How Graphical Processing Unit Specifications Influence Gaming Performance

Nate Wade

October 2025

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 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.

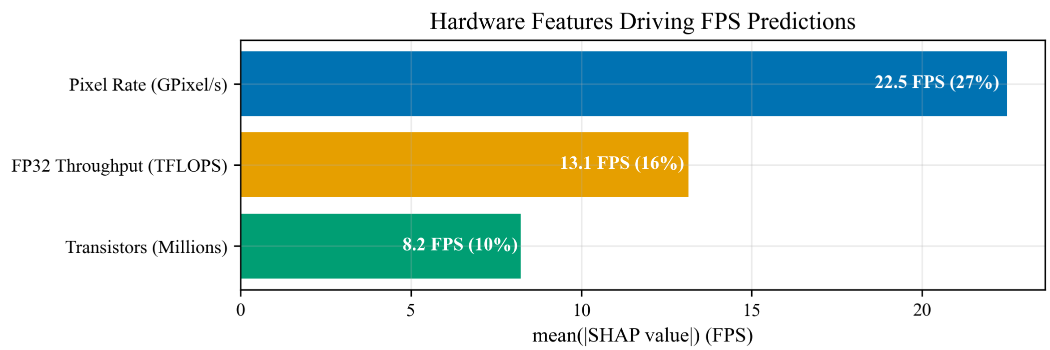

Figure 1: The three strongest contributors—pixel fill rate, FP32 compute throughput, and transistor count—represent the GPU's rendering and computational capacity. Collectively, they explain over half of hardware-driven FPS variation, highlighting that raw throughput and architectural scale dominate real-world gaming performance.

2 Research Question

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

3 Data and Features

The data set used in this analysis includes GPU hardware specifications along with measured FPS values across multiple games, resolutions, and graphical settings scraped from the public Web. Benchmarks provide outcome measurements for the target variable, ***avg_FPS***, grouped by ***GPU_name***, ***resolution***, and ***setting***. The specification table includes core throughput and capacity indicators such as ***pixel_rate***, ***fp32_throughput***, ***transistor_count***, ***memory_size***, and ***clock_speed***. In addition, the dataset retains categorical descriptors like ***architecture*** and ***memory_type*** that capture family-level differences in design and memory technology.

In the main analysis, we emphasize ***pixel_rate***, ***fp32_throughput***, and ***transistors*** as primary predictive features for ***avg_FPS*** across context of ***resolution*** (e.g., 1920×1080, 2560×1440, or 3840×2160), and ***setting*** (e.g., Low, Medium, High, or Ultra). Remaining recorded specifications are summarized briefly in Appendix A.

GPU_Name	Architecture	FP32	Pixel_Rate	TDP
AMD RX 6600 XT	RDNA 2.0	10.6	165.6	160
AMD RX 6800 XT	RDNA 2.0	20.7	238.0	300
AMD RX 5700 XT	RDNA 1.0	9.8	180.4	225
AMD RX Vega 64	GCN 5.0	12.7	109.0	295
RTX 2060	Turing	6.5	80.6	160
RTX 3060 Ti	Ampere	16.2	164.0	200
RTX 3080	Ampere	29.8	164.0	320
RTX 4070	Ada Lovelace	29.2	203.0	200

Table 1: **Hardware Specification** Data Sample

GPU_Name	Game_Name	Avg_FPS	Setting	Resolution
AMD RX 6600 XT	Elden Ring	98.5	Ultra	2560×1440
AMD RX 6800 XT	Horizon Zero Dawn	113.2	High	3840×2160
AMD RX 5700 XT	Cyberpunk 2077	74.6	Ultra	2560×1440
AMD RX Vega 64	Tomb Raider	103.4	High	1920×1080
RTX 2060	RDR2	54.2	High	3840×2160
RTX 3060 Ti	Forza H5	106.8	Ultra	2560×1440
RTX 3080	Doom Eternal	138.1	Ultra	3840×2160
RTX 4070	Starfield	117.9	High	2560×1440

Table 2: **Benchmark Performance** Data Sample

4 Preprocessing and Feature Engineering

The dataset was assembled from two primary raw data sources: a hardware specifications table and a benchmark FPS table. The hardware specifications table was compiled by scraping specification data for each unique ***gpu_name*** found in the FPS benchmark dataset. These values were stored in a dictionary structure, where each entry corresponded to one GPU and its associated specifications. The benchmark FPS table, meanwhile, was obtained as a raw JSON file downloaded from the web.

Feature selection and cleaning were performed independently for each table. Within the hardware specifications table, several nonessential variables were removed to retain only features relevant to performance modeling. Units were extracted from all columns and appended to the column headers to preserve clarity, and every numeric column was converted to a floating-point type for consistency. In the benchmark FPS table, unique ***gpu_name*** values were extracted, and the dataset was constrained to include only the most meaningful variables: ***gpu_name***, ***game_name***, ***avg_FPS***, ***setting***, and ***resolution***. The ***gpu_name*** column was set as the index, resulting in one entry per unique combination of ***game_name*** X ***setting*** X ***resolution***, with the corresponding ***avg_fps*** recorded. On average, each GPU contained approximately 820 observations.

The two tables were then merged using an inner join, aligning entries from the benchmark data with their respective hardware specifications. The resulting dataset used ***gpu_name*** as the index to unify performance measurements and technical attributes within a single analytical frame.

Table 3: **Excerpt of Combined Data** with selected hardware specifications and observed FPS performance (***game_name*** removed)

GPU_Name	Architecture	Memory_Type	FP32 (TFLOPS)	Pixel_Rate (GPixel/s)	TDP (W)	Resolution	Setting
AMD Radeon RX 6600 XT	RDNA 2.0	GDDR6	10.6	165.6	160	2560×1440	Ultra
AMD Radeon RX 6800 XT	RDNA 2.0	GDDR6	20.7	238.0	300	3840×2160	High
AMD Radeon RX 5700 XT	RDNA 1.0	GDDR6	9.8	180.4	225	2560×1440	Ultra
AMD Radeon RX Vega 64	GCN 5.0	HBM2	12.7	109.0	295	1920×1080	High
NVIDIA GeForce RTX 2060	Turing	GDDR6	6.5	80.6	160	3840×2160	High
NVIDIA GeForce RTX 3060 Ti	Ampere	GDDR6	16.2	164.0	200	2560×1440	Ultra
NVIDIA GeForce RTX 3080	Ampere	GDDR6X	29.8	164.0	320	3840×2160	Ultra
NVIDIA GeForce RTX 4070	Ada Lovelace	GDDR6X	29.2	203.0	200	2560×1440	High

Subsequent feature engineering steps ensured that categorical and numerical predictors were correctly structured for modeling. Categorical variables such as ***architecture***, ***memory_type***, ***game_name***, ***resolution***, and ***setting*** were retained as strings. Both ***architecture*** and ***memory_type*** were limited to the five most common categories, with all remaining values grouped under the label "other." Distinct feature groupings were created for numerical hardware features, categorical hardware features, software features (***resolution*** and ***setting***), all hardware features combined, and the complete set of predictors. One-hot encoding was applied to all categorical variables to facilitate numerical modeling. Finally, the outcome variable ***avg_FPS*** and contextual variable ***game_name*** were dropped from the predictor matrix to focus the analysis on the relationship between hardware specifications, ***resolution***, and ***setting***.

5 Exploratory Analysis

To establish a baseline understanding of how individual GPU specifications relate to gaming performance, a preliminary analysis computed the raw Pearson correlation between each hardware feature and ***avg_FPS***. This step provided an interpretable, model-agnostic view of linear associations before introducing multivariate or nonlinear modeling.

As shown in Figure 2, the features most strongly correlated with performance include ***pixel_rate***, ***texture_rate***, ***transistors***, and ***fp32*** throughput, each exhibiting coefficients above 0.5. These variables capture aspects of pixel fill rate, texture processing capacity, and floating-point computation—core determinants of GPU rendering power. Other hardware metrics, such as ***memory_size***, ***bandwidth***, and ***density***, show moderate positive correlations, indicating that while memory and fabrication factors contribute meaningfully, their effects are secondary to raw processing throughput.

Lower correlation values for architectural or cache-level variables (***l1_cache***, ***l2_cache***, and ***memory_type***) suggest weaker direct linear relationships with FPS, potentially because their influence emerges indirectly through interactions captured more effectively by nonlinear models. Overall, the correlation structure highlights that performance scales most directly with features describing parallel rendering capacity and computational throughput.

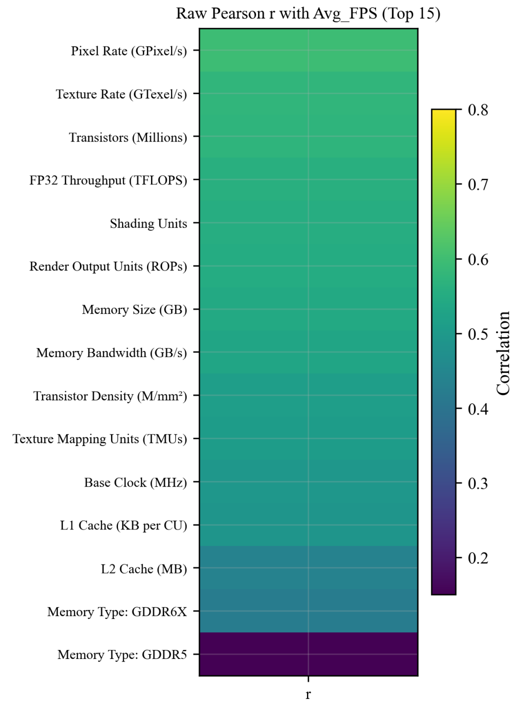

Figure 2: Raw Pearson correlation between GPU specifications and average FPS (Top 15 features).

6 Model Training

The merged dataset was divided into training (80%) and test (20%) subsets. A ***StandardScaler*** was applied to standardize numeric variables, ensuring balanced representation and preventing any feature from dominating due to scale differences.

Two models were evaluated: a linear regression baseline for interpretable feature–performance relationships and an XGBoost regressor to capture nonlinear interactions between hardware and contextual variables such as ***resolution*** and ***setting***. Hyperparameters for XGBoost, including learning rate, depth, and subsampling, were tuned through five-fold randomized cross-validation, while the linear model used default parameters for clarity.

Performance was assessed on the test set using the coefficient of determination (R^2) and mean squared error (MSE). To enhance interpretability, feature importance rankings and SHAP values were generated, quantifying how each feature contributed to FPS predictions at both global and individual levels.

7 Analysis and Results

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.

This feature set included all numerical hardware variables as well as categorical descriptors such as ***architecture*** and ***memory_type***. All numerical features were standardized by the standard deviation of the target (σ_y), allowing linear regression coefficients to be interpreted on a comparable, unitless scale. 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 XGBoost models ranked predictors differently. Specifically, the linear regression emphasized the categorical predictors, whereas XGBoost assigned most of its importance to ***pixel_rate*** and ***fp32*** (as shown in the broader analysis presented later).

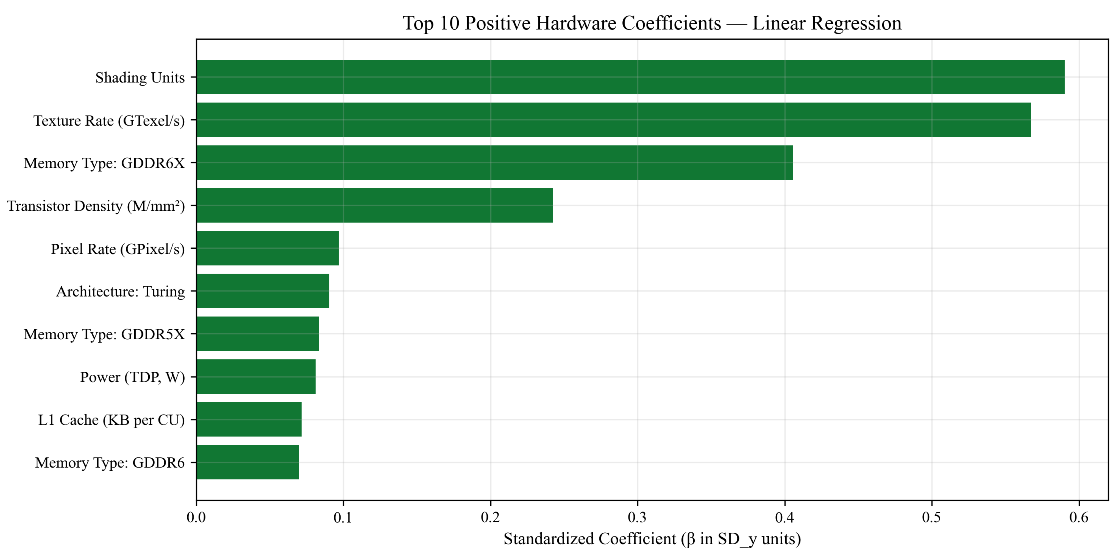

Figure 3: Feature importances (Linear Regression) using hardware features only.

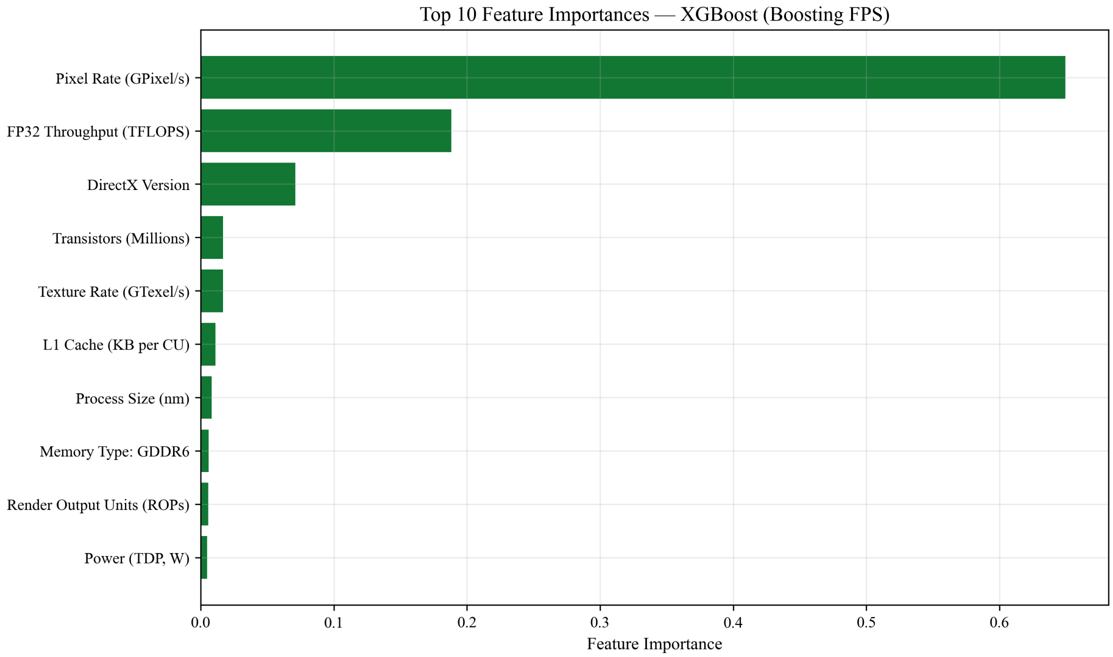

Figure 4: Feature importances (Gradient-Boosted Trees) using hardware features only.

7.1 Feature Importance Shifts Across Methods

A comparison between the exploratory correlations, linear regression coefficients, and XGBoost feature importances reveals several key shifts in how hardware variables contribute to FPS prediction. Features such as ***texture_rate*** and ***shading_units***, which showed strong linear correlations with performance, decline in relative importance once multivariate effects are introduced. This reduction occurs because these specifications are highly collinear with throughput-oriented metrics such as ***pixel_rate*** and ***fp32***, which capture much of the same underlying signal more directly.

Linear regression, by partitioning shared variance among correlated predictors, effectively redistributes importance toward the features that best explain FPS when others are held constant. As a result, variables that were strongly correlated in isolation can lose significance once overlapping contributions are controlled.

When nonlinear interactions are introduced through XGBoost, the feature hierarchy shifts again. The model emphasizes ***pixel_rate*** and ***fp32*** almost exclusively, reflecting how tree-based methods concentrate importance on variables that produce the greatest predictive gains through threshold and interaction effects. At the same time, features like ***l1_cache*** and ***process_size***—which appeared weak in linear models—gain modest importance under XGBoost, suggesting that their influence emerges only under specific performance conditions or in interaction with higher-level throughput measures.

Together, these shifts illustrate how each modeling framework captures a different layer of the performance structure: correlation identifies broad linear co-movement, regression isolates unique additive contributions, and boosting models uncover nonlinear and conditional dependencies among GPU specifications.

7.2 *Game_Name* Significance

The predictive value of ***game_name*** was first examined independently, as FPS outcomes are highly dependent on the computational demands of individual games. When trained using this feature alone, the models achieved R^2 values of 0.1854 for linear regression and 0.1790 for XGBoost. These results indicate that while game identity contributes meaningfully to FPS variance, it is insufficient as a standalone predictor. Although ***game_name*** captures substantial variability in FPS, including it as the sole feature would dilute the contribution of hardware specifications, which are central to the purpose of this study.

7.3 Overall Analysis (Hardware features + Resolution + Setting)

When hardware and contextual features were combined, specifically the complete set of numerical hardware variables along with ***resolution*** and ***setting***, the linear regression model achieved an R^2 of 0.6236, capturing a substantial portion of FPS variance. This demonstrates that hardware specifications, resolution, and setting collectively explain over 60% of observed variability, a notable improvement compared to models using only hardware or game identity. The XGBoost model provided an even stronger fit. With default parameters, it achieved an R^2 of 0.7007, outperforming linear regression by nearly eight percentage points, roughly a 16.7% relative improvement. After hyperparameter tuning, XGBoost achieved a marginally higher R^2 of 0.7034, indicating that most performance gains were realized even before optimization.

Together, these results underscore two main points: first, that combining hardware and contextual features substantially improves predictive accuracy, and second, that tree based models such as XGBoost are more effective than linear models in capturing complex non linear relationships among predictors. However, the modest benefit from tuning suggests the model is approaching the explanatory ceiling imposed by the available features, leaving some variance unaccounted for by hardware and resolution alone.

7.4 Model Interpretability (SHAP Analysis)

To better interpret these findings, SHAP (SHapley Additive exPlanations) analysis was used to evaluate the tuned XGBoost model. SHAP provides a unified framework for interpreting complex machine-learning models by attributing each prediction to individual feature contributions based on cooperative game theory. This allows the relative importance of features to be quantified while also explaining their direction of influence on specific outcomes. Global feature-importance rankings based on mean absolute SHAP values (Figure 5) revealed that ***setting***, ***resolution***, ***pixel_rate***, ***fp32***, ***transistors***, and ***directX_version*** were the strongest contributors to FPS predictions. The prominence of ***setting*** and ***resolution*** is expected given their direct influence on performance, but the significant weights assigned to hardware factors such as ***pixel_rate***, ***fp32***, and ***transistors*** emphasize the role of raw computational power and energy capacity in explaining frame-rate variation.

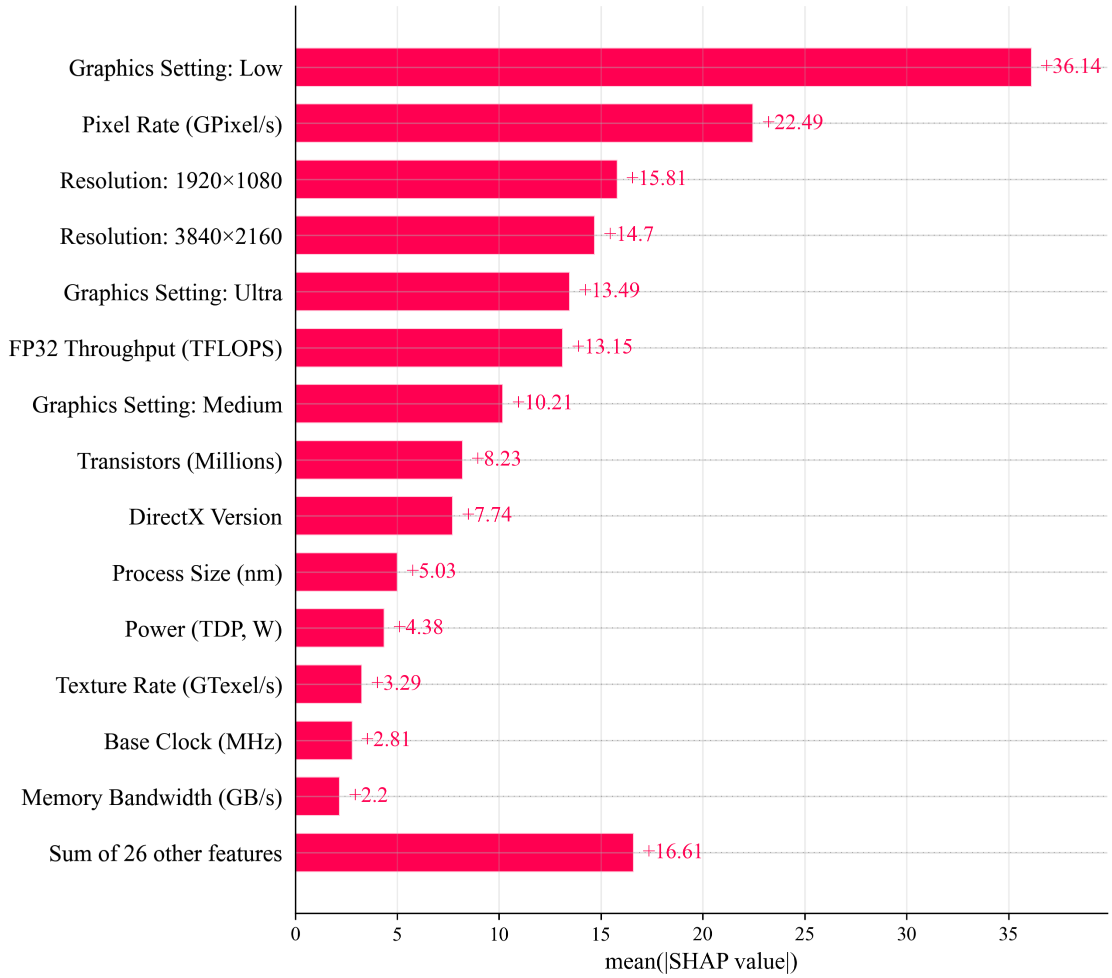

Figure 5: Mean absolute SHAP values for the tuned XGBoost model, showing the strongest global contributors.

7.5 Distribution of Contributions

Distributional SHAP analyses further clarify these relationships. In the SHAP beeswarm plot (Figure 6), ***pixel_rate***, ***fp32***, and ***transistors*** show wide variability across samples, suggesting that their influence on FPS is highly context dependent. These features may dominate predictions at certain resolutions or settings but play smaller roles in others. By contrast, ***direct_X*** exhibits narrower SHAP distributions, implying that while it may not always be the single most influential variable, it provides a steady and consistent contribution to FPS predictions across configurations.

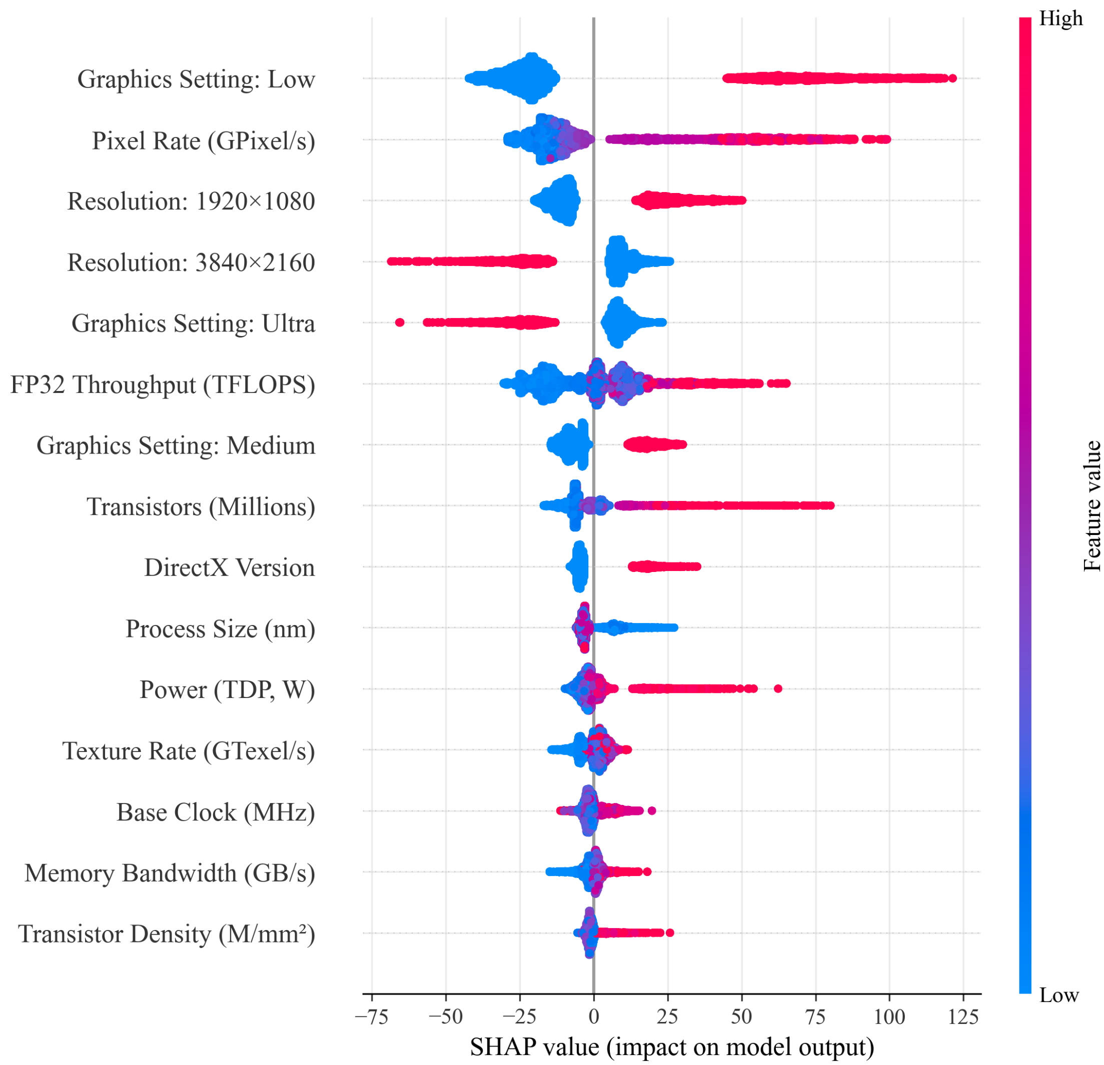

Figure 6: SHAP beeswarm plot for the tuned XGBoost model, showing the distribution of feature contributions across observations.

7.6 Contextual Trends by Resolution and Setting

To examine how hardware contributions shift under different contexts, SHAP values were aggregated by ***resolution*** and ***setting***. These trend plots (Figure 7) display both the magnitude (mean absolute SHAP) and the direction (signed SHAP) of feature effects. Aggregation by resolution revealed that ***pixel_rate***, ***fp32***, ***transistors***, and ***direct_X*** dominate overall importance, though their directional influence diminishes at lower resolutions such as 1920×1080. Importance peaks around 3440×1440—particularly for ***pixel_rate*** and ***fp32***—before declining again at 4K, suggesting a saturation point or bottleneck where other resources, such as CPU performance, constrain further gains. Signed SHAP values indicate that GPU-related features exert their strongest positive contributions at mid-to-high resolutions, with more variable and weaker effects at 4K.

7.7 Setting

Across graphical settings, from low to ultra, feature effects exhibit more nuanced directional shifts than with resolution. In the bottom-right panel, the mean signed SHAP values show that ***pixel_rate***, ***TDP***, and ***texture_rate*** maintain consistently positive contributions, gradually increasing toward higher settings. This pattern indicates that as visual fidelity rises, GPUs with greater fill-rate capacity, thermal headroom, and texture-processing throughput deliver proportionally higher FPS benefits. Meanwhile, features such as ***transistors***, ***process_size***, and ***memory_bandwidth*** fluctuate more irregularly across settings, suggesting that their influence depends on secondary interactions with rendering complexity rather than directly scaling with graphical demand.

Overall, the signed SHAP trends imply that higher graphical settings redistribute computational load rather than amplifying dependence on a single hardware metric. While resolution induces larger absolute variations in feature importance, setting-based differences reveal subtler trade-offs between throughput and efficiency. Consistently, ***pixel_rate***, ***TDP***, and ***texture_rate*** remain stable positive predictors across all settings, while factors such as ***process_size*** and ***memory_bandwidth*** contribute context-dependent effects that vary with workload intensity.

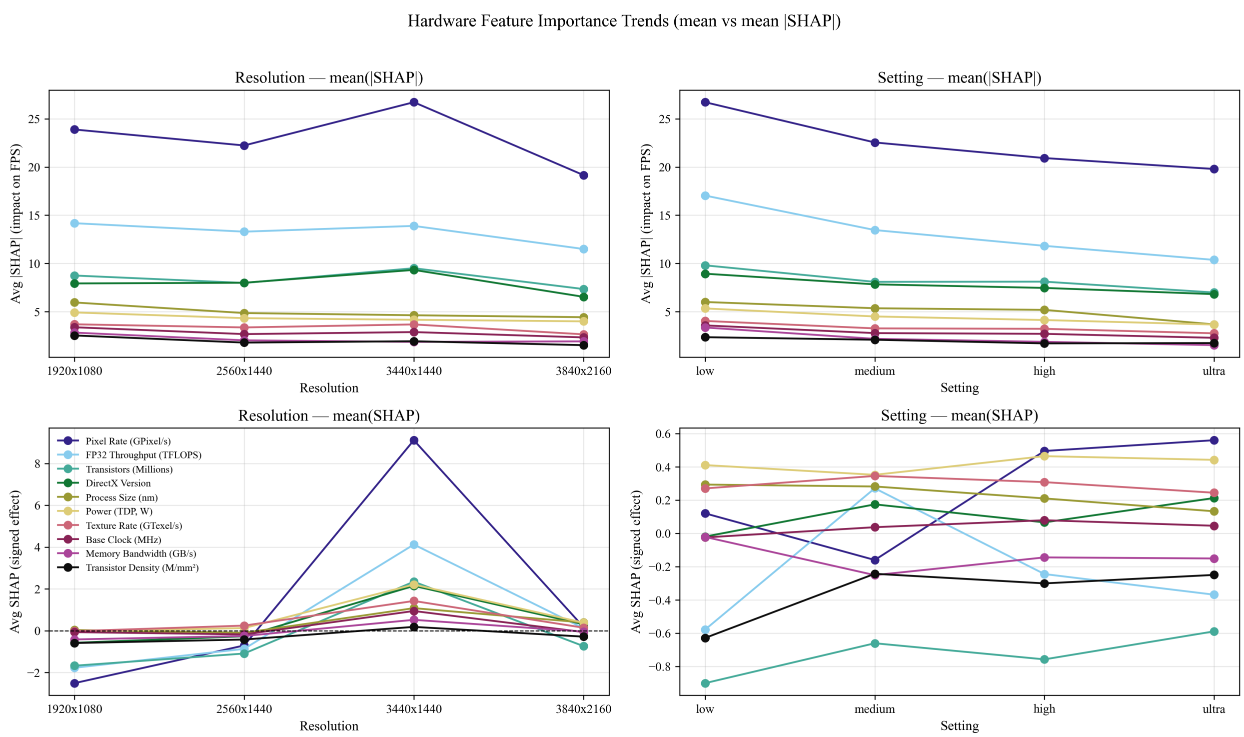

Figure 7: Absolute mean (Top) and mean (Bottom) SHAP values aggregated by resolution (left) and setting (right).

8 Limitations

8.1 Feature Selection

Although the combined model explained approximately 70% of FPS variance, a significant portion 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, or average draw calls) as generalized features that preserve comparability across titles without relying on ***game_name*** as a single dominant predictor. This approach would prevent one variable from disproportionately driving FPS variance.

8.2 Data Availability

Within the hardware specifications dataset, several GPUs contained missing values for specific components. Consequently, features such as tensor cores and CUDA support were excluded, even though they may influence performance outcomes. Additionally, the dataset was limited to the 200 most popular GPUs and did not include the latest hardware generations (e.g., NVIDIA 5000 series). This restriction reduced both the completeness and generalizability of the model, limiting its applicability to future GPU architectures.

8.3 Additional Factors

Real-world FPS performance depends on several variables outside the scope of this dataset, including CPU bottlenecks, driver optimizations, thermal conditions, and system-level processes. Because these factors were not captured, the models do not fully reflect observed gaming performance under varied real-world conditions. Furthermore, as hardware and software ecosystems evolve rapidly, the relevance of these models may diminish over time unless they are updated regularly with newer data.

9 Conclusion

This analysis investigated the relationship between GPU hardware features and FPS performance across different resolutions and settings. By excluding ***game_name***, the study focused on hardware-driven effects independent of game-specific variability. The linear regression model explained 62% of FPS variance, while the XGBoost model improved this to 70%, underscoring the advantages of non-linear modeling. SHAP analysis revealed that ***resolution***, ***setting***, ***pixel_rate***, ***fp32***, and ***transistors*** were the dominant contributors to performance outcomes.

These results demonstrate that although hardware and contextual factors account for most FPS variability, a portion remains unexplained—reflecting the influence of game-level and system-level effects. While excluding ***game_name*** limited overall predictive performance, it clarified the relative importance of hardware features, which was central to the research objective. Future work should integrate richer game-level descriptors, expand coverage to newer GPUs, and include CPU–GPU interaction terms to enhance both predictive accuracy and generalizability.

Importantly, the contextual SHAP analysis indicated that the predictability of hardware specifications depends strongly on both ***resolution*** and ***setting***. At higher resolutions, throughput-oriented features such as ***pixel_rate***, ***fp32***, and ***transistors*** decline in relative importance as broader capacity measures like ***TDP***, ***texture_rate***, and ***process_size*** become more influential. Across graphical settings, directional SHAP values show that ***pixel_rate***, ***TDP***, and ***texture_rate*** increase steadily with higher fidelity, while ***process_size*** trends negatively—implying that smaller, more efficient manufacturing nodes perform better under extreme workloads. These findings highlight that GPU performance emerges from an evolving balance between throughput, efficiency, and power capacity, mediated by both rendering resolution and graphical settings.

Appendix A: Complete Features (with short definitions)

The following specifications appear in the dataset and are defined below. Core predictors highlighted in the main text are included here for completeness. Each feature describes a property of GPU architecture, performance capacity, or configuration.

- ***architecture*** (GPU microarchitecture family, e.g., Turing, Ampere)
- ***process_size*** (fabrication node size of the GPU in nanometers)
- ***transistors*** (total transistor count in millions)
- ***density*** (approximate transistor density per unit area)
- ***die_size*** (physical silicon die area in mm^2)
- ***base_clock*** (default GPU core clock frequency in MHz)
- ***memory_clock*** (VRAM clock frequency in MHz)
- ***memory_size*** (total video memory capacity in GB)
- ***memory_type*** (VRAM technology type such as GDDR5, GDDR6, or HBM)
- ***memory_bus*** (width of the memory interface in bits)
- ***bandwidth*** (memory throughput in GB/s)
- ***shading_units*** (number of programmable shader cores or ALUs)
- ***tmus*** (texture mapping units; responsible for texture sampling throughput)
- ***rops*** (render output units; manage pixel blending and output)
- ***l1_cache*** (first-level cache per compute unit, in KB)
- ***l2_cache*** (on-chip second-level cache size, in MB)

- ***fp32*** (single-precision floating-point throughput in TFLOPS)
- ***fp64*** (double-precision floating-point throughput in TFLOPS)
- ***pixel_rate*** (maximum pixel fill rate in GPixel/s)
- ***texture_rate*** (texture fill rate in GTexel/s)
- ***tdp_W*** (thermal design power, or total board power consumption, in watts)
- ***directx*** (supported DirectX feature level)
- ***memory_bus_bit*** (explicit representation of memory bus width in bits)
- ***base_clock_MHz*** (base GPU clock frequency, expressed in MHz)
- ***memory_clock_MHz*** (memory clock frequency, expressed in MHz)
- ***game_name*** (video game used in benchmark tests)
- ***avg_fps*** (average frames per second achieved during benchmark runs)
- ***resolution*** (display resolution tested, e.g., 1920×1080 or 3840×2160)
- ***setting*** (graphical quality preset, e.g., Low, Medium, High, Ultra)