

Energy Aware Quantization for Transformer based Language Models on General Purpose GPUs

Taarana Jammula
The University of Pennsylvania
Philadelphia, USA
tjammula@seas.upenn.edu

Thomas Ngulube
The University of Pennsylvania
Philadelphia, USA
tngulube@seas.upenn.edu

Krishna Karthikeya
Chemudupati
The University of Pennsylvania
Philadelphia, USA
krishkc@seas.upenn.edu

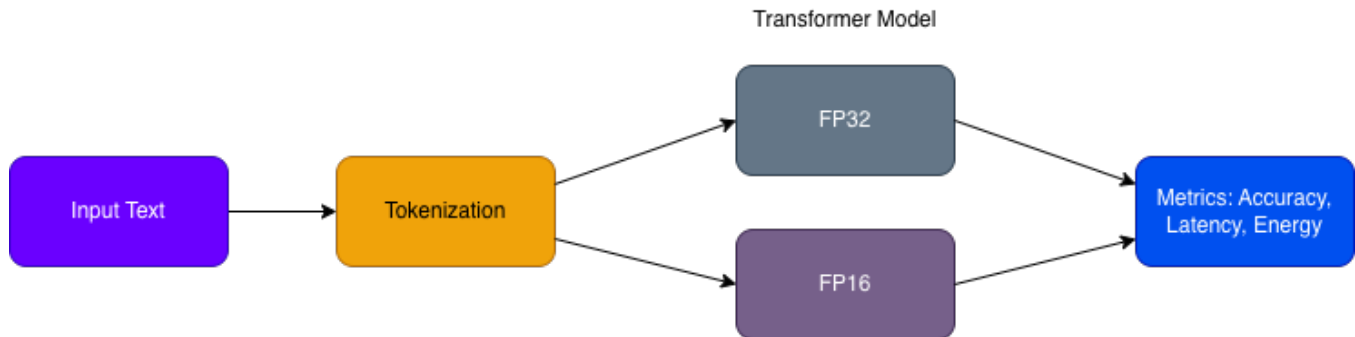


Figure 1: Overview of the experimental workflow. Text inputs are pre-tokenized and passed through a transformer model executed under multiple precision modes (FP32 and FP16). For each configuration, we measure accuracy, latency, and energy per sample using a controlled GPU power-measurement harness.

Abstract

Large language model (LLM) inference imposes substantial computational and energy costs [4], motivating the use of reduced-precision arithmetic to improve efficiency without compromising model quality. Although FP16 inference is widely supported on commodity GPUs, its end-to-end energy impact remains insufficiently characterized, particularly on mid-range inference accelerators such as the NVIDIA Tesla T4. This work provides a systematic measurement of how numerical precision affects accuracy, latency, throughput, and energy consumption for transformer-based models on the T4, which includes Tensor Cores capable of accelerating FP16 matrix operations. We develop a controlled GPU power-measurement harness and validate it using VGG16, then apply the same methodology to DistilBERT and GPT-2 Small under tightly matched workloads with fixed batch sizes, fixed sequence lengths, and fully pre-tokenized inputs to eliminate IO-dominated variability. Across both models, FP16 consistently reduces latency and energy per sample while preserving accuracy or perplexity. DistilBERT achieves a 4.7× speedup and a 5.9× reduction in energy per sample with no loss in classification accuracy. GPT-2 Small shows more modest latency improvements (1.09×) but still achieves a 1.67× reduction in energy while maintaining identical perplexity. Layer-wise analysis reveals that DistilBERT’s highest-energy FFN layers have low predictive impact—making them strong candidates for aggressive quantization—whereas GPT-2 concentrates both energy cost and predictive importance in its embedding stack and LM Head, limiting safe quantization options. These findings demonstrate that

FP16 delivers reliable efficiency gains on Tensor Core-equipped hardware and offer practical guidance for energy-aware LLM deployment using board-level GPU telemetry.

CCS Concepts

• **Computing methodologies** → **Neural networks; Natural language processing**; • **Hardware** → **Power estimation and optimization**.

Keywords

Quantization, Large Language Models, Transformer Models, Energy Efficiency, GPU Power Measurement, Post-Training Quantization, Model Compression, Sustainable Machine Learning, Inference Optimization, Neural Network Efficiency, Latency Reduction, Memory Optimization, Hardware-Software Co-Design, Performance-Accuracy Tradeoffs

1 Introduction

Large language models have become a central component of modern machine learning systems, driving applications in search, question answering, conversational agents, and document processing. Their rapid adoption has made inference efficiency a critical concern. Unlike training, which is performed once and amortized, inference runs continuously and dominates real-world computational and energy costs. Even modest per-request efficiency gains can translate into substantial savings at datacenter scale. Reduced precision is a particularly attractive strategy because it lowers memory footprint and arithmetic cost with little or no retraining.

Most prior work evaluates reduced precision in terms of accuracy, memory usage, or throughput. These metrics, while useful, do not directly capture energy consumption, which is increasingly important for sustainable LLM deployment. Transformer models also differ from convolutional networks in memory access patterns and precision sensitivity, and their end-to-end energy behavior under reduced precision is not well understood. Furthermore, real systems are influenced by IO activity, memory bandwidth, and GPU power-management state, complicating direct measurement.

This work quantifies the energy and performance impact of reduced numerical precision for transformer-based language models executed on a commodity GPU with Tensor Cores. We measure how FP16 compares to FP32 in latency, throughput, model size, accuracy or perplexity, and energy per sample under tightly controlled workloads. Although modern GPUs support additional formats such as INT8 and FP8, our environment—an NVIDIA T4—natively supports only FP32 and FP16 for standard PyTorch inference, so all comparisons are anchored to the FP32 baseline to ensure generality.

Our study is guided by two research questions. **RQ1** examines how FP32 and FP16 differ in end-to-end energy efficiency under fixed, pre-tokenized workloads. **RQ2** evaluates how transformer architecture affects the relationship between energy consumption and predictive importance through a per-layer sensitivity analysis. Addressing these questions requires careful control of the measurement environment. Power telemetry from `nvidia-smi` is coarse and reflects total board power, including memory controllers and background processes. Transformer workloads also vary substantially with sequence length and memory traffic, making IO-free experiments essential for reliable measurement.

To overcome these challenges, we make three contributions. First, we develop a zero-IO measurement harness that pre-tokenizes all inputs, loads tensors into GPU memory before benchmarking, and uses a fixed workload template. Second, we provide a comparative evaluation of FP32 and FP16 inference on DistilBERT and GPT-2 Small on an NVIDIA T4, reporting accuracy, perplexity, throughput, latency, model size, and energy per sample. Third, we perform a layer-wise energy and prediction-impact analysis, identifying components that offer the greatest opportunity for low-risk quantization. These contributions yield a reproducible methodology for transformer energy measurement and practical guidance for energy-aware deployment on commodity hardware.

2 Background

2.1 Reduced Precision for Efficient Inference

Reducing numerical precision is a widely used strategy for accelerating neural network inference. Early work such as Deep Compression [2] demonstrated that substantial redundancy exists in trained networks and that reducing model representation can yield significant improvements in efficiency. Although Deep Compression explored pruning and quantization more broadly, its central insight—that neural networks are resilient to precision reduction—motivates the use of lower-precision arithmetic in modern inference pipelines.

Subsequent work established methods for quantization-aware training and post-training quantization that preserve accuracy under reduced numerical precision. Jacob et al. [3] introduced a systematic framework for quantization within TensorFlow, showing that neural networks retain high accuracy when activations and weights are represented with lower precision. While their work focused on integer inference for mobile hardware, the underlying analysis of quantization error and robustness applies broadly to reduced-precision inference.

For transformer architectures, reduced precision is especially effective because modern GPUs provide specialized hardware support (Tensor Cores) for high-throughput FP16 computation. FP16 reduces data movement, enables larger batch sizes, and increases arithmetic density compared to FP32. Empirical studies across BERT, GPT, and T5-style models consistently show that FP16 inference preserves accuracy while substantially improving throughput.

2.2 Energy Considerations in Deep Learning Inference

Energy consumption in deep learning workloads arises from both computation and memory movement. FP16 reduces activation and weight sizes by half relative to FP32, decreasing memory bandwidth demand and improving the likelihood that data fits into faster memory hierarchies. These effects reduce DRAM access, which is often the dominant contributor to total energy cost.

Recent system-level analyses such as Caravaca et al. [1] emphasize the necessity of measuring energy at the GPU board level rather than inferring it from FLOP counts. Their findings show that a substantial portion of GPU power originates from memory controllers, HBM/VRAM refresh, and background system activity, all of which complicate interpretation of power measurements. This underscores the need for controlled workloads and careful experimental design when comparing precision modes, especially for transformer inference, which is sensitive to sequence length, memory traffic, and irregular kernel structure.

Together, these observations motivate a direct empirical comparison of FP32 and FP16 for transformer inference under conditions that minimize IO variability and isolate compute-level effects.

3 Measurement Setup and Methodology

This section describes the hardware and software environment, the zero-IO dataset design, the energy and latency measurement infrastructure, and the experimental protocol used to evaluate FP32 and FP16 inference for transformer-based models.

3.1 Hardware and Software Environment

All measurements were performed on a pair of NVIDIA T4 GPUs (16 GB GDDR6 each) available in our compute environment. The T4 is based on the Turing architecture (compute capability 7.5) and includes Tensor Cores capable of accelerating mixed-precision matrix operations. As a result, FP16 inference on the T4 benefits from both reduced memory traffic and hardware-accelerated fused GEMM operations, making the device suitable for studying the impact of reduced precision on transformer inference under realistic serving conditions.

Although two T4 GPUs were available, all experiments were executed on a single device to ensure consistent and isolated power and latency measurements. No model parallelism or data parallelism was used. The host system ran a modern Linux distribution with an x86_64 CPU and sufficient system memory to preload all datasets, models, and pre-tokenized tensors into memory before benchmarking. Experiments used PyTorch 2.6.0, CUDA 12.4, and the transformers library, with all software versions fixed to ensure reproducibility. All floating-point operations adhered to the IEEE 754 standard.

3.2 Dataset Preparation and Zero-IO Design

To eliminate disk IO interference during measurement, all datasets were pre-tokenized once and saved as PyTorch tensors. For DistilBERT, 50 samples from SST-2 were tokenized to a fixed length of 128 tokens and stored as .pt files containing input_ids, attention_mask, and labels. For GPT-2, 100 sequences from WikiText-2 were split into 128-token segments and saved analogously.

At runtime, inputs were loaded using:

```
torch.load(..., map_location=device)
```

so that all tensors were placed directly into GPU memory before benchmarking. No disk reads occurred inside timed sections. The total footprint of pre-tokenized tensors was a few tens of megabytes, well within available GPU memory.

3.3 Energy Measurement Infrastructure

3.3.1 Power Monitoring via nvidia-smi. GPU power was measured using the NVIDIA System Management Interface (nvidia-smi), which exposes instantaneous board-level power in Watts via on-device sensors. These readings reflect total GPU board consumption, including SMs, memory controllers, GDDR6 refresh, PCIe subsystems, and auxiliary circuitry. The sampling frequency is limited to approximately 50–100 ms depending on driver and hardware.

Power samples were collected using:

```
nvidia-smi --query-gpu=power.draw \
  --format=csv,noheader,nounits --id=0
```

To avoid subprocess overhead inside inference timing, power measurement was performed asynchronously in a background thread. A PowerLogger class invoked subprocess.run() at fixed intervals, parsed numerical output, and appended samples to a thread-safe buffer using a Lock. A sampling interval of 50 ms yielded roughly 20 samples per second. Before experiments, the logger was validated by confirming stable idle and sustained-load power ranges.

3.3.2 Energy Computation. Let P_i denote the i -th sampled power value and N the total number of samples. Mean power was computed as

$$\bar{P} = \frac{1}{N} \sum_{i=1}^N P_i. \quad (1)$$

If the benchmark ran for duration T , total energy consumption was:

$$E_{\text{total}} = \bar{P} T. \quad (2)$$

For a benchmark consisting of num_iters iterations with batch size B , the per-sample energy was:

$$E_{\text{sample}} = \frac{E_{\text{total}}}{\text{num_iters} B}. \quad (3)$$

All energy values are reported in millijoules (1 J = 1000 mJ).

4 Results

This section presents a comprehensive evaluation of how numerical precision influences latency, throughput, accuracy, perplexity, model size, and energy consumption for both DistilBERT and GPT-2 Small. All measurements were collected under tightly controlled conditions using pre-tokenized datasets, fixed batch sizes, and a standardized GPU power-measurement harness to ensure consistency across trials. Although the NVIDIA T4 includes Tensor Cores and supports several reduced-precision formats such as FP16 and BF16, it does not provide native INT8 execution for general-purpose PyTorch inference without TensorRT or custom kernel paths. As a result, FP32 and FP16 were the only precision modes that could be used directly and consistently within our experimental framework. Our evaluation therefore focuses on these two modes, with all recommendations anchored to the FP32 baseline. Framing the analysis in terms of FP32 versus FP16 enables the results to generalize to more capable quantization schemes on other hardware: any alternative precision format (e.g., INT8 via TensorRT, FP8 on Hopper, or mixed-precision Tensor Core kernels) can be interpreted relative to the same FP32 reference point. This allows the methodology to extend beyond the constraints of the specific GPU and software stack used in this study.

4.1 DistilBERT Inference Results

Figure 2 summarizes the performance of DistilBERT under FP32 and FP16 execution on the NVIDIA T4. Unlike older Pascal-class GPUs, the T4 includes Tensor Cores that provide hardware acceleration for FP16 matrix operations. As a result, FP16 speedups arise from both reduced memory traffic and increased arithmetic throughput on Tensor Core-optimized kernels. These effects jointly contribute to the substantial latency and energy improvements observed in our experiments.

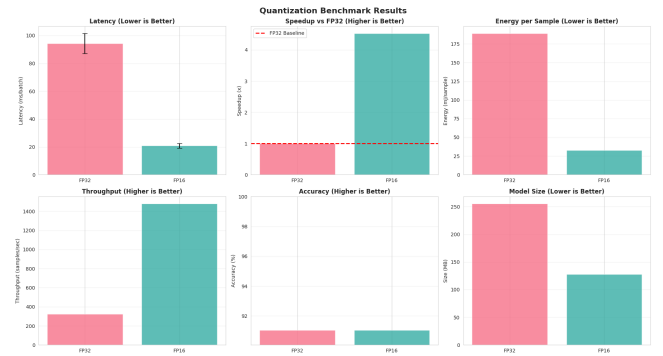


Figure 2: DistilBERT benchmark results comparing FP32 and FP16 across latency, speedup, energy per sample, throughput, accuracy, and model size.

4.1.1 Latency and Throughput. FP16 reduced end-to-end latency from approximately 94 ms per batch (FP32) to 20 ms per batch, corresponding to a 4.7 \times speedup. This improvement is consistent across trials, as indicated by the narrow error bars for FP16. The throughput increased proportionally, from roughly 300 samples/s in FP32 to 1,450 samples/s in FP16. These results suggest that DistilBERT inference on GPUs remains memory-bandwidth-limited, and halving element width significantly improves data movement efficiency.

4.1.2 Energy per Sample. FP16 also provided a substantial reduction in energy consumption. The per-sample energy decreased from approximately 190 mJ in FP32 to 32 mJ in FP16, yielding a 5.9 \times reduction. Because average power draw remained comparable across precisions, the majority of energy savings arose from reduced execution time. This confirms that FP16 acceleration translates directly into lower energy requirements for DistilBERT on commodity GPUs.

4.1.3 Accuracy. Accuracy on SST-2 remained unchanged across precision modes. Both FP32 and FP16 achieved approximately 91%, demonstrating that DistilBERT is robust to reduced precision on inference workloads when sequence lengths and input statistics are well behaved. No degradation was observed across the repeated trials.

4.1.4 Model Size and Memory Footprint. The FP32 checkpoint occupied roughly 250 MB, whereas the FP16 model required approximately 130 MB, a 48% reduction. The smaller footprint translated to lower GPU memory usage during inference, consistent with reduced activation and parameter storage. This reduction is beneficial for memory-constrained deployments and allows larger batch sizes or concurrent model hosting.

4.1.5 Speed–Accuracy Trade-off. Figure 3 summarizes the joint relationship between inference latency and classification accuracy for FP32 and FP16. The horizontal axis reports latency per batch (lower is better), and the vertical axis reports SST-2 accuracy (higher is better). FP16 occupies the lower-left region of the plot, indicating simultaneously faster execution and unchanged accuracy.

The FP32 baseline achieved approximately 93 ms per batch with an accuracy of 91.0%. FP16 reduced latency to roughly 20 ms per batch while maintaining essentially identical accuracy at 91.0%. Since both precision modes lie on the same accuracy contour, no accuracy degradation was observed. The *Pareto-optimal* point is therefore FP16, which strictly dominates FP32 in both metrics on the tested workload.

This result confirms that, for DistilBERT on the Tesla T4, FP16 provides a clear operational advantage: it improves speed by more than 4 \times at no measurable loss in predictive quality.

4.2 GPT-2 Small Inference Results

Figure 4 summarizes the quantitative performance of GPT-2 Small under FP32 and FP16 execution using fixed sequence lengths and pre-tokenized evaluation batches. Because the NVIDIA T4 includes Tensor Cores, FP16 inference benefits from both reduced memory bandwidth pressure and hardware-accelerated mixed-precision

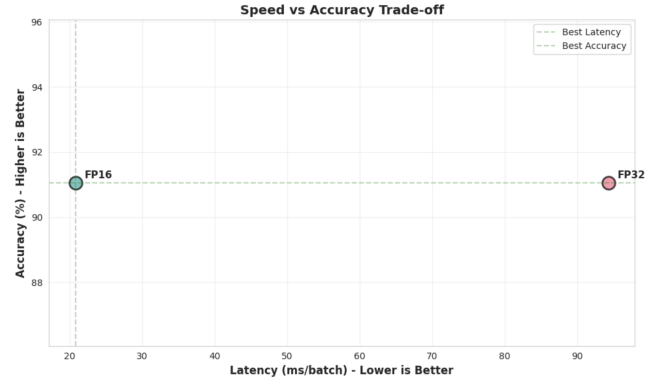


Figure 3: Speed–accuracy trade-off for DistilBERT on SST-2 under FP32 and FP16 precision. Latency is reported in milliseconds per batch, and accuracy is reported as classification accuracy on the pre-tokenized evaluation subset. FP16 achieves substantially lower latency while maintaining identical accuracy relative to FP32.

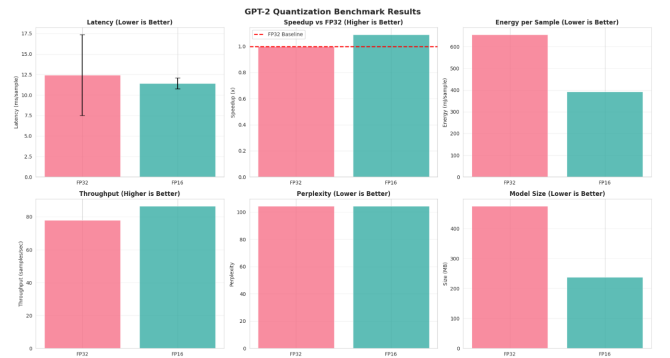


Figure 4: GPT-2 Small benchmark results comparing FP32 and FP16 across latency, speedup, energy per sample, throughput, perplexity, and model size.

matrix operations. However, GPT-2’s autoregressive structure produces smaller overall speedups than those observed for DistilBERT, since its generation pipeline is dominated by sequential decoder computations rather than large, fully parallelizable encoder blocks.

4.2.1 Latency and Throughput. FP16 reduced per-sample latency from approximately 12.4 ms in FP32 to 11.4 ms in FP16, corresponding to a modest 1.09 \times speedup. The variance in FP32 latency is noticeably higher, reflecting greater sensitivity to GPU clock state and kernel scheduling. Throughput increased from roughly 78 samples/s in FP32 to 88 samples/s in FP16, consistent with the small but predictable memory-bandwidth advantage of half-precision operands.

4.2.2 Energy per Sample. Energy consumption decreased from approximately 650 mJ per sample in FP32 to around 390 mJ in FP16, yielding a 1.67 \times improvement. These savings arise primarily from reduced execution time rather than reduced power draw, as power levels remained similar across precision modes. Although absolute

energy values are higher than DistilBERT due to GPT-2’s larger parameter count and heavier attention mechanisms, FP16 consistently improves energy efficiency.

4.2.3 Perplexity. FP16 preserved generative quality. Perplexity on the pre-tokenized subset of WikiText-2 was approximately identical across precision modes (~ 105 in FP32 versus ~ 105 in FP16). No degradation was observed over multiple trials, indicating that GPT-2 Small is robust to FP16 inference on GPUs.

4.2.4 Model Size. The model size decreased from approximately 480 MB in FP32 to 240 MB in FP16, a $2\times$ reduction consistent with halving parameter precision. This smaller footprint reduces memory pressure and enables larger batch sizes or multiple concurrent models within the same GPU memory budget.

4.2.5 Summary. Relative to DistilBERT, GPT-2 exhibits smaller speedups but comparable energy-per-sample reductions when switching from FP32 to FP16. The results demonstrate that even without Tensor Cores, FP16 delivers consistent efficiency benefits for autoregressive transformer models, while maintaining perplexity and substantially reducing model memory footprint.

4.3 DistilBERT Layer-wise Energy and Prediction Impact Analysis

To investigate which components of DistilBERT offer the most promising targets for precision reduction, a detailed layer-wise analysis was conducted. For each layer, two quantities were measured: (1) energy consumption per forward pass, estimated using the proportional attribution method described in Section 3, and (2) the prediction impact score, computed by ablating the layer output and quantifying the resulting change in model behavior via KL divergence and logit L_2 shift. Together, these metrics identify layers that incur high energy cost while contributing minimally to the final prediction, making them strong candidates for quantization or more aggressive compression.

4.3.1 Energy Distribution Across Layer Types. Figure 5 shows that the feed-forward network (FFN) sublayers dominate overall energy usage. The first and second FFN sublayers account for the largest share of compute energy, followed by the attention projection layers. These findings are consistent with the structure of Transformer blocks, where FFN layers typically contain $4\text{--}8\times$ more parameters than the self-attention mechanism and thus incur larger matrix multiplications.

LayerNorm, embeddings, and dropout operations contribute negligible energy by comparison, each residing several orders of magnitude below the FFN layers. Although low-energy layers alone do not meaningfully reduce the energy footprint, their low cost suggests that quantizing them to lower precision produces minimal risk.

4.3.2 Prediction Impact Across Layer Types. Figure 6 reports the prediction impact scores. Embeddings, LayerNorm, and the output classifier head show the highest impact values, indicating that perturbations to these layers substantially degrade model behavior. In contrast, both FFN sublayers and the attention projection/output layers show relatively low impact scores, even though they consume

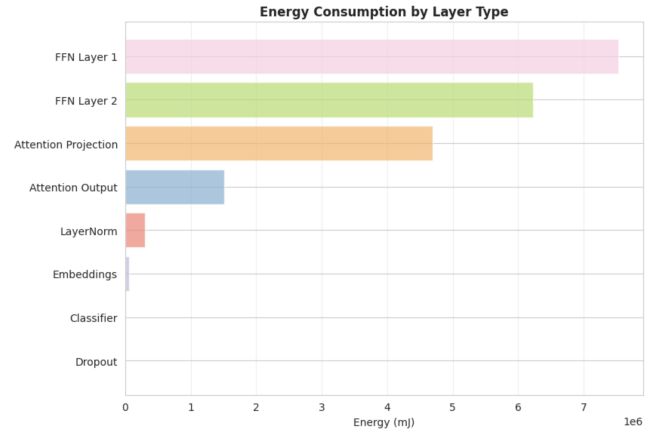


Figure 5: Energy consumption across DistilBERT layer types. FFN Layer 1 and FFN Layer 2 dominate total energy usage, followed by attention projection and attention output layers. Embeddings, LayerNorm, dropout, and classifier layers contribute minimally.

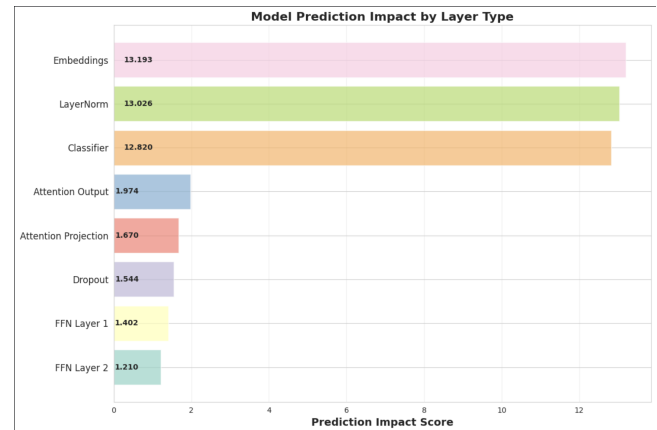


Figure 6: Prediction impact by layer type. Embeddings, LayerNorm, and the classifier head cause the largest change in model predictions when ablated, while FFN and attention layers show substantially lower impact.

the majority of the model’s energy. This misalignment between energy and importance is a key signal for identifying safe quantization targets.

4.3.3 Energy–Impact Correlation. A joint view of all 67 measured layers is shown in Figure 7, while the cross-metric correlation matrix in Figure 8 quantifies these relationships. The correlation between energy and prediction impact is moderately negative ($r = -0.494$), confirming that layers consuming more energy tend, on average, to contribute less to prediction fidelity. This inversion is unusual compared to convolutional networks and reflects the compact role that attention plays relative to the parameter-heavy FFN modules in Transformer architectures.

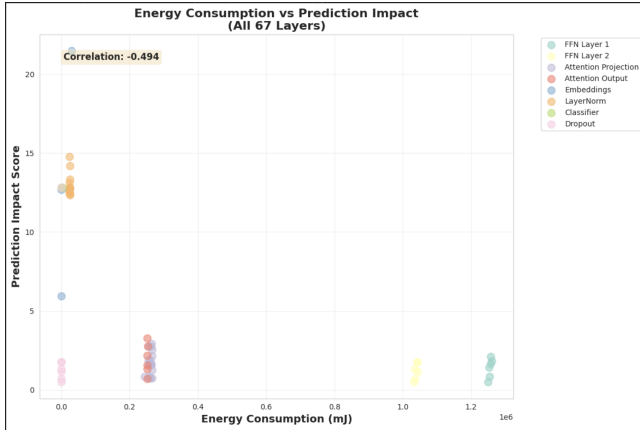


Figure 7: Scatter plot of energy consumption versus prediction impact for all 67 DistilBERT layers. A moderate negative correlation ($r = -0.494$) indicates that high-energy layers often have low predictive importance.

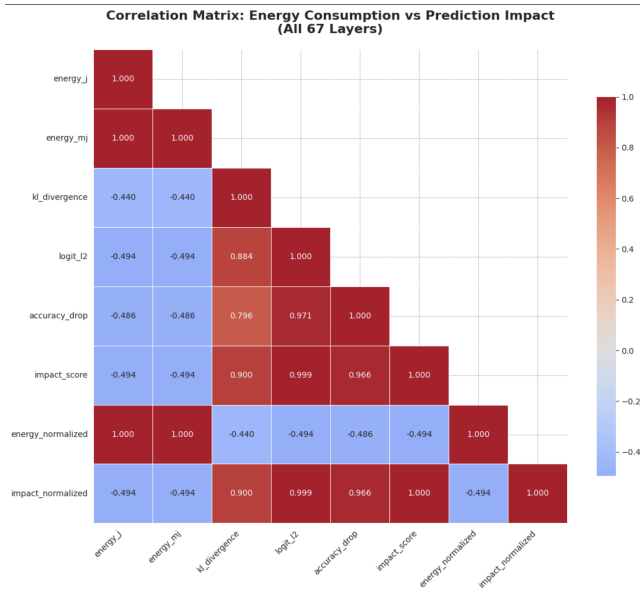


Figure 8: Correlation matrix comparing energy metrics and prediction-impact metrics across DistilBERT layers. Energy is negatively correlated with KL divergence, logit L_2 , accuracy drop, and impact score.

4.3.4 Quantization Candidates. Based on the combined analysis, the most suitable quantization targets for DistilBERT are:

- **FFN Layer 1 and FFN Layer 2**, which consume the highest energy but exhibit the lowest prediction impact. These layers form the clearest opportunity for reducing compute cost without harming accuracy.
- **Attention Projection and Attention Output** layers, which show moderate energy use and similarly low impact scores, making them strong secondary candidates.

Conversely, embeddings, LayerNorm, and the classifier head should remain at higher precision due to their high prediction impact despite their modest energy consumption.

Overall, DistilBERT’s per-layer profile indicates that FFN sublayers dominate energy cost but play a comparatively muted role in prediction sensitivity. This structure aligns with prior observations in Transformer compression work and suggests that aggressively quantizing FFN layers (e.g., INT8 or mixed-precision FP16/FP8) is likely to yield meaningful energy savings with minimal accuracy degradation.

4.4 GPT-2 Layer-wise Energy and Prediction Impact Analysis

A layer-wise analysis was conducted for GPT-2 Small to determine which components consume the most energy and how strongly each layer influences next-token predictions. While the methodology mirrors the DistilBERT study, the decoder-only architecture of GPT-2 produces a distinct pattern: the output projection (*LM Head*) and embedding stack dominate both energy consumption and predictive sensitivity. This structural concentration is characteristic of autoregressive Transformers, where every prediction passes through the embeddings at the input and the LM Head at the output.

Energy Distribution Across Layer Types. Figure 9 reports energy consumption aggregated by layer category. The **LM Head** is the largest contributor by a wide margin, followed by **LayerNorm** and **dropout** operations. The LM Head alone requires more than 35 kJ per forward pass, reflecting the expensive projection of hidden states onto GPT-2’s vocabulary space. Because this projection operates at every token step and must compute logits across thousands of vocabulary entries, it naturally becomes the dominant energy bottleneck on a GPU.

Embeddings and residual components consume significantly less energy. This contrasts sharply with DistilBERT, where FFN layers dominated energy usage. In GPT-2, the autoregressive decoding structure shifts the cost to the LM Head, which aggregates the computation produced by all decoder blocks.

Prediction Impact Across Layer Types. Figure 10 illustrates prediction-impact scores computed via layer ablation and measurement of KL divergence and logit L_2 shift. As expected for a decoder-only Transformer, the **LM Head** shows the largest impact by a wide margin, and **embeddings** also exhibit high predictive sensitivity. This alignment arises from architectural necessity: embeddings define the input representation for all subsequent attention operations, and the LM Head is solely responsible for producing next-token logits. Perturbing either layer directly disrupts GPT-2’s generative behavior.

LayerNorm also demonstrates moderate predictive impact due to its repeated occurrence across all decoder blocks. Dropout and residual categories, by contrast, exhibit very low impact scores.

Energy-Impact Relationship. Figure 11 provides a joint visualization of energy usage and prediction impact for all 53 GPT-2 layers. Unlike DistilBERT, which exhibited a negative correlation between energy and predictive importance, GPT-2 displays a **positive correlation** ($r = 0.570$): layers that consume more energy also tend to

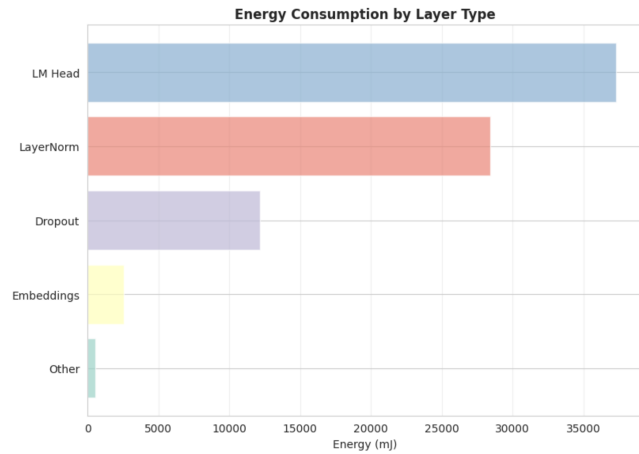


Figure 9: GPT-2 energy consumption by layer type. The LM Head dominates energy use, followed by LayerNorm and dropout layers. Embeddings and other components contribute minimally.

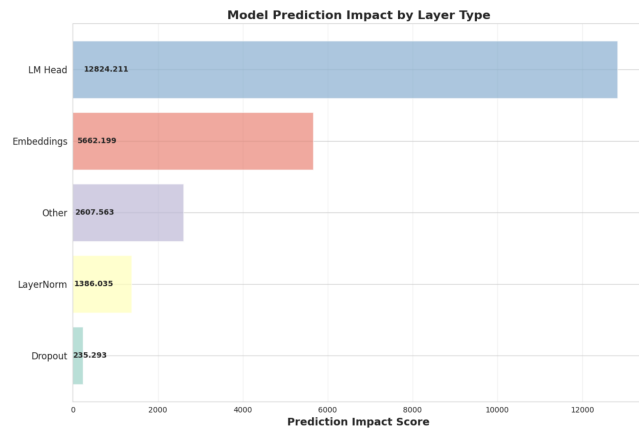


Figure 10: GPT-2 prediction-impact scores across major layer categories. LM Head and embeddings exhibit the highest impact, consistent with their architectural role in autoregressive decoding.

be more important for prediction quality. This correlation is reinforced in the heatmap in Figure 12, which shows that both energy (in mJ) and mean per-layer execution time correlate strongly with logit L_2 shift and impact score.

This behavior is a natural outcome of autoregressive Transformers: the LM Head, which is both computationally expensive and prediction-critical, aggregates the contributions of all upstream layers and maps them to the vocabulary distribution. Consequently, GPT-2 concentrates both energy demands and predictive influence in the same architectural components.

Quantization Recommendations for GPT-2. Because GPT-2’s highest-energy layers are also those with the highest predictive impact, quantization flexibility is more limited than in DistilBERT. Based

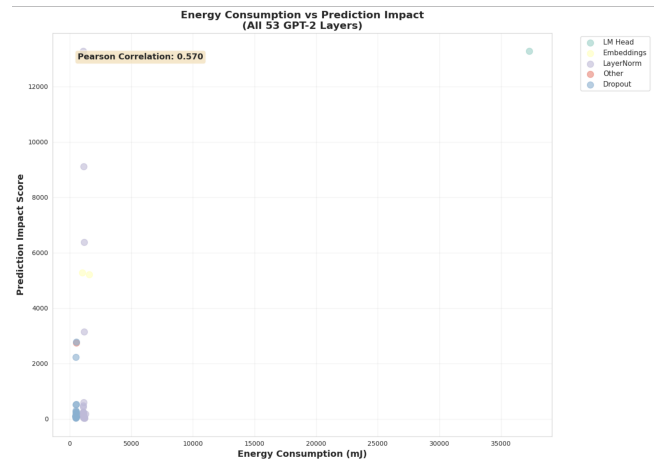


Figure 11: Energy consumption versus prediction impact for all GPT-2 layers. A positive correlation ($r = 0.570$) indicates that high-energy layers also carry high predictive importance.

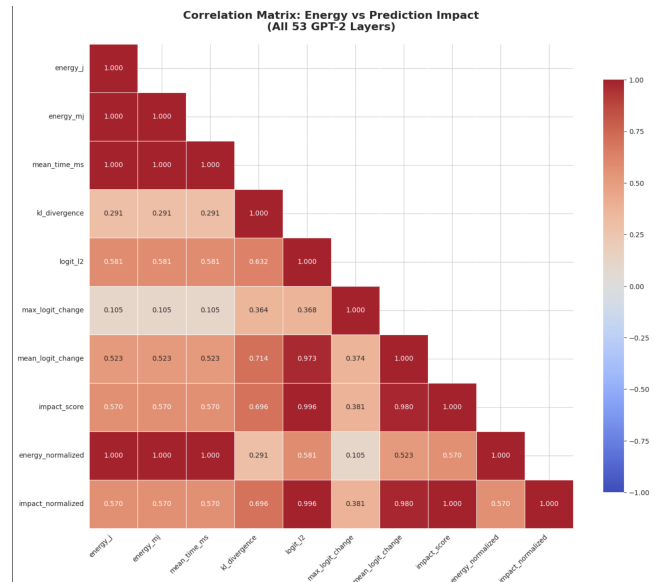


Figure 12: Correlation matrix comparing energy, timing, and prediction-impact metrics across all GPT-2 layers. Positive correlations between energy metrics and logit-shift metrics highlight the alignment between computational cost and predictive importance.

on the measured energy and importance profiles, the following guidelines emerge:

- **Avoid aggressive quantization** of the LM Head, embeddings, and LayerNorm layers. These components exhibit both high energy usage and high impact scores; reducing their precision is likely to degrade perplexity.

- **Light quantization** (e.g., FP16 or PTQ-calibrated INT8) may be applied safely to internal attention sublayers and MLP components, which show lower predictive impact relative to the LM Head but provide modest energy savings.
- **Dropout and miscellaneous residual layers** offer negligible energy savings and low prediction impact. While they can be quantized safely, doing so does not materially improve efficiency.

Overall, GPT-2’s layer-wise structure indicates that the most computationally expensive layers are inherently prediction-critical. This observation aligns with the theoretical design of decoder-only Transformers, where all generative probability mass flows through the embedding stack at the input and the output projection at the end. As a result, GPT-2 offers fewer low-risk quantization targets than encoder-only models such as DistilBERT.

5 Discussion

The results across both encoder-only and decoder-only transformer models show that reduced numerical precision can meaningfully improve inference efficiency on commodity GPUs equipped with Tensor Cores. On the NVIDIA T4, FP16 consistently reduced latency, increased throughput, and lowered energy per sample relative to FP32, with no measurable loss in accuracy for DistilBERT or perplexity for GPT-2 Small. These gains arise from reduced memory traffic and accelerated mixed-precision matrix operations, and although average board-level power remains similar, shorter runtimes translate directly into lower energy costs.

The two models, however, displayed sharply different internal energy-importance profiles. DistilBERT’s FFN layers dominated total energy while contributing little to prediction fidelity, producing a negative correlation between energy and impact metrics and creating a wide margin for safe quantization. GPT-2 Small showed the opposite pattern: its LM Head and embedding stack were both the most energy-intensive and the most critical for prediction quality, yielding a positive energy-importance correlation and limiting opportunities for aggressive precision reduction.

These observations emphasize that quantization strategies must be architecture-aware. Encoder-only models distribute predictive importance more evenly and place most computation in parameter-heavy FFNs, making them well suited for lower-precision inference. Decoder-only autoregressive models concentrate both computation and sensitivity in a small number of layers, restricting potential energy savings and requiring more conservative quantization choices. Although our hardware supported only FP32 and FP16 for standard PyTorch kernels, anchoring all comparisons to an FP32 baseline allows the conclusions to generalize to richer formats such as INT8, FP8, or mixed-precision Tensor Core kernels.

Finally, the study highlights limitations of GPU power measurement. Board-level telemetry from `nvidia-smi` provides coarse samples of total card power, including contributions from memory controllers and background activity, rather than isolated compute-unit consumption. While this restricts the precision of absolute energy estimates, it remains reliable for relative comparisons when workloads are fixed, pre-tokenized, and free of IO variation. The

methodology developed here therefore provides a practical template for evaluating and deploying reduced-precision transformer inference on commodity hardware.

6 Future Work

Several research directions follow naturally from this study. First, evaluating additional quantization formats such as INT8, FP8, and mixed-precision Tensor Core modes on newer architectures (Ampere, Hopper) would clarify how quantization interacts with hardware acceleration. Although the T4 includes Tensor Cores, standard PyTorch kernels do not expose full integer inference paths, so studying GPUs with native INT8 and FP8 support would provide a more complete picture of low-precision efficiency gains. Second, per-layer energy profiling can be extended to larger models and sparsity-aware kernels to examine how architectural scale changes the distribution of compute and predictive importance. Autoregressive models in particular may respond differently to structured sparsity or activation compression than encoder-only architectures. Third, integrating higher-frequency or on-chip power measurement tools (CUPTI, NVML polling hooks, Nsight traces) would improve temporal resolution and enable finer attribution of energy to compute, memory, and individual kernels, addressing the limitations of coarse board-level telemetry. Finally, evaluating complete serving pipelines—covering batching, caching, prompt formatting, and dynamic sequence-length adaptation—would yield a more realistic picture of energy behavior in deployed systems. Extending the zero-IO methodology to these settings would help guide future hardware-software co-design for energy-efficient LLM inference.

7 Conclusion

This work presented a systematic evaluation of how numerical precision affects the energy, latency, and accuracy characteristics of transformer-based language models on an NVIDIA T4 GPU with Tensor Cores. Using a controlled measurement harness with pre-tokenized inputs, zero-IO execution, and standardized timing and power collection, we isolated compute-level effects from noise sources such as tokenization, disk access, and host-device transfers. Across DistilBERT and GPT-2 Small, FP16 consistently reduced latency and energy per sample relative to FP32 while preserving accuracy or perplexity, demonstrating that substantial efficiency gains are achievable on commodity hardware.

Layer-wise analysis showed that model architecture strongly shapes quantization opportunities. DistilBERT’s FFN layers dominated compute energy yet had low predictive impact, making them strong candidates for aggressive precision reduction. GPT-2 Small, in contrast, concentrated both energy cost and predictive importance in its embedding stack and LM Head, limiting the amount of safe quantization without degrading generative quality.

Although the T4 exposes only FP32 and FP16 through standard PyTorch kernels, anchoring all comparisons to an FP32 baseline allows the conclusions to extend naturally to richer formats such as INT8, FP8, or mixed Tensor Core paths on newer GPUs. The methodology and findings provide practical guidance for energy-aware LLM deployment on widely accessible hardware and highlight the need for improved power telemetry and hardware-software co-design to further reduce inference energy at scale.

References

- [1] Francisco Caravaca, Ángel Cuevas, and Rubén Cuevas. 2025. From Prompts to Power: Measuring the Energy Footprint of LLM Inference. *arXiv:2511.05597* (2025).
- [2] Song Han, Huizi Mao, and William J. Dally. 2016. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In *International Conference on Learning Representations (ICLR)*.
- [3] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [4] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 3645–3650.

Received 3 December 2025