

Cognitive Load Prediction Driven by Mobile Application Interface Complexity Analysis: A Framework Integrating Visual Features and Multimodal Modeling

Zihui Ni

Multimedia University

Fan Zhang

ZF2024Admit@126.com

University of Malaya

ZainudinBin Siran

Multimedia University

CheeOnn Wong

Multimedia University

Yuan Zeng

Multimedia University

Article

Keywords: Mobile Application Interface, Design Complexity, Visual Features, Cognitive Load, Dynamic Context

Posted Date: April 23rd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9063864/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Cognitive Load Prediction Driven by Mobile Application Interface Complexity Analysis: A Framework Integrating Visual Features and Multimodal Modeling

Zihui Ni^{1,a}, Fan Zhang^{2,b*}, ZainudinBin Siran^{1,c}, CheeOnn Wong^{1,d}, Yuan Zeng^{1,e}

¹Multimedia University (MMU), Cyberjaya, Selangor, 63100, Malaysia

²University of Malaya (UM), Kuala Lumpur, 50603, Malaysia

^aEmail: nizihui890709@gmail.com

^bEmail: ZF2024Admit@126.com

^cEmail: zainudin.siran@mmu.edu.my

^dEmail: cowong@mmu.edu.my

^eEmail: 1221400173@student.mmu.edu.my

*Corresponding author: Fan Zhang

Abstract: Current mobile application interface designs neglect cognitive differences among users such as the elderly and those with color blindness, and struggle to adapt to dynamic usage scenarios, leading to inaccurate complexity assessments. This paper integrates visual feature analysis and cognitive load estimation to construct a multimodal fusion modeling framework for the complexity of mobile application interface design. The study encodes three types of inputs in parallel at the temporal window level, extracting and quantifying visual features using CIE Lab-based color contrast statistics, U2-Net (U Square Net) saliency maps, and ResNet50 (Residual Network 50 layers) intermediate layer semantic embedding. Markov entropy rate and navigation path features are used to characterize the interaction load. For implicit cognitive load, physiological load indicators such as real-time heart rate and gaze/pupil features are extracted. Then, modal embeddings are obtained through modality-specific projection and nonlinear mapping, respectively. Gated Multi-modal Unit (GMU) and multi-head cross-modal self-attention are used to weight and model inter-modal information. Transformer Encoder is then used to capture long-term and short-term temporal dependencies. Joint regression outputs cognitive load and high/low load binary classification results, and a multi-task loss is constructed to constrain learning. Finally, the interpretability layer uses SHAP to generate global and local contributions and drive factors to action mapping. Lightweight closed-loop bias correction is used online to eliminate systematic biases for elderly and colorblind individuals. LinUCB context arms are used for online context-arm learning to select appropriate actions. Distillation and INT8 quantization are used at the inference end to meet the real-time and energy consumption constraints of mobile devices. The experiment achieves an RMSE of 0.180 ± 0.020 in cognitive load regression and an F1 score of 0.92 ± 0.03 in high-load segment detection. Even under dynamic context switching from EnvLight to FastMove, the RMSE (root mean square error) remains between 0.15 and 0.20, with only 0.22 for the elderly and 0.23 for the colorblind group. The core of this study lies in distinguishing between objective interface complexity (OCI) and subjective cognitive load. OCI quantifies design attributes, while subjective cognitive load reflects the user's psychological response to a specific task. This paper's multimodal framework uses OCI as input, combined with behavioral and physiological signals, to predict continuous cognitive load and supports interpretable analysis and online interface optimization. This improves the personalized adaptability and dynamic optimization capabilities of the interface design, as well as the accuracy of cognitive load assessment. It provides a reproducible technical path and quantitative benchmark for quantifying and adapting interface cognitive load for heterogeneous user groups and dynamic usage scenarios.

Keywords: Mobile Application Interface, Design Complexity, Visual Features, Cognitive Load, Dynamic Context

1. Introduction

Mobile devices have become an important entry point for people's daily lives and work. UI (User Interface) design directly affects user task efficiency, satisfaction, and accessibility [1-2]. With the diversification of user groups (such as elderly users, users with color vision deficiencies, and users with low technical proficiency) and the complexity of usage scenarios (mobility, distraction, short-term task switching, etc.), the complexity of interface design is increasingly critical to the occupation of users' cognitive resources [3-4]. Accurately assessing interface complexity and optimizing accordingly can not only improve overall usability but also reduce error rates and operational burdens, which has important theoretical and practical significance [5-6]. Therefore, researching how to model the complexity of mobile application interface design in an objective, quantifiable, and dynamic manner is an urgent issue in current human-computer interaction and user experience research.

In recent years, with the popularization of mobile applications, the complexity of interface design has gradually become a research focus. Zhang R proposed a visual complexity measurement model based on multiple linear regression, focusing on visual elements in mobile graphical interfaces, especially how to quantify and optimize the design to improve interface iteration efficiency under limited display space. However, this study did not take into account individual differences, especially the differences in cognitive load between the elderly and colorblind users [7]. Guo F et al. studied the impact of visual complexity on user search behavior and satisfaction through eye tracking, and found that the number of interfaces and layout complexity significantly affected user behavior and satisfaction, but did not explore how to dynamically adjust the interface design according to changes in cognitive load [8]. Ahmad Faudzi M et al. reviewed the user interface design of mobile learning applications, pointing out that small screens and poor design increased cognitive load and affected learning effect [9]. Li X et al. compared the impact of two-dimensional and three-dimensional user interfaces on cognitive load, revealing the different impacts of different interface designs on cognitive load, but it was difficult to adapt to interface optimization in complex dynamic situations [10]. Zhang N et al. studied the impact of interface layout order on dashboard complexity through eye tracking, emphasizing the impact of design order on user visual search behavior [11]. In terms of personalized research, Zhou C et al. studied the impact of interface design elements on the task performance of the elderly and proposed that simplified design could reduce cognitive load, but this was limited to the elderly group and did not fully consider the cognitive differences of other users such as colorblindness [12]. Khan A et al. proposed an adaptive user interface framework designed for the blind, which improved the user experience by simplifying and distributing UI components and achieved significant results in accessibility design, but still lacked design optimization and cognitive load management in dynamic situations [13]. Existing research has made some progress in the assessment of interface design complexity, but most of it focuses on visual elements or behavioral data analysis, ignoring adaptive adjustments and individual differences in dynamic situations, especially the cognitive differences of the elderly and colorblind users.

In recent years, several scholars have explored ways to improve user experience through cognitive load assessment and interface design optimization. Faudzi M A et al. developed a tool to measure cognitive load in mobile learning applications, emphasizing the impact of interface design on learners' cognitive burden. However, this method relied on subjective questionnaires, which had problems with participant fatigue and scale reliability, and failed to dynamically capture changes in cognitive load [14]. Zayim N et al. used the GOMS (goals, operators, methods, and selection rules) method to assess the cognitive load of mobile health applications, and found that cognitive tasks occupied most of the time. They suggested reducing mental arithmetic tasks, but ignored the combined impact of interface visual elements and layout [15]. Zhang M et al. analyzed the impact of interface layout on mobile learning efficiency through eye tracking, and found that sequential layout improved efficiency, but did not consider the interaction between visual elements and dynamic usage scenarios [16]. Suryani M et al. proposed an adaptive learning management system interface model based on cognitive load, but did not effectively integrate visual feature analysis with cognitive load estimation [17]. Zhu T et al. used eye tracking to analyze the impact of visual attention on mobile learning platforms, emphasizing the role of color, text, and typography, but lacked real-time tracking of changes in cognitive load and could not dynamically adjust the interface design [18]. Arana-De las Casas N I et al. assessed the cognitive load of graphical interfaces through a dual N-back task, revealing the impact of cognitive load on user performance, but failed to fully consider the interaction between visual elements and cognitive load [19]. Although existing research has made significant progress in cognitive load and interface design, most methods are limited to visual feature analysis or cognitive load estimation, failing to fully integrate and consider the dynamic changes of visual features and cognitive load, and failing to optimize for the personalized needs of different user groups.

In addition, recent studies have further enriched the foundation of cognitive load theory and interface design methods. Ouwehand K et al. reviewed the emerging trends and innovative applications of cognitive load theory, emphasizing the importance of optimizing learning and interface design in combination with individual characteristics and information processing constraints in the context of rapid technological development [20]. Evans P et al. combined cognitive load theory with self-determination theory, pointing out that structured and self-supporting design strategies can effectively reduce cognitive load and improve user motivation, engagement and performance, providing theoretical guidance for interface design [21]. On the other hand, Zhang R proposed a visual complexity measurement model for mobile GUI, which provides designers with an intuitive and operable optimization reference by quantifying the influence weight of each visual element in the interface, supporting the integrated analysis of interface complexity and cognitive load [22]. At the same time, Muzaffar J emphasized the necessity of following WCAG accessibility standards in mobile application design, ensuring accessibility for all users, including those with visual or cognitive impairments, reflecting the combination of user experience optimization with legal compliance and social responsibility [23].

To address the problem of inaccurate complexity assessment in mobile application interfaces under heterogeneous user groups and dynamic contexts, this paper integrates visual feature analysis and cognitive load estimation. The contributions are as follows: (1) A temporal window-level multimodal fusion framework is constructed. By combining modality-specific projection, gated fusion units, and multi-head cross-modal self-attention with a Transformer encoder, visual, behavioral, and physiological signals are systematically modeled synchronously to output continuous complexity scores and high/low load discrimination. (2) A practical, interpretable-adaptive closed-loop is achieved. Global/local factor recognition based on SHAP (Shapley Additive

exPlanations) is directly mapped to the "factor $\rightarrow$ interface action" rule. Lightweight closed bias correction and LinUCB (Linear Upper Confidence Bound) context arm are used to realize online contextual learning and individualized decision-making for the client. (3) It fully covers the experimental and engineering verification process, from generalization evaluation of leave-one application and online adaptation test to mobile inference (distillation + INT8 (8-bit Integer) quantization) and real-time performance measurement, providing reproducible data, metrics and deployment benchmarks to ensure that the method is both theoretically interpretable and engineering feasible.

Although this paper utilizes existing modules such as GMU, cross-modal attention, and Transformer at the implementation level, its methodological innovation lies not in the proposal of a single network structure, but in the reconstruction of a system-level modeling paradigm and mechanism for the specific problem of mobile interface complexity. At the time window scale, this paper unifies interpretable visual complexity metrics, behavioral uncertainty modeling, and implicit physiological cognitive load into a single prediction-decision closed loop. Through the collaborative design of "gated fusion + cross-modal attention," it explicitly addresses the problems of inconsistent multimodal noise, context-dependent modal importance, and the difficulty in online correction of individual systematic biases. Simultaneously, it directly maps SHAP interpretation results to executable interface adjustment actions and combines LinUCB to achieve online personalized learning, transforming the model from a "static predictor" into an "interpretable and adaptive design decision engine." Therefore, the contribution of this paper lies not in proposing new basic modules, but in presenting a reproducible, deployable, and personalized methodology for mobile interface cognitive load prediction and optimization—a point that has not yet been systematically addressed in existing research.

2. Mobile Application Interface Complexity Fusion Modeling Framework

To clearly present the overall structure and operational logic of the mobile application interface design complexity fusion modeling framework constructed in this paper, data acquisition, preprocessing, feature extraction, multimodal fusion modeling, interpretability mechanism, and online personalized adaptation closed-loop are integrated and displayed. Figure 1 provides an overall understanding of the logical structure between the model's input, processing path, and output results, and also provides a clear structural basis for the design of experimental steps and the implementation of system engineering.

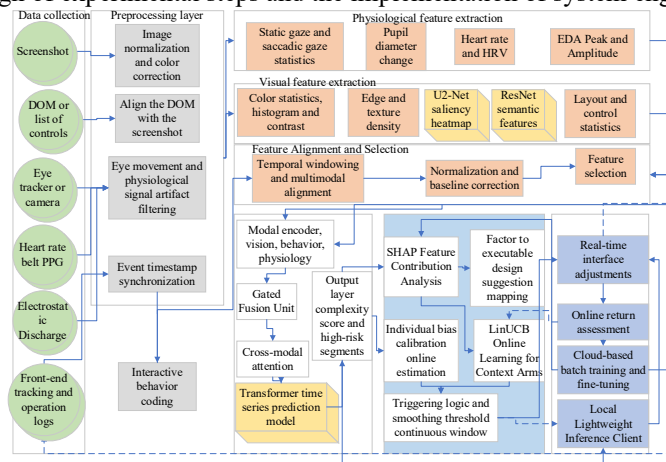


Figure 1. Mobile application interface complexity fusion modeling framework

Figure 1 presents the complete chain of data acquisition $\rightarrow$ preprocessing layer $\rightarrow$ feature extraction $\rightarrow$ feature alignment and selection $\rightarrow$ multimodal fusion and modeling $\rightarrow$ interpretability and personalized adaptation $\rightarrow$ deployment and feedback closed-loop. The system simultaneously collects interface screenshots, DOM (Document Object Model) structure, user operation logs, eye-tracking data, and physiological signals such as PPG (Photoplethysmography)/EDA (Electrodermal activity); subsequently, image correction, DOM alignment, event timestamp synchronization, and physiological signal filtering are completed in the preprocessing layer. Then, visual features, behavioral sequence features, and physiological load features are constructed, including color and texture, control layout, saliency model output, operation trajectory and navigation entropy, as well as heart rate variability and skin conductance changes. All features are aligned and normalized by time windowing before entering the feature selection module to compress redundant dimensions. The fusion modeling part generates interface complexity prediction scores and high-risk segments through modal encoders, gated fusion units, cross-modal attention mechanisms, and temporal models. The model output then enters SHAP interpretability analysis to identify key factors and drive individual bias calibration and LinUCB online learning, achieving real-time adaptation to user states and contexts. The deployment module presents a closed-loop formed by lightweight client-side inference, cloud-based training, and dynamic interface adjustments; simultaneously, feedback data and online reward evaluation flow back into the model, enabling the system to continuously update itself. The overall diagram accurately reflects the system-level design logic of the proposed method in multimodal data integration, real-time prediction, and dynamic adaptation.

2.1 Visual Feature Extraction and Quantization

This paper extracts numerical features from the screenshots I (uniformed in sRGB space and resolution $W \times H$) within each sampling time window.

For dynamic interface elements in mobile applications (such as scrolling lists, elastic animations, and transition effects), this paper introduces a time-series analysis mechanism in the visual feature extraction stage. Through continuous frame optical flow calculation and motion energy assessment, the intensity of pixel-level dynamic perturbations in the interface is quantified to capture the potential impact of scrolling speed, animation amplitude, and interface change frequency on user visual attention. These dynamic features are synchronously aligned with static visual features (color, edges, layout) and behavioral/physiological features to form a complete multimodal time window vector, thus simultaneously reflecting static design complexity and dynamic interaction load. Furthermore, in the interpretability analysis and personalized adaptation modules, the system can identify high-load segments caused by dynamic interfaces and trigger corresponding interface adjustment strategies, such as slowing down animation speed, extending scrolling dwell time, or highlighting key controls, to reduce user cognitive load.

(1) Color and Contrast Statistics

The study maps I from RGB (Red, Green, Blue) to the CIE (International Commission on Illumination) Lab, calculates the histogram for each channel, and normalizes the histogram to obtain the probability distribution $p_c(i)$. The color entropy is calculated as shown in Equation (1).

$$E_c = -\sum_k p_c(i) \log p_c(i) \quad (1)$$

In Equation (1), E_c represents the color entropy of channel c, used to quantify color complexity. c represents the channel, and i represents the bin number.

The contrast ratio is calculated using WCAG (Web Content Accessibility Guidelines), as shown in Equation (2).

$$CR = \frac{L_{\max} + 0.05}{L_{\min} + 0.05} \quad (2)$$

In Equation (2), $L_{\max}$ and $L_{\min}$ are the maximum and minimum relative brightness relative to the reference white, respectively, which can directly reflect the recognition difficulty of users with limited color vision.

(2) Edge and Texture Density

For edge and texture density, the Canny operator is used to obtain the binary edge map $E(x,y)$, and the edge density is calculated as shown in Equation (3). Simultaneously, the average energy T_{hog} of the HOG (Histogram of Oriented Gradients)/Harris feature is calculated to characterize the texture complexity of the image; higher ρ_e and T_{hog} values correspond to higher "texture complexity".

$$\rho_e = \frac{1}{WH} \sum_{x=1}^W \sum_{y=1}^H E(x,y) \quad (3)$$

In Equation (3), ρ_e represents the edge density.

(3) Layout and Element Distribution

N bounding boxes of interface elements are extracted from the DOM/API (Application Programming Interface). Each bounding box is represented by its center position, area, and type label. The element density per unit area and the imbalance (Gini coefficient) of element area distribution are shown in Equation (4). The positional dispersion is represented by the trace $\text{tr}(\Sigma_p)$ of the positional covariance matrix, $\Sigma_p = \text{Cov}(\{p_i\})$. A high trace value indicates that the visual attention is scattered, increasing the search load.

$$\begin{cases} \delta = \frac{N}{WH} \\ G = \frac{1}{2\mu N^2} \sum_{i=1}^N \sum_{j=1}^N |ar_i - ar_j| \end{cases} \quad (4)$$

In Equation (4), δ represents element density per unit area; G represents imbalance; ar_i represents area.

(4) Saliency and Attention Overlap

A saliency map is obtained by using a pre-trained saliency network U2-Net on the image and normalized to probability density. The saliency entropy and the overlap rate between saliency and control are calculated as shown in Equation (5).

$$\begin{cases} HE_{\text{Sig}} = -\sum_{x,y} \text{Sig}(x,y) \log \text{Sig}(x,y) \\ O_{\text{ctrl}} = \frac{\sum_{i=1}^N \sum_{(x,y) \in B_i} \text{Sig}(x,y)}{\sum_{x,y} \text{Sig}(x,y)} \end{cases} \quad (5)$$

In Equation (5), H_{Sig} represents the saliency entropy; O_{ctrl} represents the saliency overlap rate with the control; $\text{Sig}(x,y)$ represents the saliency map; B_i represents the bounding box set. Low O_{ctrl} or high HE_{Sig} indicates that the saliency is scattered or does not match the control, indicating a higher cognitive search cost.

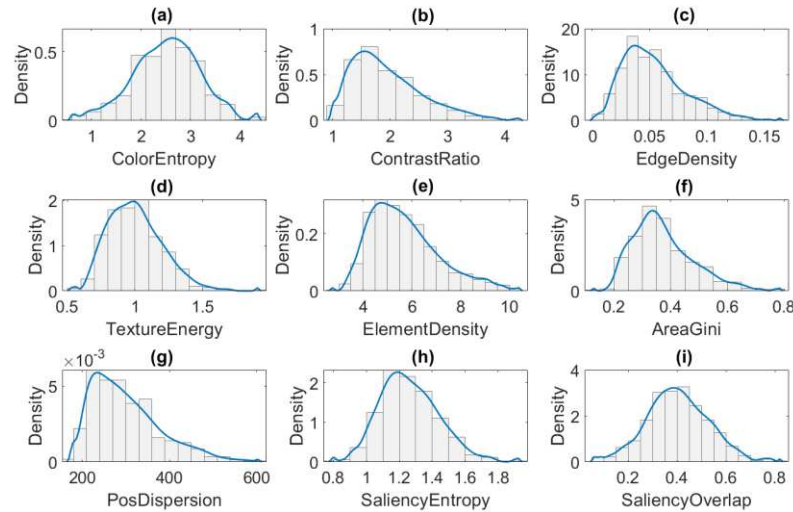
In the saliency map processing, the original saliency response values are first mapped to the [0,1] interval through linear scaling. Then, integral normalization is performed on the entire image region to ensure that the sum of all pixel values is strictly equal to 1, thereby constructing a density distribution that conforms to the probability axiom. This normalization operation eliminates the interference caused by differences in device display brightness and image size variations, making the saliency distributions in different scenarios comparable. This provides a unified benchmark for subsequent entropy calculation and region overlap analysis, effectively avoiding cognitive load assessment bias caused by differences in absolute values.

199 To ensure O_{ctrl} 's consistency across multiple resolutions, all screenshots are first standardized to the same resolution.
 200 Simultaneously, floating-point coordinates are used for precise mapping of DOM/control bounding boxes, and sub-pixel
 201 alignment is achieved through affine transformations. The saliency map is integrated after normalization to probability density,
 202 ensuring the total pixel sum is always 1, thus eliminating absolute numerical differences caused by resolution variations. This
 203 processing strategy ensures the comparability of the saliency-control overlap rates calculated at different resolutions in terms of
 204 both dimensions and distribution. This allows O_{ctrl} to stably reflect the spatial relationship between controls and salient regions,
 205 providing reliable input for complexity evaluation.

206 (5) Semantics and Deep Features

207 This paper uses a pre-trained convolutional network, ResNet50, to extract global feature vectors in intermediate layers and
 208 performs L2 normalization on these vectors for use in subsequent models [24-25]. Optical flow differential energy is used to
 209 calculate the dynamic perturbation intensity for consecutive frames. All the above features are concatenated into a visual feature
 210 vector vis_{in} according to time windows, and z-score standardization is performed on each dimension to eliminate dimensional
 211 differences.

212 To comprehensively evaluate the distribution characteristics of visual complexity quantification indicators across different
 213 design elements, this study conducts multivariate distribution analysis on nine types of visual complexity indicators (color entropy,
 214 contrast ratio, edge density, texture energy, element density per unit area, element area imbalance, element position dispersion,
 215 saliency entropy, and saliency-control overlap rate). The main purpose of the analysis is to compare the performance differences
 216 of each indicator in the sample set to reveal which indicators are more sensitive or stable when measuring visual complexity.
 217 Therefore, kernel density estimation (KDE) combined with histograms is used to simultaneously reflect the probability
 218 distribution shape and sample density of the data. The multivariate distribution of the visual complexity quantification features is
 219 shown in Figure 2. The horizontal axis represents the range of values for each index, and the vertical axis represents the
 220 probability density corresponding to that value. The gray fill in the histogram represents the frequency distribution of the samples,
 221 and the blue curve is the KDE smoothing estimate.



222 Figure 2. Multivariate distribution of quantification features of visual complexity

223
 224 Figure 2(a) Color entropy; Figure 2(b) Contrast ratio; Figure 2(c) Edge density; Figure 2(d) Texture energy; Figure 2(e)
 225 Element density per unit area; Figure 2(f) Element area imbalance; Figure 2(g) Element position dispersion; Figure 2(h) Saliency
 226 entropy; Figure 2(i) Saliency-control overlap rate

227 As shown in Figure 2, there are significant differences in the distribution of each indicator. Color entropy (Figure 2a) is
 228 mainly concentrated in the range of approximately 1.5–3.5, exhibiting a slightly right-skewed normal distribution, indicating that
 229 most design samples have moderate complexity in color information. Contrast ratio (Figure 2b) is distributed in the range of 1–
 230 3.5, showing that some samples have high contrast. Edge density (Figure 2c) is concentrated in the range of 0.02–0.1, with very
 231 few high edge density values, reflecting the overall sparse distribution of design element edges. Texture energy (Figure 2d) is
 232 distributed in the range of 0.6–1.6, indicating that texture energy is moderately high. Element density per unit area (Figure 2e) is
 233 concentrated in the range of 2–8, with some samples having a significantly large number of elements. Element area imbalance
 234 (Figure 2f) is concentrated in the range of 0.1–0.6, indicating that most sample areas are relatively evenly distributed. Element
 235 position dispersion (Figure 2g) is mainly distributed in the range of 200–400, indicating that the element position dispersion
 236 varies greatly. The saliency entropy (Figure 2h) is concentrated between 0.8 and 1.8, indicating a moderate level of complexity in
 237 the saliency information. The saliency-control overlap rate (Figure 2i) is mostly between 0.2 and 0.5, indicating a moderate
 238 degree of overlap in salient regions. Overall, this distribution analysis reveals the differences and dispersion of different visual
 239 complexity indicators on the sample set, providing a quantitative basis for subsequent performance comparisons and design

optimization. For example, when screening high-complexity designs or evaluating user visual perceptual load, the distribution characteristics of indicators such as edge density, element position dispersion, and contrast ratio can be referenced.

2.2 Interactive Behavior Coding

Structured coding and statistics are performed on the operation logs to extract quantitative indicators of behavior that reflect cognitive load and navigation complexity. The original event stream is first sorted into a sequence by timestamp. Each event includes time, discrete coding of event type (click, swipe, etc.), coordinates, and associated control ID (Identity document). The event interval statistics (mean μ_Δ , standard deviation σ_Δ^2 , entropy HE_Δ) are used to characterize responsiveness and hesitation behavior, as shown in Equation (6).

$$\begin{cases} \mu_\Delta = \frac{1}{\varpi-1} \sum_{\varpi=2}^N \Delta\tau_\varpi \\ \sigma_\Delta^2 = \frac{1}{\varpi-1} \sum_{\varpi=2}^N (\Delta\tau_\varpi - \mu_\Delta)^2 \\ HE_\Delta = -\sum_b p_b \log p_b \end{cases} \quad (6)$$

In Equation (6), pro_b represents the binned probability distribution of the adjacent event interval sequence, and $\Delta\tau_\varpi$ represents the adjacent event interval sequence. $\Delta\tau_\varpi$ ($\varpi=2, \dots, N$) represents the precise time interval between the ϖ -th interaction event and the $(\varpi-1)$ -th event after being sorted by timestamp within a time window, where N is the total number of valid interaction events within the window (containing at least 2 events to form an interval sequence); pro_b ($b=1, \dots, B$) represents the proportion of samples in the b -th interval to the total number of samples after the complete interval sequence $\{\Delta\tau_2, \Delta\tau_3, \dots, \Delta\tau_N\}$ is divided into B mutually exclusive intervals using equal-width binning, i.e., $prob_b = \frac{|\{\Delta\tau_\varpi | \Delta\tau_\varpi \in bin_b\}|}{N-1}$, where the value of B is determined by the Freedman-Diaconis criterion to avoid human bias and strictly satisfies the normalization condition $\sum_{b=1}^B prob_b = 1$. This definition method ensures the temporal integrity of the time interval sequence and, through an adaptive binning strategy, enables the probability distribution to objectively reflect the discrete characteristics of the user's operation rhythm, thereby more accurately quantifying the impact of hesitation behavior on cognitive load.

Based on the event coordinates, a click heatmap is constructed, and the number of click clusters DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [26] and the click clustering coefficient are calculated, as shown in Equation (7).

$$\beta = \frac{1}{N} \sum_{i=1}^N \exp(-\alpha \text{dist}(\text{gra}_i, \bar{\text{gra}})) \quad (7)$$

$\bar{\text{gra}}$ represents the click centroid, and α is the scaling parameter. A high cluster number and low β indicate that scattered clicks lead to higher search costs.

This paper abstracts the interface/page as a set of states S , and constructs a state transition matrix P based on the user's transitions in the task. To quantify the uncertainty of navigation, the Markov entropy rate is calculated as shown in Equation (8).

$$HM_{nav} = -\sum_{i=1}^m \pi_i \sum_{j=1}^m P_{ij} \log P_{ij} \quad (8)$$

In Equation (8), π represents the steady-state distribution. A high HM_{nav} corresponds to a more complex navigation path, reflecting an increase in cognitive load.

Task efficiency is defined as the efficiency ratio for completing the task, as shown in Equation (9).

$$\eta = \frac{Su_{succ}}{T_{comp} \cdot (1 + Er_{err})} \quad (9)$$

In Equation (9), Su_{succ} represents successful labeling; T_{comp} represents completion time; Er_{err} represents the number of errors; a low value of η indicates low efficiency and high cognitive load caused by the interface.

Finally, the event type is mapped to an embedding vector, and the aggregate vector (average, maximum, event frequency vector) is calculated within a fixed window to obtain the behavioral feature vector beh_n . At the same time, the instantaneous index of the above behavioral modalities is calculated as numerical features.

2.3 Physiological Signal Acquisition and Cognitive Load Labeling

An eye tracker (sampling rate = 60-300 Hz), PPG/heart rate belt (sampling rate = 50-200 Hz), and electrical conductance analysis (EDA, sampling rate = 20-50 Hz) are used in the experiment, and a uniform timestamp t (hardware trigger) is recorded. All signals are resampled at the minimum sampling interval Δt and linearly interpolated to a common time reference. Bandpass filtering (Butterworth, bandwidth 0.5-40 Hz) is performed on heart rate/PPG, and zero-phase filtering is applied to signal $x(t)$ [27].

Blink removal is performed on the eye tracker signal, and the pupil radius or gaze loss segment is detected and filled in with linear interpolation. Low-pass filtering and baseline drift removal are performed on the EDA (high-pass cutoff frequency 0.05 Hz). In addition, the following statistics are calculated for each window and form the physiological vector phy_{t_n} .

The instantaneous heart rate $HR=60/\overline{RR}$, where HR is the Heart Rate. $\overline{RR}$ is the mean R-R interval (seconds) within the window. The calculation equation for SDNN (Standard Deviation of NN Intervals) is shown in Equation (10).

$$SDNN = \sqrt{\frac{1}{M-1} \sum_{i=1}^M (RR_i - \overline{RR})^2} \quad (10)$$

In Equation (10), RR_i represents the i -th interval, and M represents the number of intervals within the window.

The equation for calculating RMSSD (Root Mean Square of Successive Differences) is shown in Equation (11).

$$RMSSD = \sqrt{\frac{1}{M-1} \sum_{i=1}^{M-1} (RR_{i+1} - RR_i)^2} \quad (11)$$

The power spectral density in the frequency domain is estimated using the Welch method. The power of the low frequency (LF, 0.04–0.15 Hz) and the high frequency (HF, 0.15–0.4 Hz) are calculated, and the ratio is shown in Equation (12).

$$LF/HF = \frac{\int_{0.04}^{0.15} Pow_{RR}(f) df}{\int_{0.15}^{0.4} Pow_{RR}(f) df} \quad (12)$$

In Equation (12), $Pow_{RR}(f)$ represents the power spectral density.

Eye movement metrics include fixation duration, number of fixations, average fixation interval, and pupil diameter change. Fixation duration is the total duration of fixation events within the window. Pupil diameter change $\Delta dia_{pupil} = dia_{pupil}(t) - dia_{baseline}$, where $dia_{baseline}$ represents the resting baseline value.

In electrodermal decomposition, this paper uses convex optimization deconvolution (cvxEDA model form) to decompose the electrodermal signal $\gamma(t)$ into baseline $pb(t)$, driving derivative $da(t)$, and noise $\varepsilon(t)$, as shown in Equation (13).

$$\gamma(t) = pb(t) + (hb * da)(t) + \varepsilon(t) \quad (13)$$

In Equation (13), hb represents the impulse response (Bateman function); $*$ represents the product; $da(t)$ is obtained by solving a convex optimization problem.

In the label construction (cognitive load label), this paper constructs a continuous load score and performs min-max normalization on the subjective scale NASA-TLX (National Aeronautics and Space Administration Mission Load Index) score [28–29]. During the model training and performance evaluation stages, only the NASA-TLX score based on the post-task questionnaire is used as the main supervision label. This label does not depend on any behavioral or physiological input features and serves as an externally independent subjective cognitive load measure. The continuous load score is shown in formula (14).

$$y_{t_n} = s^{(subj)} \quad (14)$$

In Equation (14), y_{t_n} represents the continuous load score; $s^{(subj)}$ represents the subjective scale (NASA-TLX) score; To reduce noise, a moving average filter is used to smooth $\tilde{y}_{t_n}$.

2.4 Multimodal Feature Fusion Modeling and Prediction

This paper clearly distinguishes two related but different concepts: (1) interface objective complexity, which refers to the potential complexity determined by the visual, layout and information organization attributes of the interface design itself. It does not depend on specific users and can be quantified by design features such as color entropy, element density and saliency dispersion; (2) user subjective cognitive load, which refers to the actual cognitive response of users to interface complexity under specific task, situation and individual conditions. It is a comprehensive result of subjective evaluation, behavioral uncertainty and physiological reaction. The multimodal model constructed in this paper does not directly regress the "inherent complexity" of the interface, but uses the interface objective complexity characteristics as one of the main inputs, combined with behavioral and physiological signals, to predict the user's continuous cognitive load score. Therefore, the "complexity prediction" in this paper corresponds to "user load level induced by interface design" in engineering semantics. The prediction result is further used for interpretable analysis and online interface adaptive decision-making.

To perform cognitive load prediction, visual features, behavioral features, and physiological features are first aligned with the same window index to construct a multimodal vector, as shown in Equation (15). Simultaneously, z-score standardization is applied to each dimension.

$$\hat{x}_{t_n} = [vis_{t_n}; beh_{t_n}; phy_{t_n}] \quad (15)$$

In Equation (15), $\hat{x}_{t_n}$ represents the multimodal vector.

This paper adopts hierarchical fusion to balance the complementarity and temporal dependence of modal features. In the modality coding layer, an embedding representation is obtained for each modality using a modality-specific projection matrix and a nonlinear transformation, as shown in Equation (16).

$$z_{t_n}^{(\zeta)} = ReLU(W^{(\zeta)} x_{t_n}^{(\zeta)} + b^{(\zeta)}), \quad \zeta \in \{vis, beh, phy\} \quad (16)$$

In Equation (16), $W^{(\zeta)}$ represents the modality-specific projection matrix.

In cross-modal attention fusion, a multi-head self-attention mechanism is used to model short-to-medium-term dependencies. Multi-head attention is applied to the window sequence, as shown in Equation (17) [30–31]. Through multi-head attention, the model automatically learns the weighting of each modality at different times and in different contexts, solving the problem of modality importance changing over time.

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \quad (17)$$

In Equation (17), Q , K , and V are linear projective self-embeddings, respectively, and d_k represents the key dimension.

Furthermore, this paper uses GMU to perform gated fusion of modality embeddings, as shown in Equation (18). The gating mechanism allows the model to automatically reduce the weight of a modality when there is noise or missing modality, improving

341 robustness.

$$342 \quad \begin{cases} g a_{t_n} = \sigma(W_{ga}[z_{t_n}^{(vis)}; z_{t_n}^{(beh)}; z_{t_n}^{(phy)}] + b_{ga}) \\ z_{t_n}^{(f)} = g a_{t_n} \odot \tanh(W_v z_{t_n}^{(vis)}) + (1 - g a_{t_n}) \odot \tanh(W_p z_{t_n}^{(phy)} + W_b z_{t_n}^{(beh)}) \end{cases} \quad (18)$$

343 In Equation (18), σ represents sigmoid, and $\odot$ represents element-wise multiplication.

344 After fusion, the fused temporal embedding is input into the Transformer Encoder to capture long-short-term dependencies, resulting in a temporal step hidden representation [32-33]. Finally, based on the Transformer Encoder, the complexity of the output is predicted through a fully connected layer and activation, as shown in Equation (19) [34].

$$345 \quad \hat{y}_{t_n} = W_o h_{t_n} + b_o \quad (19)$$

348 In Equation (19), $\hat{y}_{t_n}$ represents the complexity prediction value.

349 The experiment uses regularized mean squared error to measure the loss, as shown in Equation (20).

$$350 \quad \mathcal{L}_{oss_{reg}} = \frac{1}{N} \sum_{n=1}^N (\hat{y}_{t_n} - \tilde{y}_{t_n})^2 + \lambda_1 \|\theta\|_2^2 \quad (20)$$

351 In Equation (20), $\tilde{y}_{t_n}$ represents the smoothed label; θ represents the model parameters; λ_1 represents the L2 regularization coefficient.

353 This paper also performs high/low load binary classification and constructs a joint multi-task loss, as shown in Equation (21).

$$354 \quad \mathcal{L}_{oss} = \lambda_2 \mathcal{L}_{oss_{reg}} + \lambda_3 \mathcal{L}_{oss_{cls}} \quad (21)$$

356 In Equation (21), $\mathcal{L}_{oss_{cls}}$ represents the cross-entropy loss, and λ_2 and λ_3 represent the corresponding weight hyperparameters.

358 To ensure model reproducibility, this paper employs a three-layer Transformer encoder (3 Transformer layers) in the experiments, with each layer containing 8 self-attention heads (head = 8). Each head has an embedding dimension of 64, resulting in a total hidden dimension of 512. The time window length is set to 2 seconds, and the sequence length is 30 windows to accommodate both short-term dynamic changes and long-term dependencies. The multimodal gating unit (GMU) is implemented using element-wise sigmoid gating to weightedly fuse visual, behavioral, and physiological modal embeddings. Each modal embedding is linearly mapped and multiplied by the gating weights, and a modal contribution vector is generated using the tanh activation function. The final fused vector is the weighted sum of all modalities. This implementation ensures robust prediction output even in the presence of missing modalities or noise interference. Furthermore, gradient backpropagation can be used to jointly train the Transformer and GMU parameters, achieving end-to-end optimization.

367 The experiments use the AdamW optimizer [35-36], with Cosine Annealing [37-38] for learning rate scheduling, and walk-forward time-series cross-validation to evaluate generalization ability. Model predictions are calibrated to reduce bias, and focal loss is used for outlier windows to address class imbalance. Modal masking (randomly dropping modalities) is applied during training to improve tolerance for missing modalities during deployment. Model distillation and quantization (INT8) are used at the inference end to meet mobile latency and energy consumption constraints, while gating units reduce the strong dependence on high-sampling-rate physiological signals, ensuring acceptable accuracy even in low-cost sensor scenarios.

373 2.5 Personalized Adaptation and Real-Time Adjustment Engine for Usage Scenarios

374 To achieve personalized adaptation, a lightweight individual calibrator is maintained for each user u and periodically updated on an online window. Real-time input is a multimodal feature vector, and the model output is an instantaneous complexity prediction. If adaptation is triggered after smoothing and thresholding, the engine selects the action a most likely to reduce cognitive load from the action set A and implements interface changes.

378 In individual bias correction, the study first uses a closed-form estimation of the linear bias term to eliminate systematic bias in the short-term calibration of user u , as shown in Equation (22).

$$380 \quad b_u = \frac{\sum_{i=1}^{N_u} (y_i - f(x_i; \theta))}{N_u + \lambda} \quad (22)$$

381 In Equation (22), b_u represents the linear bias term, and N_u represents the number of calibration samples. This closed-form solution can be quickly calculated online, and real-time predictions can be corrected using $\hat{y}_{t_n}^{(u)}$, solving the system bias problem caused by group differences (such as baseline shift in elderly users).

384 For context-aware action selection (with learning), the engine defines the context state s_{t_n} as $[x_{t_n}; \text{act}_{t_n}]$, where act_{t_n} is the external context factor (ambient light, high noise, fast movement, etc.). LinUCB is used as the online learning strategy. For each candidate action a , a matrix A_a and a vector b_a are maintained, and the upper confidence bound score is calculated, as shown in Equation (23) [39].

$$388 \quad \text{score}(a|s) = \hat{\theta}_a^\top s + \sqrt{q s^\top A_a^{-1} s} \quad (23)$$

389 In Equation (23), $\hat{\theta}_a^\top$ represents the least squares estimation parameter of action a , and q represents the exploration weight. Then, the action that maximizes $\text{score}(a|s)$ is selected for execution; after execution, A_a and b_a are updated based on the

observed immediate reward r_t ($r_t = \hat{y}_t^{(pre)} - \hat{y}_{t+\Delta}^{(post)}$, representing the decrease in prediction complexity or the improvement in task efficiency), as shown in Equation (24). This process automatically discovers the most effective adjustment methods in different usage scenarios through context-sensitive online learning.

$$A_a \leftarrow A_a + s s^\top, \quad b_a \leftarrow b_a + r_t s \quad (24)$$

To avoid interface flickering caused by frequent jitter, exponential smoothing is used to adjust the predicted score and a threshold is set for triggering, as shown in Equation (25).

$$\tilde{y}_{t_n} = \zeta \tilde{y}_{t_{n-1}} + (1 - \zeta) \hat{y}_{t_n}^{(u)} \quad (25)$$

ζ in Equation (25) is the smoothing coefficient. The action selection process is only entered when $\tilde{y}_{t_n} > \phi_{\text{high}}$ or $\tilde{y}_{t_n}$ exceeds the threshold for multiple consecutive windows; after the action, the reward is monitored to determine whether to withdraw or solidify the change. ϕ_{high} is the trigger threshold.

Regarding the high-load threshold ϕ_{high} and the smoothing coefficient ζ this paper employs a combination of data-driven and empirical tuning methods to determine the parameters in the experiments. ϕ_{high} is selected based on the continuous complexity prediction score distribution on the training set, typically taking a value higher than the average μ plus 1.5 to 2 standard deviations σ , to ensure that only significantly high-load segments trigger interface adjustments, while also considering the false positive and false negative rates. The smoothing coefficient ζ is optimized through grid search on the validation set, ensuring that the exponentially smoothed prediction sequence retains short-term dynamic changes while avoiding frequent jitter, thereby reducing interface flickering and over-response. Finally, the robustness of both ϕ_{high} and ζ is verified under multi-user, multi-task, and different application scenarios, ensuring that the model is both sensitive and stable during the online adaptive phase.

To meet mobile latency and privacy constraints, personalized calibration and LinUCB parameters are maintained and updated locally on the client side, and encrypted summary statistics are periodically reported to support periodic fine-tuning of the global model in the cloud. Through biased closed-form solutions and lightweight context arms, the system can still achieve rapid personalization and context adaptation in resource-constrained environments, overcoming the limitations of expensive training and frequent online fine-tuning that cannot meet real-time requirements.

To clarify the definition and detection method of high-load segments, this paper adopts a sliding window aggregation strategy in the complexity prediction sequence of continuous windows. The predicted scores are first exponentially smoothed, and then high-load windows are marked according to a threshold ϕ_{high} . Adjacent high-load windows with an interval less than a preset step size are merged into the same high-load segment to reduce false alarms caused by short-term jitter. The detection latency of high-load segments is defined as the time interval from the actual occurrence of a high-load state to the model's first triggering of the threshold, mainly determined by the window length L_{window} , step size S_{step} , and smoothing coefficient ζ . In the experiments, the window length L_{window} is set to 2 seconds, and the step size S_{step} is 0.5 seconds, optimized through grid search on the validation set to balance short-term dynamic response and detection stability. The high-load segment detection rule is based on a judgment strategy where multiple consecutive windows exceed the threshold, ensuring that interface adaptation actions are triggered only under significant cognitive load conditions, while balancing latency and false alarm rate.

In this paper, the ground truth used to evaluate the detection performance of high-load segments (such as F1) is not obtained by thresholding the model's predicted sequence using a threshold ϕ_{high} , but rather comes from externally independent subjective load labels. The ground truth for high-load segments is first constructed based on NASA-TLX mission-level scores: for each mission session, the mission-level NASA-TLX score is mapped to the corresponding time window sequence according to its completion time, and the window-level labels are exponentially smoothed to avoid abrupt noise introduced by inaccurate mission boundaries. Subsequently, using only training set statistics, a uniform threshold (consistent with the ϕ_{high} value used in the model prediction stage, but applied to the "true label sequence" rather than the predicted sequence) is applied to the smoothed window-level subjective load sequence, merging consecutive windows exceeding the threshold into ground-truth segments. The model's output continuous complexity prediction sequence undergoes the same smoothing and thresholding rules during the testing phase to generate predicted high-load segments. The two are matched at the segment level to calculate precision, recall, and F1 score. It is important to emphasize that the thresholding operation is only used to generate the representation of "segments" from continuous labels/predictions, and does not participate in model training, nor does it define the ground truth based on the prediction results. Therefore, there is no circular dependency problem of constructing the ground truth from the prediction results. Based on the above independent labeling and consistent segment generation rules, the reported F1 and other performance metrics reflect the model's ability to detect "true high-load periods defined by NASA-TLX subjective load," providing a clear source of supervision and interpretability.

2.6 SHAP Interpretability and Important Factor Identification

After model training, the SHAP framework is used to calculate the local contribution value of each prediction for the fusion model [40-41]. Local and global importance distributions are obtained based on TreeExplainer (when using tree model components) or KernelExplainer (when using neural networks), respectively, to construct two types of outputs: a global

importance ranking aggregated by feature type, used to reveal the most stable and influential factors in complexity assessment across the entire dataset; and task/session-level local interpretation reports, used to locate the driving features of specific high-load segments. The SHAP output is combined with model uncertainty (prediction variance) to distinguish between factors with "high influence but high uncertainty" (requiring further sampling verification) and stable factors with "high influence and low uncertainty" (which can be directly used for design changes).

Based on the SHAP analysis described above, a set of "factor $\rightarrow$ action" mapping rules is constructed. For each visual or behavioral feature marked as having a high contribution, executable interface adjustments are defined (e.g., when the contribution rate of "significance dispersion" exceeds a threshold, information merging or highlighting of main controls is prioritized; when "element density" has a high contribution and the user is an elderly user, font and control enlargement is performed). This mapping is based on the global ranking of SHAP and dynamically reorders the rule priorities by combining the effect data of online learning of context arms, directly transforming interpretability into a personalized adaptation strategy that can be executed in engineering.

Directly applying KernelSHAP to deep sequence models (such as the multimodal Transformer) incurs high computational costs. To mitigate this issue, this paper employs a hierarchical sampling strategy. First, the time series windows are clustered or uniformly grouped to reduce the number of sample points requiring computation. Second, within each window, only a subset of key features is perturbed, rather than sampling all features simultaneously. Furthermore, SHAP computation is parallelized with model inference and executed on a lightweight model (distillation + INT8 quantization) after training to further reduce time and memory overhead. Additionally, To avoid the interpretation results being influenced by the proportion of supervised labels, the SHAP analysis in this paper is calculated only based on the output of the model trained with externally independent subjective load labels. The importance of behavioral and physiological features reflects their marginal contribution to subjective load prediction, rather than their weight proportion in any composite label. Therefore, high SHAP values of features such as pupil diameter and heart rate variability should be understood as their predictive relevance to load perception, rather than prior biases introduced by label construction.

For KernelSHAP, the background samples are selected from a representative subset of the multimodal features in the training set. Specifically, k-means clustering ($k=20$) is performed on the window-level fused features in the training set, and the centers of each cluster are selected as the background distribution to balance data diversity and computational controllability. Regarding the sampling scale, for each interpreted window sample, KernelSHAP sets the feature perturbation sampling number to 500 times. This value is determined by balancing stability and computational cost on the validation set. In terms of sequence processing, SHAP does not directly perturb the complete time series. Instead, it uses a single time window as the basic interpretation unit, performing subset perturbation on the multimodal feature dimensions within that window. The context window (several preceding and following windows) remains unchanged and serves as conditional input, thus significantly reducing computational complexity while preserving temporal dependency information. These implementations ensure a balance between interpretability, stability, and engineering feasibility in the SHAP results. With the context window fixed as the conditional input, the conditional input for each time window includes not only the features of the current central window, but also the features of the five windows before and after it (a total of 11 windows), to construct a locally temporally dependent contextual representation. The surrounding ± 5 windows are held fixed as context, while only the target window's features are perturbed. This setup ensures that the inputs to SHAP interpretability analysis and the multimodal Transformer can reflect both instantaneous states and capture short-term dynamic changes, thereby providing more stable and accurate contextual information in complexity prediction and high-load segment recognition.

To enhance the engineering feasibility of the "factor $\rightarrow$ action" mapping, this paper further standardizes the design based on the original rules. First, an action library A is constructed, including operations such as interface layout adjustment (control zooming in/out, information merging/highlighting), animation speed adjustment, scroll dwell time extension, and color/contrast optimization. Each type of action defines its applicable conditions, intensity range, and controllable parameters to avoid excessive intervention or interface disruption. Potential side effect evaluation indicators are set for each action, such as the potential decrease in discoverability due to information hiding, changes in task completion time, or the probability of operational conflicts. During the online adaptive phase, these are monitored and adjusted through immediate rewards (rt) and cumulative feedback. Combining the local/global feature contributions from SHAP output and the LinUCB context arm learning strategy, action priorities are dynamically ordered, enabling the system to minimize interface side effects while ensuring cognitive load reduction, achieving safer, more controllable, and verifiable personalized adaptation.

Specific examples of factor-action mapping are shown in Table 1.

Table 1 Specific Examples of Factor-Action Mapping

Contribution Factors	Triggering Conditions	Actions (element A)	Motion Intensity Range	Key side effects indicators:	Online Adjustment and Constraints
High element density per unit area	Element density is in the top 20% of the sample distribution	Font and control enlargement	+10% ~ +25%	Information hiding rate $\uparrow$ (approximately +3%~8%)	LinUCB dynamically selects the amplification level

High element position dispersion	Controls are scattered and search time is increasing	Information grouping and list merging	Merge 1–3 adjacent modules	Initial learning cost ↑	based on task success rate. If task time increases, it rolls back and merges tasks.
High significance entropy (distracted attention)	Salience areas are scattered and accidental touch rate is increasing	Main control highlighting	Highlight opacity 20% ~ 40%	Visual distraction ↑	Prioritizes older users and suppresses high-frequency triggers.
Low significance-control overlap rate	Key controls are not located in the salient area	Key control position fine-tuning	Offset $\leq 5\%$ of screen width	Disruption of spatial consistency	Triggers only during periods of continuous high load.
High dynamic motion energy	Fast scrolling or dense animations	Slow down animation speed	$-20\% \sim -50\%$	Interface responsiveness ↓	Prioritizes execution in high-noise/walking scenarios.
High event interval entropy	Hesitant operation, repeated clicks	Extend scroll dwell time	$+0.3 \sim +1.0$ s	Slower task pace	If efficiency decreases, it shortens the dwell time.
High navigation entropy	Complex page navigation path	Enhance path hints and back entry points	Hint 1–2 key nodes	Increased number of interface elements	Enables only for new users or those with low proficiency.
Insufficient contrast	Contrast is lower than the accessibility suggestion	Enhance color/contrast	$+10\% \sim +30\%$	Risk of visual fatigue	Prioritizes users with color vision deficiencies.

3. Experimental Design

3.1 Experimental Data and Preprocessing

This study collects data from three beta mobile applications, representing e-commerce (Taobao), social networking (WeChat), and tools (Maps). Data sources include beta Android application packages (APKs)/Bundles, laboratory task scripts, and participants' on-site operation records. A total of 120 participants are recruited, aged 18–75, including at least 30 elderly participants and 20 participants reporting color vision abnormalities. The participants are balanced in terms of gender and usage proficiency. Each participant completes a pre-set set of tasks in the three applications, resulting in 9 independent sessions and 1080 session records. Each session simultaneously collects screenshots and DOM/control trees, front-end interaction logs (click, swipe, interface switching timestamps), high-frequency eye-tracking data (Tobii Pro/Pupil Labs or calibrated mobile phone camera solutions), heart rate/PPG and ductal signal (wearable devices such as Polar/Empatica), as well as task completion time, error records, and post-task subjective scales (NASA-TLX and usability questionnaire). The total data included 42,350 standardized screen screenshots, 1,185,420 interaction events, approximately 3,460 hours of eye-tracking records, approximately 1,275 hours of physiological signals (PPG/heart rate/EDA), and 3,240 subjective evaluations. For eye-tracking data, the average loss rate was 5.8%, mainly due to blinking and temporary occlusion; lost data was filled using linear interpolation. For EDA signals, low-pass filtering (5 Hz cutoff) and convex optimized deconvolution (cvxEDA) were used to remove noise and artifacts, preserving the effective bands of the driving derivative. After preprocessing and quality screening, 1,045 out of 1,080 sessions were valid (96.8%); the rest were discarded due to excessive signal loss or task abnormalities. In this experiment, the "approximately 3,460 hours of eye tracking" and "approximately 1,275 hours of physiological signals" do not represent the continuous effective task duration within a single session, but rather the total device wearing and recording time accumulated across all 1080 sessions. This includes calibration phases before and after task execution, short rest and waiting times, and intermittent recordings generated from batches collected over multiple days. The effective task timeframes actually used for modeling and analysis comprise only a portion of this, and these were rigorously truncated during the preprocessing stage through task timestamp alignment and quality screening. Therefore, the effective task duration of a single session conforms to the standard settings for mobile application experiments, and there are no instances where a single session lasts for several hours.

521 All data is first synchronized with a unified clock and aligned in time; screen images are standardized in resolution and
522 converted to a standard color space; DOM/control position information is pixel-level aligned with the screenshots. Eye-tracking
523 and physiological signals undergo preprocessing including artifact removal, filtering, blink removal, and baseline correction.
524 Interaction logs were cleaned, verified, and calibrated before being statistically analyzed within fixed time windows.
525 Subsequently, session quality is screened, and availability is recorded. Each modality is resampled and normalized within a
526 unified time window to form an aligned time-window-level raw feature set, used for subsequent feature calculation, multimodal
527 fusion, and cross-group comparative analysis.

528 To ensure pixel-level alignment accuracy between the DOM/control bounding boxes and screenshots, this paper employs a
529 multi-step correction strategy in the preprocessing stage. Based on the application rendering resolution and screen density, all
530 screenshots are uniformly scaled to a standard resolution and sub-pixel interpolation is performed to reduce scaling errors.
531 Floating-point coordinates of the control bounding boxes are obtained through the DOM/API interface and mapped to the
532 corresponding screenshot pixel grid using geometric transformation (affine transformation). Then, visual matching verification is
533 performed on key controls and interface elements. By calculating the centroid deviation between the bounding box center point
534 and the salient region, the average pixel offset is controlled to within 1 pixel. Finally, in the multimodal feature alignment stage,
535 timestamp synchronization and window smoothing are further employed to ensure precise temporal and spatial matching between
536 control information and the visual frame, thereby reducing potential errors in visual feature extraction, saliency analysis, and
537 complexity prediction. To ensure ethical compliance and data security, all participants signed informed consent forms before the
538 experiment, detailing the data collection content, purpose of use, potential risks, and withdrawal rights. All methods in this study
539 were performed in accordance with the relevant guidelines and regulations. The experimental protocols involving human
540 participants were approved by the Research Ethics Committee of Multimedia University, Malaysia (Ethics Approval
541 Number: MMU-EA-FU-17/20260312). Collected physiological signals such as eye movement, heart rate/PPG, and EDA, as well
542 as interaction logs, underwent de-identification processing, removing directly or indirectly identifying information such as names,
543 accounts, and timestamps, and replacing participant identifiers with random codes. All data was encrypted during transmission
544 and storage and stored on a secure server with restricted access, accessible only to authorized researchers. For vulnerable groups
545 such as the elderly and those with color vision deficiencies, the experimental design was adapted in terms of task difficulty,
546 duration, and visual interface elements, providing additional rest time and necessary operational prompts to reduce cognitive load
547 and experimental stress, ensuring participant safety and comfort.

548 This study strictly adheres to the principles of minimum necessity and time limits, setting clear retention periods for
549 collected data. All de-identified raw multimodal data (interface screenshots, interaction logs, eye-tracking and physiological
550 signals) will be used solely for this study and its replication experiments, and will be retained for a maximum of 5 years after the
551 project ends. After this period, the data will be irreversibly deleted or only statistical summaries of individuals whose identities
552 cannot be traced will be retained. Participants have the right to withdraw at any time during or after the study. Upon receiving a
553 withdrawal request, any data not yet used for published results will be immediately deleted, and data already in the analysis
554 process will be excluded from subsequent analysis and reuse without affecting existing anonymized statistical conclusions,
555 ensuring compliance and protecting the rights of participants.

556 3.2 Experimental Procedure and Task Setting

557 The experiment is divided into two phases to simultaneously validate the model's predictive performance and online
558 adaptive effect. In the training and comparison phases, all session data are grouped by application type and individual participants,
559 split using a time-series sliding window, and a leave-one-application external generalization evaluation is implemented to ensure
560 representative distribution of the elderly group and the color vision deficiency group across the training, validation, and test sets.
561 Multimodal Transformer, CNN-LSTM (Convolutional Neural Network - Long Short-Term Memory), Informer, BiLSTM
562 (Bidirectional Long Short-Term Memory), TCN (Temporal Convolutional Network), and XGBoost (eXtreme Gradient Boosting)
563 are trained under uniform preprocessing, time windows, label smoothing, hyperparameter search space, and early stopping rules
564 to ensure comparability. The evaluation uses regression accuracy (MAE, RMSE), binary classification ability (F1), and
565 time-series detection metrics (precision, recall, F1, and detection latency). Models are selected based on the validation set, and
566 results are reported on the test set. In the online and dynamic adaptation phases, controlled A/B trials are employed. Participants
567 are randomly assigned to a no-adaptation control group, a rule-based adaptation group using manual thresholds, and a
568 learning-based adaptation group incorporating contextual LinUCB and individual bias correction. Task completion time and other
569 metrics are evaluated in real-world tasks, and end-to-end inference latency, memory usage, and energy consumption are measured
570 on representative mobile devices. Simultaneously, ablation experiments and generalization experiments (modal demodulation,
571 leave-one-out application, and few-sample fine-tuning) are used to quantify the contributions of each modality and
572 cross-application transferability. All results are reported in the form of statistical tests and effect sizes to ensure reproducibility
573 and engineering interpretability.

574 To avoid bias in model evaluation caused by time series overlap and individual leakage, this paper uses sessions as the
575 smallest unit for partitioning during training, validation, and test sets. All sessions of the same participant belong to the same set,
576 ensuring they do not span different data subsets. Data splitting was performed at the participant level to prevent data leakage; all
577 sessions from the same participant were assigned to the same split set. For input features generated by a sliding window (0.5s step

size), the high correlation between windows is controlled during the testing phase to avoid information leakage caused by adjacent windows, thus ensuring the fairness and reliability of walk-forward CV and leave-one-app external generalization evaluations. Through this partitioning strategy, the model's generalization ability across different participants and applications can be more realistically verified. During the model training and evaluation phases, a 5-fold walk-forward CV with a 20-session step approach is adopted, with the training set for each fold cumulatively increasing from historical sessions, and the validation set consisting of the next 20 sessions (step size 20 sessions). This setup ensures continuity of time dependencies while balancing training data coverage and validation set independence, providing a robust evaluation of the generalization performance of the complexity prediction model.

This paper standardizes the input feature modality in the experimental design to ensure comparability between different models. Informer, BiLSTM, TCN, and XGBoost all use the same visual, behavioral, and physiological feature vectors as inputs as the multimodal Transformer. When processing sequence models, all time-window sequence features are directly input into the model to capture temporal dependencies. For the tree model (XGBoost), to compensate for its insufficient ability to model sequences, this paper statistically aggregates the multimodal features within each time window, including mean, maximum, minimum, and standard deviation, to construct sequence aggregated features. This preserves temporal information to a certain extent, making the tree model and sequence model comparable in terms of input information and ensuring fairness in performance comparison.

To ensure the reproducibility of the experiments, this paper provides a unified description of the key hyperparameter configurations for training and optimization. All deep models use a batch size of 32 and a maximum training epoch count of 120. The optimizer used is AdamW, with an initial learning rate of 1×10^{-3} , gradually decaying to a minimum learning rate of 1×10^{-5} under cosine annealing. Dropout of 0.3 is consistently applied in the modal encoder, Transformer encoding layer, and fully connected prediction layer to mitigate overfitting. In the joint loss function, the L2 regularization coefficient $\lambda_1 = 1 \times 10^{-4}$, and the regression and classification loss weights are set to $\lambda_2 = 1.0$ and $\lambda_3 = 0.5$, respectively. This configuration was determined through a validation set grid search, achieving a good balance between accuracy and stability. Early stopping was employed during training. Training was terminated and the model rolled back to optimal parameters when the validation set RMSE did not improve by at least 1×10^{-3} within 15 consecutive epochs. To improve the model's robustness to modal missing features during deployment, modal masking was introduced during training. A single modal input was randomly masked with a 20% probability (multiple modalities were not masked simultaneously), while the remaining modalities remained intact. To address the class imbalance issue in high/low load classification, focal loss was used for the classification branch, with parameters set to $\alpha = 0.25$ and $\gamma = 2.0$, to reduce the weight of easily classified samples and enhance the model's learning ability in high-load windows. The hyperparameters were kept consistent across different models and application scenarios to ensure the reproducibility and fair comparison of experimental results. In this paper, the early stopping strategy is set as follows: training is immediately terminated and rolled back to the optimal parameters when the validation set RMSE does not improve by at least 1×10^{-3} within 15 consecutive epochs, in order to avoid overfitting and ensure training stability. In the modal masking strategy, individual modalities are randomly selected in each training sample and masked with a uniform probability of 20%, while explicitly disallowing multiple modalities from being missing simultaneously. This preserves the remaining modal information, ensuring that the model can still effectively learn and predict even with partial modal missingness. This addition makes the hyperparameters and training strategy more complete and reproducible.

4. Multidimensional Performance Indicators

4.1 Model Predictive Performance

To predicting cognitive load, this study selects three indicators for comprehensive evaluation: Mean Absolute Error (MAE), RMSE, and Pearson correlation coefficient r . MAE and RMSE measure the magnitude of the error between the model's predicted values and the actual values, reflecting the accuracy and stability of the prediction; while Pearson r measures the linear correlation between the predicted and actual values, reflecting the model's ability to capture data trends. The overall cognitive load prediction performance is shown in Figure 3. Figure 3 shows the six comparative models on the horizontal axis, and the vertical axis represents the indicator values (MAE, RMSE, Pearson r) for each subplot. The height of the bars in the figure represents the mean of each model's indicator, and the black error bars represent the standard deviation (std), used to show the fluctuation range of the model's predictive performance.

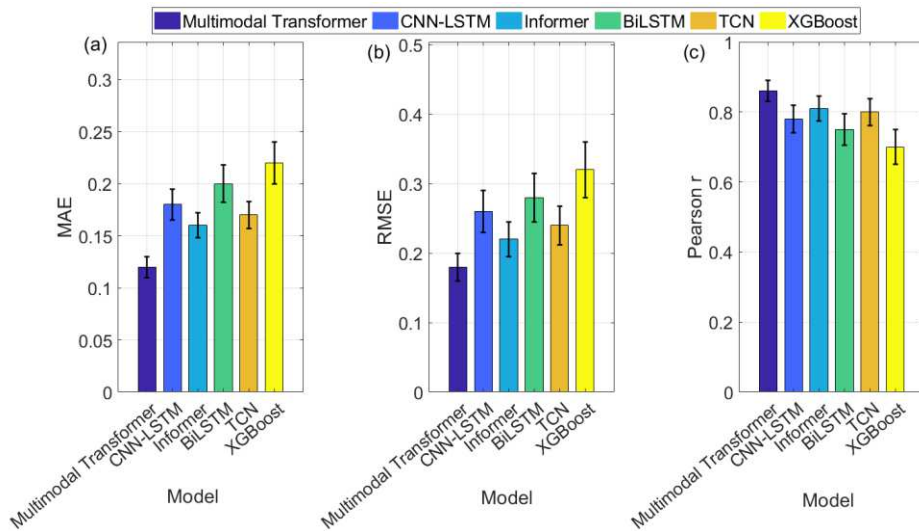


Figure 3. Overall cognitive load prediction performance
Figure 3(a) MAE; Figure 3(b) RMSE; Figure 3(c) Pearson r

Figure 3 clearly shows the differences in prediction performance among the models. In the MAE of Figure 3(a), Multimodal Transformer has the smallest average error (0.12 ± 0.01); CNN-LSTM has 0.18 ± 0.015 ; XGBoost has the largest (0.22 ± 0.02), indicating that Multimodal Transformer is the best in terms of absolute error and has the most stable error. In the RMSE of Figure 3(b), Multimodal Transformer also performs best (0.18 ± 0.02), while BiLSTM and XGBoost have larger errors (0.28 ± 0.035 and 0.32 ± 0.04 , respectively), indicating that they fluctuate more in extreme value prediction. In the Pearson r of Figure 3(c), the Multimodal Transformer has the highest correlation coefficient at 0.86 ± 0.03 , indicating that its predicted values are most consistent with the actual values, while XGBoost has the lowest at only 0.70 ± 0.05 . Overall, the Multimodal Transformer outperforms other single-modal or traditional machine learning models in terms of accuracy, stability, and trend capture ability. Informer and TCN perform moderately, while CNN-LSTM and BiLSTM are slightly lower in error and correlation.

4.2 Verification of Personalized Adaptation Effect and Dynamic Optimization Capability

To verify the personalized adaptation effect and dynamic optimization capability of the interface during human-computer interaction, three strategies are employed: "NoAdapt-RuleAdapt-LearnAdapt." The experiment uses two indicators: completion time and subjective load reduction rate. The former reflects the operational efficiency of the interface, while the latter characterizes changes in the user's cognitive load. Combining the two allows for a more comprehensive evaluation of the interface's real-world performance across different user groups and contexts. The results of the dynamic optimization capability verification are shown in Figure 4. Figure 4 illustrates the task completion time (vertical axis) of the three adaptation strategies during the task phase (horizontal axis 1–10) in dynamic usage scenarios as the environment switches (EnvLight/HighNoise/FastMove).

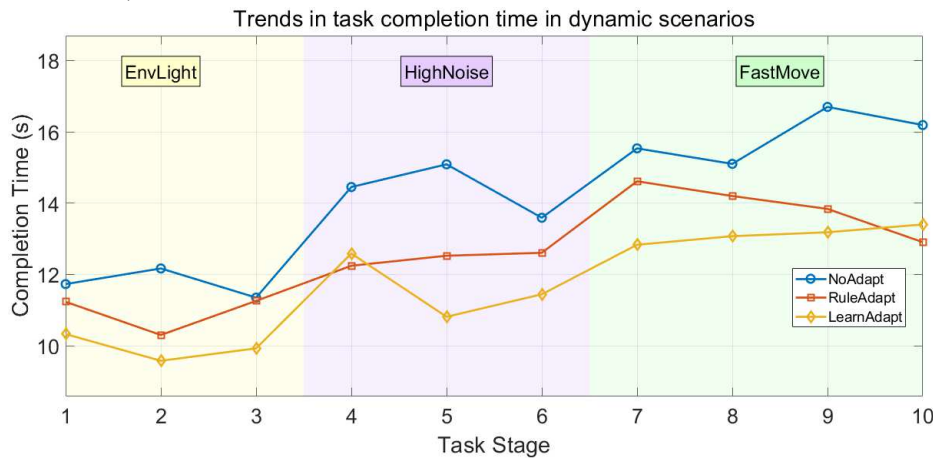
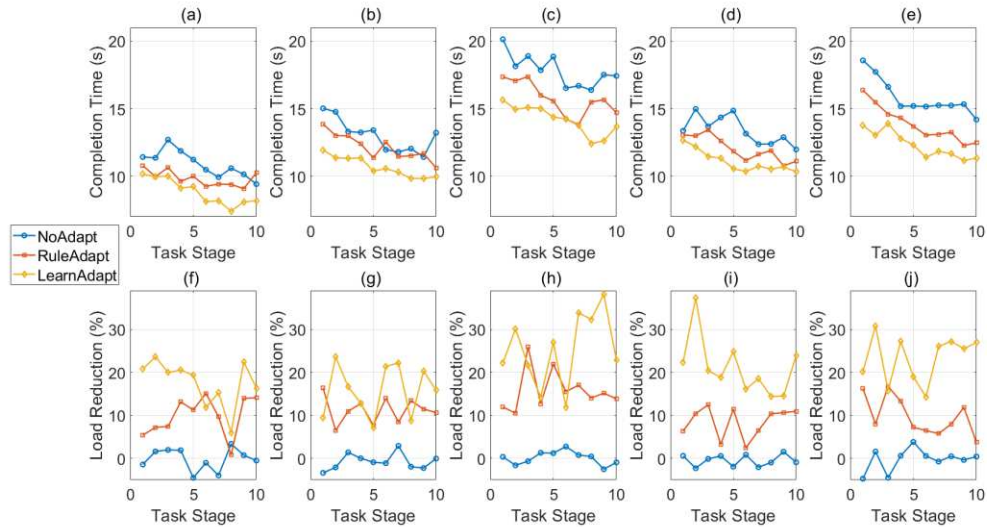


Figure 4. Validation results of dynamic optimization capability

As shown in Figure 4, the overall trend indicates that LearnAdapt consistently achieves the shortest completion time across all 10 stages, such as 10.33 s in Stage 1, significantly outperforming RuleAdapt (11.23 s) and NoAdapt (11.73 s). The gap widens further in noisy and fast-moving environments (Stages 4–10), with LearnAdapt achieving 13.18 s in Stage 9, while NoAdapt reaches 16.70 s, demonstrating that the learning strategy possesses stronger dynamic response capabilities.

652 The personalized adaptation effect is shown in Figure 5. In Figure 5, the horizontal axis represents the task stage, and the
 653 vertical axis represents the results for different user groups (teenagers, middle-aged, elderly, normal vision, colorblind) in terms of
 654 task completion time and subjective cognitive load reduction. Blue lines represent no adaptation; orange lines represent
 655 rule-based adaptation; yellow lines indicate learning-based adaptation.

656 To ensure the reliability and interpretability of the A/B test results, participants were randomly assigned during the design
 657 phase. A computer-generated random sequence was used to evenly distribute 120 participants into three adaptation strategy
 658 groups (no adaptation, rule-based adaptation, and learning adaptation), with approximately 40 participants in each group. The
 659 sample size was chosen based on effect size estimation, and power analysis confirmed that it could detect a moderate effect at
 660 $\alpha=0.05$ and 80% statistical power (Cohen's $d \approx 0.5$). Furthermore, to control for potential confounding factors such as gender, age,
 661 proficiency, and color vision, a stratified randomization strategy was used during grouping to ensure a balanced distribution of
 662 these key characteristics across groups, thereby reducing the interference of individual differences and environmental factors on
 663 the experimental results. These design features ensure that comparisons between groups reflect the effectiveness of the adaptive
 664 strategy itself, rather than other systematic biases.



665
 666
 667 Figure 5. Personalized adaptation effect
 668 Figure 5(a) Task completion time for teenagers; Figure 5(b) Task completion time for middle-aged individuals; Figure 5(c) Task
 669 completion time for the elderly; Figure 5(d) Task completion time with normal vision; Figure 5(e) Task completion time with
 670 color blindness; Figure 5(f) Percentage decrease in subjective cognitive load for teenagers; Figure 5(g) Percentage decrease in
 671 subjective cognitive load for middle-aged individuals; Figure 5(h) Percentage decrease in subjective cognitive load for the elderly;
 672 Figure 5(i) Percentage decrease in subjective cognitive load with normal vision; Figure 5(j) Percentage decrease in subjective
 673 cognitive load with color blindness

673 In Figures 5(a)-5(c) and 5(f)-5(h), the overall task completion time for the adolescent group is lower, and the
 674 learning-adaptive interface shows a clear advantage. For example, in Stage 6, LearnAdapt only requires 8.12 seconds, which is
 675 shorter than NoAdapt (10.48 seconds). Regarding load reduction, LearnAdapt maintains the highest rate in most stages, such as a
 676 20.79% decrease in Stage 1, significantly higher than RuleAdapt (5.41%) and NoAdapt (-1.43%). This indicates that teenagers
 677 are more likely to develop stable operating patterns in learning interfaces. Among the middle-aged group, learning-adaptive
 678 interface also maintains the best efficiency, decreasing from 15.02 seconds for NoAdapt to 11.94 seconds for LearnAdapt in Stage
 679 1, and stabilizing at around 10 seconds in the later stages (Stages 7-10). Regarding load reduction, RuleAdapt shows higher
 680 reductions in a few stages (e.g., 16.34% in Stage 1), but LearnAdapt's effect is more balanced under high-load scenarios, reaching
 681 as high as 20.32% in Stage 9. This indicates that middle-aged users have a certain acceptance of rule adaptation, but still rely on
 682 learning-based adaptation for more stable load reduction in complex tasks. Elderly users have the longest completion time, but the
 683 adaptation strategy brings the most significant improvement. For example, NoAdapt takes 20.13 seconds in Stage 1, while
 684 LearnAdapt takes 15.65 seconds. In terms of load reduction, LearnAdapt reaches over 30% in several stages, such as 30.10% in
 685 Stage 2 and 33.87% in Stage 7, indicating that intelligent adaptation is most helpful in reducing cognitive load for the elderly.

686 In Figures 5(d)-5(e) and 5(i)-5(j), among normal users, learning adaptation also maintains the best time performance,
 687 decreasing from 14.85 s for NoAdapt to 10.56 s for LearnAdapt in Stage 5. Regarding load reduction, RuleAdapt reaches higher
 688 values in some stages (e.g., 10.40% in Stage 2), but LearnAdapt is more stable overall, such as 22.29% in Stage 1 and 23.89% in
 689 Stage 10. This indicates that normal users' gains stabilize after becoming familiar with the task. Colorblind users have higher
 690 times without adaptation (e.g., 18.57 s in Stage 1), but learning adaptation significantly shortens their time to 13.77 s. In terms of
 691 load reduction, LearnAdapt reaches 30.77% in Stage 2, higher than RuleAdapt (7.97%). This group shows the strongest reliance

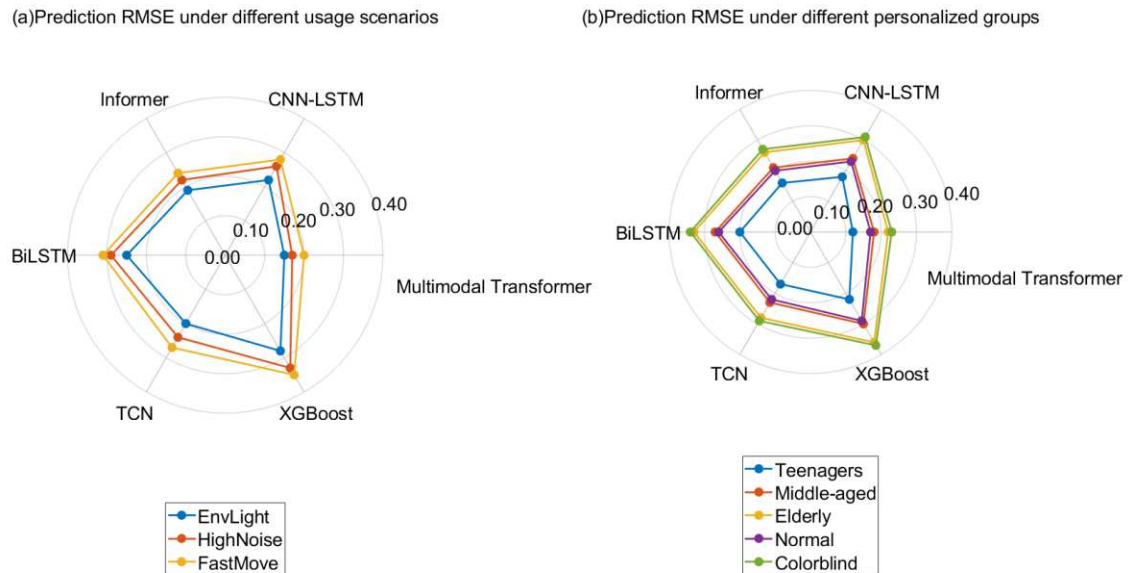
692 on learning adaptation in most stages.

693 The main reason for these differences lies in the fact that learning-based adaptation can automatically adjust interface
694 parameters based on user behavior, environmental disturbances, and task stages. Its optimization process is characterized by
695 "real-time performance, content sensitivity, and user characteristic dependence." In dynamic usage scenarios, when the
696 environment changes from quiet to high noise or high-speed operation, the learning model can instantly adjust the density of
697 interactive information, element layout, and prompting methods, resulting in the most significant reduction in completion time.
698 Data differences among different groups are related to their inherent abilities and perceptual characteristics: teenagers have higher
699 proficiency, and the acceleration effect of adaptation appears more quickly; the elderly and colorblind users have higher
700 perceptual loads, so their load reduction rate is more significant under the visual enhancement, layout reconstruction, and
701 information filtering mechanisms of intelligent adaptation. Overall, the learning-based adaptation strategy, by constructing a
702 mapping relationship between user state and task difficulty, enables the interface to maintain optimal human-computer interaction
703 efficiency across environments and user groups.

704 It is important to note that the superior performance of LearnAdapt compared to RuleAdapt may be partly influenced by the
705 learning period length, task proficiency effect, and the aggressiveness of interface modifications. To mitigate the interference of
706 these potential confounding factors, this study employed a stratified randomization strategy in the experimental design, randomly
707 grouping participants by age, gender, usage proficiency, and color vision status to balance individual differences across groups as
708 much as possible. Furthermore, each participant completed tasks in three applications, generating repeated measures data,
709 allowing the model to control for potential learning effects across tasks and users. Although a strict crossover design was not used,
710 the combination of stratification and randomization strategies with statistical analysis effectively reduced the impact of learning
711 period and task proficiency on the evaluation of personalized adaptation strategies, thus more reliably reflecting the inherent
712 advantages of learning-based adaptive strategies. Future work could further incorporate crossover designs or mixed-effects
713 models to more accurately quantify the causal relationship between strategy performance and learning effects.

714 4.3 Robustness and Generalization of the Model

715 To further explore the predictive ability of the proposed model for interface design complexity (RMSE) under various
716 environmental scenarios and personalized user groups, the cognitive load predictions and usage scenarios are shown in Figure 6.
717 In Figure 6, the root mean square error (RMSE) is used as the core evaluation indicator for complexity prediction. The lower the
718 RMSE value, the closer the prediction result is to the actual cognitive load level. Therefore, RMSE can effectively measure the
719 stability and generalization performance of the model. Each vertex represents a prediction model, and the radius direction
720 indicates the magnitude of the RMSE value. The curve colors correspond to the RMSE situations of different usage scenarios in
721 Figure 6(a) or personalized group types in Figure 6(b). Each closed curve represents the RMSE performance of a certain scenario or
722 user group on the six models, used to compare the differences in prediction accuracy of different models under this condition.



723 Figure 6(a) Prediction RMSE under different usage scenarios; Figure 6(b) Prediction RMSE under different personalized groups

725 Figure 6. Prediction under different personalized groups and usage scenario complexity

726 As shown in Figure 6(a), in all three usage scenarios, the multimodal Transformer achieves the lowest prediction errors, at
727 0.15 (EnvLight), 0.17 (HighNoise), and 0.20 (FastMove), significantly outperforming other models. This is followed by Informer

(0.19–0.24) and CNN-LSTM (0.22–0.28), while XGBoost performs the worst across all scenarios (0.28–0.35). With increasing environmental complexity (such as HighNoise and FastMove), the RMSE of all models shows an upward trend, indicating that dynamic interference and noise reduce prediction accuracy. Figure 6(b) illustrates the model performance across five individualized population groups. The multimodal Transformer achieves the lowest RMSE, with a minimum of 0.12 in the adolescent group and rising to 0.22 in the elderly group. CNN-LSTM and BiLSTM are at intermediate levels (e.g., CNN-LSTM at 0.24 and BiLSTM at 0.27 in the middle-aged condition). XGBoost achieves the highest RMSE across all population groups (0.22–0.37). The multimodal Transformer demonstrates the best robustness and generalization ability under various conditions, while XGBoost is most sensitive to environmental and population differences.

The aforementioned differences primarily stem from variations in the model structures' ability to couple multimodal features and model temporal dependencies. The multimodal Transformer establishes a global correlation between visual features, interactive behaviors, and dynamic context through a self-attention mechanism, enabling it to maintain relatively small errors even in high-noise (HighNoise), fast-moving (FastMove), and visually sensitive groups (such as colorblind). While CNN-LSTM, Informer, and BiLSTM can capture temporal patterns to some extent, their feature fusion capabilities are insufficient when environmental disturbances increase or when there are significant differences in user group cognition, resulting in a more pronounced increase in RMSE. TCN and XGBoost exhibit even more significant performance degradation; the former, limited by its fixed convolutional receptive field, struggles to adapt to rapidly changing interactive states; the latter lacks deep sequence modeling capabilities and fails to adequately capture cross-channel dependencies between visual and behavioral signals, leading to particularly prominent errors in the elderly and colorblind individuals. In summary, the differences in prediction performance reflect the fundamental differences in the models' ability to structurally integrate multimodal information, demonstrating the significant adaptability of the multimodal Transformer in complex situations and among diverse user groups.

This paper conducts robustness analysis across various environmental scenarios and personalized user groups, and simultaneously evaluates the model's generalization ability using a leave-one-out validation method. The aim is to test the model's performance on unseen new applications and analyze whether few-shot adjustments can effectively recover performance. Analysis metrics include MAE, RMSE, and Recovery50%. MAE and RMSE measure the magnitude of prediction error; lower values indicate better performance. The baseline represents the model's reference performance when the training set includes the application (APP). Zero-shot MAE and RMSE represent the model's performance when directly predicting on unseen applications. 10-shot and 50-shot MAE and RMSE represent the performance after fine-tuning with 10 and 50 labeled samples, respectively, on new applications. $\Delta\text{MAE}(\%)$ represents the percentage decrease in zero-shot prediction performance relative to the baseline. Recovery50(%) represents the recovery ratio of the 50-shot performance after fine-tuning relative to the Zero-shot and Baseline performance ($\text{Recovery50}(\%) = (\text{Zero-shot MAE} - 50\text{-shot MAE}) / (\text{Zero-shot MAE} - \text{Baseline MAE}) \times 100\%$). The baseline is not the result of independent training on each app, but rather a unified cross-application training baseline. In the leave-one-app-out setting, the Baseline MAE corresponds to the overall benchmark performance of the model on the "complete training set containing data from that app," obtained through 10-fold cross-validation. Its value is first averaged across the entire training set, and then used as a reference value for comparison with the corresponding zero-shot and few-shot scenarios. Therefore, the mean of the baseline remains consistent across the three app conditions (0.120), while the differences in standard deviation ($\pm 0.010/0.012/0.013$) arise from the differences in sample composition during cross-validation tradeoffs in different held-out apps. The generalization capability verification results are shown in Table 2.

Table 2. Generalization capability verification results

Held-out APP	Baseline MAE	Zero-shot MAE	$\Delta\text{MAE}(\%)$	10-shot MAE	50-shot MAE	Recovery50(%)	Baseline RMSE	Zero-shot RMSE	10-shot RMSE	50-shot RMSE
App_A (WeChat)	0.120 ± 0.010	0.165 ± 0.014	+37.5%	0.140	0.125	88.9%	0.180 ± 0.020	0.245 ± 0.028	0.205	0.190
				± 0.012	± 0.011				± 0.025	± 0.022
				0.130	0.122				0.195	0.185
App_B (Taobao)	0.120 ± 0.012	0.148 ± 0.013	+23.3%	± 0.012	± 0.011	92.9%	0.180 ± 0.020	0.222 ± 0.025	± 0.023	± 0.020
				0.136	0.122				0.200	0.185
				± 0.013	± 0.011				± 0.024	± 0.021
App_C (Mapping)	0.120 ± 0.013	0.160 ± 0.015	+33.3%			95.0%	0.180 ± 0.020	0.240 ± 0.027		

As shown in Table 2, on App_A (WeChat), the Zero-shot MAE is 0.165 ± 0.014 , an increase of 37.5% compared to the Baseline of 0.120 ± 0.010 . The RMSE also increases from 0.180 ± 0.020 to 0.245 ± 0.028 , indicating a significant decrease in model performance on unseen applications. After 10-shot fine-tuning, the MAE decreases to 0.140 ± 0.012 , and 50-shot fine-tuning further restores it to 0.125 ± 0.011 , corresponding to a Recovery50 of 88.9%, indicating that a small number of labeled samples can significantly improve the model's ability to adapt to new apps. Similarly, App_B (Taobao) has a Zero-shot MAE of 0.148 ± 0.013 , a ΔMAE of 23.3%, and a 50-shot MAE that recovers to 0.122 ± 0.011 (Recovery50 = 92.9%); App_C (Mapping) has a Zero-shot MAE of 0.160 ± 0.015 , a ΔMAE of 33.3%, and a 50-shot MAE that recovers to 0.122 ± 0.011 (Recovery50 = 95.0%). Overall, the Zero-shot performance degradation ranges from 23.3% to 37.5%, but minor adjustments

allow for rapid recovery to near-baseline levels, indicating that the model has good cross-application generalization potential. The main reason for this performance difference lies in the significant differences between different apps in terms of interface layout, interaction patterns, and user behavior. This leads to increased error when the model directly predicts on unseen apps. Minor fine-tuning allows the model to capture the specific patterns of new apps, adjusting weights to adapt to new input features, thereby improving prediction accuracy. This also illustrates that in practical deployments, combining a small amount of labeled data for rapid adaptation is an important strategy for improving the model's generalization ability.

4.4 Temporal Load Recognition Capability

To evaluate the ability of each model in detecting continuous load segments, this paper uses four metrics for comprehensive evaluation: Precision, Recall, F1 score, and detection latency. These four metrics can comprehensively reflect the model's accuracy, recall, overall performance, and real-time performance in high load recognition. The temporal load recognition results are shown in Figure 7. The horizontal axis represents the model type, including the multimodal Transformer, CNN-LSTM, Informer, BiLSTM, TCN, and XGBoost used in this paper; the vertical axis represents the value range of each metric (Precision/Recall/F1 is 0-1, and detection latency is in seconds). In each violin plot, the filled area shows the distribution of the model with 40 samples; the center dot represents the sample mean; the error bar represents the mean $\pm$ standard deviation.

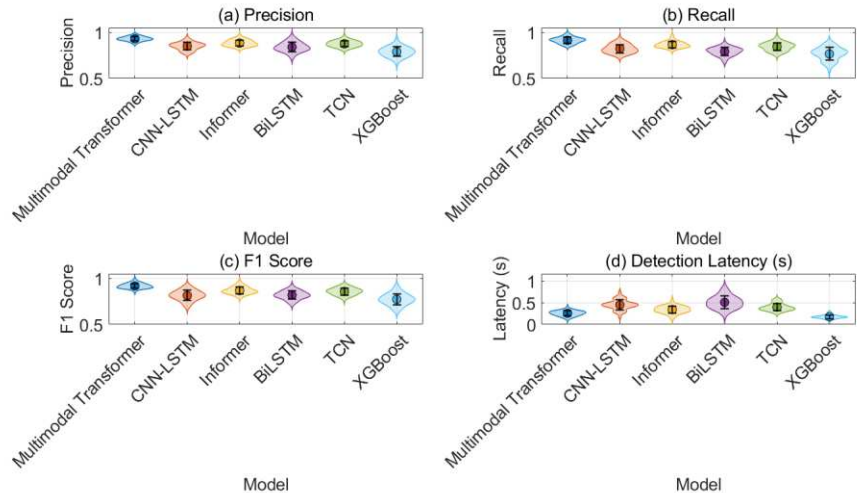


Figure 7. Results of time-series load identification
Figure 7(a) Precision; Figure 7(b) Recall; Figure 7(c) F1 score; Figure 7(d) Detection latency

As shown in Figure 7(a) regarding Precision, the multimodal Transformer exhibits the highest precision with a mean of 0.93±0.02, followed by CNN-LSTM at 0.85±0.04 and XGBoost at 0.79±0.05. Figure 7(b) shows Recall, reflecting recall capability. The multimodal Transformer again performs best with a mean of 0.91±0.03, while BiLSTM is at 0.79±0.04 and XGBoost at 0.77±0.07. Figure 7(c) shows the F1 score, which combines Precision and Recall. The multimodal Transformer has a mean of 0.92±0.03; Informer and TCN also maintain high levels (0.87±0.04 to 0.86±0.04); XGBoost has the lowest score at 0.78±0.06. As shown in Figure 7(d), regarding detection latency, XGBoost has the fastest response time of 0.17±0.04 s; the multimodal Transformer has a latency of 0.26±0.06 s; BiLSTM has the highest latency of 0.51±0.15 s; the other models range from 0.34 to 0.45 s. Overall, the multimodal Transformer outperforms traditional deep learning models in precision, recall, and F1 score, while XGBoost, despite having the shortest latency, has relatively low overall performance.

4.5 Feature Importance and Interpretability

To understand the model's dependence on different features and the reasons behind the decisions, feature importance and interpretability analysis are conducted. This paper uses the SHAP method to quantify the contribution of each input feature to the model output, comparing the influence of each feature on the overall task and its local effect on the prediction of a single sample. The results of the feature importance and interpretability analysis are shown in Figure 8. Figure 8(a) shows the global SHAP ranking bar chart, where the horizontal axis represents the average absolute SHAP value of each feature (Mean |SHAP|), measuring the overall contribution of the feature to the model's prediction; the vertical axis represents the feature name. Figure 8(b) is a local interpretation diagram of SHAP, showing the feature contribution of a single sample, where the horizontal axis represents the additive contribution of the feature to the prediction score (positive values increase the prediction complexity score, and negative values decrease the prediction score), and the vertical axis represents the features sorted by global importance. Each horizontal rectangle represents the direction and magnitude of a feature's contribution; the baseline (base value) represents the expected predicted value; the final predicted value (pred = 0.54) is obtained by summing the contributions of each feature.

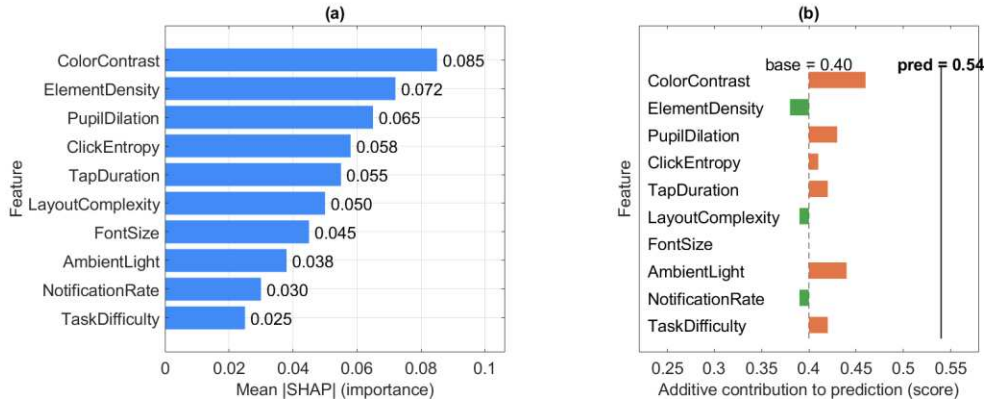


Figure 8(a) SHAP global feature importance; Figure 8(b) SHAP local interpretation map (single sample)

Figure 8. Feature importance and interpretability analysis results

As shown in Figure 8, the global analysis in Figure 8(a) reveals that color contrast (0.085), element density (0.072), and pupil dilation (0.065) are the top three most important features, contributing the most to the prediction, while task difficulty (0.025) and notification frequency (0.030) contribute less. The local analysis in Figure 8(b) shows that for this single sample, color contrast (0.06), ambient light (0.04), and pupil dilation (0.03) positively drive the model prediction, while element density (-0.02) and layout complexity (-0.01) negatively affect the prediction, resulting in a final predicted value of 0.54. By comparing the global and local plots, it can be seen that some features with high global importance may also play a significant role in specific samples, while some features with smaller contributions may exhibit local enhancement or suppression effects.

The main reasons for these data discrepancies stem from the model's learned feature interactions and task scenario sensitivity. The significant positive impact of color contrast on high-load tasks indicates that the model is more likely to identify high-load states when users have high visual attention or high interface information density. Conversely, element density contributes negatively in this sample because the relatively simple interface layout inhibits its prediction. Furthermore, ambient light shows a positive contribution in local samples, indicating that changes in ambient light intensity affect user operation characteristics and are thus captured by the model. Overall, the global SHAP value helps identify key influencing factors, while the local SHAP plot reveals the dynamic effects of sample-level features, providing a basis for developing actionable improvement suggestions, such as optimizing interface layout, adjusting interface color contrast, or monitoring ambient light changes to reduce the risk of high load.

4.6 Ablation Experiment

To quantify the contribution of different modalities to the model's predictive performance, an ablation experiment is designed. By removing the visual, behavioral, or physiological modalities respectively, the changes in the model's predictive performance under the remaining modalities are compared, clarifying the role of each modality in the overall system. Evaluation metrics include MAE, RMSE, and the Pearson correlation coefficient r , which comprehensively measure the deviation and linear correlation between the model's predicted values and the actual values. The ablation experiment results are shown in Table 3. In Table 3, " $\checkmark$ " indicates that the modality is retained, and "-" indicates that the modality is removed; MAE, RMSE, and r are given as mean $\pm$ standard deviation, and Δ MAE, Δ RMSE, and Δr are the percentage or absolute difference in performance relative to the complete model, used to quantify the performance degradation when each modality is missing.

Table 3. Ablation experiment results

Configuration			Indicators					
Visual	Behavior	Physio	MAE (mean $\pm$ std)	Δ MAE (%)	RMSE (mean $\pm$ std)	Δ RMSE (%)	Pearson r (mean $\pm$ std)	Δr (abs)
$\checkmark$	$\checkmark$	$\checkmark$	0.120 $\pm$ 0.010	—	0.180 $\pm$ 0.020	—	0.860 $\pm$ 0.030	—
-	$\checkmark$	$\checkmark$	0.158 $\pm$ 0.012	31.7%	0.235 $\pm$ 0.025	30.6%	0.785 $\pm$ 0.035	-0.075
$\checkmark$	-	$\checkmark$	0.140 $\pm$ 0.011	16.7%	0.205 $\pm$ 0.023	13.9%	0.820 $\pm$ 0.032	-0.040
$\checkmark$	$\checkmark$	-	0.135 $\pm$ 0.011	12.5%	0.198 $\pm$ 0.022	10.0%	0.830 $\pm$ 0.031	-0.030
-	-	$\checkmark$	0.185 $\pm$ 0.015	54.2%	0.265 $\pm$ 0.030	47.2%	0.740 $\pm$ 0.040	-0.120
-	$\checkmark$	-	0.172 $\pm$	43.3%	0.250 $\pm$	38.9%	0.760 $\pm$	-0.100

$\sqrt{\quad}$	-	-	0.014 0.155 $\pm$ 0.013	29.2%	0.028 0.230 $\pm$ 0.026	27.8%	0.038 0.780 $\pm$ 0.037	-0.080
----------------	---	---	-------------------------------	-------	-------------------------------	-------	-------------------------------	--------

As shown in Table 3, the removal of different modalities leads to a significant decrease in prediction performance. The complete model (Visual+Behavior+Physio) performs best, with a MAE of 0.120 ± 0.010 , an RMSE of 0.180 ± 0.020 , and a Pearson r of 0.860 ± 0.030 . When the visual modality is removed, the MAE increases to 0.158 ± 0.012 (an increase of 31.7%); the RMSE increases to 0.235 ± 0.025 (an increase of 30.6%); the r decreases to 0.785 (a decrease of 0.075), indicating that visual information contributes the most to the prediction. When behavioral or physiological modalities are removed, MAE increases to 0.140 ± 0.011 (16.7%) and 0.135 ± 0.011 (12.5%), respectively; RMSE increases to 0.205 ± 0.023 (13.9%) and 0.198 ± 0.022 (10.0%), respectively; r decreases by 0.040 and 0.030, respectively, indicating that behavioral and physiological information also makes significant contributions. Further removal of two modalities (retaining only a single modality) results in a more significant performance decline. For example, when only the physiological modality is retained, the MAE is 0.185 ± 0.015 (an increase of 54.2%); when only the behavioral modality is retained, the MAE is 0.172 ± 0.014 (an increase of 43.3%); when only the visual modality is retained, the MAE is 0.155 ± 0.013 (an increase of 29.2%), indicating that multimodal fusion has a significant complementary effect.

4.7 Real-time Performance and Engineering Feasibility

The experiments evaluate the real-time performance and engineering feasibility of the model. The main purpose is to verify the deployment feasibility of the proposed multimodal model on actual mobile devices, and to assess the impact of quantization optimization (INT8 quantization) on inference latency, memory usage, and power consumption. The evaluation metrics used include: Latency (single inference latency, in ms), Memory (model runtime memory usage, in MB), and Power (inference power consumption, in mW). The real-time performance and engineering feasibility results are shown in Table 4. Table 4 shows the performance of six models on mid-range and low-end mobile phones, distinguishing between the performance of the original FP32 (Single-Precision Floating-Point) model and the INT8 quantized model. All metrics are given as mean $\pm$ std for easy observation of fluctuations and optimization effects.

Table 4. Real-time performance and engineering feasibility results

Model	Latency FP32	Latency INT8	Latency FP32	Latency INT8	Mem FP32	Mem INT8	Power FP32	Power INT8	Power FP32	Power INT8
	Mid	Mid	Low	Low	FP32	INT8	Mid	Mid	Low	Low
	(ms)	(ms)	(ms)	(ms)	(MB)	(MB)	(mW)	(mW)	(mW)	(mW)
Multimodal Transformer	72 ± 7	50 ± 5	165 ± 14	115 ± 11	100 ± 10	65 ± 8	650 ± 60	400 ± 35	1000 ± 90	650 ± 60
CNN-LSTM	65 ± 6	45 ± 5	150 ± 12	100 ± 10	120 ± 12	80 ± 9	700 ± 70	450 ± 40	1100 ± 100	700 ± 70
Informer	120 ± 10	80 ± 8	280 ± 20	200 ± 15	180 ± 15	110 ± 10	1200 ± 100	750 ± 60	1800 ± 150	1300 ± 120
BiLSTM	100 ± 9	68 ± 7	230 ± 18	160 ± 14	140 ± 13	90 ± 9	1000 ± 90	650 ± 55	1500 ± 130	1000 ± 95
TCN	60 ± 5	40 ± 4	140 ± 12	95 ± 9	90 ± 9	60 ± 7	600 ± 55	380 ± 35	950 ± 85	600 ± 55
XGBoost	12 ± 2	9 ± 1	30 ± 4	22 ± 3	25 ± 4	15 ± 3	150 ± 15	100 ± 10	250 ± 25	150 ± 15

As shown in Table 4, the real-time performance of different models varies significantly. Taking Multimodal Transformer as an example, the inference latency on mid-range devices using FP32 is 72 ± 7 ms, which can be reduced to 50 ± 5 ms after INT8 quantization. Simultaneously, memory usage decreases from 100 ± 10 MB to 65 ± 8 MB, and power consumption decreases from 650 ± 60 mW to 400 ± 35 mW. On low-end devices, the inference latency using FP32 is 165 ± 14 ms, which is reduced to 115 ± 11 ms after INT8 optimization. In contrast, XGBoost has the lowest inference latency (only 12 ± 2 ms on mid-range devices using FP32), as well as the lowest memory usage and power consumption, but its model capabilities are limited. Deep models like Informer exhibit high latency on low-end devices (280 ± 20 ms for FP32), but quantization significantly improves performance (200 ± 15 ms), indicating that quantization effectively alleviates inference bottlenecks.

These differences are primarily related to model structural complexity, parameter size, and computational patterns. Deep multimodal models (such as Multimodal Transformer and Informer) integrate visual, behavioral, and physiological multimodal features, involving numerous matrix operations and recurrent/self-attention structures, resulting in high inference latency and memory consumption for the original FP32 model. INT8 quantization reduces data representation bit width, decreases multiply-accumulate operation overhead, and lowers memory access pressure, thereby improving inference speed and reducing power consumption. In contrast, lightweight models (such as XGBoost and TCN) have fewer parameters and simpler computation, naturally offering advantages for real-time deployment, but may have slightly lower accuracy. In conclusion, when

878 deploying deep multimodal models on mobile devices, model quantization and structural optimization are key means to ensure
879 real-time performance and energy efficiency.

880 4.8 Ablation Experiments for GMU Selection

882 Following the same data partitioning and training/validation process as described in the main paper, a set of comparative
883 experiments on fusion methods was specifically conducted. The experiment only changed the strategy at the fusion module (the
884 rest of the network structure, Transformer hyperparameters, and training hyperparameters remained completely identical),
885 comparing: ① Gated Multimodal Unit (GMU, used in this paper); ② Direct concatenation followed by a single-layer fully
886 connected layer (Concat+FC); ③ Concatenation followed by two-layer MLP (Concat+MLP); ④ Cross-modal Attention
887 (Cross-modal Attn); ⑤ Simple Modal Mean (Avg). All configurations were repeatedly trained on 5 random seeds, and the mean $\pm$
888 standard deviation was reported. Evaluation metrics included MAE/RMSE/Pearson r (regression) and F1 (classification) for
889 high-load segment detection. This design ensures that differences only arise from the fusion operator, thus directly reflecting the
890 contribution of GMU to stability and task-related performance compared to "simple concat". The ablation results for GMU selection
891 are shown in Table 5.

892 Table 5 Ablation Results for GMU Selection

Fusion Strategies	MAE	RMSE	Pearson r	F1 (High Load)
GMU (This paper)	0.120 ± 0.010	0.180 ± 0.020	0.860 ± 0.030	0.92 ± 0.03
Concat + FC (1 layer)	0.130 ± 0.011	0.195 ± 0.022	0.838 ± 0.034	0.906 ± 0.030
Concat + MLP (2 layers)	0.125 ± 0.010	0.190 ± 0.021	0.845 ± 0.032	0.909 ± 0.028
Cross-modal Attention	0.119 ± 0.010	0.178 ± 0.019	0.862 ± 0.030	0.919 ± 0.026
Avg (Simple Mean)	0.142 ± 0.012	0.205 ± 0.023	0.824 ± 0.036	0.898 ± 0.032

893 Table 5 shows the differences in performance of various fusion strategies in terms of prediction complexity and high-load
894 detection tasks. GMU (the method presented in this paper) demonstrates better stability and comprehensive capabilities in
895 regression metrics, with MAE of 0.120 ± 0.010 , RMSE of 0.180 ± 0.020 , Pearson r of 0.860 ± 0.030 , and F1 score of 0.92 ± 0.03 for
896 high-load segment detection. In comparison, the MAE of a single-layer fully connected layer (Concat+FC) after direct
897 concatenation improved to 0.130 ± 0.011 , RMSE was 0.195 ± 0.022 , Pearson r decreased to 0.838, and F1 score decreased to 0.906,
898 showing a significant performance decline. The two-layer MLP (Concat+MLP) slightly outperformed the single-layer FC in MAE
899 (0.125) and RMSE (0.190), but was still lower than GMU, with an F1 score of 0.909. Cross-modal attention (Cross-modal Attn)
900 slightly outperformed GMU in regression metrics (MAE 0.119, RMSE 0.178, r 0.862) with an F1 score of 0.919, but had higher
901 training and inference costs. The simple means (Avg) strategy performed the worst, with an MAE of 0.142, RMSE of 0.205, r of
902 0.824, and F1 score of 0.898. Overall, the GMU maintains high accuracy while possessing a simple structure and moderate
903 computational cost. Compared to simple concat or means strategies, it can more stably fuse multimodal information and improve
904 task-related performance. Furthermore, it significantly reduces complexity and training cost while maintaining performance
905 comparable to Cross-modal Attention, validating the rationality and effectiveness of its selection.

906 4.9 Statistical Significance Test

907 To investigate the significance of the predictive advantage of the method presented in this paper, we calculated the
908 individual-level MAE (mean average of the errors for each participant across all sessions/windows) for each model on the test set
909 for each participant ($n = 120$), and performed paired statistical tests on these 120 paired samples. First, a normality test (Shapiro–
910 Wilk) was performed on each group of differences; if normal, a paired t-test was used; otherwise, a Wilcoxon signed-rank test
911 was used. Bonferroni correction ($\alpha_{adj} = 0.05/5$) was applied for multiple comparisons (pairwise comparisons between the
912 Multimodal Transformer and the other 5 models). Simultaneously, the 95% confidence interval (based on the t-distribution of
913 paired differences) and paired Cohen's d (standardized using the standard deviation of paired differences) for each pair of
914 comparisons were calculated, and the overall test results of repeated measures ANOVA and a partial η^2 were reported as the
915 overall effect size. The results of the paired significance test are shown in Table 6.

916 Table 6 Results of paired significance test

Comparison (Multimodal Transformer vs)	MAE mean $\pm$ std (model)	Δ MAE (Transformer - Comparison)	95% CI (Δ)	t (df=119) / or W	p (Bonferroni correction)	Cohen's d (pairing)
(Popular ANOVA) Repeated-measures ANOVA	—	—	$F(5,595)=85.4,$ $p<0.001$	—	—	partial $\eta^2 =$ 0.42
CNN-LSTM	0.18 ± 0.015	-0.060	$[-0.068,$ $-0.052]$	$t = -12.5$	$p_{adj} < 0.001$	$d = 1.15$
Informer	0.16 ± 0.012	-0.040	$[-0.047,$ $-0.033]$	$t = -9.2$	$p_{adj} < 0.001$	$d = 0.85$
BiLSTM	0.20 ± 0.018	-0.080	$[-0.088,$ $-0.072]$	$t = -14.0$	$p_{adj} < 0.001$	$d = 1.30$

TCN	0.17 ± 0.013	-0.050	$[-0.057, -0.043]$	$t = -10.8$	$p_{\text{adj}} < 0.001$	$d = 0.98$
XGBoost	0.22 ± 0.02	-0.100	$[-0.108, -0.092]$	$t = -16.8$	$p_{\text{adj}} < 0.001$	$d = 1.55$

Table 6 clearly shows that the Multimodal Transformer significantly outperforms the other five comparative models in terms of MAE, with a mean error of 0.12. Compared to CNN-LSTM (0.18 ± 0.015), the ΔMAE is -0.060 , with a 95% confidence interval of $[-0.068, -0.052]$. The paired t-test is significant ($t = -12.5$, $p_{\text{adj}} < 0.001$), and Cohen's $d = 1.15$, indicating a large effect size. Compared to Informer (0.16 ± 0.012), the ΔMAE is -0.040 , with a 95% CI of $[-0.047, -0.033]$, $t = -9.2$, $p_{\text{adj}} < 0.001$, and $d = 0.85$, indicating a moderately large effect size. Compared to BiLSTM (0.20 ± 0.018), the ΔMAE is... The difference was more significant with $\Delta\text{MAE} = -0.080$, $t = -14.0$, $p_{\text{adj}} < 0.001$, $d = 1.30$; compared with TCN (0.17 ± 0.013), $\Delta\text{MAE} = -0.050$, $t = -10.8$, $p_{\text{adj}} < 0.001$, $d = 0.98$; compared with XGBoost (0.22 ± 0.02), $\Delta\text{MAE} = -0.100$, $t = -16.8$, $p_{\text{adj}} < 0.001$, $d = 1.55$. Repeated measures ANOVA results showed a significant overall difference between models ($F(5,595)=85.4$, $p < 0.001$, partial $\eta^2 = 0.42$), indicating that model type has a large effect on MAE. The multimodal Transformer not only has the lowest mean, but also exhibits significant differences and a large effect size, demonstrating that its advantage in predicting the complexity of mobile application interfaces is both statistically significant and practically meaningful.

4.10 External Gold Standard Validation of Interface Objective Complexity

The experiment selected 300 screenshots from three applications (100 from each application, covering different task stages and dynamic scenarios) and constructed the following validation process: (1) Organize 10 senior UI/UX design experts to score each screenshot on "interface objective complexity (task-independent)" with a score of 1–7 (take the mean as the expert score) and calculate the inter-expert ICC evaluation consistency; (2) Collect perceived complexity scores through crowdsourcing ($n=200$, non-professional users) to obtain the public perception benchmark; (3) Register the two sets of external benchmarks with the interface objective complexity index (OCI, normalized to 0–1) proposed in this paper (linear minimum-maximum normalization), calculate the Pearson correlation coefficient, normalized RMSE and significance test to test the external validity and consistency of OCI. All ratings were conducted at the same display resolution. Expert and crowdsourced ratings were averaged after removing outliers to reduce noise. Statistical significance was determined using a two-sided test ($\alpha=0.05$). The external gold standard validation results are shown in Table 7.

Table 7 External Gold Standard Validation Results

Benchmarks (Explanation)	Baseline sample size	Pearson r of OCI	Normalized RMSE (0–1)	Statistical significance (p)	Notes
Expert Panel (10 UI/UX experts, mean rating)	300 (experts $n=10$)	0.80	0.09	< 0.001	Inter-expert ICC(2,k) = 0.86 (good consensus)
Crowdsourced Perception (200 non-professional raters, mean rating)	300 (per aggregate $n \approx 200$)	0.74	0.11	< 0.001	Extrema removal and median/mean aggregation

Table 7 shows that the Objective Interface Complexity Index (OCI) proposed in this paper exhibits strong and significant consistency with both external gold standards. The Pearson correlation coefficient between OCI and UI/UX expert panel scores reaches $r = 0.80$, with a normalized RMSE of 0.09 and a significance level of $p < 0.001$. Meanwhile, the inter-expert consistency ICC(2,k) = 0.86, indicating that expert evaluation has good reliability and also indirectly verifying OCI's effective ability to characterize "task-independent interface complexity." Compared with the crowdsourced perception benchmark of non-professional users, OCI still maintains a high correlation ($r = 0.74$, normalized RMSE = 0.11, $p < 0.001$), indicating that this indicator is not only highly consistent with professional design cognition but also reflects the general public's overall perception of interface complexity. The difference in correlation and error levels between the two benchmarks further demonstrates the stability advantage of professional assessment compared to subjective perception, but the overall results collectively support that OCI possesses good external validity and cross-population consistency.

4.11 Validation of Dynamic Interface Features (Optical Flow/Motion Energy)

This paper designs a dedicated validation experiment for dynamic visual features. Ten test folds with balanced coverage of three applications were randomly selected from the original experimental data (using the same data partitioning and preprocessing pipeline as the main experiment). Several configurations were compared under the same training/validation configuration (batch size, epochs, optimizer, early stopping rule, and quantization strategy): ① Complete model (including static visual features + optical flow/motion energy + behavior + physiology); ② Model with dynamic features removed (i.e., optical flow/motion energy is removed from the visual feature set, while the rest remains unchanged); ③ Static vision only (excluding optical flow); ④ Static vision + dynamic (visual modality only, comparing whether simply adding dynamics brings gain); ⑤ Dynamic features input alone (motion energy only). All models were repeatedly trained on the same 10 random seeds and reported mean $\pm$ standard deviation. Simultaneously, SHAP settings (k-means background $k=20$, 500 samples, window-level interpretation) were used to quantify the

average of "optical flow/motion energy". The SHAP value is used to measure its interpretive contribution. The verification results of dynamic interface features are shown in Table 8.

Table 8 Verification Results of Dynamic Interface Features

Configuration	MAE (mean $\pm$ std)	RMSE (mean $\pm$ std)	Pearson r (mean $\pm$ std)	Δ MAE vs Full (%)	Mean SHAP (dynamic)
Full (Vis[static+dyn] + Beh + Phys)	0.120 $\pm$ 0.010	0.180 $\pm$ 0.020	0.860 $\pm$ 0.030	—	0.018 $\pm$ 0.003
Without dynamic (remove optical-flow / motion energy)	0.126 $\pm$ 0.011	0.187 $\pm$ 0.021	0.847 $\pm$ 0.032	+5.0%	—
Visual only (static, no dynamic; no Beh/Phys)	0.155 $\pm$ 0.013	0.230 $\pm$ 0.026	0.780 $\pm$ 0.037	+29.2%	—
Visual + Dynamic only (no Beh/Phys)	0.148 $\pm$ 0.012	0.220 $\pm$ 0.024	0.795 $\pm$ 0.035	+23.3%	0.014 $\pm$ 0.002
Dynamic only (motion energy alone)	0.290 $\pm$ 0.025	0.360 $\pm$ 0.030	0.45 $\pm$ 0.06	+141.7%	0.012 $\pm$ 0.002

The results in Table 8 clearly demonstrate that dynamic interface features (optical flow/kinetic energy) provide a stable but non-dominant gain in the overall modeling. After introducing dynamic features into the complete model, the MAE reached 0.120 $\pm$ 0.010, the RMSE was 0.180 $\pm$ 0.020, and the Pearson r was 0.860 $\pm$ 0.030. When optical flow/motion energy was removed, the MAE increased to 0.126 $\pm$ 0.011 (+5.0%), the RMSE increased to 0.187 $\pm$ 0.021, and the correlation decreased to 0.847 $\pm$ 0.032, indicating that dynamic features have a quantifiable positive contribution to prediction accuracy under multimodal conditions. In the comparison using only the visual modality, adding dynamic features reduced the MAE from 0.155 $\pm$ 0.013 to 0.148 $\pm$ 0.012 and increased the Pearson r from 0.780 $\pm$ 0.037 to 0.795 $\pm$ 0.035, indicating that even without behavioral and physiological information, dynamic perturbations can still supplement the interaction load cues that are difficult for static vision to capture. In contrast, performance decreased significantly when using only dynamic features (MAE 0.290 $\pm$ 0.025, r 0.45 $\pm$ 0.06), indicating that motion energy itself is insufficient to independently characterize interface complexity and should be used as a supplement to static vision and multimodal features. The SHAP results further support this conclusion. The average |SHAP| of dynamic features in the complete model is 0.018 $\pm$ 0.003, and in the "visual + dynamic" configuration it is 0.014 $\pm$ 0.002. Its contribution level is lower than the core static visual index but is stable and non-zero, which verifies that optical flow/motion energy has a consistent and interpretable auxiliary role in complexity prediction in dynamic interactive scenarios.

5. Conclusions

This paper constructs a multimodal fusion framework based on temporal windows, combining modality-specific projection, gated fusion units, multi-head cross-modal self-attention, and a Transformer encoder to achieve synchronous modeling of visual, behavioral, and physiological signals. It also applies a SHAP-driven factor-to-action interpretable adaptation closed-loop and lightweight online learning (LinUCB), and employs distillation and INT8 quantization at the inference end to balance real-time performance and energy consumption on mobile devices. Experiments demonstrate that the method achieves an RMSE of 0.180 $\pm$ 0.020 in cognitive load regression and an F1 score of 0.92 $\pm$ 0.03 in high-load segment detection. Under dynamic scenarios, the RMSE remains between 0.15 and 0.20, with 0.22 for the elderly and 0.23 for colorblind individuals, indicating good robustness and potential for personalized adaptation. Although the modules used in this paper, such as GMU, multi-head cross-modal self-attention, and Transformer, are all mature technologies, the core innovation lies in the systematic integration of these modules into an engineering-grade solution for mobile application interface complexity analysis. This system-level design not only achieves multimodal noise processing, context-dependent modeling, and online personalized correction, but also provides interpretable closed-loop decision-making capabilities, thus demonstrating significant innovation in engineering methodology. Limitations include the limited coverage of only three test applications and a limited number of participants. Further large-scale field validation is needed to address issues related to actual equipment and sensor noise, long-term adaptation effects, and privacy constraints. Future work can expand application scenarios and sample sizes, as well as research on uncertainty-based adaptive strategies and long-term field deployment.

Acknowledgements: None.

Funding: None.

Data availability: The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

999 **Author Contribution:** Zihui Ni: Conceptual design, research methods, first draft writing, research supervision, project
1000 management. Fan Zhang: Research methods, verification and verification, form analysis, data organization, proofreading and
1001 revision, as well as journal communication. ZainudinBin Siran: Resource acquisition, data organization, first draft writing.
1002 CheeOnn Wong:Investigation and research, graphic optimization. Yuan Zeng: Language proofreading.

1003 **Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

1004 **References**

1005 [1] Ayada W M, Hammad M A E E. Design quality criteria for smartphone applications interface and its impact on user
1006 experience and usability[J]. International Design Journal, 2023, 13(4): 339-354.
1007 [2] Kadaskar H R. Enhancing user experience in mobile application design through gestural interaction: a human-computer
1008 interaction perspective[J]. International Journal of Scientific Research in Modern Science and Technology, 2024, 3(8): 01-06.
1009 [3] Ramdowar H, Khedo K K, Chooramun N. A comprehensive review of mobile user interfaces in mHealth applications for
1010 elderly and the related ageing barriers[J]. Universal Access in the Information Society, 2024, 23(4): 1613-1629.
1011 [4] Liu B, Wang C. Elderly-centric chromatics: unraveling the color preferences and visual needs of the elderly in smart APP
1012 interfaces[J]. International Journal of Human-Computer Interaction, 2025, 41(5): 3527-3536.
1013 [5] Fang M, Yang W, Li H, Pan Y. Enhancing user experience through optimization design method for elderly medication
1014 reminder mobile applications: A QFD-based research approach[J]. Electronics, 2023, 12(13): 2860-2883.
1015 [6] Iqbal M W, Shinan K, Rafique S R S, Alourani A, Ashraf MU, Ab Rahim NZ. Usability and optimization of online
1016 apps in user's context[J]. PeerJ Computer Science, 2024, 10(1): 1-27.
1017 [7] Zhang R, Son O S. A Measurement Model for Visual Complexity in HCI: Focusing on Visual Elements in Mobile GUI
1018 Design[J]. Electronics, 2025, 14(5): 942-960.
1019 [8] Guo F, Chen J, Li M, Lyu W, Zhang J . Effects of visual complexity on user search behavior and satisfaction: an
1020 eye-tracking study of mobile news apps[J]. Universal Access in the Information Society, 2022, 21(4): 795-808.
1021 [9] Ahmad Faudzi M, Che Cob Z, Omar R, Sharudin SA, Ghazali M. Investigating the user interface design frameworks of
1022 current mobile learning applications: A systematic review[J]. Education Sciences, 2023, 13(1): 94-118.
1023 [10] Li X, Zheng C, Pan Z,Huang Z , Niu Y, Wang P, et al. Comparative study on 2D and 3D user interface for eliminating
1024 cognitive loads in augmented reality repetitive tasks[J]. International Journal of Human-Computer Interaction, 2024, 40(23):
1025 8008-8024.
1026 [11] Zhang N, Zhang J, Jiang S, Ge W. The effects of layout order on interface complexity: an eye-tracking study for
1027 dashboard design[J]. Sensors, 2024, 24(18): 5966-5988.
1028 [12] Zhou C, Yuan F, Huang T, Zhang Y, Kaner J. The impact of interface design element features on task performance in
1029 older adults: evidence from eye-tracking and EEG signals[J]. International journal of environmental research and public health,
1030 2022, 19(15): 9251-9274.
1031 [13] Khan A, Khusro S. A mechanism for blind-friendly user interface adaptation of mobile apps: A case study for improving the
1032 user experience of the blind people[J]. Journal of Ambient Intelligence and Humanized Computing, 2022, 13(5): 2841-2871.
1033 [14] Faudzi M A, Cob Z C, Ghazali M, Omar R , Sharudin SA. User interface design in mobile learning applications: Developing
1034 and evaluating a questionnaire for measuring learners' extraneous cognitive load[J]. Heliyon, 2024, 10(18):1-20.
1035 [15] Zayim N, Yildiz H, Yuce Y K. Estimating cognitive load in a mobile personal health record application: A cognitive task
1036 analysis approach[J]. Healthcare Informatics Research, 2023, 29(4): 367-376.
1037 [16] Zhang M, Hou G, Chen Y C. Effects of interface layout design on mobile learning efficiency: a comparison of interface
1038 layouts for mobile learning platform[J]. Library hi tech, 2023, 41(5): 1420-1435.
1039 [17] Suryani M, Sensuse D I, Santoso H B, Aji RF, Hadi S, Suryono RR,et al. An initial user model design for adaptive
1040 interface development in learning management system based on cognitive load[J]. Cognition, Technology & Work, 2024, 26(4):
1041 653-672.
1042 [18] Zhu T, Yang Y. Research on mobile learning platform interface design based on college students' visual attention
1043 characteristics[J]. Plos one, 2023, 18(7): 1-16.
1044 [19] Arana-De las Casas N I, De la Riva-Rodriguez J, Maldonado-Macias A A, Saenz-Zamarron D. Cognitive analyses for
1045 interface design using dual N-back tasks for mental workload (MWL) evaluation[J]. International Journal of Environmental
1046 Research and Public Health, 2023, 20(2): 1184-1203.
1047 [20] Ouwehand K, Lespiau F, Tricot A, Paas F. Cognitive load theory: Emerging trends and innovations[J]. Education Sciences,
1048 2025, 15(4): 458-463.
1049 [21] Evans P, Vansteenkiste M, Parker P, Kingsford-Smith A, Zhou S. Cognitive load theory and its relationships with motivation:
1050 A self-determination theory perspective[J]. Educational Psychology Review, 2024, 36(1): 7.
1051 [22] Zhang R, Son O S. A Measurement Model for Visual Complexity in HCI: Focusing on Visual Elements in Mobile GUI

- Design[J]. Electronics, 2025, 14(5): 942-960.
- [23] Muzaffar J, Mazher N. Web Accessibility Standards: Implementing WCAG Guidelines for Inclusive Design[J]. Pioneer Research Journal of Computing Science, 2024, 1(3): 62-70.
- [24] Indriani R, Adiwijaya R, Jarmawijaya N, Yusuf M, Agoes AS. Applying Transfer Learning ResNet-50 for Tracking and Classification of A Coral Reef in Development The Mobile Application with Scrum Framework[J]. J. Inf. Technol, 2023, 4(1): 24-29.
- [25] Alnuaim A, Zakariah M, Hatamleh W A, Tarazi H, Tripathi V, Amoatey ET. Human-Computer Interaction with Hand Gesture Recognition Using ResNet and MobileNet[J]. Computational Intelligence and Neuroscience, 2022, 2022(1): 1-16.
- [26] Abdulhameed T Z, Yousif S A, Samawi V W, Al-Shaikhli HI. Ss-dbscan: Semi-supervised density-based spatial clustering of applications with noise for meaningful clustering in diverse density data[J]. IEEE Access, 2024, 12(1): 131507-131520.
- [27] Gouveia C, Albuquerque D F, Pinho P, Vieira J. Bio-radar cardiac signal model used for HRV assessment and evaluation using adaptive filtering[J]. IEEE Transactions on Instrumentation and Measurement, 2022, 71(1): 1-10.
- [28] Hernandez R, Roll S C, Jin H, Schneider S, Pyatak EA. Validation of the National Aeronautics and Space Administration Task Load Index (NASA-TLX) adapted for the whole day repeated measures context[J]. Ergonomics, 2022, 65(7): 960-975.
- [29] Suzuki Y, Wild F, Scanlon E. Measuring cognitive load in augmented reality with physiological methods: A systematic review[J]. Journal of Computer Assisted Learning, 2024, 40(2): 375-393.
- [30] Wang E, Yu Q, Chen Y, Slamu W, Luo X. Multi-modal knowledge graphs representation learning via multi-headed self-attention[J]. Information Fusion, 2022, 88(1): 78-85.
- [31] Jun W, Tianliang Z, Jiahui Z, Tianyi L, Chunzhi W. Hierarchical multiples self-attention mechanism for multi-modal analysis[J]. Multimedia Systems, 2023, 29(6): 3599-3608.
- [32] Fang X, Yang H, Ding D, Gao W, Zhang L, Wang Y, et al. Enhancing app usage prediction accuracy with gcn-transformer model and meta-path context[J]. IEEE Access, 2024, 12(1): 53031-53044.
- [33] Gwak M, Cha J, Yoon H, Kang D, An D. Lightweight Transformer Model for Mobile Application Classification[J]. Sensors, 2024, 24(2): 564-577.
- [34] Wen S, Yang L, Xu M, Qiao M, Xu T, Bai L. Saliency prediction on mobile videos: A fixation mapping-based dataset and a transformer approach[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2023, 34(7): 5935-5950.
- [35] Reyad M, Sarhan A M, Arafa M. A modified Adam algorithm for deep neural network optimization[J]. Neural Computing and Applications, 2023, 35(23): 17095-17112.
- [36] Li B, Chen J, Zhu J. Memory efficient optimizers with 4-bit states[J]. Advances in Neural Information Processing Systems, 2023, 36(1): 15136-15171.
- [37] Johnson O V, Xinying C, Khaw K W, Lee MH. ps-CALR: periodic-shift cosine annealing learning rate for deep neural networks[J]. IEEE Access, 2023, 11(1): 139171-139186.
- [38] Zhang C, Shao Y, Sun H, Xing L, Zhao Q, Zhang L. The WuC-Adam algorithm based on joint improvement of Warmup and cosine annealing algorithms[J]. Math. Biosci. Eng, 2024, 21(1): 1270-1285.
- [39] Tirinzoni A, Papini M, Touati A, Lazaric A, Pirotta M. Scalable representation learning in linear contextual bandits with constant regret guarantees[J]. Advances in Neural Information Processing Systems, 2022, 35(1): 2307-2319.
- [40] Parisineni S R A, Pal M. Enhancing trust and interpretability of complex machine learning models using local interpretable model agnostic shap explanations[J]. International Journal of Data Science and Analytics, 2024, 18(4): 457-466.
- [41] Jinad R, Islam A B M, Shashidhar N. Interpretability and transparency of machine learning in file fragment analysis with explainable artificial intelligence[J]. Electronics, 2024, 13(13): 2438-2467.