.76	0.50 s
Study-room focused periods	OCAPL	0.835	0.214	0.94	0.38 s
	Knowledge-only	0.792	0.188	0.91	0.95 s

3.5 Online adaptation driven by implicit feedback and long-term effects

To analyze the effects and resource overhead of different continual adaptation strategies during long-term learning, a comparison of learning gain, adaptation speed, knowledge retention rate, communication volume, and memory increment was performed for low-rank adaptation online updating, no-update, on-device full fine-tuning, and periodic cloud retraining, and the results are shown in Figure 2 and Table 5. A balance among effectiveness, overhead, and communication is of concern in long-term adaptation. A four-week learning gain of 0.231 was achieved by low-rank adaptation online updating, which is close to the 0.238 obtained by on-device full fine-tuning, but a memory increment of only 2 MB was observed. A memory increment of 86 MB was observed for full fine-tuning, making it unsuitable for

mid-range and low-end devices. For the no-update model, a learning gain of only 0.142, 45 cold-start adaptation rounds, and a knowledge retention rate of 0.72 were observed, indicating that offline deployment rapidly deviates from the learner's state. For periodic cloud retraining, a learning gain of 0.226 was achieved, but a daily communication volume of 1.2 GB was observed, which is much higher than the 12 MB for low-rank adaptation. Considering both effectiveness and overhead, better cost-effectiveness is demonstrated by low-rank adaptation online updating in mobile-side continual adaptation.

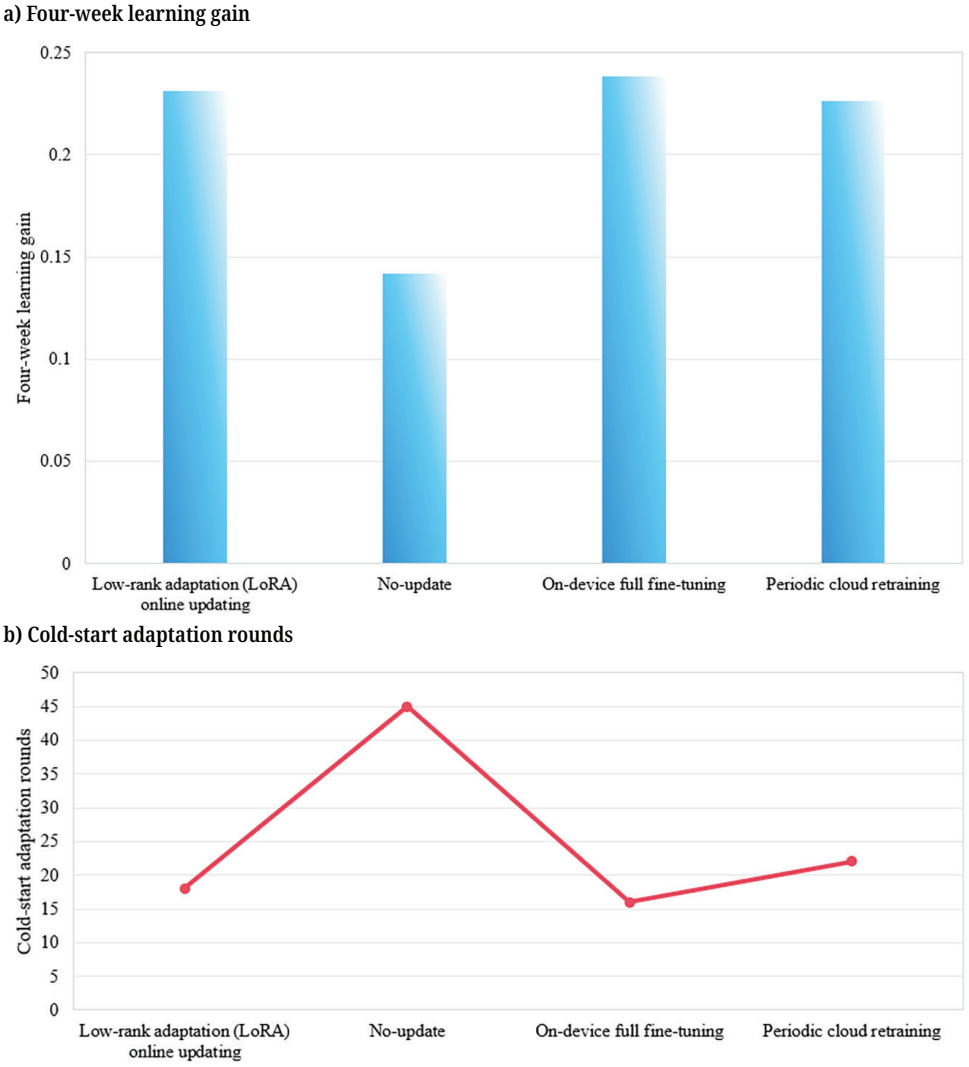


Fig. 2. Four-week learning gains and cold-start adaptation rounds obtained under different methods

Table 5. Four-week online adaptation performance

Method	Knowledge Retention Rate	Daily Communication Volume	Memory Increment
Low-rank adaptation online updating	0.86	12 MB	2 MB
No-update	0.72	0 MB	0 MB
On-device full fine-tuning	0.87	8 MB	86 MB
Periodic cloud retraining	0.84	1.2 GB	4 MB

3.6 Ablation, robustness, and user study

To further identify the impact of each core module on overall system performance and to test stability under complex operating conditions, ablation analyses of touch interaction, device context, implicit reward, path coherence constraint, and resource constraint were conducted, and the results are listed in Table 6. The necessity of each module is further confirmed by the ablation results. The highest NDCG@5 and learning gain were achieved by full OCAPL, and the lowest path jump was obtained while relatively low latency was maintained. After touch interaction was removed, NDCG@5 decreased to 0.781, and learning gain decreased to 0.161. After device context was removed, average latency increased to 0.55 s and path jump increased to 0.18. After the implicit reward was removed, learning gain decreased to 0.149. After the coherence penalty was removed, path jump increased to 0.27, which was the highest among all ablations. After the Lagrangian constraint was removed, latency was 0.78 s, which exceeded the real-time budget. The robustness average row shows that under network jitter, low battery, cross-device, and cross-disciplinary conditions, NDCG@5 was 0.791, learning gain was 0.168, average latency was 0.51 s, and path jump was 0.13; performance degradation was controllable.

Table 6. Ablation and robustness results

Configuration	NDCG@5	Learning Gain	Average Latency	Path Jump Kullback–Leibler Divergence
Full On-device Context-Adaptive Personalization Loop (OCAPL)	0.812	0.183	0.42 s	0.11
Without touch interaction	0.781	0.161	0.41 s	0.13
Without device context	0.768	0.152	0.55 s	0.18
Without implicit reward	0.774	0.149	0.42 s	0.12
Without coherence penalty	0.795	0.171	0.41 s	0.27
Without Lagrangian constraint	0.806	0.176	0.78 s	0.15
Robustness average	0.791	0.168	0.51 s	0.13

On the basis of objective performance evaluation, to further examine the actual user experience of the system, OCAPL was compared with the cloud baseline in terms of usability, trust, cognitive load, privacy perception, and willingness to continue use; the results are listed in Table 7. Supplementary subjective evidence was provided by the user study. System usability scale scores of 84.2 and 72.4 were obtained for OCAPL and the cloud baseline, respectively; trust ratings of 4.3 versus 3.5, cognitive load ratings of 2.6 versus 3.4, privacy perception ratings of 4.5 versus 3.1, and willingness-to-continue-use ratings of 4.2 versus 3.6 were observed. The largest effect size was found for privacy perception, with Cohen's $d = 1.12$, indicating the practical value of the design in which inference is performed locally and raw data are retained on-device. The decrease in cognitive load indicates that additional burden caused by mismatched recommendations was reduced by context adaptation. Consistent with the objective indicators, comprehensive advantages of OCAPL in usability, trust, and privacy perception are supported by the user study results.

Table 7. Subjective scale results from the user study

Metric	On-Device Context-Adaptive Personalization Loop (OCAPL)	Cloud Baseline	p Value	Cohen's d
Usability of the system usability scale	84.2	72.4	<0.01	0.82
Trust	4.3	3.5	<0.01	0.91
Cognitive load	2.6	3.4	<0.01	0.78
Privacy perception	4.5	3.1	<0.001	1.12
Willingness to continue use	4.2	3.6	<0.01	0.74

4 CONCLUSION

Mobile personalized learning is advanced by OCAPL from a cloud master–slave architecture to an on-device closed loop, and state inference, path generation, and continual adaptation are unified under a sequential decision-making framework with resource constraints. As a result, latency, memory, energy consumption, and communication are transformed from post-deployment engineering constraints into optimization variables during training, and a unified trade-off between recommendation quality and mobile cost is formed. It is experimentally demonstrated that stable performance is achieved by the closed loop in on-device inference latency, peak memory, hourly energy consumption, completion rates under weak-network and low-battery scenarios, learning gains, path coherence, and long-term low-rank updating. These results indicate that, under the resource constraints of personal mobile terminals, real-time, context-sensitive, continually evolving, and privacy-friendly personalized learning can be simultaneously achieved. For higher education platforms, dependence on network stability and additional edge infrastructure is reduced by this design, synchronization communication and energy overhead are decreased, and raw touch and response data are retained on-device, which is helpful for improving trust and willingness to continue use. Current validation is still limited by the coverage of disciplines, courses, and devices, and individual differences in touch features, noise in implicit rewards, and update scheduling under extreme resource conditions require further examination. Future work can be directed toward multi-device collaboration, federated teacher updating, interpretable state inference, and cross-disciplinary transfer so that on-device context-adaptive personalization is advanced from a methodological prototype to sustainable deployment.

5 ACKNOWLEDGEMENT

This paper was funded by the Project of the Association of Fundamental Computing Education in Chinese Universities: "Exploration and Research on the Teaching Reform of C Language Programming Integrating Curriculum-Based Ideological and Political Education under the Background of Emerging Engineering Education" (Grant No.: 2025-AFCEC-255).

6 REFERENCES

[1] S. K. B. Syed Ali and K. Bai, "Personalized education via mobile technology flexible learning frameworks," *International Journal of Interactive Mobile Technologies*, vol. 18, no. 18, pp. 146–156, 2024. https://doi.org/10.3991/ijim.v18i18.50659

[2] P. Vallejo-Correa, J. Monsalve-Pulido, and M. Tabares-Betancur, "A systematic mapping review of context-aware analysis and its approach to mobile learning and ubiquitous learning processes," *Computer Science Review*, vol. 39, Art. no. 100335, 2021. https://doi.org/10.1016/j.cosrev.2020.100335

[3] S. Sophonhiranrak, "Features, barriers, and influencing factors of mobile learning in higher education: A systematic review," *Heliyon*, vol. 7, no. 4, Art. no. e06696, 2021. https://doi.org/10.1016/j.heliyon.2021.e06696

[4] S. Sun and Y. Fu, "The role of mobile education technology in promoting personalized learning in higher education," *International Journal of Interactive Mobile Technologies*, vol. 19, no. 4, pp. 93–107, 2025. https://doi.org/10.3991/ijim.v19i04.54217

[5] E. B. Cloude, R. Azevedo, P. H. Winne, G. Biswas, and E. E. Jang, "System design for using multimodal trace data in modeling self-regulated learning," *Frontiers in Education*, vol. 7, Art. no. 928632, 2022. https://doi.org/10.3389/feduc.2022.928632

[6] G. Qu, Q. Chen, W. Wei, Z. Lin, X. Chen, and K. Huang, "Mobile edge intelligence for large language models: A contemporary survey," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 6, pp. 3820–3860, 2025. https://doi.org/10.1109/COMST.2025.3527641

[7] N. Szántó, G. D. Monek, and S. Fischer, "Development of an immersive, digital twin-supported smart reconfigurable educational platform for manufacturing training: A proof of concept," *Journal of Engineering Management and Systems Engineering*, vol. 3, no. 4, pp. 199–209, 2024. https://doi.org/10.56578/jemse030402

[8] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020. https://doi.org/10.1109/COMST.2020.2986024

[9] A. K. Pallikonda, V. K. Bandarapalli, and V. Aruna, "Enhancing performance and reducing latency in autonomous systems through edge computing for real-time data processing," *Mechatronics and Intelligent Transportation Systems*, vol. 4, no. 3, pp. 154–165, 2025. https://doi.org/10.56578/mits040305

[10] C. Huang, W. Jiang, K. Li, J. Wu, and J. Zhang, "Enhancing learning process modeling for session-aware knowledge tracing," *Knowledge-Based Systems*, vol. 309, Art. no. 112740, 2025. https://doi.org/10.1016/j.knosys.2024.112740

[11] C. Romero and S. Ventura, "Educational data mining and learning analytics: An updated survey," *WIREs Data Mining and Knowledge Discovery*, vol. 10, no. 3, Art. no. e1355, 2020. https://doi.org/10.1002/widm.1355

[12] K. Matsuda, Y. Sasaki, C. Xiao, and M. Onizuka, "Benchmark for personalized federated learning," *IEEE Open Journal of the Computer Society*, vol. 5, pp. 2–13, 2024. https://doi.org/10.1109/OJCS.2023.3332351

[13] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning in mobile edge networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3606–3621, 2021. https://doi.org/10.1109/JSAC.2021.3118436

[14] R. Chen, L. Li, K. Xue, C. Zhang, M. Pan, and Y. Fang, "Energy efficient federated learning over heterogeneous mobile devices via joint design of weight quantization and wireless transmission," *IEEE Transactions on Mobile Computing*, vol. 22, no. 12, pp. 7451–7465, 2023. https://doi.org/10.1109/TMC.2022.3213766

[15] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5362–5383, 2024. https://doi.org/10.1109/TPAMI.2024.3367329

[16] O. D. Incel and S. Ö. Bursa, "On-device deep learning for mobile and wearable sensing applications: A review," *IEEE Sensors Journal*, vol. 23, no. 6, pp. 5501–5512, 2023. https://doi.org/10.1109/JSEN.2023.3240854

[17] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 869–904, 2020. https://doi.org/10.1109/COMST.2020.2970550

7 AUTHORS

Weijing Li graduated from Hebei University in 2010. Works at the College of Computer and Information Technology, Cangzhou Jiaotong College. Engaged in the research of algorithms, image recognition, and computer applications (E-mail: wjli@czjtu.edu.cn).

Tingting Fu graduated from Hebei University in 2010. Works at the College of Computer and Information Technology, Cangzhou Jiaotong College. Engaged in research in machine learning, artificial intelligence, information retrieval, software engineering, and other related fields (E-mail: futingting198403@163.com).