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.v20i21.63624

PAPER

Content Scheduling and Interaction Optimization for Adult Mobile Learning in Fragmented Learning Contexts

Min Guo[1], Junjiao Li[2], Xiwen Hu[3] (✉)

[1]Hebei Open University, Shijiazhuang, China

[2]Hengshui Open University, Hengshui, China

[3]Cangzhou Open University, Cangzhou, China

czhxw@hebnetu.edu.cn

ABSTRACT

Adult mobile learning is frequently conducted during fragmented periods, such as commuting and lunch breaks, in which available learning windows fluctuate substantially and interruptions are inherently unpredictable. Existing pre-segmentation-based scheduling paradigms fix content units before learning begins, restricting the scheduler to sequencing predefined micro-content and thereby limiting its ability to adapt to real-time time constraints and unexpected interruptions. To address these limitations, a friction-aware generative content scheduling framework was proposed, in which learning friction was decomposed into startup friction, interruption friction, and restart friction and jointly optimized with knowledge gain through an online objective. Under this formulation, real-time content generation, dynamic scheduling, and interaction control enabled by large language models were governed by a unified objective. The framework integrates interruptible content-tree generation, on-device prediction of fragmented time and cognitive readiness, interruption-risk awareness and proactive cognitive closure, edge-cloud collaborative joint scheduling, and progressive interaction optimization. During content generation, semantic closure points were marked within an interruptible content tree, thereby providing scheduling with truncatable semantic interfaces. Interruption risk was estimated by jointly incorporating elapsed time, anomalous interaction patterns, and device-level events. An optimal closure node was selected and a lightweight closure interaction was executed before interruption. Experiments conducted on real mobile devices under representative fragmented-learning conditions for adult learners demonstrate that the friction-aware generative content scheduling framework reduced interruption and restart friction, shortened recovery time, and improved knowledge gain per unit of fragmented learning time, session completion rate, and knowledge retention. These improvements were achieved while satisfying the requirements for end-to-end latency, energy consumption, and interaction responsiveness. Collectively, the proposed framework establishes a unified approach to content scheduling and interaction optimization in fragmented mobile learning.

KEYWORDS

fragmented learning, mobile learning, content scheduling, large language models, learning friction, interruption management, device-cloud collaboration, interaction optimization

Guo, M., Li, J., Hu, X. (2026). Content Scheduling and Interaction Optimization for Adult Mobile Learning in Fragmented Learning Contexts. *International Journal of Interactive Mobile Technologies (iJIM)*, 20(21), pp. 135–149. https://doi.org/10.3991/ijim.v20i21.63624

Article submitted 2026-07-28. Revision uploaded 2026-09-28. Final acceptance 2026-10-11.

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

1 INTRODUCTION

Adult and workforce-oriented continuing education increasingly requires flexible learning arrangements that can be accommodated alongside employment and other daily responsibilities [1]. Against this background, adult mobile learning is typically conducted during fragmented intervals such as commuting, lunch breaks, and gaps between meetings, with durations fluctuating between two and 10 min. Unlike classroom learning, fragmented mobile learning is characterized by discontinuous learning periods and dynamically varying cognitive demands, making the organization of learning resources and the management of cognitive load particularly important [2]. Although anytime access is afforded by mobile devices [3], existing learning systems remain dominated by pre-segmented content and scheduler-based selection and ordering [4], giving rise to two systematic mismatches: a content-granularity mismatch, in which cognitive closure cannot be readily achieved at arbitrary interruption points for pre-segmented micro-videos or micro-texts, and an interaction-rhythm mismatch, in which static feedback cannot be readily adapted to learners' time-varying attentional states [5]. The technical root of these problems can be summarized as learning friction, i.e., additional cognitive costs incurred by learners for startup, interruption, and restart in each session [6]. With the rapid development of generative artificial intelligence, mobile learning systems are increasingly able to provide dynamically generated and personalized learning support rather than relying solely on predefined learning resources [7]. However, the unification of such generation capability with mobile-side scheduling, interruption management, and interaction control has not been achieved and remains an open problem [8]. Learning friction is thereby transformed from an experiential concept into a computable and optimizable objective function, providing an end-to-end technical framework for fragmented mobile learning.

A pre-segmentation-then-scheduling paradigm has mainly been adopted in existing microlearning content scheduling, in which content units are fixed before learning [9], and only existing micro-content can be selected and ordered by the scheduler [10]; real-time temporal fluctuations and unexpected interruptions therefore cannot be readily accommodated. More generally, scheduling under uncertain temporal conditions requires variability in available processing time to be incorporated directly into the optimization process rather than treated as a fixed scheduling parameter [11]. In educational content generation with large language models, linear content or question-answering generation has mostly been focused on [7], whereas interruptibility, cognitive closure, and on-device constraints in mobile fragmented scenarios have not been considered [12]. More generally, real-time inference on resource-constrained edge platforms requires lightweight models that balance computational complexity, latency, and deployment feasibility [13]. Mobile interaction interruption management has mostly been performed as recovery after an interruption has occurred, and pre-interruption risk awareness and proactive cognitive closure mechanisms have been lacking [14]. Overall, content generation, scheduling decisions, and interaction strategies have not been unified under a single optimization objective in existing work, nor has learning friction been explicitly modeled. This gap is particularly critical in mobile technology research, because temporal constraints, interruption risk, and on-device resource limitations in fragmented scenarios are mutually coupled, and effective independent optimization of any single module is unlikely to be achieved [15]. Related studies on computation offloading

have shown that resource-constrained devices can distribute computational workloads between local, edge, and cloud resources to address the joint requirements of processing capacity and latency [16].

A friction-aware generative content scheduling framework is proposed, in which learning friction is decomposed into startup friction, interruption friction, and restart friction and is jointly formulated with knowledge gain as an online optimization objective, such that real-time generation, dynamic scheduling, and interaction control by large language models are aligned under a unified objective function. The method is composed of interruptible content tree generation, on-device prediction of fragmented time and cognitive readiness, interruption-risk awareness and proactive cognitive closure, device-cloud collaborative joint scheduling, and progressive interaction optimization. In the interruptible content tree, semantic closure points are marked during generation, and a truncatable semantic interface is thereby provided for scheduling. In interruption-risk awareness, time consumption, interaction anomalies, and device events are integrated; an optimal closure node is selected before an interruption, and a lightweight closure interaction is executed. In device-cloud collaboration, generation quality, latency, and energy consumption are balanced under mobile-side resource constraints. Multiple experiments on real mobile devices and in adult fragmented-learning scenarios, covering prediction accuracy, generation quality, proactive closure, joint scheduling ablation, device-cloud performance, and user studies, are conducted to validate the effectiveness of the friction-aware generative content scheduling framework in reducing interruption and restart friction, shortening recovery time, and improving knowledge gain per unit of fragmented time. The problem formulation and the overall friction-aware generative content scheduling framework are presented in Section 2; technical details and core formulations of the method are described in Section 3; experimental design and result analysis are provided in Section 4; and conclusions are drawn in Section 5, where future directions are indicated.

2 PROBLEM FORMULATION AND THE FRICTION-AWARE GENERATIVE CONTENT SCHEDULING FRAMEWORK

A fragmented learning session is denoted by $s = \langle \hat{T}, \rho, C, u \rangle$, where $\hat{T}$ is a conditional quantile estimate of the time available at session startup, and the median together with the upper and lower bounds of the prediction interval are simultaneously output on the device side; the depth and breadth of the content tree are determined by the scheduler accordingly. The term $\rho \hat{I}[0,1]$ denotes cognitive readiness and reflects the ease with which the learner can currently enter an effective learning state. C denotes a knowledge coherence constraint, and u denotes the learner profile, which includes historical preferences, mastered knowledge points, cognitive load tolerance, and device usage habits. In the content tree $T = (V, E)$, a node $v \hat{I} V$ represents a micro-content segment that is semantically relatively independent, and an edge represents a prerequisite dependency among knowledge points. Node attributes include duration t_v, cognitive load c_v, semantic completeness $s_v \hat{I}[0,1]$, knowledge point coverage k_v, and closure point marker $z_v \hat{I}\{0,1\}$. For a learning path $P = \{v_1, \ldots, v_n\}$, the cumulative time is $T(P) = \sum_{i=1}^{n} t_{v_i}$, which must lie within the session time window. The temporal constraints, cognitive state, and content

structure of a fragmented session are unified within a single symbolic system by this representation.

Learning friction is concentrated at three moments: session startup, interruption occurrence, and restart after interruption. Startup friction is defined as:

$$L_{start} = -\log(\rho + \varepsilon) \tag{1}$$

where, ε is a numerical smoothing term. The lower the cognitive readiness, the higher the startup cost, and the nonlinear growth of startup cost as cognitive readiness decreases is characterized by the negative logarithmic form. Interruption friction is defined as:

$$L_{interrupt} = \left(1 - s_{v_j}\right) + \lambda \frac{\left|T_j - \hat{T}\right|}{\hat{T}} \tag{2}$$

where the interruption point is denoted by v_j, the elapsed time at the interruption is denoted by T_j, and the time-deviation balancing coefficient is denoted by λ. The first term represents the loss of semantic completeness, and the second term represents the deviation penalty between the elapsed time at the interruption and the predicted available time. Restart friction is defined as:

$$L_{reentry} = 1 - \exp(-\Delta t / \tau) \cdot s_{v_j} \tag{3}$$

where the interval from interruption to restart is denoted by Δt, and the forgetting time constant is denoted by τ. The higher the semantic completeness at the interruption point and the shorter the interval, the lower the restart friction. The friction cost of a single session is given by $F(s, P) = \alpha L_{start} + \beta L_{interrupt} + \gamma L_{reentry}$, where the scenario weights are denoted by α, β, and γ. In commuting scenarios, β and γ can be increased, whereas in lunch-break scenarios, α can be increased.

Knowledge gain is defined as $K(P) = \sum_{i=1}^{n} k_{v_i} \eta^{i-1}$, where $0 < \eta \leq 1$ is the forgetting decay factor and reflects the fact that the absorption rate of knowledge points in later segments of the path is reduced by attentional decay and accumulated cognitive load. The unified optimization objective of the friction-aware generative content scheduling framework is:

$$\min_{T,P,\pi} \mathrm{E}_{s\sim D}[\lambda_F F(s, P) - K(P)] \tag{4}$$

where, λ_F is the trade-off coefficient between friction and gain, and π is the interaction policy. Constraints include the time constraint $T(P) \leq \hat{T} + \varepsilon_T$, the cognitive load constraint $c(P) \leq c_{\max}$, the knowledge coherence constraint $S(P, P_{\mathrm{ref}}) \geq \delta$, the end-to-end latency constraint $L_{e2e} \leq L_{max}$, and the energy consumption constraint $E \leq E_{max}$, where ε_T is the time tolerance, P_{ref} is the target knowledge path, and $S()$ is the path coherence similarity. Under this objective, content generation quality, scheduling decisions, and interaction policy are aligned with the same friction-gain trade-off. Interruption risk and temporal constraints are required to be internalized by the large language model during content tree generation, and the information exposure rhythm of the interaction layer is required to be considered by the scheduler during path selection.

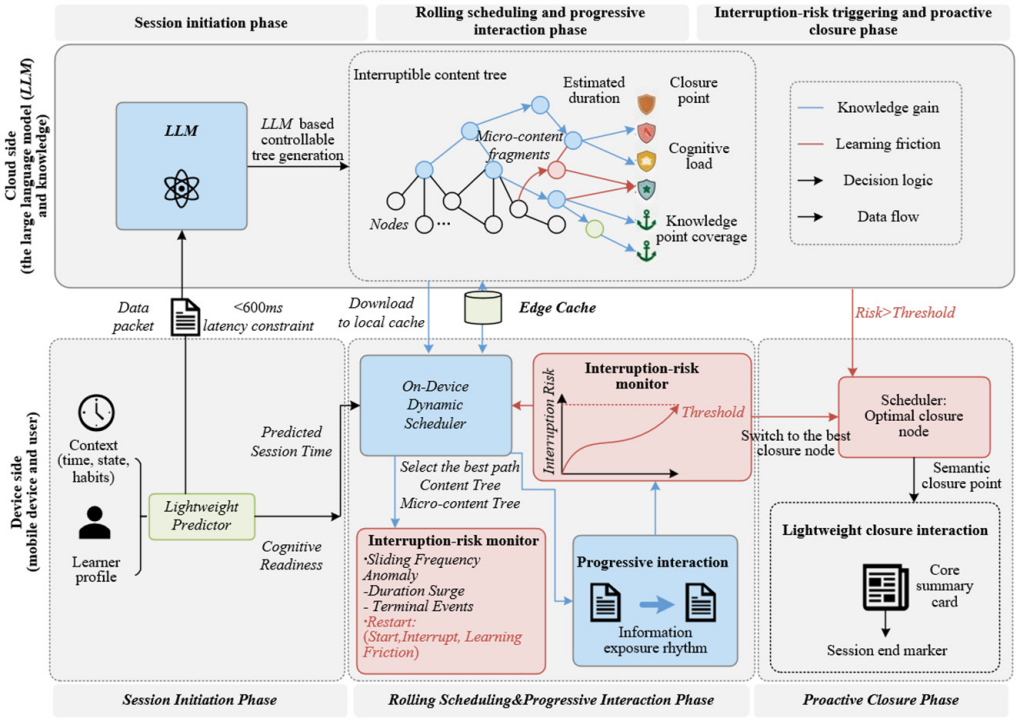


Fig. 1. The device-cloud collaborative architecture and dynamic workflow of the friction-aware generative content scheduling framework

The overall architecture of the friction-aware generative content scheduling framework is composed of an on-device perception layer, a cloud generation layer, an on-device scheduling layer, and an interaction layer, as shown in Figure 1. A lightweight predictor is run in the on-device perception layer, $\hat{T}$ and ρ are estimated, and interruption risk is computed in real time during the session:

$$R(t) = \sigma\left(w_0 + w_1 \frac{t}{\hat{T}} + w_2 \Delta_{slide} + w_3 \Delta_{dwell} + w_4 1_{term} \right) \tag{5}$$

where the elapsed time is denoted by t, the sliding-frequency anomaly is denoted by Δ_{slide}, the dwell-time surge is denoted by Δ_{dwell}, the terminal event indicator function is denoted by 1_{term}, and the weights are denoted by w_0 to w_4. Session context and constraints are received by the cloud generation layer, and a content tree with closure point markers is generated. Beam search is executed on the content tree by the on-device scheduling layer, a path satisfying real-time constraints is selected, and switching to the optimal closure node v_{opt} is performed when $R(t)$ exceeds a threshold. Progressive content presentation and lightweight closure interaction are executed by the interaction layer, and scheduling decisions are translated into interface behavior. The same friction-gain objective function is shared across layers, rather than being configured as a series of independent modules.

Temporal fluctuations, cognitive states, content structures, and interruption risks in fragmented learning are incorporated into a unified representation by the above formulation, such that content generation, scheduling, and interaction can be coordinated under a single optimization framework. A semantically truncatable interface for dynamic scheduling is provided by the interruptible content tree; temporal and cognitive baselines for real-time decision-making are provided by

on-device prediction; cognitive closure is completed by the proactive cognitive closure mechanism before an interruption occurs; and generation quality, latency, and energy consumption are balanced by device-cloud collaboration under mobile-side resource constraints. A unified problem definition and objective function are provided by this framework for content tree generation, interruption-risk awareness, joint scheduling, and interaction optimization in subsequent methods.

3 FRICTION-AWARE GENERATIVE SCHEDULING

3.1 Online solution of the unified objective

On the basis of the problem formulation in Section 2, content tree generation, path selection, and interaction control are organized by the friction-aware generative content scheduling framework into a single online decision-making loop. At session startup, the time budget, cognitive state, and knowledge coherence constraint are initialized on the device side according to $s = \langle \hat{T}, \rho, C, u \rangle$, and a candidate content tree T is generated accordingly on the cloud side. One-shot ranking of fixed micro-content is no longer performed on the device side; instead, a set of paths satisfying time and cognitive load constraints is continuously maintained on the candidate tree, and the current learning path P is updated in a rolling manner according to session state.

Path evaluation is still jointly determined by knowledge gain and learning friction. For candidate paths and interaction policies, the objective is equivalently maximized as $E[K(P) - \lambda_F F(s, P)]$, where startup friction is used to constrain initial content difficulty, interruption friction is used to suppress session termination at positions with low semantic completeness, and restart friction is used to evaluate the potential cost of re-localization after an interruption. As a result, the same evaluation scale is used in the generation, scheduling, and interaction stages, and objective inconsistency caused by separate optimization of local metrics in individual modules is avoided.

As the session proceeds, interruption risk $R(t)$ is continuously updated on the device side. When risk remains below the threshold, the current knowledge path is preferentially expanded by the scheduler. When the threshold is reached, closure nodes with higher semantic completeness and lower cognitive load are searched within the allowable time margin, and the path is switched to the corresponding node so that closure is completed. Through this mechanism, interruption management is shifted from post hoc recovery to preemptive risk response, while consistency between the path and the target knowledge structure is maintained.

On-device perception, cloud generation, on-device scheduling, and interaction control thus constitute a closed loop: temporal and state estimates are provided by the perception layer, truncatable candidate content structures are provided by the generation layer, real-time path selection is performed by the scheduling layer, and path changes are mapped by the interaction layer to progressive presentation and lightweight closure. The inputs and outputs of all layers are constrained by the unified objective, and specific implementations of content tree generation, interruption-risk awareness, and device-cloud joint scheduling are provided in the subsequent methods.

3.2 Interruptible content tree generation

The generation capability of the large language model is moved upstream into scheduling constraints by interruptible content tree generation. Inputs to the large

language model include a textualized description of the learner profile u, a context summary of the current knowledge path position p, an available time estimate $\hat{T}$ together with its prediction interval, and a level description of cognitive readiness ρ. In the prompt, content is required to be organized by the large language model in a tree structure, and a title, body, estimated duration, cognitive load label, and closure point marker are contained in each node. A closure point marker $z_v \hat{I}\{0,1\}$ indicates whether the node can serve as a natural closure point for a session. During supervised fine-tuning, a multi-task loss is adopted:

$$L_{LLM} = -\sum_v \log P(v \mid \cdot) + \lambda_1 \sum_v (s_v - s_v^{ref})^2 + \lambda_2 (T(P_{ref}) - \hat{T})^2 + \lambda_3 \max(0, c(P_{ref}) - c_{max}) \quad (6)$$

where, s_v^{ref} is the semantic completeness label, P_{ref} is a fixed reference path obtained by annotation and constraint filtering, and λ_1, λ_2, and λ_3 are loss weights. Beam search is used only for candidate path selection at the inference stage and is not involved in backpropagation of this loss. At the inference stage, multiple candidate branches are generated and closure points are marked; a path is selected in the tree by the scheduler, and the optimal closure node is selected when risk is triggered, such that an interruptible semantic interface is provided for content generation.

3.3 On-device prediction and interruption-risk awareness

Temporal baselines and closure bases for online scheduling are provided by on-device prediction and interruption-risk awareness. Fragmented time estimation is performed using a conditional quantile regression model, with the current time period, historical session duration distribution, device state, and calendar events as features; $\hat{T} = Q_{0.5}(T \mid x_t)$ is predicted, and $Q_{0.05}$ and $Q_{0.95}$ are output to form a 90% prediction interval. Cognitive readiness is predicted using $\rho = \exp(-\Delta t/\tau) \cdot (1 - c_{last}/c_{max}) \cdot s_{last}$, where Δt is the interval from the end of the last session to the current moment, τ is the learner-specific forgetting time constant, and c_{last} and s_{last} are the cognitive load and semantic completeness at the end of the last session, respectively. Behavior-aware adaptive systems have demonstrated that interaction signals can be continuously monitored to estimate the risk of undesirable user actions and trigger timely interventions through threshold-based predictive control [17]. Time consumption ratio, sliding frequency anomaly, dwell time surge, and terminal events are integrated by interruption-risk awareness, and risk probability $R(t)$ is output. Proactive closure is triggered when $R(t) \geq \theta$, and the closure node is selected as:

$$v_{opt} = \arg\max_{v \hat{I} P,\, T_v \leq t+\Delta} \left[s_v - \mu \frac{t - T_v}{\hat{T}} - \nu c_v \right] \quad (7)$$

where, Δ is the allowed time margin, and μ and ν are penalty coefficients. A lightweight closure interaction is then pushed, and the learner is guided to end the session in a conscious state.

3.4 Joint scheduling and device-cloud collaboration

Content generation and path selection are integrated by joint scheduling and device-cloud collaboration into an online decision problem, $\max_{P \subseteq T, \pi} E[K(P) - \lambda_T F(s, P)]$, with constraints as previously defined. A receding-horizon strategy is adopted for

online execution, and path selection is updated every 15–30 s according to the latest $R(t)$ and remaining time. When risk remains below the threshold, deeper progression is continued; when the threshold is exceeded, switching to v_{opt} is performed and closure interaction is executed. A lightweight predictor and beam search are run on the device side, whereas large language model-based content tree generation is run on the cloud side. Local fragments of the most recently generated content tree are cached on the device side, and service is degraded to cached content or templated micro-content when network latency causes the end-to-end latency to exceed L_{max}. Progressive disclosure is adopted in the interaction layer; the amount of information exposure is expressed as $I(d) = I_{max}(1 - e^{-\kappa d})$, and cognitive load is approximated as $c(d) = c_0 + c_1 I(d)$, where d denotes sliding depth and κ denotes the disclosure-rate parameter. The interaction policy is optimized as $\pi_{opt} = \text{argmax}_{\pi} E[K(\pi) - \lambda_F F(\pi)]$, and the constraint is $c(d) \leq c_{max}$. The beam width is b, the average branching factor is q, and the search depth is d, the on-device beam search complexity is $O(bqd)$. Under a typical configuration, the computational latency of on-device prediction and path search is maintained below 100 ms. When the cache hit ratio is above 80%, the end-to-end latency, including network and cloud generation, is maintained on the order of approximately 600 ms, which is within the same range as the subsequently measured 620 ms. Only a limited number of candidate paths are updated per round by the receding-horizon strategy, and online computational overhead is thereby maintained at a controllable level.

4 EXPERIMENTAL DESIGN AND RESULTS ANALYSIS

4.1 Experimental settings

The experiments were composed of on-device temporal and cognitive state prediction, quality evaluation of interruptible content tree generation, interruption-risk and proactive closure evaluation, joint scheduling ablation, mobile-side performance testing, and user studies. For the prediction models, training, validation, and test data were split chronologically from the same historical session records, and the test records were not used in parameter fitting; identical input features and test sets were adopted for moving average, survival analysis, quantile regression, and on-device Transformer. In the content generation and scheduling experiments, consistent learning tasks, time budgets, and cognitive load constraints were used; semantic completeness and knowledge coverage were normalized to the interval [0, 1], and closure point F1 was calculated against manually annotated closure point labels. The 90% prediction interval coverage rate was defined as the proportion of samples in which the actual session duration fell between the lower quantile $Q_{0.05}$ and the upper quantile $Q_{0.95}$.

For mobile-side performance, three deployment modes—cloud-only, device-only, and device-cloud collaboration of the friction-aware generative content scheduling framework—were compared, and end-to-end latency, energy consumption, memory usage, cache degradation success rate, interaction fluency, and response time were recorded under the same task set. In the user study, a paired design was adopted with the same cohort of 28 adult learners, and fragmented learning tasks were completed under the baseline condition and the friction-aware generative content scheduling framework, respectively; the tasks covered three content types: programming, language, and management. Continuous metrics were analyzed using two-sided paired tests, with the significance level set at 0.05, and Cohen's d was reported as a standardized effect size.

4.2 On-device prediction accuracy and efficiency

Table 1. Prediction performance for fragmented time and cognitive readiness

Method	$\hat{T}_{MAE/s}$	$\hat{T}_{RMSE/s}$	Interval Coverage (%)	ρ_{MAE}	Inference Latency (ms)
Moving average	68.3	92.1	–	0.148	2.1
Survival analysis	52.7	74.6	81.2	0.121	6.4
Quantile regression	41.2	58.9	89.7	0.093	8.7
On-device Transformer	39.8	56.2	88.3	0.089	22.5

The prediction performance for fragmented time and cognitive readiness is summarized in Table 1. For the quantile regression model, $\hat{T}_{MAE}$ was 41.2 s, $\hat{T}_{RMSE}$ was 58.9 s, and the 90% prediction interval coverage rate was 89.7%, ρ_{MAE} was 0.093, and inference latency was 8.7 ms. Compared with moving average, the mean absolute error was reduced by 39.7%; compared with survival analysis, the mean absolute error was reduced by 21.8%. Although the mean absolute error was further reduced to 39.8 s by the on-device Transformer, inference latency was increased to 22.5 ms, approximately 2.6 times that of quantile regression. The prediction interval coverage rate was close to the target level of 90%, indicating that interval calibration was largely consistent with the specified coverage level. When both accuracy and on-device computational constraints are considered, quantile regression is more feasible on mobile devices, and a stable temporal baseline can be provided for subsequent content tree scale determination and interruption risk judgment.

4.3 Interruptible content tree generation quality

The quality of content tree generation is presented in Table 2. For the full friction-aware generative content scheduling framework, semantic completeness was 0.84, closure point F1 was 0.79, time deviation was 6.3%, cognitive load exceedance rate was 3.2%, and knowledge coverage was 0.87. For large language model-based linear generation, semantic completeness was only 0.61, time deviation was 18.7%, the exceedance rate was 12.3%, and knowledge coverage was 0.74. In comparison, semantic completeness was increased by 37.7%, time deviation was reduced by 66.3%, and cognitive load exceedance rate was reduced by 74.0% in the full configuration. The closure point F1 for unconstrained decoding was 0.58, indicating that a substantial role in closure point identification is played by constrained decoding. These differences indicate that semantic self-consistent closure is more easily formed at available closure nodes by tree organization and closure point marking, thereby providing a truncatable interface for dynamic scheduling.

Table 2. Interruptible content tree generation quality

Configuration	Semantic Completeness s_v	Closure Point F1	Time Deviation (%)	Cognitive Load Exceedance Rate (%)	Knowledge Coverage
Large language model-based linear generation	0.61	–	18.7	12.3	0.74
Without the closure point prediction	0.67	–	14.2	9.8	0.78
Unconstrained decoding	0.72	0.58	11.5	8.1	0.81
Full friction-aware generative content scheduling framework	0.84		6.3	3.2	0.87

4.4 Interruption-risk awareness and proactive closure effects

The effects of proactive closure are shown in Figures 2 and 3. Under the passive interruption strategy, interruption friction was 0.72, restart friction was 0.65, the knowledge retention rate was 54.3%, the session completion rate was 61.2%, and recovery time was 28.4 s. With proactive closure, interruption friction was reduced to 0.31, a reduction of 56.9%; restart friction was reduced to 0.27, a reduction of 58.5%. The knowledge retention rate was increased to 79.2%, the session completion rate was increased to 84.6%, and recovery time was shortened to 9.3 s. Even when compared with the post-interruption recovery strategy, interruption friction was still reduced by 46.6%, restart friction was reduced by 47.1%, and recovery time was shortened by 52.8%. It is indicated by these results that, when cognitive closure is completed before an interruption occurs, the re-localization and reactivation costs incurred by the learner when re-entering the learning state can be significantly reduced.

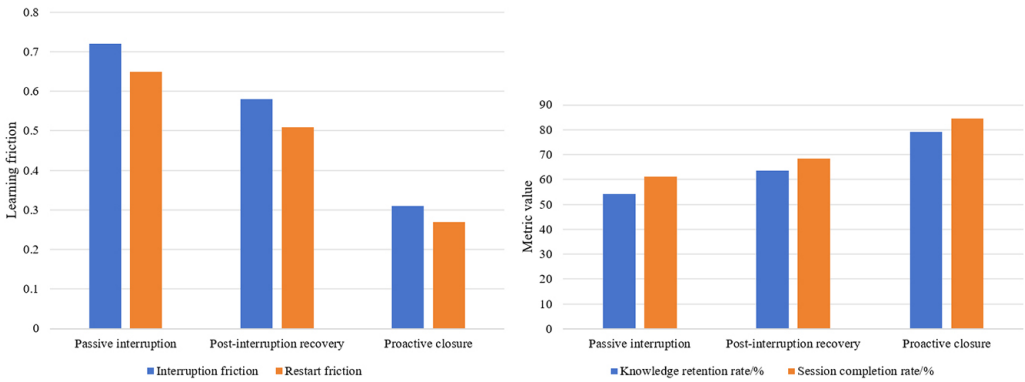


Fig. 2. Effects of interruption-risk awareness and proactive closure under different strategies

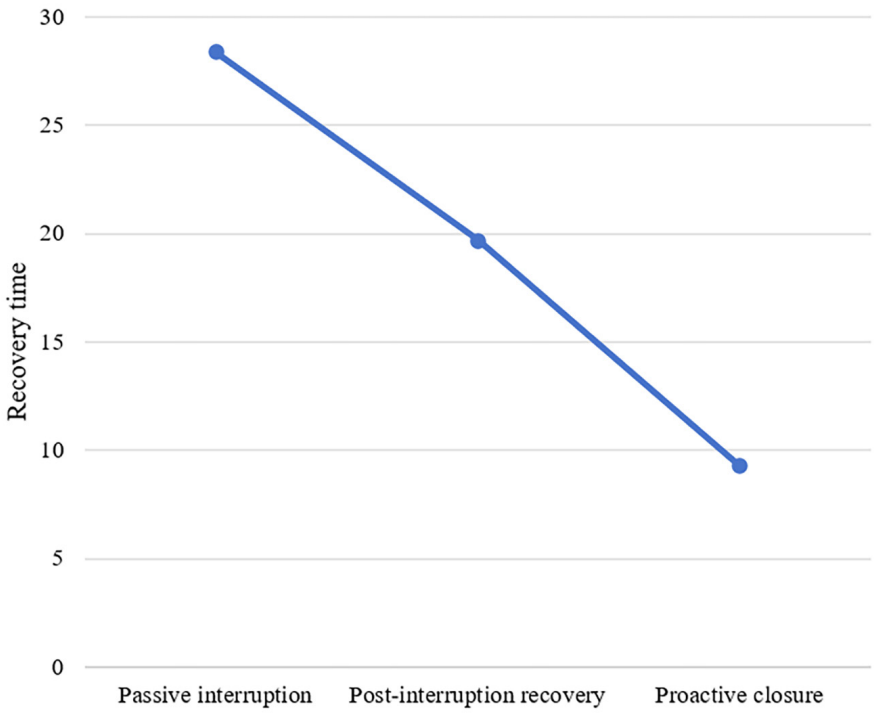


Fig. 3. Recovery time under different strategies

4.5 Ablation study of joint scheduling and interaction optimization

The ablation results in Figure 4 reveal the contribution of each component to the unified objective. For the full configuration, a knowledge gain of 0.87 was achieved, with a friction cost of 0.42, a session completion rate of 84.6%, and a knowledge retention rate of 79.2%. When large language model regeneration was removed, the knowledge gain was reduced to 0.71, the friction cost was increased to 0.68, the completion rate was reduced to 69.3%, and the retention rate was reduced to 62.1%, indicating that a critical role in adapting to fragmented time windows is played by real-time generation. When $R(t)$ was removed, the completion rate was decreased by 11.8 percentage points, and the retention rate by 12.8 percentage points, demonstrating that session continuity and knowledge retention are directly affected by interruption-risk awareness. When progressive disclosure was removed, a knowledge gain of 0.79 and a friction cost of 0.53 were observed, with a retention rate of 71.5%, confirming the influence of interaction rhythm on knowledge absorption. When device-cloud collaboration was removed, a knowledge gain of 0.76 and a friction cost of 0.59 were observed, with a completion rate of 74.2% and a retention rate of 68.7%, its impact being more reflected in mobile-side usability rather than learning outcomes themselves. The full configuration was optimal across all metrics, indicating that the components work in concert around the same friction objective.

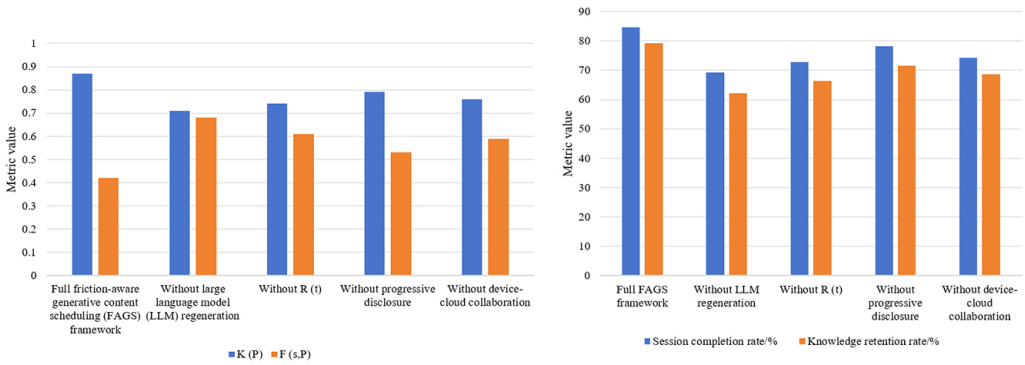


Fig. 4. Ablation results for joint scheduling and interaction optimization

4.6 Device-cloud collaborative mobile-side performance and user study

Mobile-side performance is presented in Table 3. An end-to-end latency of 620 ms was achieved by device-cloud collaboration of the friction-aware generative content scheduling framework, compared with 1850 ms for cloud-only, a reduction of 66.5%; 320 ms was achieved by device-only, but memory usage was as high as 186 MB, making deployment on low-end devices difficult. Energy consumption for the friction-aware generative content scheduling framework was 11.6 mAh/session, a reduction of 37.0% compared with cloud-only, and memory usage was 48 MB, a reduction of 74.2% compared with device-only. A cache degradation success rate of 94.3% indicates that service continuity can still be maintained when the network is unstable. Interaction fluency was 57 frames per second, close to 58 frames per second for cloud-only, and response time was 118 ms, below the 200 ms threshold for fluent interaction. A compromise suitable for mobile devices among latency, energy consumption, memory, and fluency is achieved by device-cloud collaboration.

User study results are reported in Table 4. A decrease in the National Aeronautics and Space Administration Task Load Index (NASA-TLX) from 62.4 to 43.7 was observed, with Cohen's d = 0.82, indicating a medium-to-large effect for the reduction in cognitive load. Satisfaction increased from 4.1 to 5.8, with Cohen's d = 1.12, corresponding to a large effect. Learning performance increased from 58.3% to 74.6%, with Cohen's d = 0.67, indicating that improved experience was not achieved at the expense of learning outcomes. Perceived friction decreased from 5.2 to 3.4, with Cohen's d = 0.95, directly corroborating the effectiveness of the friction minimization objective at the subjective level. The session completion rate increased from 63.5% to 82.1%, with Cohen's d = 0.78; daily average learning duration increased from 12.4 min to 18.7 min, with Cohen's d = 0.54. Statistical significance was reached for all comparisons, and effect sizes ranged from medium to large, indicating that both learning experience and learning behavior in fragmented scenarios were improved under the friction-aware generative content scheduling framework.

Table 3. Device-cloud collaborative mobile-side performance

Metric	Cloud-Only	Device-Only	Device-Cloud Collaboration of the Friction-Aware Generative Content Scheduling Framework
End-to-end latency (ms)	1850	320	620
Energy consumption (mAh/session)	18.4	8.2	11.6
Memory usage (MB)	12	186	48
Cache degradation success rate (%)	–	–	94.3
Interaction fluency (frames per second)	58	42	57
Response time (ms)	210	95	118

Table 4. User study results

Metric	Baseline	Proposed Friction-Aware Generative Content Scheduling Framework	p-Value	Cohen's d
National Aeronautics and Space Administration Task Load Index (NASA-TLX)	62.4	43.7	<0.001	0.82
Satisfaction	4.1	5.8	<0.001	1.12
Learning performance (%)	58.3	74.6	0.002	0.67
Perceived friction	5.2	3.4	<0.001	0.95
Session completion rate (%)	63.5	82.1	<0.001	0.78
Daily average learning duration (min)	12.4	18.7	0.008	0.54

4.7 Parameter sensitivity and generalization

Parameter sensitivity is presented in Table 5. An optimal balance was achieved at $\lambda_F = 0.4$, with $K(P)$ varying by ±0.06 and $F(s, P)$ varying by ±0.09. An optimal value of 0.7 was obtained for θ, with friction varying by ±0.11, indicating that a relatively pronounced effect on friction cost is exerted by the closure threshold. An optimal

value of 0.90 was obtained for η, with knowledge gain varying by ±0.08, indicating that personalized estimation of the forgetting decay factor is relatively important. An optimal compromise was achieved for τ near 25 min, with friction varying by ±0.07. Overall, performance fluctuations within a reasonable parameter range were found to be controllable under the friction-aware generative content scheduling framework, and fine-grained parameter tuning was not required.

Cross-disciplinary testing covered three content types: programming, language, and management. Knowledge gains of 0.86, 0.88, and 0.85 were obtained, respectively, and friction costs of 0.43, 0.41, and 0.44 were obtained, respectively. In cross-scenario testing, session completion rates of 84.2% for commuting, 86.1% for lunch breaks, and 81.3% for meeting gaps were achieved, all of which were superior to the corresponding baselines. These results indicate that relatively stable applicability is exhibited by the framework across the tested disciplines and scenarios.

Table 5. Parameter sensitivity and generalization

Parameter	Range	Optimal Value	$K(P)$ Variation	$F(s, P)$ Variation
λ_F	0.1–1.0	0.4	±0.06	±0.09
θ	0.5–0.9	0.7	±0.04	±0.11
η	0.80–0.98	0.90	±0.08	±0.03
τ (min)	5–60	25	±0.05	±0.07

5 CONCLUSION

To address the core contradiction between fluctuating fragmented time windows and unpredictable interruptions in adult mobile learning, a friction-aware generative content scheduling framework was proposed. In this framework, learning friction was decomposed into startup, interruption, and restart components and jointly formulated with knowledge gain as a unified online optimization objective, such that real-time content generation by large language models, dynamic scheduling under fragmented time, and interaction control share the same objective function rather than being configured as a series of independent modules. Semantic closure points were marked during generation in the interruptible content tree, and a truncatable semantic interface was thereby provided for scheduling. Available time and cognitive readiness were estimated with low latency by on-device prediction. Time consumption, interaction anomalies, and terminal events were integrated by interruption-risk awareness; the optimal closure node was selected before an interruption, and lightweight closure interaction was executed. Generation quality, latency, and energy consumption were balanced by device-cloud collaboration under mobile-side resource constraints. Experimental results on real mobile devices and in adult fragmented-learning scenarios demonstrate that, under the friction-aware generative content scheduling framework, interruption friction and restart friction were reduced, interruption recovery time was shortened, knowledge gain per unit of fragmented time, session completion rate, and knowledge retention rate were improved, and the mobile learning operational requirements specified in this study were satisfied in terms of end-to-end latency, energy consumption, and interaction fluency.

A unified technical pathway from friction modeling to generative scheduling and then to interaction optimization is provided by the friction-aware generative content scheduling framework for fragmented mobile learning. Its value lies in the

transformation of learning friction from an experiential concept into a computable and optimizable objective function, enabling content generation, scheduling decisions, and interaction policies to be coordinated within a single framework. The implication for mobile technology research is that temporal constraints, interruption risk, and on-device resource limitations in fragmented scenarios are mutually coupled, and effective independent optimization of a single module is unlikely to be achieved; however, considerable experience and learning benefits can be achieved under resource-constrained conditions through device-cloud collaboration and proactive closure mechanisms. Future work will be centered on factual verification of large language models, lightweighting of on-device models, and privacy preservation, and the sustained effectiveness of the framework will be validated over longer periods and across cross-cultural user populations.

6 REFERENCES

[1] A. Fakhrorazi, H. Hartini, R. Omar, and R. B. Ikhsan, "Executive education as a tool for workforce capacity development: Evidence from Malaysian private sector employees," *Central Community Development Journal*, vol. 5, no. 2, pp. 20–40, 2025. https://doi.org/10.55942/ccdj.v5i2.1046

[2] L. Hu and D. Liu, "A cognitive load theory-based approach to integrating mobile fragmented learning resources," *International Journal of Interactive Mobile Technologies*, vol. 19, no. 16, pp. 60–76, 2025. https://doi.org/10.3991/ijim.v19i16.57605

[3] Y. Lee, I. Jahnke, and L. Austin, "Mobile microlearning design and effects on learning efficacy and learner experience," *Educational Technology Research and Development*, vol. 69, no. 2, pp. 885–915, 2021. https://doi.org/10.1007/s11423-020-09931-w

[4] J. Dai, X. Gu, and J. Zhu, "Personalized recommendation in the adaptive learning system: The role of adaptive testing technology," *Journal of Educational Computing Research*, vol. 61, no. 3, pp. 523–545, 2022. https://doi.org/10.1177/07356331221127303

[5] L. Tetzlaff, F. Schmiedek, and G. Brod, "Developing personalized education: A dynamic framework," *Educational Psychology Review*, vol. 33, no. 3, pp. 863–882, 2020. https://doi.org/10.1007/s10648-020-09570-w

[6] J. Guo, T. Chen, Z. Xie, and C. K. Or, "Effects of interventions to reduce the negative consequences of interruptions on task performance: A systematic review, meta-analysis, and narrative synthesis of laboratory studies," *Applied Ergonomics*, vol. 97, p. 103506, 2021. https://doi.org/10.1016/j.apergo.2021.103506

[7] Z. L. Liu, "Generative AI and mobile learning in higher education: Comparing student and faculty perspectives on employability impact," *International Journal of Interactive Mobile Technologies*, vol. 19, no. 1, pp. 34–45, 2025. https://doi.org/10.3991/ijim.v19i01.51325

[8] E. Kasneci *et al.*, "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, p. 102274, 2023. https://doi.org/10.1016/j.lindif.2023.102274

[9] D. Ma *et al.*, "Learning path recommendation with multi-behavior user modeling and cascading deep Q networks," *Knowledge-Based Systems*, vol. 294, p. 111743, 2024. https://doi.org/10.1016/j.knosys.2024.111743

[10] F. Zhang, X. Feng, and Y. Wang, "Personalized process-type learning path recommendation based on process mining and deep knowledge tracing," *Knowledge-Based Systems*, vol. 303, p. 112431, 2024. https://doi.org/10.1016/j.knosys.2024.112431

[11] A. Stanković, N. Simić, and M. Stanković, "Metaheuristic optimization for stochastic job scheduling in parallel machine systems with uncertain processing and setup times," *Journal of Engineering Management and Systems Engineering*, vol. 4, no. 3, pp. 206–217, 2025. https://doi.org/10.56578/jemse040304

[12] 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

[13] Q. L. Chen, L. Zhang, J. L. Ma, and A. Khadka, "A lightweight real-time vision framework for road infrastructure monitoring in intelligent transportation systems," *Mechatronics and Intelligent Transportation Systems*, vol. 5, no. 1, pp. 1–10, 2026. https://doi.org/10.56578/mits050101

[14] S. A. Powers and M. W. Scerbo, "Examining the effect of interruptions at different breakpoints and frequencies within a task," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 65, no. 1, pp. 22–36, 2021. https://doi.org/10.1177/00187208211009010

[15] R. Pelánek, "Adaptive learning is hard: Challenges, nuances, and trade-offs in modeling," *International Journal of Artificial Intelligence in Education*, vol. 35, no. 1, pp. 304–329, 2024. https://doi.org/10.1007/s40593-024-00400-6

[16] S. Sheela, K. R. Nataraj, and S. Mallikarjunaswamy, "A comprehensive exploration of resource allocation strategies within vehicle ad-hoc networks," *Mechatronics and Intelligent Transportation Systems*, vol. 2, no. 3, pp. 169–190, 2023. https://doi.org/10.56578/mits020305

[17] M. Savenko, "A systems engineering framework for real-time user error prediction and adaptive user experience control," *Journal of Engineering Management and Systems Engineering*, vol. 5, no. 3, pp. 319–339, 2026. https://doi.org/10.56578/jemse050303

7 AUTHORS

Min Guo is an Associate Professor at the School of Science and Technology, Hebei Open University, and was selected as a third-tier candidate of the "Three-Three-Three Talent Project" of Hebei Province. Member of the Teaching and Research Center Group for the Computer Science and Technology Program at the Open University of China. Has led or participated in multiple provincial-level research projects, published several academic papers, one academic monograph, and co-edited two textbooks (E-mail: guomin@Hebnetu.edu.cn).

Junjiao Li is a young backbone teacher at the School of Science and Technology, Hengshui Open University. She has presided over and participated in a number of provincial and municipal research projects, published one EI indexed paper, compiled one textbook as the chief editor, and co-authored two academic monographs (E-mail: lijunjiao0402@163.com).

Xiwen Hu obtained his bachelor's degree in computer science education from Hebei Normal University and his master of engineering degree in network security technology from the University of Science and Technology Beijing. He is currently a lecturer and deputy director of the Modern Educational Technology Center at Cangzhou Open University. He has published numerous papers on adult education and computer technology. Having long worked on the front line of adult education, he has gained in-depth understanding and research insights into the characteristics of adult education teaching as well as computer-subject teaching. He presided over the research project Problems and Countermeasures for the Digital Transformation of Adult Education in Cangzhou and participated in a number of other research projects related to adult education and computer technology (E-mail: czhxw@hebnetu.edu.cn).