Architecture Features Affecting GPU performance*

Ali Jabir

December 10, 2025

This study investigates which GPU architectural features most strongly influence performance (TFLOP/s). Using a dataset of modern GPUs, we fit a multiple linear regression model and a ridge regression model to quantify the contribution of transistor count, process size, die area, memory capacity, memory bandwidth, and power budget. While the linear regression achieved high explanatory power, severe multicollinearity made individual coefficients unstable. Ridge regression mitigated this issue and produced more reliable estimates. The results show that GPU performance scales most significantly with transistor count and memory bandwidth, while die size, memory capacity, and process technology act primarily as indirect or secondary factors. These findings highlight the architectural attributes most predictive of GPU performance and offer guidance for performance-oriented GPU design.

Introduction

Graphics Processing Units (GPUs) play a fundamental role in modern high-performance computing. Originally designed to accelerate graphics rendering, GPUs have evolved into highly parallel processors capable of executing thousands of calculations simultaneously. This capability has made them critical for gaming, visualization, scientific simulations, data processing, and most notably training and deploying large language models (LLMs). Models such as GPT, LLaMA, and other transformer-based architectures rely heavily on GPU acceleration because their matrix-multiplication–intensive operations demand massive parallel computation. As LLMs grow in size and complexity, GPU performance becomes a major limiting factor for training speed, inference latency, energy consumption, and the overall cost of model development.

*Project repository available at: https://github.com/syedalijabir/gpu-performance.

Despite the rapid increase in GPU capabilities, performance can vary substantially across models due to differences in architectural and hardware characteristics. Factors such as transistor count, memory bandwidth, thermal design power (TDP), manufacturing process and die size all potentially influence how well a GPU handles computationally intensive workloads. Understanding which of these characteristics contribute most to performance is essential for system designers and researchers when selecting hardware for LLM training or deployment.

In this study, we use linear regression to investigate the relationship between GPU performance and a set of measurable hardware attributes. The central question we aim to answer is: Which GPU specifications have the strongest impact on performance?

By analyzing a dataset of GPUs with features such as manufacturer, transistor count, memory size, memory bandwidth, TDP, die size, and price, our regression model quantifies how changes in each attribute relate to changes in performance while holding others constant. This allows us to identify the most influential factors of GPU performance.

Our analysis indicates that memory bandwidth and transistor count are the primary drivers of performance, while memory size, and TDP have smaller but directionally consistent effects on GPU performance scaling.

The paper is structured as follows. The data section describes the dataset and its limitations. The methods section explains the regression framework used to analyze the relationship of interest. The results section presents the regression findings. Finally, the discussion section interprets the results of the analysis find comments on the findings.

Data

The data for this analysis comes from Epoch AI (Epoch AI 2024). It is a collection of GPUs used to develop and deploy machine learning models gathered on top of (Epoch AI Models 2025). Epoch AI's data is free to use, distribute, and reproduce under the Creative Commons Attribution license with authors credit.

The dataset records a wide range of hardware characteristics. Each row corresponds to a specific GPU model, identified by its full hardware name, such as "NVIDIA H100 SXM5 80GB." Additional columns describe the release date and launch price, which provide contextual information about each GPU's market positioning at the time of introduction. Performance-related attributes include floating-point throughput at various numerical precisions (FP64, FP32, and FP16), which represent non-tensor compute performance. Integer throughput at INT16, INT8, and INT4 precision, which captures performance on quantized operations. Beginning in 2017, many GPUs introduced specialized tensor cores designed to accelerate tensor operations essential for modern machine learning workloads, and the dataset therefore includes maximum machine-learning throughput (ML OP/s), measured in floating-point or integer operations per second depending on format width.

Memory characteristics are represented through the total on-board memory and the memory bandwidth, the latter quantifying how quickly data can be transferred between memory and the processor. Additional fields describe both intranode and internode bandwidth, which capture communication performance within and across server nodes that may host multiple CPUs, GPUs, and storage components.

Physical and architectural attributes include die size, measured in square millimeters. The thermal design power (TDP), which represents the maximum sustainable power dissipation and the base clock frequency. The semiconductor process size, measured in nano-meters, describes the manufacturing technology used to fabricate the chip. Finally, the number of transistors, reported in millions, provides a proxy for architectural complexity and density.

Hardware specifications were collected primarily from official datasheets, which provided key information such as computational throughput, die size, memory characteristics, and manufacturing details. The full dataset contains 169 GPUs with 39 recorded attributes. However, not all specifications are available for every device, leading to many missing entries across the dataset. After filtering out GPUs with incomplete information for the variables required in our analysis, we retain a subset of 40 GPUs. Figure Figure 1 illustrates the release year of these GPUs on the horizontal axis and their launch price in US dollars on the vertical axis.

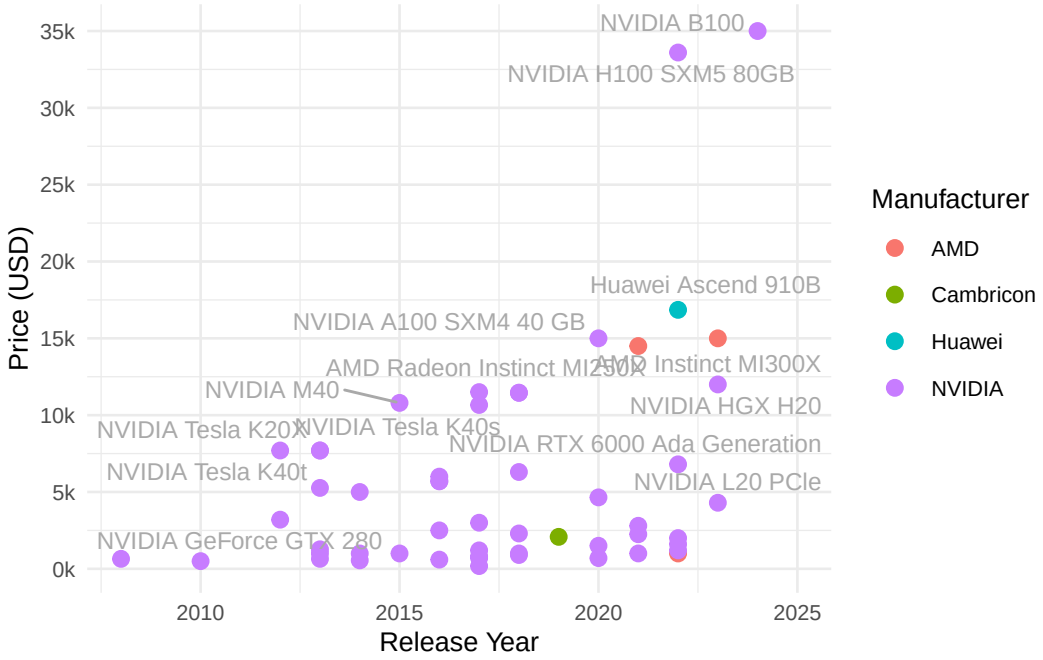

Figure 1: GPU prices by release year

A notable feature of the cleaned dataset is its imbalance towards a single manufacturer: nearly all GPUs originate from Nvidia. Although Nvidia currently dominates the GPU market, this

imbalance limits the diversity of hardware profiles in the sample and should be considered when interpreting the regression results.

Several of the hardware attributes in the dataset, such as transistor count, memory bandwidth, FP32/FP16/ML throughput, and price span multiple orders of magnitude. Working directly with such large values can create numerical instability during model fitting, especially when combining predictors with vastly different ranges in the same regression. Variables measured in millions or trillions can dominate the optimization process, causing coefficient estimates to become sensitive to scaling and making the resulting model difficult to interpret. Applying a logarithmic transformation standardizes these quantities onto a more comparable scale, ensuring that no single feature overwhelms the regression purely due to its units or magnitude.

Methods

Multiple Linear Regression

To quantify how different hardware characteristics contribute to GPU performance, we begin by constructing a multiple linear regression model. Multiple regression provides a statistical framework for examining the relationship between a single response variable and several explanatory variables simultaneously. In this study, the response variable is ML operations per second. The multiple regression model assumes that the expected GPU performance can be expressed as a linear combination of these predictors. Formally, for a GPU i, the performance can be written as:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + ... + \beta_p x_{ip} + \epsilon_i, \text{where } i = 1, ..., n \text{ and p is \# of predictors}$$

where y_i denotes the log performance of GPU i. The β_js are the coefficients to capture the contribution of each attribute to performance where $j = 0, ..., p$. ϵ_i is an error term accounting for unobserved factors and measurement noise. We assume that the error terms are independent and identically distributed with mean zero and a constant variance i.e. $\epsilon_i \backsim N(0, \sigma^2)$. And x_{ij} terms used are listed below:

- x_{i1} is log of the total number of transistors of the i^{th} GPU
- x_{i2} is the process size in nano-meters of the i^{th} GPU
- x_{i3} is log of the size of the die in square millimeters of the i^{th} GPU
- x_{i4} is log of the memory size per board in bytes of the i^{th} GPU
- x_{i5} is log of the memory bandwidth in bytes per second of the i^{th} GPU
- x_{i6} is the thermal design power in Watts of the i^{th} GPU

The model parameters β_j are estimated by minimizing the sum of squared residuals via the `lm()` function in the R programming language (R Core Team 2025) `version 2025.09.0+387`. All the visualizations were conducted in R using the `dplyr` and `ggplot2` packages.

While the multiple linear regression model provides a baseline for understanding how different hardware attributes relate to GPU performance, it is sensitive to multicollinearity. Multicollinearity is a situation where predictor variables are highly correlated with one another. This issue is particularly relevant for GPU hardware data because architectural features tend to co-evolve i.e. GPUs with larger die sizes typically contain more transistors, higher memory bandwidth often accompanies larger memory capacity. Newer process technologies also correlate with higher performance metrics. Such correlations inflate the variance of the estimated coefficients under ordinary least squares (OLS), making the model unstable and the individual effects of predictors difficult to interpret. To address this limitation, we extend our analysis using a regularized model in the next sub-section.

Ridge Regression

Ridge regression is a regularized extension of linear regression that stabilizes coefficient estimates in the presence of multicollinearity. Ridge regression modifies the OLS objective function by adding an L_2 penalty on the magnitude of the coefficients, shrinking them toward zero in a controlled manner. Formally, ridge regression estimates β by minimizing:

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

where $\lambda \geq 0$ is a tuning parameter that governs the strength of the penalty. When $\lambda = 0$, the model reduces to standard OLS regression and as λ increases, the coefficients are increasingly shrunk toward zero. This shrinkage reduces variance at the cost of a small increase in bias, producing a more robust model when predictors are correlated or when the sample size is limited relative to the number of features. We use the same predictor and response variables as multiple linear regression.

To select the optimal value of λ, we use cross-validation, choosing the value that minimizes the prediction error on held-out data. The resulting ridge model allows us to assess the relative importance of each hardware feature while accounting for the complex inter-dependencies inherent in GPU design.

The parameters were estimated by minimizing the penalized sum of squared residuals using the `glmnet` package in R. The tuning parameter λ was selected through 10-fold cross-validation via the `cv.glmnet()` function, which identifies the value that yields the lowest prediction error on held-out data.

Both the OLS and ridge models assume that the relationship between performance and the predictors is linear in parameters. Residuals are assumed to be independent across GPUs.

The models require homoscedasticity and approximately normal residuals for valid inference under OLS; ridge regression is less sensitive to these assumptions but still relies on linearity.

The primary limitation arises from multicollinearity and the design interdependencies within GPUs. Ridge regression mitigates, but does not eliminate, interpretive challenges. Coefficients represent partial effects, even when predictors logically co-occur.

Results

The multiple linear regression model was fitted giving us the regression equation:

$$\begin{aligned}\log(\text{Performance}) = 0.77 \\ &+ 0.79 \times \log(\text{Transistors}) \\ &- 0.03 \times \text{Process\_Size (nm)} \\ &+ 0.41 \times \log(\text{Die\_Size}) \\ &- 0.08 \times \log(\text{Memory\_Size}) \\ &+ 0.81 \times \log(\text{Memory\_Bandwidth}) \\ &+ 0.001 \times \text{TDP}\end{aligned}$$

The model explains a substantial proportion of the variation in GPU performance, with an $R^2 = 0.9$, indicating that 90% of the variance in performance is captured by the six hardware characteristics included in the model. The overall model is highly significant with $F_{(6,33)} = 49.76$, $p < 0.001$, but none of the individual coefficients reach statistical significance at the 5% level. This outcome is consistent with the high multicollinearity among predictors, which inflates standard errors and reduces the apparent significance of individual effects even when the model fits well. Variance inflation factor values as shown in Figure 2 confirm this, with log_transistors (`21.9`), log_mem_bw (`17.0`), and process_size_nm (`11.3`) showing severe multicollinearity. These strong interdependencies reflect the correlation of the features.

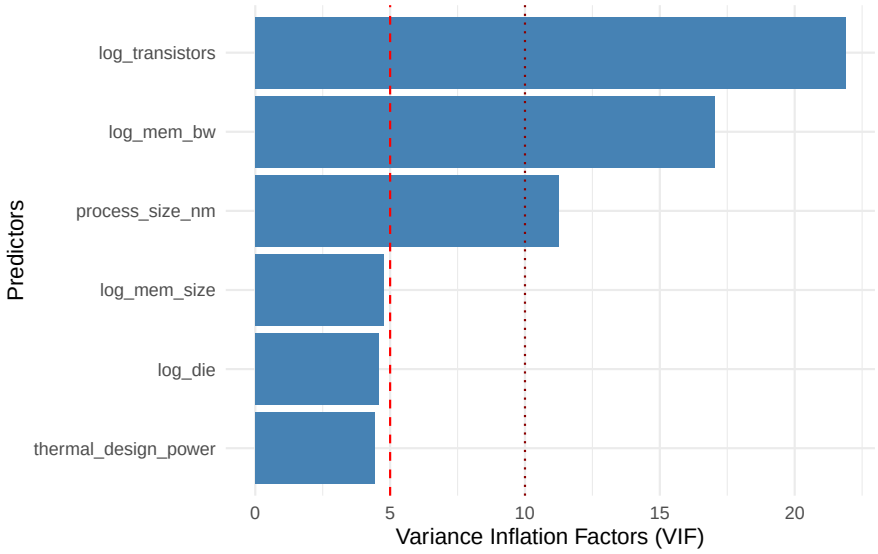

Figure 2: Variance inflation factors for multiple linear regression model

Despite the lack of individual statistical significance, the direction and magnitude of the coefficients provide meaningful information. The estimated slope coefficient for number of transistors is `0.79` which can be interpreted as a 1% increase in transistor count corresponds to an approximate 0.79% increase in GPU performance while holding others constant. This aligns with the role of transistor density in enabling larger computational arrays and wider execution units. Memory bandwidth has a similarly strong positive association with performance. A 1% increase in memory bandwidth corresponds to roughly a 0.81% increase in GPU performance while holding others constant. Because modern deep-learning workloads are bandwidth-bound, this effect size is consistent with architectural expectations. Larger die sizes tend to increase performance, though with a smaller effect (`0.4`) than transistor count or memory bandwidth. Similarly, a 1 nano-meter increase in process technology corresponds to roughly `3.3%` decrease in performance while holding others constant, which is consistent with industry practice of moving towards smaller process technology.

Overall, the regression coefficients align with architectural intuition i.e. GPUs with more transistors, greater memory bandwidth, and larger dies tend to achieve higher performance. The primary limitation of this model is extreme multicollinearity among predictors, which prevents precise estimation of individual effects.

To address the substantial multicollinearity present in the multiple linear regression model, a ridge regression model was estimated. Ridge regression directly mitigates this issue by shrinking correlated coefficients jointly, producing smoother and more interpretable estimates. The model was fit using 10-fold cross-validation to select the regularization parameter λ. The coefficient path are shown in Figure 3. When $\lambda = 0$, the model reduces to standard OLS

regression and as λ increases, the coefficients are increasingly shrunk toward zero. The optimal value of λ was identified as `0.3109`

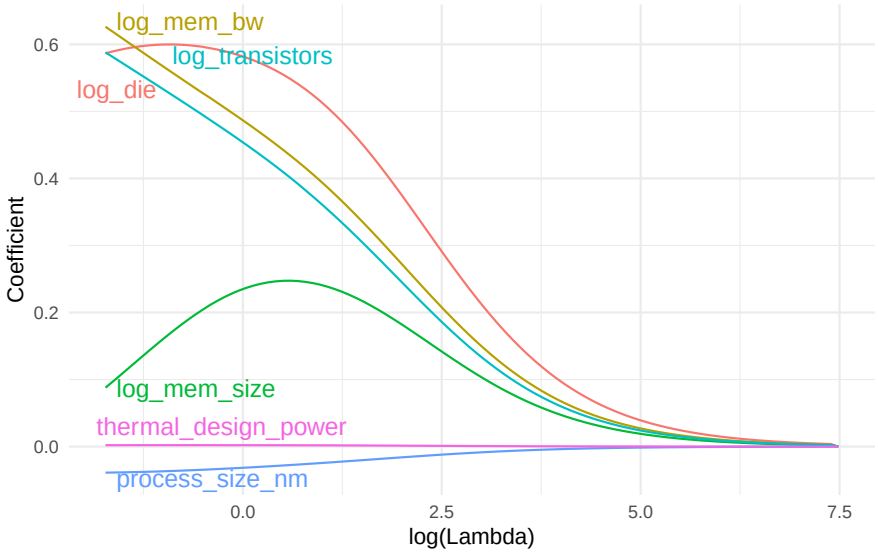

Figure 3: Ridge regression coefficient paths

The optimal coefficients were identified as follows:

$$\begin{aligned}\log(\text{Performance})_{ridge} = 2.94 \\ &+ 0.546 \times \log(\text{Transistors}) \\ &+ 0.144 \times \log(\text{Memory\_Size}) \\ &+ 0.581 \times \log(\text{Memory\_Bandwidth}) \\ &+ 0.002 \times \text{TDP} \\ &+ 0.599 \times \log(\text{Die\_Size}) \\ &- 0.037 \times \text{Process\_Size (nm)}\end{aligned}$$

The coefficients provide a stable set of numerical relationships between GPU hardware characteristics and performance. The coefficient on transistors is approximately `0.5458`, meaning that a 1% increase in transistor count is associated with a 0.55% increase in performance. The coefficient on memory bandwidth is `0.5806`, so a 1% increase in memory bandwidth corresponds to about a 0.58% increase in performance. The effect of die size is similar, with a coefficient of `0.5986`, indicating that expanding the silicon die area by 1% raises performance by roughly 0.60%. The coefficient on memory size is smaller at `0.1441`, implying that a 1% increase in memory capacity produces only a 0.14% improvement in performance, suggesting diminishing returns once baseline memory requirements are met. Thermal design power

(TDP) has a coefficient of `0.002093`, which means that for each additional watt, GPU performance increases by approximately 0.21%. The process size coefficient is `-0.03748`, so every 1 nano-meter reduction in semiconductor process node increases the performance by roughly a 3.7%, which is consistent with the efficiency and density gains from smaller fabrication technologies.

We see that the GPU performance scales most strongly with memory bandwidth, die area, and transistor count, more modestly with memory capacity and power budget, and negatively with process node size.

Overall, the ridge model provides a clearer and more reliable picture of how architectural features contribute to GPU performance. Although the multiple linear regression model achieved a high R^2, its coefficients were too unstable to interpret meaningfully due to severe multicollinearity. Ridge regression, by contrast, optimizes cross-validated predictive accuracy and yields parameter estimates that generalize better beyond the training sample. In this context, the ridge model is the preferable specification. It produces coefficients with sensible directions and magnitudes, reduces variance without excessively biasing the estimates, and better captures the underlying relationships between GPU design characteristics and performance.

Discussion

This study examined the factors that predict GPU performance (TFLOP/s), using multiple linear regression and ridge regression. After transforming several predictors to stabilize variance and reduce skew, the analysis compared traditional OLS with penalized approach to address multicollinearity and identify the most influential hardware characteristics.

By integrating domain knowledge from GPU architecture with the regression results, we can better interpret which hardware features truly drive performance. Although die size appears as a predictor in the model, die area itself does not directly increase computational throughput. Rather it is the logic implemented within the die that determines performance. For this reason, die size can be viewed as a secondary structural variable and may reasonably be excluded in favor of transistor count which actually implements logic.

Similarly, the analysis shows that increasing memory capacity alone does not meaningfully improve GPU performance. Modern GPU workloads are typically bottlenecked not by storage capacity on the device, but by the rate at which data can be moved to and from the compute cores. This is consistent with our findings that memory bandwidth, not memory size, is the stronger and more meaningful predictor of performance. The process size is also not a dominant factor, since smaller fabrication technologies primarily matter insofar as they enable higher transistor densities, an effect already captured by the transistor count variable.

Taken together, both the empirical models and architectural reasoning point to a clear conclusion that the two hardware features that most strongly influence GPU performance are the number of transistors and the memory bandwidth. These components directly reflect

the computational capability of the chip and its ability to feed data to the compute units efficiently.

The study has several limitations. First, the dataset is relatively small, which increases sensitivity to sampling noise and limits model complexity. Second, several predictors (e.g., price, TDP, die size) are themselves partially determined by unobserved engineering trade-offs, making their interpretation less straightforward. Third, even after log transformation, the relationships may remain nonlinear. Fourth, the dataset mostly consists of Nvidia GPUs with complete information, so the results should only be interpreted as such for these GPUs.

Future work could expand the dataset by incorporating more GPUs across generations and vendors. Adding the missing information to dataset, architectural-level features such as CUDA core count, tensor core count, cache hierarchy details, and PCIe vs. NVLink interconnect, would allow for a more fine-grained analysis.

References

Epoch AI. 2024. "Data on Machine Learning Hardware." https://epoch.ai/data/machine-learning-hardware.

Epoch AI Models. 2025. "Data on AI Models." https://epoch.ai/data/ai-models.

R Core Team. 2025. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.