European Journal of Engineering and Technologies

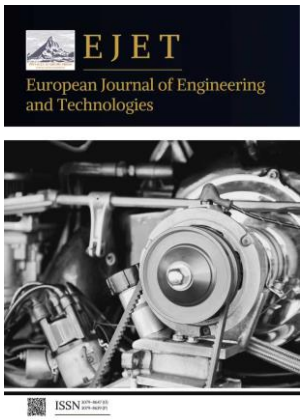
Vol. 1 No. 2 2025

Article **Open Access**

Key Indicators and Data-Driven Analysis Methods for Game Performance Optimization

Xia Hua [1,*]

[1] SMU Guildhall, Southern Methodist University, Texas, 75205, USA
* Correspondence: Xia Hua, SMU Guildhall, Southern Methodist University, Texas, 75205, USA



Received: 15 October 2025
Revised: 30 November 2025
Accepted: 10 December 2025
Published: 13 December 2025

Abstract: With the continuous enhancement of user experience requirements in the gaming industry, performance optimization has become a core component of modern game development. However, traditional experience-driven tuning methods and isolated performance-based optimization strategies are no longer sufficient to address the increasingly heterogeneous device environments, diverse usage scenarios, and the growing complexity of user interaction patterns. As game ecosystems evolve toward higher fidelity, larger scale, and cross-platform deployment, developers urgently need systematic and data-driven mechanisms to identify performance bottlenecks and ensure stable, high-quality gameplay experiences. This paper focuses on the construction of a comprehensive key indicator system for game performance optimization, the design of efficient data collection and preprocessing pipelines, and the development of data-driven performance tuning strategies. By integrating multi-dimensional data sources-including frame rate stability, resource consumption metrics, latency information, and user-perceived smoothness-the study establishes a new comprehensive discriminant model that evaluates performance by jointly considering technical indicators and user experience indicators. This dual-dimension assessment enables a more balanced and accurate understanding of performance quality. Furthermore, the paper proposes a complete empirical workflow for cross-platform performance improvement. By incorporating visualization-based diagnostic tools, real-time feedback mechanisms, and adaptive optimization loops, the framework supports rapid identification of performance issues, iterative refinement, and systematic optimization decisions. The approach not only improves performance tuning efficiency but also enhances the scalability and transparency of the process, providing actionable guidance for optimizing game performance across devices and platforms.

Keywords: game performance optimization; key indicators; data-driven; visualization analysis

1. Introduction

Game performance issues have become a key factor influencing both the competitiveness of the gaming market and user satisfaction. As modern games evolve toward higher visual complexity, richer interactive behaviors, and larger-scale content production, the challenge of maintaining stable performance becomes increasingly significant. Meanwhile, the rapid proliferation of heterogeneous terminal devices-ranging from high-end PCs to mobile phones, tablets, and cloud-gaming platforms-has created diverse hardware environments that traditional single-method optimization approaches can no longer handle smoothly or efficiently. The variability in CPU/GPU capabilities, memory constraints, thermal management, and system configurations demands a more systematic and adaptable optimization framework.

In recent years, data-driven methods have achieved remarkable success across various technological fields, demonstrating strong capabilities in pattern discovery, anomaly detection, and predictive decision-making. When applied to game performance optimization, these methods provide clear advantages in identifying latent bottlenecks, guiding resource allocation, and supporting real-time adaptive tuning. By leveraging large-scale performance data, user behavior signals, and multidimensional metrics, developers can establish a more objective, quantifiable, and reproducible foundation for optimization, transforming the traditional trial-and-error workflow into a more scientific and iterative process.

The primary aim of this research is to summarize the key techniques and methodological paths necessary for effective game performance optimization. This includes establishing a comprehensive and hierarchical evaluation system, exploring a scientific and scalable data management model, and introducing data mining approaches to extract actionable insights from complex performance datasets. Through these efforts, the study seeks to promote refined performance analysis, enhance optimization decision quality, and ultimately provide reliable scientific guidance and operational references for game production teams. It is expected that the proposed framework will support developers in addressing cross-platform performance challenges and improving overall user experience with greater efficiency and precision.

2. Research Background and Development Trends of Game Performance Optimization

2.1. The Development of the Game Industry and the Necessity of Performance Optimization

In recent years, the game industry has developed rapidly and is increasingly evolving towards an integrated and interactive development form that transcends the boundaries of single computer games. With the development of various platforms such as computing devices, PC computers, smart phones, and tablets, graphics processing technology, physical simulation accuracy, and interaction design have become increasingly refined. The increasing number of game products and the complexity of the operating environment require us to have more terminals and more user-friendly terminals to choose from; higher resolution and the pursuit of timely response have become a trend, and higher requirements have been put forward for the performance of games [1].

Under such circumstances, optimizing performance becomes increasingly crucial. If the performance control is not properly managed and controlled, issues such as frame rate fluctuations, download delays, and stutters may occur. This is an extremely unpleasant experience for users and will significantly undermine the product's reputation and commercial benefits. To ensure the smooth operation of the game on multiple hardware platforms and to reduce development costs and release risks, a comprehensive performance optimization plan must be established at the very beginning of the development process, serving as a powerful support throughout the entire lifecycle of the game.

2.2. The Transformation of Performance Optimization Model by Data-Driven Approaches

The traditional methods that rely on developers' experience to adjust and optimize usually cannot cover various performance problems brought by complex systems and their effects are not good with too slow response speed. However, the data-driven approach can obtain important indicator information immediately through automatic data collection and performance monitoring [2]. It provides quantitative references for performance issues and timely responses. The application of this method completely changes the process of performance optimization. Developers can quickly find bottleneck factors by using data-driven tools, design unique optimization schemes, and complete the optimization process transparently and with version comparison. More importantly, the data-driven approach makes performance management no longer a post-event handling

but a process control. Doing so, it speeds up development speed, improves test accuracy, and lays a technological foundation for creating high-quality and highly adaptable products.

3. Analysis of the Key Performance Indicators System for Game Performance

3.1. Classification and Measurement Methods of Objective Technical Indicators

Objective technical indicators form the core data basis for evaluating the running status of a game, mainly including frame rate (FPS), memory usage, CPU and GPU utilization rates, loading time, etc. These indicators can be quantitatively collected through various performance analysis tools such as Profiler, RenderDoc, Perfetto, etc. To ensure the uniformity and comparability of the data, it is necessary to set unified test conditions and fixed data collection rates. However, different types of categories have different responses to each indicator, and the evaluation standards need to be determined based on the positioning of the product [3].

Table 1 presents the definitions and measurement methods of mainstream objective indicators, providing a quantitative basis for subsequent performance optimization. It helps to unify the assessment standards and enhance the efficiency of data comparison.

Table 1. Classification and Explanation of Objective Technical Performance Indicators.

Indicator Name	Measurement Unit	Function Description	Tool Support
Frame Rate	Frames Per second	Measure the smoothness of the picture flow	Unity Profiler,FRAPS
CPU Utilization Rate	%	Reflect the occupation of logical computing resources	Windows Performance Monitor
GPU Occupancy Rate	%	Indicate the usage status of rendering computing resources	RenderDoc,NVIDIA NSight
Memory Usage	MB	Indicate the demand of the game for system memory	Android Studio,Xcode Instruments
Loading Time	Second	Used to measure the efficiency of scene or resource loading	Custom event tracking tool, log analysis

3.2. Modeling and Application of User Experience-Related Indicators

The evaluation of game user experience is based on the subjective feelings of the users themselves. Whether the game can meet the users' own demands through its operation includes issues such as lagging in the game, response speed of the graphics, heating of the mobile phone and smoothness, etc. An information collection system of different dimensions can be established through user behavior statistics, user feedback and device monitoring information collection to ensure a comprehensive understanding of the user experience situation.

Clean and unify the collected data to remove noise and correct the problem of scale imbalance. Then, by combining technical parameters and subjective evaluations with weighted rating methods and machine learning strategies, an experience evaluation model is constructed and used to assess the performance in different situations and on different devices. Finally, the conclusions derived from the above model are applied in various scenarios (such as for version update recommendations, for performance boundary identification, for user profile construction, etc.), making it possible to solve performance experience problems through data analysis-based approaches, and enhancing the scientificity of performance tuning and the personalized adaptability to user experience.

Figure 1 illustrates the complete path of data collection, processing, analysis and feedback for the game user experience. Through multi-dimensional data fusion and modeling methods, it is possible to accurately quantify the experience and provide us with effective and specific guidance for efficiency improvement. The process shown in Figure 1 is from the reflection of user behavior, to the translation of traits, then entering the algorithm model, and finally feedback for strategy adjustment. This is the basic model for forming "optimization oriented towards user experience".

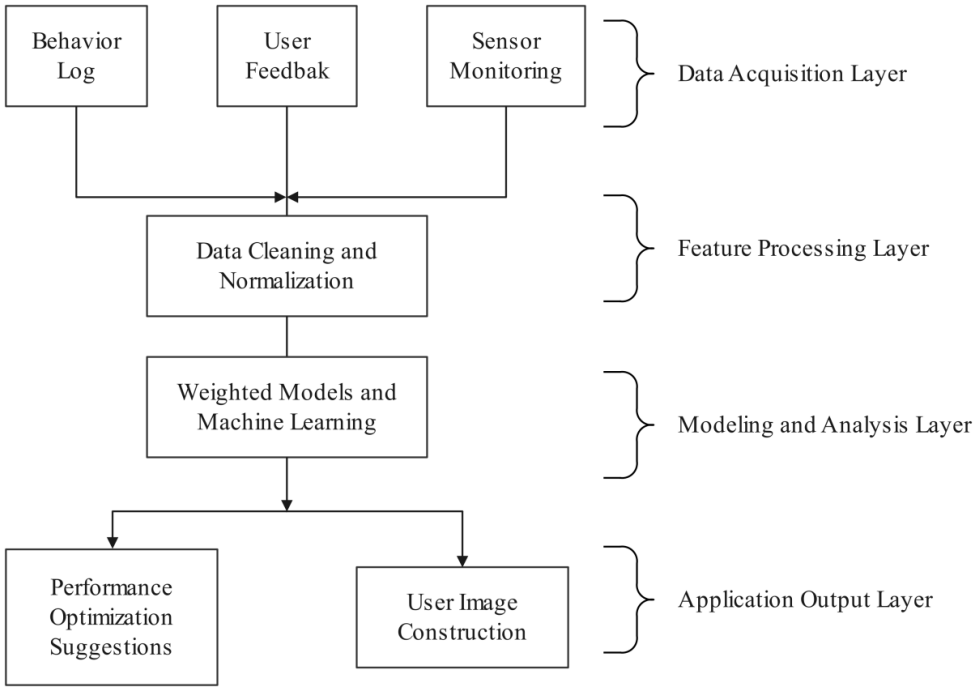


Figure 1. The entire process of collecting, modeling and optimizing feedback related to user experience indicators from start to finish.

3.3. Weight Distribution and Evaluation Mechanism of Multi-Dimensional Indicators

In games, the consideration of various performance indicators mostly involves a compromise assessment of various factors. Moreover, the importance of different game types' weights varies. Through a proper weight setting, a unified performance evaluation index can be better constructed. Commonly, the weights of each indicator in the total performance score can be determined by methods such as AHP (Analytic Hierarchy Process) or entropy weight method, forming a good overall performance evaluation model [4].

Table 2 presents the distribution of importance of different indicators in typical game categories. Of course, for developers, they can adjust the weights according to the product situation, thereby achieving the best effect in terms of resource allocation during performance processing and the satisfaction of user experience.

Table 2. Distribution of Weight for Key Indicators under Different Game Types (Illustrative).

Game Genre	Frame Rate Stability	Memory Control	Response Time Delay	Loading Time	User Experience Rating
Competitive Category	0.30	0.10	0.35	0.05	0.20
Relaxing Puzzle Game Type	0.15	0.20	0.15	0.20	0.30
Open-world Type	0.25	0.25	0.10	0.20	0.20

4. Methods for Performance Data Acquisition, Processing and Visualization

4.1. Design of Performance Tracking Points and Data Collection Architecture

The accurate collection of performance data relies on scientific event-point strategies and stable data transmission architectures. In the design of event points, it is necessary to cover the core performance stages, such as frame rate refresh, memory usage, resource loading, network requests, etc., and take into account the collection rate to meet the requirements of balancing system load and accuracy. We usually establish the architecture by adopting the methods of client-side event points, server-side relay, log aggregation, and asynchronous transmission to minimize the impact on real-time processes. During the data collection process, the performance load within the defined sampling period is defined as:

$$P(t) = \frac{1}{n}\sum_{i=1}^{n} M_i(t) \quad (1)$$

Among them, the average performance load within time t represents the performance indicators (such as CPU occupancy rate) of the i-th collection point, and n is the number of collection points. This formula can calculate the system's average load in real time, serving as the fundamental basis for judging whether the system exhibits periodic stalling or sudden resource bottlenecks, and being used by the subsequent monitoring module.

4.2. Data Cleaning, Feature Extraction and Index Aggregation Strategies

The original performance data often contain noise, missing values and outliers. Therefore, data cleaning and feature extraction are necessary to enhance the efficiency of performance analysis. Data cleaning involves outlier removal, filling of missing values, normalization of time axes, etc. Feature extraction mainly includes the extraction of important performance features such as stability features, volatility features, and extreme frequency features. The fluctuation range of a certain performance indicator (such as FPS) is defined as:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2} \quad (2)$$

Among them, the actual frame rate for each frame is μ, which is the average frame rate within the given period, and σ represents the standard deviation of the frame rate fluctuation. By using the standard deviation, the jitter condition of the picture can be evaluated. The larger the fluctuation value is, the more it indicates that the performance experience is inconsistent. Further investigation is needed to determine whether there are problems with resource allocation or the rendering pipeline.

The aggregation strategy can reduce the dimensionality of multi-dimensional data through window moving average, range evaluation, etc., thereby enhancing the efficiency of comprehensive judgment and providing support for the a priori threshold of performance warning methods.

4.3. Visualization Analysis and Dynamic Monitoring Mechanism of Performance Data

Visual analysis not only enhances the comprehensibility of data, but is also highly beneficial for programmers in quickly identifying performance bottlenecks. Common methods include presenting the variation of frame rate over time in the form of curve charts, and displaying the CPU/GPU usage in the form of heat maps; presenting the sluggish core areas in the form of distribution charts. Therefore, it is necessary to have the capabilities of real-time refreshing, flexible marking, and post-event review record to establish a data loop monitoring system for performance. In real-time charts, to highlight performance anomalies, an anomaly judgment formula can be set:

$$E(t) = \begin{cases} 1, \text{if } |x(t) - \mu| > k\sigma \\ 0, \text{otherwise} \end{cases} \quad (3)$$

Herein, x(t) represents the performance value at the current moment, μ and σ are the historical mean and standard deviation respectively, k is the abnormal amplification

factor, usually taking values between 2 and 3. The abnormal state setting is as follows: once the performance at a certain time point exceeds the predetermined value, it will be marked as an abnormal state. The monitoring system will automatically track logs or conduct automatic analysis, and through automatic anomaly location, improve the efficiency of adjustment and optimization.

5. Data-driven Performance Optimization Strategies

5.1. Optimization Practices for Scene Rendering and Resource Scheduling

As the game environment becomes increasingly complex, a large number of images need to be rendered. If the scheduling is improper, it will lead to lag and frame flickering. To reduce the burden of graphics computing, a dynamic calling mechanism that can dynamically adjust according to the situation must be constructed to reasonably allocate and render. For the optimization of scene rendering, one is the advanced settings in terms of models and textures, and the other is the elimination processing of invisible objects. The LOD (Level of Detail) technology can automatically switch between advanced and primary models when the camera moves, achieving more refined control of graphic burden; the use of occlusion elimination and perspective occlusion is to eliminate the objects that need to be rendered outside the current view range, which can significantly reduce the number of scene calls. In addition, for the presentation of effects such as shadows, lighting, and particle effects, hierarchical rendering control can be carried out using quality levels to achieve a balance between visual performance and performance efficiency.

From the perspective of resource management, the big data-driven model will guide the asynchronous execution of resource loading schemes based on the variation characteristics of frame rates. For instance, it will predictively load a large number of textures or special effect files to avoid loading blockage problems during the game process and promptly release the space occupied by temporarily unused resources during the game. On the basis of applying dynamic hotspot domain analysis technology, it will perform local fine processing and progressive loading of the dense content in some hotspot domains, and improve the overall stability and efficiency of rendering. Based on runtime monitoring information, it can dynamically adjust the priority order of resource loading and compression levels in a timely manner, forming an adaptive resource allocation mechanism that can flexibly respond to hardware capabilities and real-time performance conditions.

5.2. Improvements in Network Latency and Data Transmission Methods

In online games, the influence of network performance on the smoothness of interaction is mainly manifested in key processes such as frame synchronization mechanism, input delay and status update frequency. These processes are directly related to the players' experience. Data-based optimization can conduct real-time monitoring and analysis of data in these processes and provide real-time optimization suggestions for network transmission strategies. Common optimization methods include switching communication protocols, data compression, differential update, and network architecture reconfiguration. The following are the performance comparison results of a multiplayer battle game under different transmission strategies:

The data in Table 3 shows that when using traditional TCP communication, the delay and packet loss rate are relatively high, which affects the smoothness of operations. After switching to UDP, the network delay is reduced. Coupled with differentiated data processing and data compression, the communication efficiency is further optimized. The total data traffic has been reduced by approximately 60%, and the perception evaluation of players has also improved significantly. We established a model for the characteristics of the mobile data and redesigned the packet format. In this restructured sending process,

it can not only reduce the load on the server but also improve the stability of data in high-frequency usage scenarios.

Table 3. Comparative Effect Analysis of Network Transmission Strategies.

Optimization Plan	Average Delay (ms)	Packet Loss Rate (%)	Bandwidth Occupancy (KB/s)	Player Ratings (on a 5-point scale)
Default TCP Communication	105	4.2	520	3.0
Switch to UDP Protocol	68	2.1	390	3.7
UDP + Differential Data + Compression Processing	42	1.0	210	4.5

This process reflects the effect transformation path of data-driven optimization in the field of network transmission: first, identify the impact indicators, then iterate and verify the solutions, and finally form scalable and self-adaptive optimization strategies to meet the real-time performance requirements in different regions and network environments.

5.3. Algorithm Efficiency and Performance Adjustment of Code Structure

Complex algorithm design and inefficient programming implementation are both reasons for performance bottlenecks. They are particularly evident in areas such as character action management, path planning, and physical simulation. The data-driven mode can assist developers in identifying bottleneck parts by using indicators such as function execution time consumption, memory usage size, and the number of calls. It can then carry out targeted architecture optimization.

The optimization strategies mainly include splitting logical modules, algorithm substitution methods, and parallel design. For example, traditional A* path finding can be replaced with Jump Point Search, which can effectively reduce search costs in complex path finding scenarios. For granular objects or collision logic, an octree or spatial partition structure can be constructed to reduce invalid traversals and calculation times. At the same time, using coroutines and object pooling mechanisms can reduce the load on the main thread, and implementing asynchronous events and object recycling can significantly enhance work efficiency.

For systems driven by scripts, if a certain behavior logic is called frequently and involves multiple calculations, the calculation results during the process can be saved and the triggering factors modified. This can reduce the number of calls and improve resource utilization. The ultimate goal is to optimize the algorithm path and improve algorithm performance while maintaining the integrity of the main functions of the system, thereby achieving the optimal performance, robustness, and energy efficiency optimization of the game system.

6. Conclusion

Game performance optimization has become an important aspect for meeting consumer demands and enhancing product competitiveness. This article systematically explores the core indicators system construction, data collection and processing methods, visualization analysis mechanism, and data-driven optimization strategies for performance optimization. On one hand, quantitative factors such as frame rate, latency, and resource occupation are used as indicators to measure user experience while integrating performance. On the other hand, the optimization process is compared and analyzed through data-oriented optimization methods, combining experience and models. However, performance optimization should be flexibly set according to different game

types and application environments when formulating monitoring architectures and optimization strategies, becoming a performance management mechanism that can continuously evolve and return. The intelligent optimization system based on artificial intelligence analysis, edge computing, and multi-platform collaboration will undoubtedly become an important driving force for the development of game technology in the future.

References

1. S. M. Andersen, S. Chen, and R. Miranda, "Significant others and the self," Self and identity, vol. 1, no. 2, pp. 159-168, 2002. doi: 10.1080/152988602317319348
2. R. S. Nickerson, "How we know-and sometimes misjudge-what others know: Imputing one's own knowledge to others," Psychological bulletin, vol. 125, no. 6, p. 737, 1999.
3. L. Tiefer, "Sex is not a natural act & other essays," Westview Press, 2004.
4. M. Sishi, and A. Telukdarie, "Adoption of Data-Driven Automation Techniques to Create Smart Key Performance Indicators for Business Optimization," Applied System Innovation, vol. 8, no. 1, p. 10, 2025. doi: 10.3390/asi8010010