write_up

October 9, 2025

1 How GPU Specifications Influence Gaming Performance

1.0.1 1. Introduction

The relationship between hardware and gaming performance is inherently complex, shaped by multiple factors including hardware specifications, display resolution, graphical settings, and the characteristics of individual games. Within the hardware domain, both the central processing unit (CPU) and the graphics processing unit (GPU) contribute significantly to performance outcomes. However, the GPU plays the dominant role in modern gaming, as it is primarily responsible for rendering complex visuals and executing the parallelized computations required for real-time graphics.

This study isolates the GPU and investigates how its technical specifications influence frames per second (FPS), a common measure of gaming performance. By focusing on GPU-level features, the analysis aims to clarify the specific contribution of graphics hardware while controlling for the broader complexity of full system configurations.

1.0.2 2. Research Question

How do GPU specifications influence gaming performance (FPS), and how does this relationship change across different resolutions and graphics settings?

1.0.3 3. Data and Features

The dataset used in this analysis includes GPU hardware specifications alongside measured FPS values across multiple games, resoultions, and graphical settings. FPS benchmarks were derived from https://www.kaggle.com/datasets/baraazaid/gpus-fps-on-games?resource=download and the hardware specifications were scraped from https://www.techpowerup.com/.

- Hardware Features (continuous variables)
  - Process Size
  - Transistors
  - Density
  - Die_size
  - Base Clock

    - Memory Size
    - Memory Type
    - Memory Bus
    - Bandwidth
    - Shading Units
    - Texture Mapping Units (TMUS)
    - Render Output Units (ROPs)
    - L1_cache
    - L2_cache
    - Directx
    - Thermal Design Power (TDP)
    - Memory Clock
    - Fp32
    - Fp64
    - Pixel Rate
    - Texture Rate
- Categorical Features
    - Architecture (build type)
    - Memory Type
    - Game Title
    - Grapic Setting (low, med, high, ultra)
    - Resolution (e.g. 1920x1080)
- Target Variable (continious)
    - Average Frames Per Second (Hz)

1.0.4 4. Preprocessing and Feature Engineering

1. **Raw Data Sources**
    - **Hardware Specifications Table**
        - Raw data scraped for each unique GPU Name found in the FPS Benchmark dataset
        - Data stored in a dictionary with one entry per GPU and its corresponding specifications
    - **Benchmark FPS Table**
        - Downloaded raw json from Kaggle

2. **Feature Selection and Cleaning per Table**
    - **Hardware Specifications**
        - Removed several unwanted features (kept relavant performance modeling features)
        - Extracted units from all columns and attached to column headers
        - Converted all numeric columns to float
    - **Benchmark FPS Table**
        - Extracted unique GPU Names
        - Included meaningful variables (*GPU_Name*, *Game_Name*, *Avg_FPS*, *Setting*, *Resolution*)
        - GPU Name set as index with one entry for each combination of (*Game_Name* X *Setting* X *Resolution*) with *Avg_FPS* listed
        - ~820 observations for each GPU
3. **Merging Tables**
    - Inner joined hardware specifications table on benchmark table
    - Set GPU Name as the index
4. **Feature Engineering**
    - Kept categorical variables (*architecture*, *memory_type*, *Game_Name*, *Resolution*, *Setting*) as strings
    - Limited *architecture* and *memory type* to the top 5 most common entries with all others grouped together as "other"
    - Categorized separate feature lists for analysis
        - Numerical hardware features
        - Categorical hardware features
        - Software features (*Resolution*, *Setting*)
        - All hardware features
        - All features
    - Applied one-hot encoding for categorical variables
    - Dropped Avg_FPS (outcome variable) from predictors
    - Dropped Game_Name from predictors
        - The focus was to analyze how hardware specifications, *resolution*, and *setting* are correlated with performance

1.0.5 5. Model Training

1. **Train/Test Splitting**
    - The merged dataset was split into training (80%) and test (20%) subsets. StandScalar was applied to ensure balanced representation of variables
2. **Modeling Approaches**
    - Two model families were evaluated:
        - Linear regression as a baseline
        - XGBoost as a non-linear, tree-based model to capture more complex feature interactions
3. **Hyperparameter Optimization**
    - The Linear model was used in its default form
    - Hyperparameters for the XGBoost model (e.g., learning rate, depth, subsampling) were tuned using five-fold randomized cross-validation
4. **Evaluation Metrics**
    - Model performance was assessed on the test set using R^2 and mean squared error (MSE)

- To enhance interpretability, feature importance rankings were generated for both models
- SHAP values were used to quantify each feature's contribution to FPS predictions

1.0.6 6. Analysis

The analysis was structured to evaluate how different subsets of features influenced predictive performance and then view them all together. For each subset, results are reported for both the baseline linear regression model and the XGBoost model, allowing for direct comparison between linear and non-linear approaches.

1. **Hardware Specifications (*Resolution* and *Setting* omitted)**
   - This feature set included all numerical hardware variables as well as categorical descriptors such as architecture and memory_type. When trained on this subset, both models produced similar results, with R^2 values of 0.4132 for linear regression and 0.4241 for XGBoost. These outcomes suggest that hardware specifications alone provide a moderate but incomplete explanation of FPS performance.
   - However, the linear regression and XG Boost models ranked predictors differently. More specifically, the linear regression labeled the categorical predictors as more important whereas the XG Boost dedicated the majority of the importance to *pixel_rate* and *fp32* (as seen in the overall analysis later).

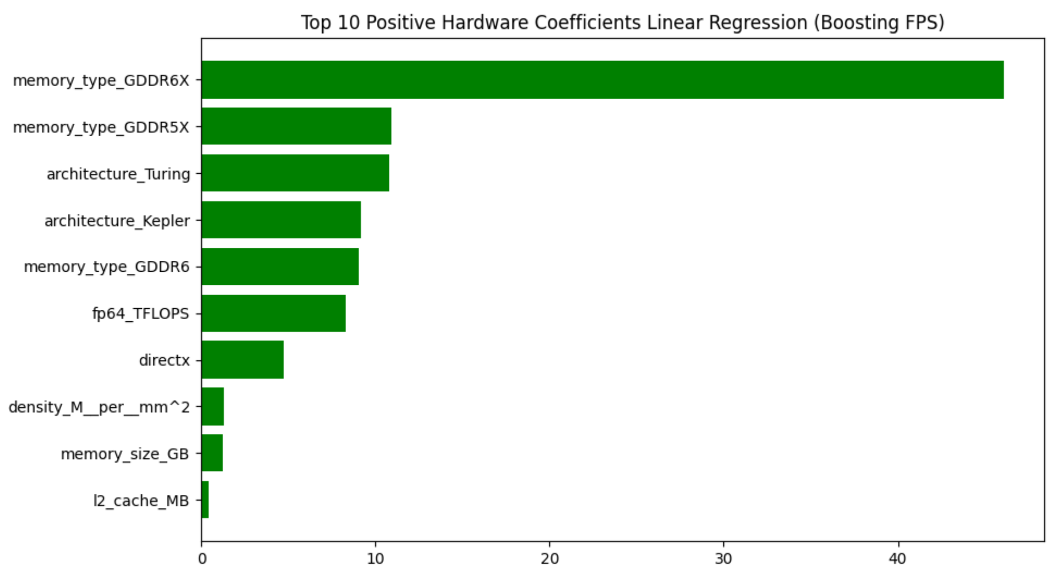

Figure 1. Feature importances (Linear Regression) for hardware-only features, highlighting memory types, architecture types, and fp64 as the most predictive.

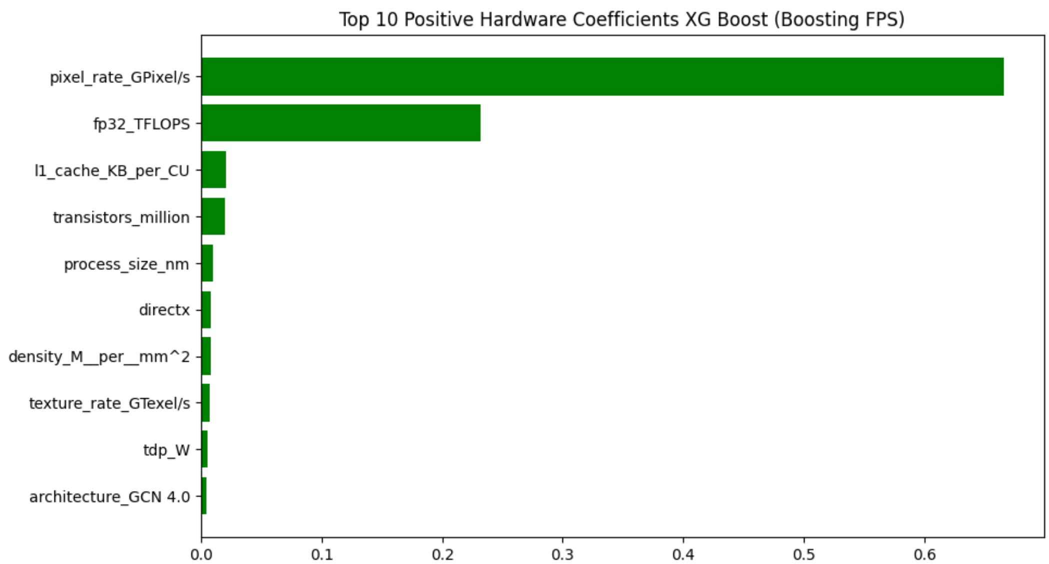

Figure 2. Feature importances (XGBoost) for hardware-only features, highlighting pixel_rate and fp32 as the most predictive.

2. **Game_Name only**

   - Because FPS outcomes are also dependent on the computational demands of individual games, the predictive value of Game_Name was analyzed independently. Using this feature alone, the models achieved R^2 values of 0.1854 (Linear Regression) and 0.1790 (XG Boost), indicating that while game identity contributes to FPS variance, it is insufficient as a standalone predictor. So for the scope of this question, this variable was dropped in the overall analysis.

3. **Overall Analysis (Hardware features + *Resolution* + *Setting*)**

   - The linear regression model was able to capture a substantial portion of the variance in FPS, achieving an R^2 value of 0.6236. This indicates that hardware specifications, resolution, and setting collectively explain over 60% of the observed variability, marking a significant improvement over using hardware or game identity alone.
   - The XGBoost model provided a stronger fit. With default parameters, XGBoost achieved an R^2 of 0.7007, outperforming linear regression by nearly eight percentage points. After hyperparameter tuning, the R^2 improved slightly further to 0.7034, suggesting that most of the performance gains were realized even before tuning.
   - These results highlight two important findings:
     - First, that combining hardware and context-related features substantially improves predictive accuracy.
     - Second, that tree-based models such as XGBoost are more effective than linear models at capturing complex, non-linear relationships in the data. Nonetheless, the relatively small gain from tuning suggests that the model is already approaching the ceiling imposed by the available features, leaving some variance unexplained by hardware and resolution alone.
   - While *Game_Name* alone produced only limited predictive power (R^2 0.18), excluding it from the overall feature set leaves roughly 18% of variance unexplained. This reflects the fact that FPS is highly game-dependent, and omitting game identity inevitably

reduces the model's ability to fully capture performance differences across titles.

4. **Model Interpretability (SHAP analysis)**

    - To better understand the drivers of FPS predictions in the tuned XGBoost model, SHAP (SHapley Additive exPlanations) analysis was conducted.
    - **Global Feature Importance**
        - A bar chart of mean absolute SHAP values (Figure 3) showed that setting, resolution, pixel_rate, fp32 throughput, transistor count, and TDP were the strongest contributors to model predictions. The prominence of setting and resolution is intuitive, given their direct role in FPS performance. More notable are the hardware-related factors—particularly pixel_rate, fp32 throughput, and transistors—which highlight the importance of raw computational power and energy capacity in explaining frame rate variability.

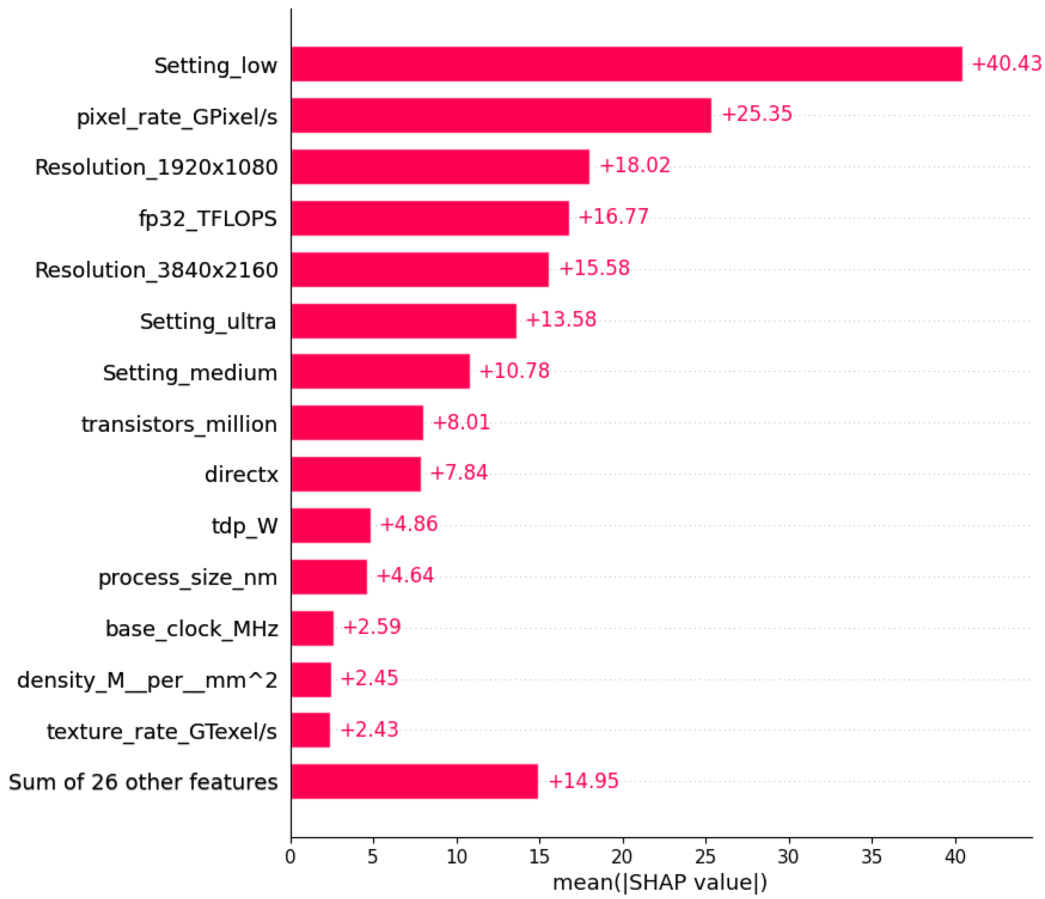

Figure 3. Mean absolute SHAP values for the tuned XGBoost model, showing the strongest global contributors.

    - **Distribution of Contributions**
        - The SHAP beeswarm plot (Figure 4) complements the global ranking by illustrating the spread of feature effects across individual samples. Features such as pixel_rate and fp32 throughput exhibit wide variation, indicating strong context-dependent effects. In contrast, features like TDP show more stable contributions, reinforcing their consistent role in performance prediction.

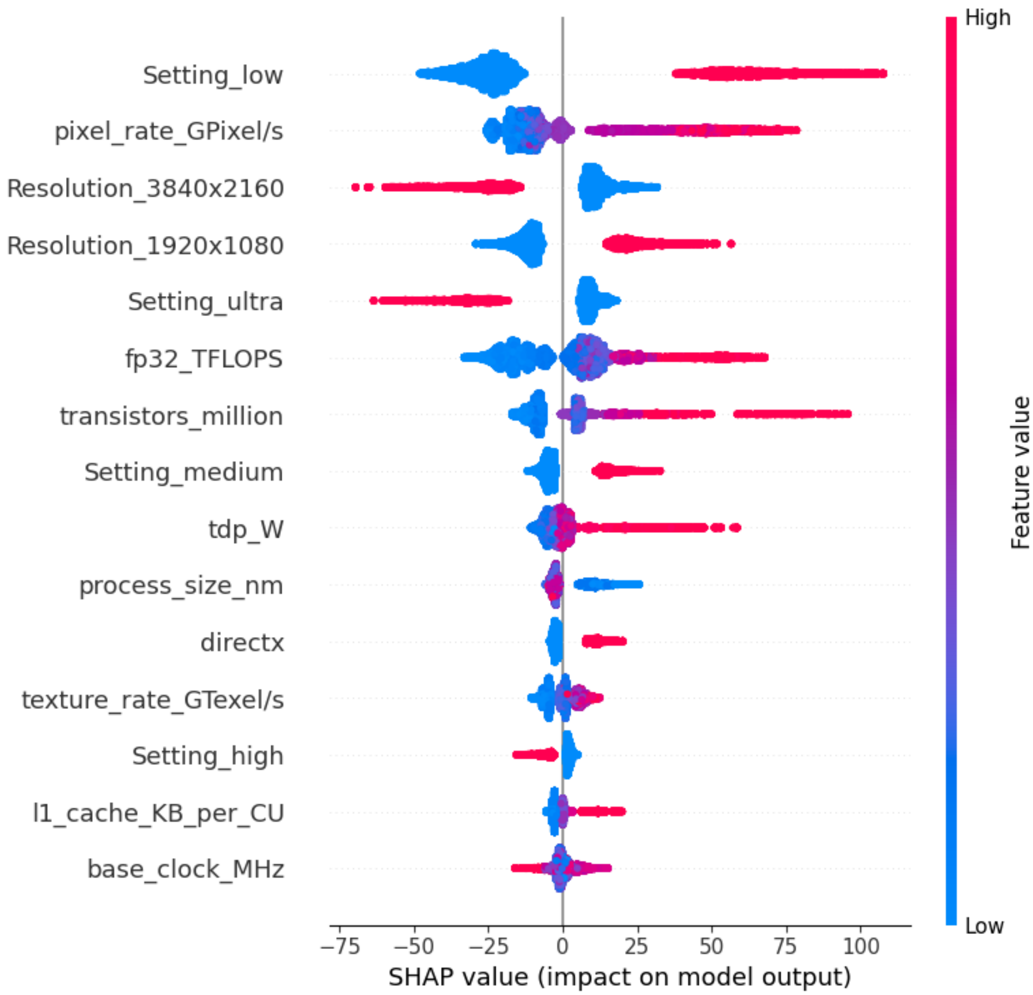

Figure 4. SHAP beeswarm plot for the tuned XGBoost model, showing the distribution of feature contributions across observations.

5. **Contextual Trends by Resolution and Setting**

   - To examine how hardware contributions shift under different usage contexts, SHAP values were aggregated by *Resolution* and *Setting*. These trend plots (Figures 5 and 6) highlight both the *magnitude* (mean |SHAP|) and *direction* (signed SHAP) of feature effects.
   - **Resolution trends**
     - *pixel_rate*, *fp32 throughput*, and *transistors* dominate overall importance. However, directionally, *pixel_rate* is less important at *1920x1080*
     - In directional importance, features peak around **3440×1440** (primarily *pixel_rate*, *fp32*, and *transistors*) and then decline at **4K**, suggesting a saturation/bottleneck where other resources (e.g., CPU) limit further gains.
     - Signed SHAP values indicate the strongest positive contributions from GPU specs at mid–high resolutions, with weaker and more variable effects at **4K**.
   - **Setting trends**

- From *low* → *ultra*, the relative importance of *pixel_rate**, fp32 throughput, transistors* **steadily** declines**.
- Broader capacity indicators—*transistor count* and *TDP*—remain **consistently important** at higher settings, helping sustain performance under heavier workloads.
- Directional (signed) SHAP effects remain relatively **stable** across settings, implying that high settings distribute demand more evenly across multiple hardware components.

- **Summary**
  - Resolution produces more pronounced shifts in feature importance than graphical setting.
  - Across all contexts, *pixel_rate*, *fp32 throughput*, and *transistors* are consistently among the strongest predictors.
  - At higher resolutions, these features show signs of diminishing returns, while broader capacity measures such as *TDP* maintain steady contributions.
  - Overall, the analysis suggests that FPS performance is shaped primarily by hardware throughput features, with resolution and setting acting as key moderators of their relative influence.

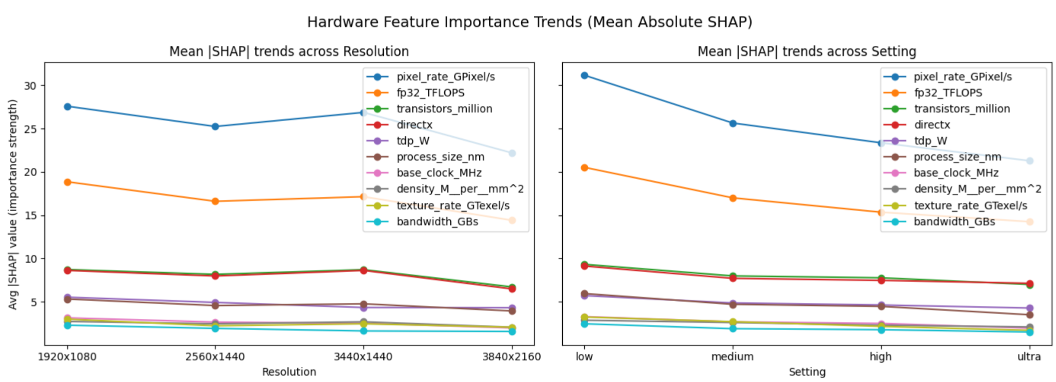

Figure 5. Mean absolute SHAP values aggregated by resolution (left) and setting (right)

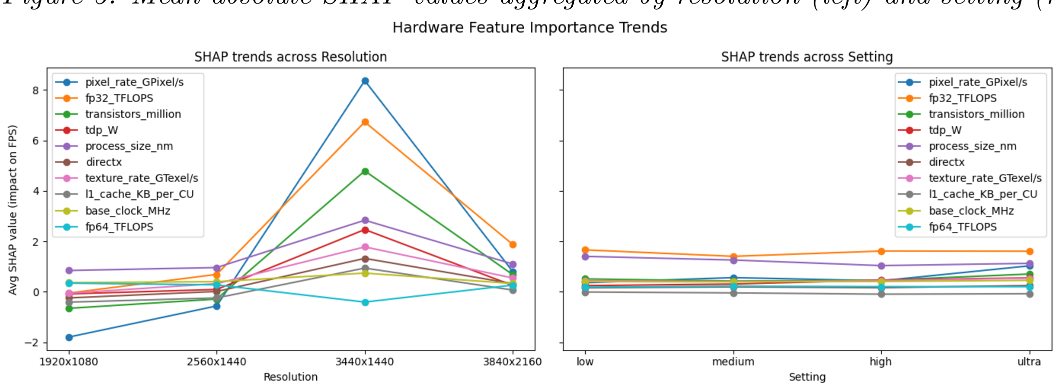

Figure 6. Mean SHAP values aggregated by resolution (left) and setting (right)

1.0.7 6. Limitations

1. **Feature Selection**

- Although the combined model explained approximately 70% of FPS variance, a significant share of the remaining variation is attributable to Game_Name. This variable was excluded to align with the research question, which focused on hardware components rather than game-specific effects. However, omitting it inevitably limited the maximum achievable performance and reduced model explainability. Future work could explore encoding game-level characteristics (e.g., engine type, graphics API, average draw calls) as generalized features that preserve comparability across titles without relying on Game_Name as a single dominant predictor. This would prevent one variable from disproportionately driving FPS variance.

2. **Data Availability**
   - Within the hardware specifications dataset, several GPUs had missing values for certain components. As a result, features such as tensor cores and CUDA were excluded, even though they may have influenced performance outcomes. Additionally, the dataset was restricted to the top 200 most popular GPUs and did not include the most recent hardware generations (e.g., NVIDIA 5000 series). This reduced both the completeness and the generalizability of the model, limiting its applicability to future GPU architectures.
3. **Additional Factors**
   - Real-world FPS performance depends on variables outside the dataset, such as CPU bottlenecks, driver optimization, thermal conditions, and system-level processes. These factors were not captured, meaning the models do not fully reflect actual gaming performance under diverse real-world conditions. Moreover, as hardware and games evolve rapidly, the relevance of these models may degrade without continual updates.

1.0.8 7. Conclusion

This analysis investigated the relationship between GPU hardware features and FPS performance across different resolutions and settings. By excluding Game_Name, the study narrowed its focus to hardware-driven effects. Linear regression explained 62% of FPS variance, while XGBoost improved this to 70%, underscoring the benefits of non-linear modeling. SHAP analysis revealed that resolution, setting, pixel rate, fp32 throughput, and transistor count were the dominant contributors to performance.

These results demonstrate that while hardware and context account for most FPS variability, a portion remains unexplained—reflecting the influence of game-level and system-level factors. Although the exclusion of Game_Name limited predictive performance, it clarified the relative importance of hardware features, which was the central research question. Future work should incorporate richer game-level descriptors, expand coverage to newer GPUs, and consider CPU–GPU interactions to improve predictive accuracy and generalizability.