Alma Mater Studiorum · University of Bologna

Department of Computer Science and Engineering
Master's Degree in Computer Science

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

The Hidden Cost of Intelligence

Understanding and Compare the Energy Consumption of AI models

Supervisor:
Prof.
Maurizio Gabbrielli

Co-supervisors:
Prof. Fabrice Huet
Prof. Dino Lopez Pacheco

Presented by:
Erik Koci

II session
Academic Year 2024/25

"Success is not in what you have,
but who you are."

Bo Bennett

Abstract

L'impronta energetica dei moderni sistemi di intelligenza artificiale sta diventando una preoccupazione sempre più rilevante. Questa tesi analizza i consumi energetici dei Large Language Model (LLM) e dei modelli di Generative AI. L'analisi prende avvio dallo studio degli strumenti di monitoraggio energetico, come le Power Distribution Units (PDU) e le metriche a livello di GPU. I primi esperimenti sono stati condotti su una singola macchina, valutando diversi LLM tramite il framework Ollama e la piattaforma Hugging Face, permettendo un confronto diretto tra le richieste energetiche dei compiti generativi testuali e visivi. Per ampliare l'analisi, modelli di dimensioni maggiori sono stati testati su un'infrastruttura più potente (2 NVIDIA RTX A6000). È stato inoltre sviluppato un sistema di monitoraggio personalizzato in grado di raccogliere dati sincronizzati e ad alta risoluzione sia dalle GPU che dalle PDU. Successivamente sono stati effettuati anche dei benchmark per calcolare l'efficienza e la precisione dei Large Language Model attraverso il benchmark Humanity's Last Exam (HLE). I risultati evidenziano differenze significative in termini di efficienza energetica e performance tra tipologie di modelli e contesti di deployment, offrendo spunti concreti per pratiche di calcolo AI più sostenibili e fornendo indicazioni utili per future strategie di ottimizzazione su larga scala.

Table of Contents

1 Introduction

The field of artificial intelligence has witnessed remarkable advancements in recent years, particularly in the domains of natural language processing and image generation. Large Language Models (LLMs) have emerged as powerful tools capable of understanding and generating human-like text, while Diffusion Models have revolutionized the field of image generation, enabling the creation of high-quality, realistic images from textual descriptions.

These advances, while impressive, have raised concerns about the environmental impact of AI systems. The high energy consumption associated with training and deploying such models contributes to significant carbon emissions, prompting the need for sustainable AI practices and energy-efficient architectures. These technological come with significant computational and energy costs that warrant careful examination.

1.1 General Project Description

1.1.1 Background and Motivation

LLMs, operate by processing and generating text through complex neural network architectures. These models are trained on vast amounts of text data, learning patterns and relationships that enable them to perform tasks ranging from text completion to complex reasoning. The power consumption of these models varies significantly based on their architecture, size, and the specific task being performed, as demonstrated in recent studies.

Diffusion Models, on the other hand, represent a different approach to AI generation, focusing on image creation through an iterative denoising process. These models, often hosted on platforms, have gained popularity for their ability to generate high-quality images from textual prompts. The power consumption patterns of these models differ from LLMs due to their distinct architectural requirements and computational needs.

1.1.2 Research Objectives

This report aims to provide a comprehensive and depth overview of the state of the art in energy efficiency for large language models and diffusion models. It will detail major architectural innovations, advanced optimization techniques, energy monitoring tools, and existing research findings, with a specific focus on energy consumption. A key contribution of this work is the development of a custom monitoring framework designed to accurately track and analyze the power usage of AI models. This framework integrates both software-level and hardware-level metrics to offer high-resolution energy profiles of generative models in operation. The analysis will highlight the most recent developments, Special attention is given to measuring the power used during inference, in order to assess the energy efficiency of diffusion models relative to large language models (LLMs). Additionally, the research explores how energy usage scales as model size increases, especially when deployed on high performance computing infrastructures. To relate energy efficiency to model quality, we also evaluate LLMs using the Humanity's Last Exam (HLE) benchmark strictly for performance assessment, reporting category-level results to understand when high compute/power is necessary and when smaller models are sufficient.

1.1.3 Methodology Overview

The research begins with an in depth analysis of power monitoring tools, focusing on Power Distribution Units (PDUs) and GPU monitoring capabilities. Initial experiments were conducted on a server machine, establishing a baseline for power consumption analysis during idle states. Following these baseline measurements, we will expand our experiments to include more powerful servers. Furthermore, our investigation will not be limited to LLMs; we will also evaluate the power consumption of diffusion models. To evaluate various models, the study employs Ollama, an open-source framework that simplifies the deployment and management of Large Language Models (LLMs) and huggingface, an open source community that has become a central hub for artificial intelligence.

The methodology employed in this study combines software-level monitoring through NVIDIA's System Management Interface (nvidia-smi) with hardware-level power tracking using Simple Network Management Protocol (SNMP). This dual approach enables parallel data collection with high temporal resolution, providing detailed insights into power consumption patterns during model inference.

1.1.4 Research Scope

The research scope is to include Diffusion Models and Text Models, sourced from Huggingface and Ollama, allowing for a comparative analysis of power consumption patterns between language models and image generation models. To evaluate the power consumption of larger models, we used another computing infrastructure, allowing experiments with more complex and resource intensive models. The results of this study are intended to inform developers and infrastructure designers about the energy implications of deploying different types of generative models.

2 State of the Art

2.1 The Context of Energy Efficiency in AI

The rapid advancement and adoption of Large Language Models (LLMs) and generative artificial intelligence have triggered a significant increase in computational demands and, consequently, energy consumption. This trend raises substantial concerns regarding sustainability and environmental impact, making energy optimization a critical priority. [9]

Projections from authoritative bodies such as the International Energy Agency (IEA) indicate that global electricity demand from data centers is set to double by the end of this decade, with AI emerging as the primary driver of this growth. In particular, the electricity demand for AI-optimized data centers is predicted to quadruple by 2030.[8] Similarly, in the United States, AI workloads are estimated to consume between 9.1% and 11.7% of total energy demand by 2030.[6]

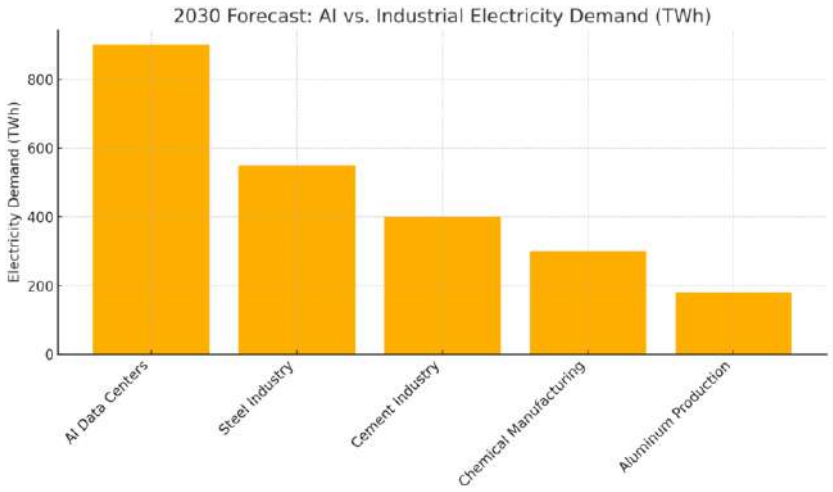


Figure 1: Estimated AI related electricity consumption across leading US states in 2030 [15]

As shown in Fig. 1, the projected electricity consumption for AI-related workloads varies significantly across different US states, with California, Texas, and New York showing the highest projected consumption. This geographic distribution reflects the concentration of data centers and AI infrastructure in these regions, highlighting the need for region-specific energy optimization strategies.

It is crucial to distinguish between energy consumed during the training phase and that during inference. Although training is a one-time, energy-intensive process, inference represents continuous demand, as models are employed to serve millions of queries daily. This continuous nature implies that the cumulative energy footprint of inference can become significantly larger over time.[1]

The scale of inference demand is substantial and growing rapidly. Recent industry reports indicate that major AI platforms handle hundreds of millions of queries daily. For instance, ChatGPT alone processes approximately 1 billion queries per day globally. This figure represents a conservative estimate that excludes specialized AI applications in healthcare, finance and other sectors.[5]

For example, GPT-3 inference, estimated at 0.0003 kWh per query, drastically increases when multiplied by millions of daily users, posing a considerable challenge. [6]

The energy challenge is of paramount importance for the sustainable development and large-scale adoption of AI technologies. Industry focus is shifting from creating larger and more capable models to developing smarter, more efficient models, thereby balancing cutting-edge capabilities with environmental responsibility. Greater efficiency in LLMs is not only an environmental imperative, reducing electricity consumption and carbon emissions, but also an economic advantage, lowering operational costs and making AI more accessible to a wide range of users and organizations. [14]

A crucial aspect to consider is the trend where individual AI models are becoming significantly more energy-efficient, as demonstrated by the 40-60% reduction in DeepSeek's consumption or Llama 3.3 70B's 120 times higher efficiency compared to older GPT-3 estimates.[3] However, in parallel, global consumption of AI energy is projected to drastically increase globally.[6]

Moreover, while training is energy intensive, inference, being a continuous and large-scale process, represents a broader and more persistent energy challenge. Meta reports that inference workloads constitute up to 70% of their AI energy consumption, and Google attributes 60% of its ML energy to inference.[6] The cumulative nature of inference energy consumption, fueled by millions of daily queries, means its long term environmental and economic impact can far outweigh one time training costs. This continuous demand places unique pressures on data center infrastructure, requiring constant power and cooling. Optimization strategies for inference, which include real time processing, batching, and low-latency requirements, differ significantly from those for training, which focus on maximizing throughput for large datasets. This understanding indicates a critical shift in the orientation of energy optimization efforts within the AI community. While training optimization remains important, the primary emphasis of research and development should increasingly focus on real world deployment efficiency, including inference engine optimization, the development of specialized hardware for serving, and the implementation of intelligent workload management.

2.2 State of the Art on LLMs and Energy Efficiency

2.2.1 Energy Efficiency Research Landscape

The field of energy-efficient AI has evolved rapidly, with research focusing on multiple optimization dimensions. Recent studies have identified several key areas where significant energy savings can be achieved:

Model Compression Techniques: Quantization has emerged as one of the most effective methods for reducing energy consumption. Post-training quantization (PTQ) can reduce model size by 75% with minimal accuracy loss, while quantization-aware training (QAT) achieves even better results. Dynamic quantization, which adapts precision based on layer importance, has shown particular promise for maintaining performance while maximizing energy savings.

Architectural Innovations: The development of sparse architectures, particularly Mixture of Experts (MoE) models, represents a paradigm shift in energy-efficient design. These models achieve significant computational savings by activating only a subset of parameters for each input, with some implementations showing 60-80% reduction in active parameters during inference.

Hardware-Software Co-optimization: The integration of specialized hardware accelerators with optimized software frameworks has enabled substantial energy efficiency gains. Tensor Processing Units (TPUs), Neural Processing Units (NPUs), and optimized GPU architectures specifically designed for transformer workloads have demonstrated 2-5x improvements in energy efficiency compared to general-purpose hardware.

Inference Optimization Strategies: Research has identified several inference-level optimizations that can significantly impact energy consumption. These include:

- **Early Exit Mechanisms:** Allowing models to terminate computation early for simple inputs
- **Adaptive Computation:** Dynamically adjusting computational resources based on input complexity
- **Knowledge Distillation:** Training smaller, more efficient models to mimic larger ones
- **Pruning:** Removing redundant parameters while maintaining model performance

2.2.2 Benchmarking and Evaluation Methodologies

The establishment of standardized benchmarks for energy efficiency has been crucial for advancing the field. Recent initiatives include:

MLPerf Inference: Provides standardized benchmarks for measuring inference performance and energy consumption across different hardware platforms and model architectures. The benchmark includes specific energy efficiency metrics that enable fair comparison between different optimization techniques.

Green AI Benchmarks: Several research groups have developed specialized benchmarks focusing on energy consumption, including measurements of carbon footprint, energy per inference, and efficiency scaling across different model sizes.

Real-world Deployment Studies: Large-scale studies of production AI systems have provided valuable insights into actual energy consumption patterns, revealing significant discrepancies between theoretical efficiency gains and real-world performance.

2.2.3 LLM Families Architectures

LLM models are rapidly evolving, with increasing attention to energy efficiency through architectural innovations and software/hardware optimizations. In the following, we list the main models and their characteristics, particularly focusing on aspects related to their energy footprint and design.

2.2.4 Llama Family

NVIDIA has strategically optimized its Llama model family for compute efficiency and maximum inference throughput when deployed on NVIDIA accelerated infrastructure. This optimization includes model compression techniques, such as pruning, followed by retraining with distillation and alignment methods. The goal is to produce smaller models that maintain high accuracy while achieving superior throughput.[11]

2.2.5 DeepSeek Family

DeepSeek models are presented as an innovative AI technology that radically redefines energy efficiency in computational intelligence. They are explicitly designed with a strong focus on sustainability, aiming to be more environmentally friendly than traditional AI models. The core of DeepSeek's revolutionary approach is its Mixture of Experts (MoE) architecture. Unlike dense models that process every input through the entire neural network, DeepSeek intelligently routes tasks to specific *expert* [1] subnets that are best suited to handle them. This targeted approach significantly reduces unnecessary computational overhead.[3]

DeepSeek's efficiency is further enhanced by specific algorithmic innovations[3]:

- **Dynamic Resource Allocation** ensures simpler tasks consume less energy than complex ones
- **Sparse Activation** means only relevant neural pathways are engaged during processing
- **Intelligent Routing Mechanisms** determine the most efficient computational path, minimizing energy waste

DeepSeek claims to reduce energy consumption by approximately 40-60% compared to traditional AI models. This directly translates into significant reductions in carbon emissions and notable operational cost savings for organizations that deploy AI technologies.

2.2.6 Mistral Family

Mistral AI distinguishes itself by developing LLMs that are both highly efficient and accessible. Their models are designed to be lightweight, efficient, and scalable, requiring fewer computational resources while maintaining state of the art accuracy.[17] The **Mixture of Experts** (MoE) architecture is a central feature for many Mistral models. Mixtral's design uses 8 experts but only activates 2 at a time for each token, leading to a significant reduction in computation per token.[1] This selective activation can theoretically reduce energy consumption by up to 75% compared to dense models, making it a significant factor for scaling AI sustainably. However, a key consideration for MoE models like Mixtral is that inactive experts still consume GPU memory and incur maintenance overhead, which can increase costs, especially when multiple MoE models are running concurrently on the same hardware.[1] Mistral also, employs innovative techniques such as Sliding Window Attention (SWA) and a Rolling Buffer Cache to enhance inference speed and reduce memory requirements. SWA allows each token to attend to a limited window of previous tokens, mitigating the complexity.[4]

2.2.7 Qwen Family

The Qwen series of LLMs, developed by Alibaba Cloud, is designed with a strong emphasis on efficiency and scalability.[2] The Qwen3 dense base models demonstrate performance comparable to Qwen2.5 base models with a higher parameter count. This efficiency gain

[1] Experts are like specialized neural subnetworks, specialized modules that work together, but in a coordinated and selective manner, to process information more efficiently and powerfully within AI models.

is attributed to advancements in model architecture, an expanded training dataset, and more effective training methodologies. The Mixture of Experts (MoE) architecture is also adopted by Qwen. Qwen3 introduces **Hybrid Thinking Modes** (a "Thinking Mode" for complex, step-by-step reasoning, and a "Non-Thinking Mode" for rapid, instant responses). This flexibility allows users to configure specific computational budgets for tasks, enabling a more optimal balance between cost-efficiency and inference quality based on task demands.[12]

2.2.8 Gemma Family

Gemma 3 models developed by Google are explicitly designed to run quickly and efficiently directly on a wide range of devices, from mobile phones and laptops to workstations. They are available in various sizes, enabling developers to select the best model for their specific hardware and performance needs. A key feature of Gemma 3 is the inclusion of official **quantized versions**, to reduce model size and computational requirements while maintaining high accuracy for example: a 4 GB model in float32 can be reduced to just 1 GB in int8, while maintaining similar performance. This makes it feasible to run locally on a laptop instead of relying solely on the cloud. This directly contributes to faster performance and lower energy consumption.[7] Significant architectural changes in Gemma 3 aim to improve memory efficiency, particularly in reducing the use of KV cache memory, which tends to grow with long context lengths. Gemma models are highly optimized for a wide range of hardware platforms. This broad hardware compatibility underscores their design philosophy for efficiency and wide portability across diverse computing environments.

The following table summarizes the efficiency characteristics of these LLM families:

LLM Family	**Key Architectural Innovations**	**Efficiency Benefits (Claimed/Measured)**
Llama	NVIDIA optimization: pruning, distillation, alignment	Smaller models, high accuracy, superior throughput
DeepSeek	MoE, Dynamic Resource Allocation, Sparse Activation, Intelligent Routing	40-60% energy reduction
Mistral	MoE (selective activation), SWA, Rolling Buffer Cache	Up to 75% energy reduction (theoretical), faster inference, less memory
Qwen	MoE, Hybrid Thinking Modes	Similar performance with fewer active parameters, flexible computational budgets
Gemma	Quantized versions, KV cache optimization, broad hardware compatibility	Fast, efficient on diverse devices, reduced compute/energy

While the table focuses on LLMs, it's worth noting the rise of text diffusion models. Unlike the token-by-token generation of traditional LLMs, these models create text by iteratively refining a noisy input. This different approach offers a speed advantage, with recent advancements like Google Gemini Diffusion and Mercury Inception Labs showing significantly faster inference speeds (e.g., over 1000 tokens/second for Mercury Coder

compared to around 200 tokens/second for optimized autoregressive models). This speed comes from their ability to process multiple tokens in parallel, rather than sequentially.

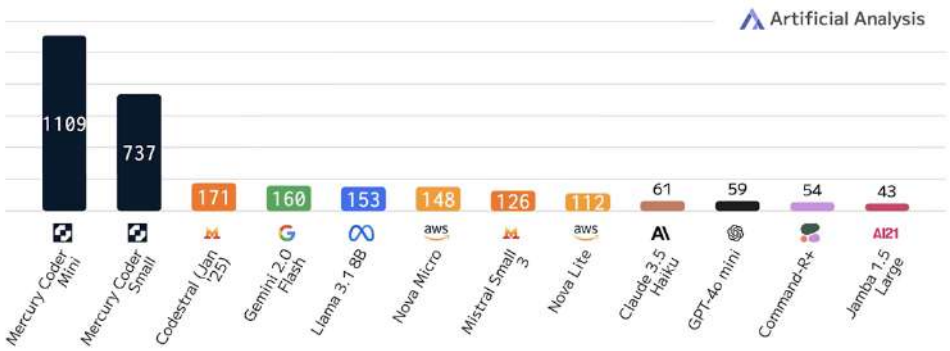


Figure 2: Speed comparison between diffusion text models and traditional autoregressive models [10]

Fig. 2 illustrates the substantial speed advantages of diffusion-based text generation compared to traditional autoregressive models. The Mercury Coder model achieves over 1000 tokens/second, representing a 5x improvement over optimized autoregressive models that typically achieve around 200 tokens/second. This performance gain directly translates to reduced energy consumption per token generated, making diffusion models an attractive alternative for energy-conscious applications.

2.2.9 Recent Advances in Energy Optimization

Transformer Optimization Techniques: Recent research has focused on optimizing the attention mechanism, which typically accounts for 30-50% of total computation in transformer models. Techniques such as sparse attention patterns, linear attention approximations, and attention pruning have shown significant energy savings while maintaining model performance.

Memory-Efficient Architectures: The development of memory-efficient transformer variants, such as FlashAttention and its successors, has reduced memory requirements by 50-80% during training and inference. These optimizations directly translate to lower energy consumption by reducing memory bandwidth requirements and enabling larger batch sizes.

Dynamic Model Scaling: Recent work on dynamic model scaling allows models to adapt their computational requirements based on input complexity. This approach can reduce average energy consumption by 40-60% for mixed-complexity workloads while maintaining performance on challenging tasks.

Neural Architecture Search (NAS) for Efficiency: Automated architecture search techniques have been applied to discover energy-efficient model architectures. These methods have identified novel architectural patterns that achieve comparable performance with significantly reduced computational requirements.

2.2.10 Industry Adoption and Production Considerations

The transition from research prototypes to production systems has revealed several important considerations for energy-efficient AI deployment:

Latency vs. Efficiency Trade-offs: Production systems often require strict latency constraints that can conflict with energy optimization goals. Research has shown that careful tuning of batch sizes, model parallelism strategies, and hardware selection can achieve both low latency and high energy efficiency.

Scalability Challenges: While individual model optimizations show promising results, scaling these techniques to large-scale production systems presents unique challenges. Issues such as load balancing, resource allocation, and system-level optimizations become critical for achieving overall energy efficiency.

Monitoring and Measurement: Accurate measurement of energy consumption in production environments requires sophisticated monitoring infrastructure. Recent work has developed standardized methodologies for measuring and reporting AI energy consumption, enabling better comparison and optimization across different systems.

2.3 State of the Art in Diffusion Models and Energy Efficiency

The core architecture of Stable Diffusion consists of three main components [13]:

- **Variational Autoencoder** (VAE): plays a crucial role in efficiency by compressing the image from a high-dimensional pixel space into a smaller, more semantically meaningful latent space. This latent representation requires substantially less memory, which translates to faster inference and reduced computational requirements
- **U-Net neural network**: is the central denoising component, operating iteratively in the latent space. It learns to progressively remove Gaussian noise from a noisy latent representation, guided by the text prompt.[36] The U-Net is often the most computationally intensive part of the Stable Diffusion pipeline and is typically memory-bound, meaning that the GPU's VRAM bandwidth can be a significant bottleneck for image generation speed.
- **CLIP text encoder**: converts the input text prompt into a numerical embedding (a vector representation) that provides semantic guidance for the U-Net during the denoising process, ensuring the generated image aligns with the text description.

The image generation process involves iteratively denoising random noise over a configurable number of steps. A critical factor influencing energy consumption is the number of these inference steps: more steps lead to longer generation times and, consequently, higher energy usage.[18] Compared to LLM decoding, diffusion models tend to be more computationally intensive.

2.3.1 Diffusion Model Families Architectures

Diffusion models, much like LLMs, are rapidly evolving, with increasing attention to energy efficiency through architectural innovations and software/hardware optimizations. In the following, we list the main model families and their characteristics.

—

2.3.2 Stable Diffusion Family

The **Stable Diffusion** family, developed by Stability AI, has democratized access to AI image generation, with a growing focus on efficiency. Subsequent versions have introduced

more complex architectures to improve quality, while also considering sustainability and accessibility. The optimization of the **latent space** has been a key characteristic since the earliest versions, allowing the model to operate on a compressed representation of the image, significantly reducing computational requirements compared to models that operate directly in pixel space.

2.3.3 Midjourney Family

Midjourney, a proprietary model, is renowned for its ability to generate high quality artistic images with a unique aesthetic. While its internal architectures are not publicly detailed, it is plausible that the model's evolution towards faster and more detailed versions involves computational optimizations. The focus on generation speed, evident in recent versions, suggests an emphasis on inference efficiency.

2.3.4 DALL-E Family

OpenAI's **DALL-E** series has played a pioneering role and continues to be at the forefront of AI image generation. More recent versions, such as DALL-E 3, have been optimized not only for image quality but also for prompt understanding efficiency, a crucial aspect for reducing iterations.

2.3.5 Imagen Family

Google AI's **Imagen** models have highlighted the importance of large language models for text understanding in image generation. While not open source, their research has influenced the entire field.

These families represent the pillars of current development in AI image generation via diffusion, each uniquely contributing to the advancement and accessibility of this technology, with an increasing emphasis on energy efficiency.
The following table summarizes the efficiency characteristics of these diffusion model families:

Model Family	**Key Architectural Features**	**Efficiency Benefits (Claimed/Observed)**
Stable Diffusion	Latent space processing, U-Net backbone, VAE compression, CLIP/dual encoders in later versions	Reduced memory and compute via latent representation, optimizations improve speed and quality over time
Midjourney	Proprietary, likely transformer-based with fine-tuned aesthetic priors	Fast inference, highly optimized for GPU runtime, artistic output with minimal prompt tuning
DALL-E	Diffusion + transformer-based prompt encoder, guided decoding, integrated with language models (GPT)	Strong prompt-image alignment reduces retries, efficient prompt understanding in DALL-E 3

Model Family	Key Architectural Features	Efficiency Benefits (Claimed/Observed)
Imagen	Large transformer-based text encoder + diffusion decoder pipeline	Leverages LLMs for superior text understanding, efficient decoding pipeline for high fidelity generation

3 Performance Evaluation

3.1 Experimental Scope

This section details the experimental setup and methodology employed to evaluate the energy consumption and performance of various Large Language Models (LLMs) and Diffusion Models. The evaluation was conducted across two distinct hardware environments to provide a comprehensive analysis of energy efficiency under varying computational capacities.

3.1.1 Server Environment

Initial experiments were conducted on a server configured with consumer grade hardware. The specifications of this server are as follows:

Component	**Specification**
Operating System	Ubuntu 22.04.5 LTS x86_64
Host	MS-7C56 1.0
Kernel	6.5.0-41-generic
CPU	AMD Ryzen 5 5600X (12 cores) @ 3.700GHz
GPU	NVIDIA GeForce RTX 3080 Lite Hash Rate
Memory	32 GB

Table 3: Server Hardware Specifications

In addition to the consumer-grade server, experiments were also executed on a second machine with the following specifications:

Component	**Specification**
Operating System	Ubuntu Linux (x86_64)
Host	gpu2
Kernel	`5.15.0-144-generic`
CPU	Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz
GPU	NVIDIA RTX A6000
Memory	100 GB

Table 4: GPU2 Machine Hardware Specifications

A wide range of LLMs and Diffusion Models were selected for testing to assess their energy footprint across different architectures and parameter sizes. The models tested are summarized in Table:

Model Family	Specific Models/Sizes
Llama	1B, 3B, 8B, 70B
DeepSeek-R1	1.5B, 7B, 8B, 70B
Mistral	7B, 12B
Qwen	1.8B, 4B, 7B, 72B
Gemma3	470M, 1B, 4B
Stable Diffusion	1B, 1.5B, 8B

Table 5: Summary of LLM and Diffusion Models Tested

3.1.2 Measurement Methodology

Custom Python and Bash scripts were developed to automate the calculation of computational metrics and energy consumption. GPU power usage was monitored using `nvidia-smi`, while overall server power consumption was tracked via a Power Distribution Unit (PDU) using `snmpget`. This dual-source monitoring provided a comprehensive view of energy draw.

The monitoring process was segmented into distinct phases to enable granular analysis of energy consumption during different operational stages. The **Baseline** phase established a power baseline by measuring the machine in an idle state. During **Server Startup**, energy consumption was tracked while initializing and loading the model onto the server. The **Inference** phase captured power draw specifically during active model inference processing. **Response Received** measured energy used between the completion of inference and the full receipt of the model's output. Finally, **Server Shutdown** tracked power consumption during the graceful termination of the model server.

All tests were executed multiple times to ensure data reliability and statistical significance. Beyond power consumption, additional critical data points were recorded, including RAM utilization, GPU temperature, and cumulative energy consumption in Joules. The generated model responses were also saved to correlate the number of generated tokens or words with the corresponding energy expenditure.

3.1.3 Statistical Reliability and Experimental Design

To ensure robust and statistically significant results, a comprehensive experimental design was implemented with multiple independent runs for each model. Each model was tested with 10 independent runs to account for system variability and ensure reproducibility. This approach allows for the calculation of confidence intervals and statistical significance testing, helping to identify and account for background process interference, memory allocation patterns, GPU clock speed fluctuations, and PDU idle power consumption.

The collected data was subjected to rigorous statistical analysis including calculation of mean and standard deviation for all energy consumption metrics across multiple runs. 95% confidence intervals were computed for all reported measurements, and timestamps were aligned using predefined intervals to ensure consistent temporal analysis across different measurement sources.

Given the multi-stage measurement process, error propagation was carefully analyzed. Baseline power consumption was measured and subtracted to account for systematic errors and system overhead, while multiple measurements helped reduce the impact of

random fluctuations. To ensure reproducibility, all measurements were conducted using automated scripts to eliminate human error, with complete experimental logs including timestamps, system states, and environmental conditions.

The framework implements some personalization, like a crafted prompt is used across all evaluations to ensure comparable workloads:

```
Write a detailed analysis of the impact
of artificial intelligence on modern healthcare, focusing
on both benefits and challenges. Include specific examples
of AI applications in diagnosis, treatment, and patient care.
```

This prompt was specifically designed to evaluate consistent reasoning and structured output generation across different model architectures. The prompt requires models to engage in complex analysis, synthesis of information, and coherent argumentation - cognitive tasks that represent typical real-world usage patterns for AI systems.

For diffusion model evaluation, a standardized image generation prompt was employed to ensure comparable computational workloads:

```
A breathtaking sunset over the French Riviera, the beach should
have the rocks, highly detailed, 4K resolution. The sky painted
in vivid red and orange tones, reflecting over the calm
Mediterranean Sea. In the foreground, a wooden sign with the
word "NIZZA" clearly written on it. Warm light, cinematic
atmosphere, photorealistic details.
```

This image generation prompt was selected to provide a standardized, computationally intensive workload that tests the full capabilities of diffusion models while maintaining consistency across different architectures. The prompt requires complex scene composition, detailed rendering, and specific textual elements, ensuring that all models perform comparable computational work during energy measurement.

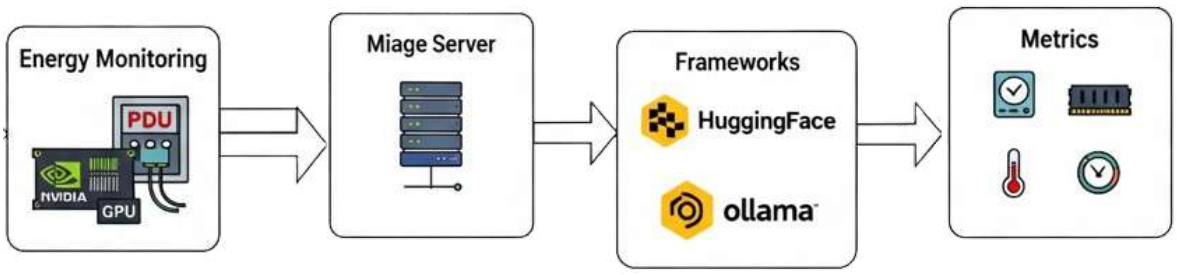


Figure 3: Experimental setup architecture

Fig. 3 presents the comprehensive experimental setup architecture, illustrating how hardware-level power monitoring via PDU integrates with software-based GPU monitoring through nvidia-smi. This dual-source monitoring approach ensures accurate measurement of both system-wide energy consumption and component-specific power draw, providing a complete picture of the energy footprint during LLM inference operations.

4 Deployment of LLMs

This chapter details the deployment and performance characteristics of various LLM and diffusion model families across different hardware configurations. The experimental evaluation was conducted in two phases: initial testing on consumer-grade hardware with 10GB GPU memory, followed by comprehensive evaluation on enterprise-grade hardware with 100GB GPU memory to accommodate larger models. This progression allows for a complete analysis of energy consumption patterns across the full spectrum of model sizes and capabilities.

4.1 Experimental Progression and Hardware Evolution

The experimental evaluation was designed to provide comprehensive coverage of energy consumption patterns across different model sizes and hardware configurations. The study progressed through two distinct phases:

Phase 1 - Consumer-Grade Hardware (10GB GPU): Initial experiments were conducted on consumer-grade hardware with 10GB GPU memory, focusing on smaller to medium-sized models (1B-8B parameters). This phase established baseline energy consumption patterns and validated the measurement methodology.

Phase 2 - Enterprise-Grade Hardware (100GB GPU): The evaluation was extended to enterprise-grade hardware with 100GB GPU memory to accommodate large-scale models (70B+ parameters). This phase enabled comprehensive analysis of energy consumption patterns across the full spectrum of model sizes.

This progression ensures that the energy consumption analysis covers the complete range of model sizes from small, efficient models suitable for edge deployment to large, state-of-the-art models requiring substantial computational resources.

4.2 Complete Monitoring phases

The complete monitoring approach captures energy consumption across all operational phases of LLM inference. This comprehensive analysis provides insights into how energy usage varies throughout the model lifecycle, from initialization to completion. The monitoring framework tracks power consumption during five distinct phases: baseline idle state, server startup, active inference, response generation, and server shutdown.

This phase-based analysis reveals critical insights into energy efficiency patterns. For instance, the startup phase typically shows a spike in power consumption as the model loads into GPU memory, while the inference phase demonstrates sustained high power usage during active computation. Understanding these patterns is essential for optimizing deployment strategies and identifying opportunities for energy savings.

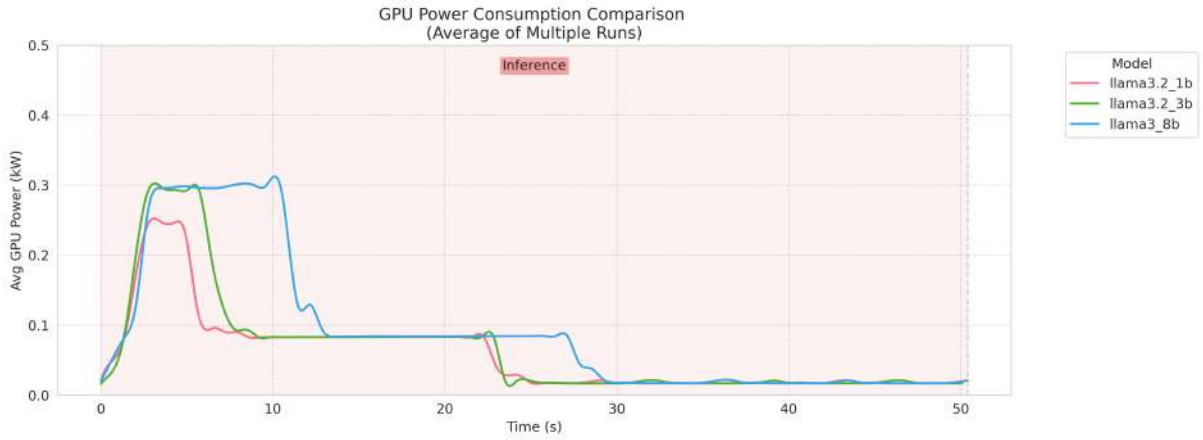


Figure 4: Average GPU power consumption (kW) across different Llama model sizes

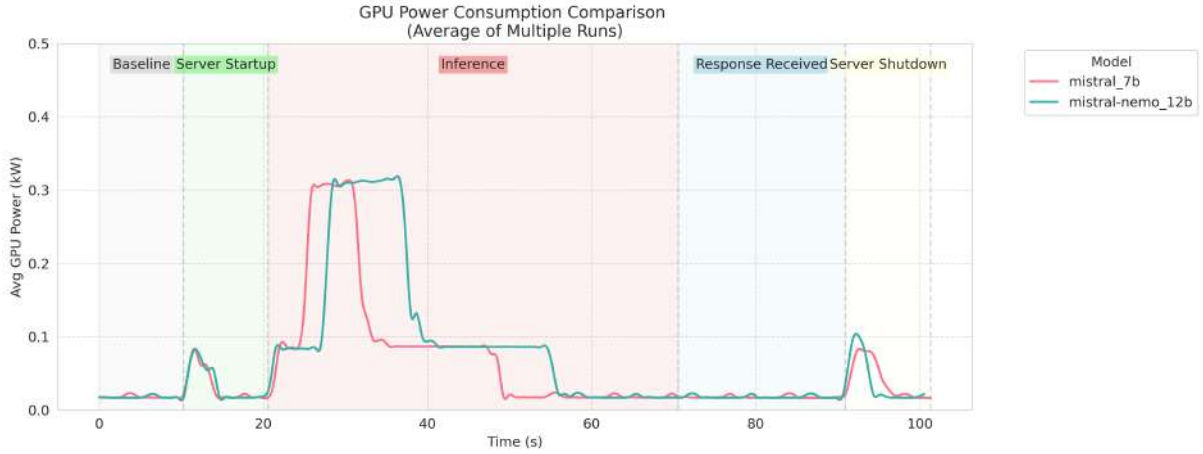


Figure 5: Average GPU power consumption (kW) across different Mistral model sizes

The power consumption analysis reveals consistent patterns across model families. Fig. 4 shows that Llama models exhibit a clear linear relationship between model size and GPU power consumption, with larger models requiring proportionally more energy. Similarly, Fig. 5 demonstrates that Mistral models follow comparable scaling patterns, confirming that energy consumption is primarily determined by model size rather than architectural differences within the same family.

4.3 Power usage of the Inference (understanding PDU & GPU inference phase)

This subsection provides a detailed analysis of power usage patterns during the inference phase, utilizing both software-based GPU monitoring (nvidia-smi) and hardware-based system monitoring (PDU). The dual-source approach provides a comprehensive view of energy consumption at both the component level and the system level.

The GPU monitoring via nvidia-smi captures detailed metrics including power draw, memory usage, temperature, and utilization during inference. This software-based approach provides granular insights into how the GPU processes the computational workload and how different model architectures utilize GPU resources.

In parallel, the PDU monitoring provides hardware-level measurements of total system power consumption, including all components such as CPU, memory, storage, and cooling systems. This hardware-based approach captures the complete energy footprint of the inference process, accounting for both direct computational costs and system overhead.

The comparison between GPU-specific and system-wide power measurements reveals important insights about the efficiency of the overall system architecture. The ratio between GPU power consumption and total system power consumption indicates how much of the energy is directly used for computation versus system overhead.

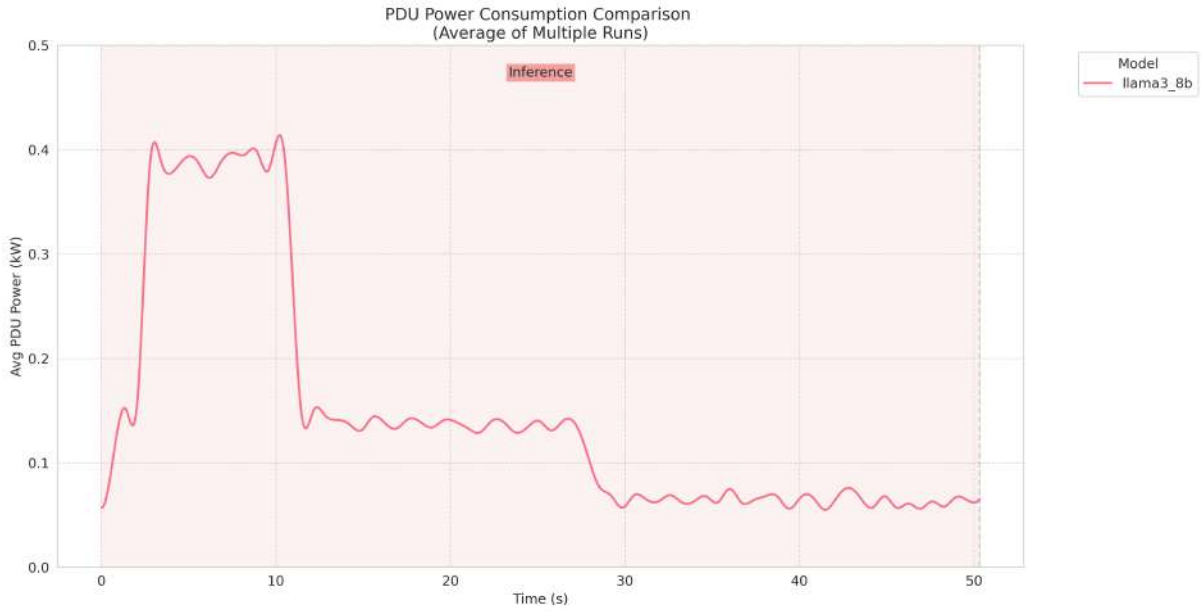


Figure 6: System-wide power consumption (kW) measured via PDU for Llama

Fig. 6 presents the system-wide power consumption measured through the Power Distribution Unit (PDU), which captures the total energy draw including CPU, memory, storage, and cooling systems in addition to GPU consumption. This comprehensive measurement reveals the complete energy footprint of the inference process, providing insights into the ratio between direct computational energy and system overhead.

4.4 Cumulative Energy Inference

The cumulative energy inference analysis provides a temporal perspective on energy consumption during the inference phase, tracking the total energy consumed in Joules over time. This approach offers valuable insights into the energy accumulation patterns and power draw consistency throughout the inference process.

The cumulative energy curves are monotonically increasing and nearly linear, indicating a steady power draw throughout inference. This linear behavior suggests that the energy consumption rate remains relatively constant during the inference phase, which is characteristic of well-optimized model inference where computational load is distributed evenly across the processing time.

The nearly linear nature of these curves also provides insights into the computational consistency of the inference process. Unlike training operations that may exhibit variable power consumption patterns, inference typically shows more predictable and steady energy usage, making it easier to estimate energy costs for production deployments.

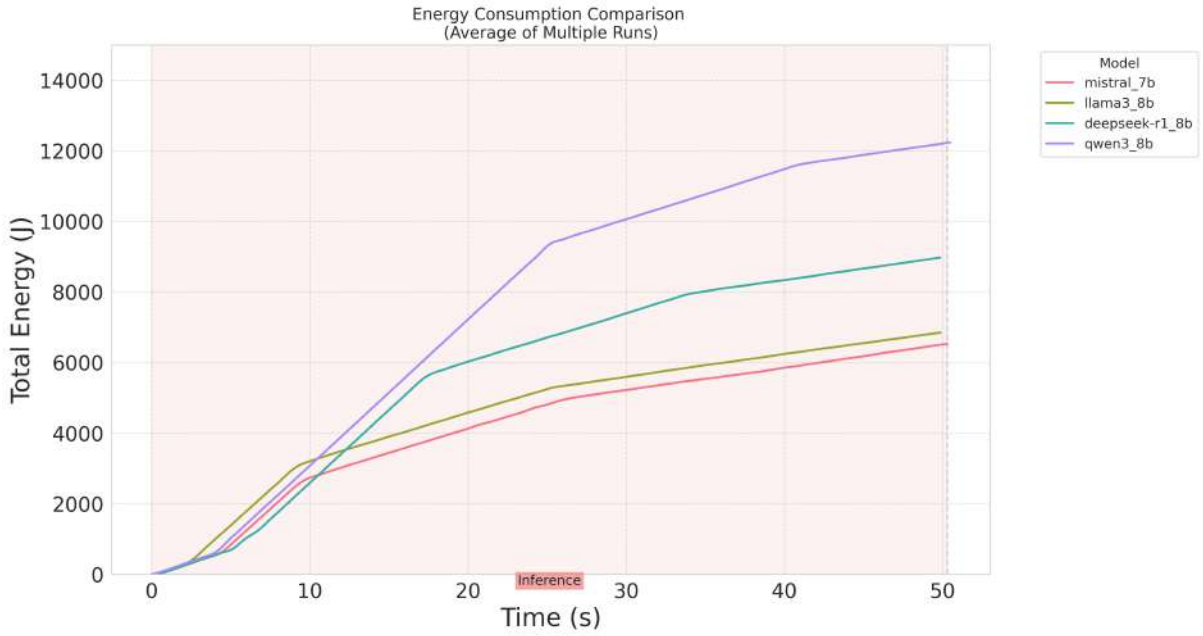


Figure 7: Total energy consumption (Joules) across different model families and sizes

Fig. 7 provides a comprehensive comparison of total energy consumption across different model families and sizes. The results clearly demonstrate the linear relationship between model size and total energy consumption, with larger models requiring proportionally more energy to complete the same inference task. This linear scaling pattern is consistent across all tested model families, providing valuable insights for energy-aware model selection.

4.5 Energy per output word (ratio between energy and words generated)

The energy per output word metric provides a normalized measure of energy efficiency that accounts for the actual output generated by each model. This ratio is calculated by dividing the total energy consumed during inference by the number of words generated in the response, offering a standardized efficiency metric across models of different sizes and capabilities.

This analysis is particularly valuable for comparing the energy efficiency of different model architectures, as it normalizes energy consumption by the actual work performed (word generation). Models that generate more content with less energy demonstrate superior efficiency in this metric.

4.5.1 Quality-Aware Energy Efficiency Analysis

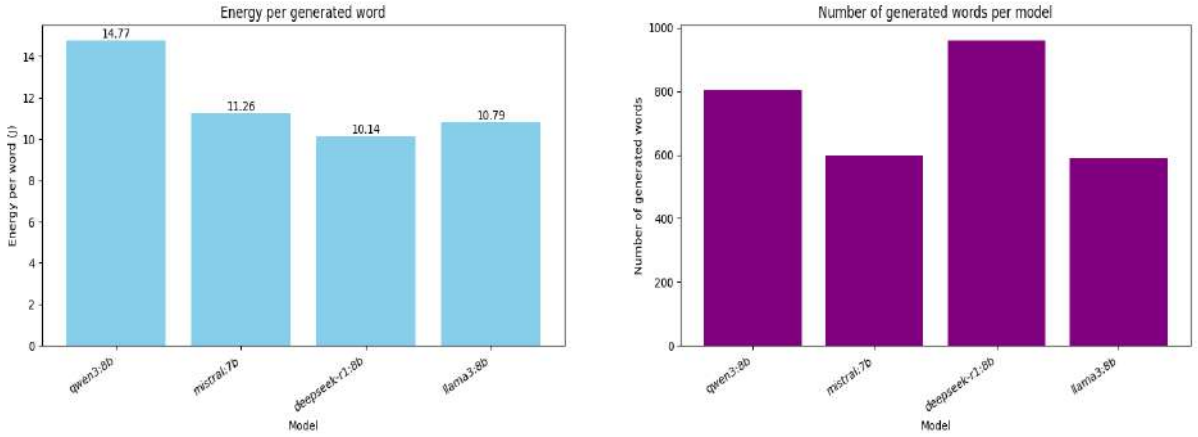
While the energy per word metric provides valuable insights into computational efficiency, it is crucial to consider the quality of the generated content when evaluating overall efficiency. A model that generates fewer words but produces higher-quality, more coherent responses may be more energy-efficient in practice than a model that generates more words of lower quality.

The relationship between energy consumption and output quality can be analyzed through several dimensions:

Coherence and Relevance: Models that generate more coherent and relevant responses often require less energy per useful word, as they avoid generating redundant or off-topic content. This is particularly evident in specialized domains where precise, accurate responses are more valuable than verbose outputs.

Task-Specific Efficiency: Different tasks require different levels of computational complexity. For example, creative writing tasks may benefit from models that can generate longer, more elaborate responses, while factual question-answering tasks may be more efficient with concise, accurate responses.

Error Correction and Iteration: Models that generate higher-quality initial responses reduce the need for multiple iterations or corrections, leading to lower overall energy consumption for achieving the desired output quality.



(a) Energy consumption per generated word across different model families and sizes

(b) Total words generated by each model during the standardized inference task

Figure 8: Energy efficiency analysis

Fig. 12 presents a comprehensive analysis of energy efficiency normalized by output quality. Fig. 8a shows the energy consumption per generated word, revealing significant variations in efficiency across different model families. Interestingly, larger models often demonstrate better energy efficiency per word due to their superior language understanding capabilities, which can lead to more coherent and efficient text generation. Fig. 8b displays the total output volume for each model, showing that while some models generate more content, the quality and coherence of that content must be considered when evaluating overall efficiency.

4.5.2 Benchmark Performance and Energy Correlation

The analysis of energy per word becomes particularly meaningful when correlated with benchmark performance results from Section 6. The Humanity's Last Exam (HLE) benchmark results provide valuable insights into how energy efficiency relates to model performance across different academic domains.

Performance-Energy Trade-offs: Models that achieve higher accuracy on the HLE benchmark often demonstrate better energy efficiency per word when generating high-quality responses. This correlation suggests that architectural optimizations that improve reasoning capabilities can also lead to more efficient text generation.

Domain-Specific Efficiency: The HLE results reveal that models excel in different academic domains. When these performance patterns are considered alongside energy consumption data, it becomes clear that energy efficiency should be evaluated in the context of the specific tasks and domains where the model will be deployed.

Scaling Efficiency: The relationship between model size, benchmark performance, and energy consumption reveals important scaling patterns. Larger models often achieve better performance per unit of energy when the quality of output is considered, suggesting that the energy investment in larger models can be justified by their superior reasoning capabilities.

4.6 Large-Scale Model Evaluation (70B+ Parameters)

To provide a comprehensive analysis of energy consumption patterns across the full spectrum of model sizes, the evaluation was extended to include large-scale models with 70B+ parameters. These models require enterprise-grade hardware with substantial GPU memory (100GB+).

Hardware Configuration for Large Models

The large-scale model evaluation was conducted on enterprise-grade hardware specifically designed to accommodate models with 70B+ parameters. This hardware configuration provides the necessary computational resources and memory capacity to run these models efficiently while maintaining accurate energy consumption measurements.

Energy Consumption Analysis for Large Models

The evaluation of large-scale models reveals important insights into the scaling behavior of energy consumption for state-of-the-art LLMs. These models represent the current frontier of AI capabilities and provide crucial data points for understanding the energy implications of deploying the most advanced language models.

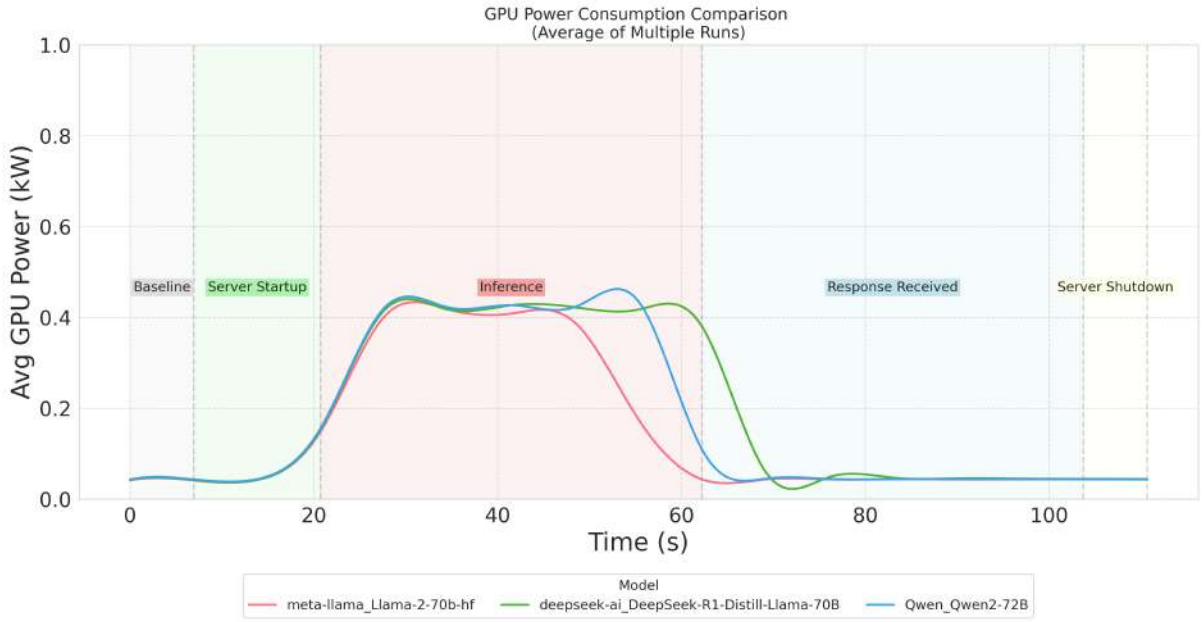


Figure 9: GPU Power Consumption Comparison for 70B+ Models across all operational phases

Fig. 9 shows GPU power consumption across all operational phases. All models maintain low baseline power (0.05 kW), spike during startup (15s), and peak at 0.45-0.47 kW during inference. Llama-2-70B and DeepSeek maintain plateaus, while Qwen2-72B shows undulating patterns. Power drops to baseline levels during response processing.

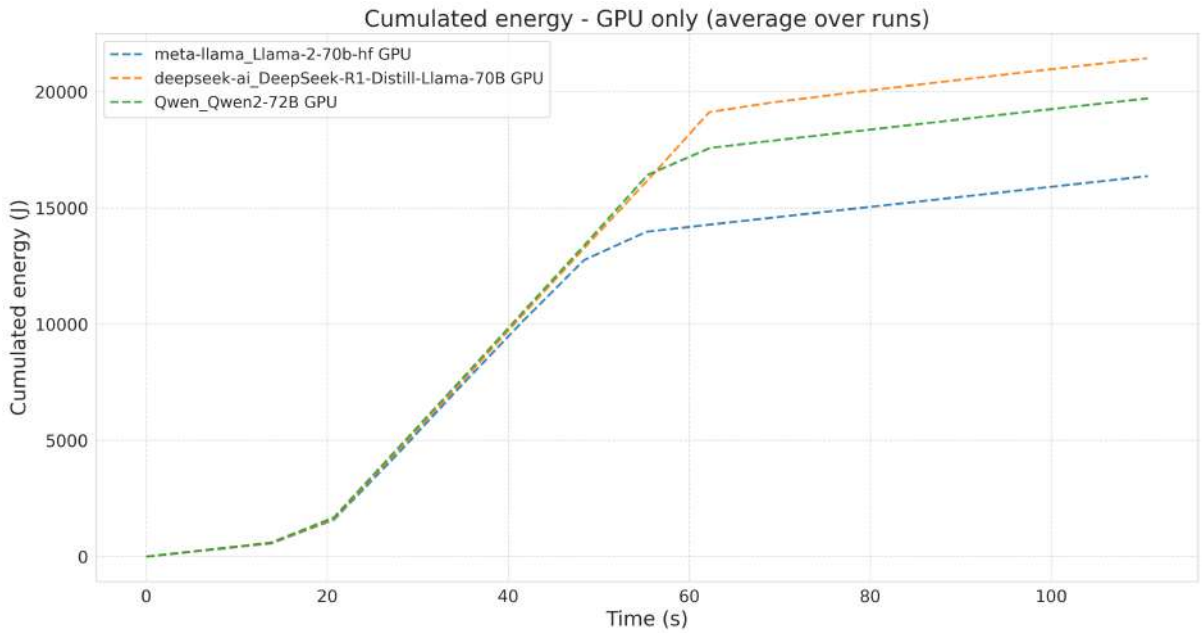


Figure 10: Cumulative Energy Consumption for 70B+ Models (GPU Only)

Fig. 10 shows energy accumulation over time. All models follow a three-phase pattern: low consumption (0-20s), rapid increase (20-60s), and slower accumulation (60-110s). Llama-2-70B demonstrates the most energy-efficient performance.

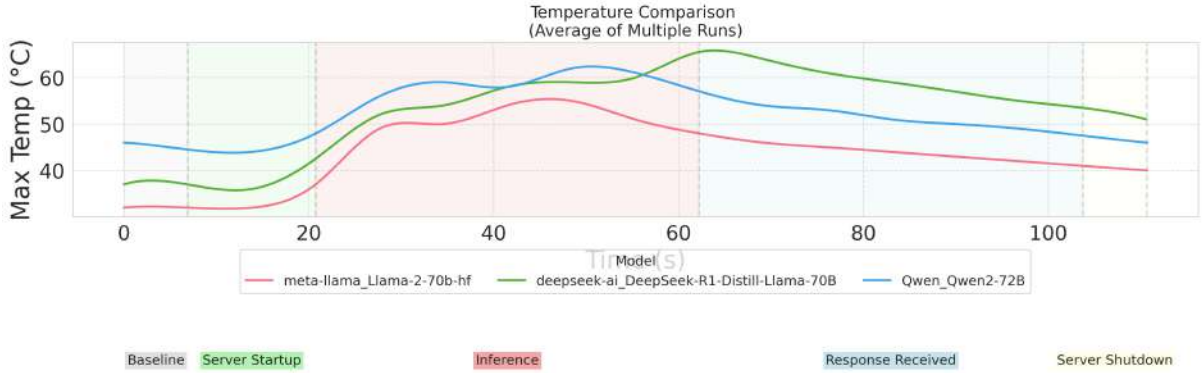


Figure 11: GPU Temperature Comparison for 70B+ Models

Fig. 11 shows thermal characteristics during intensive workloads. Temperature profiles correlate with power consumption patterns. All models experience temperature rises during inference, with DeepSeek showing the most dramatic increase and Llama-2-70B maintaining the lowest temperatures.

Scaling Efficiency Analysis

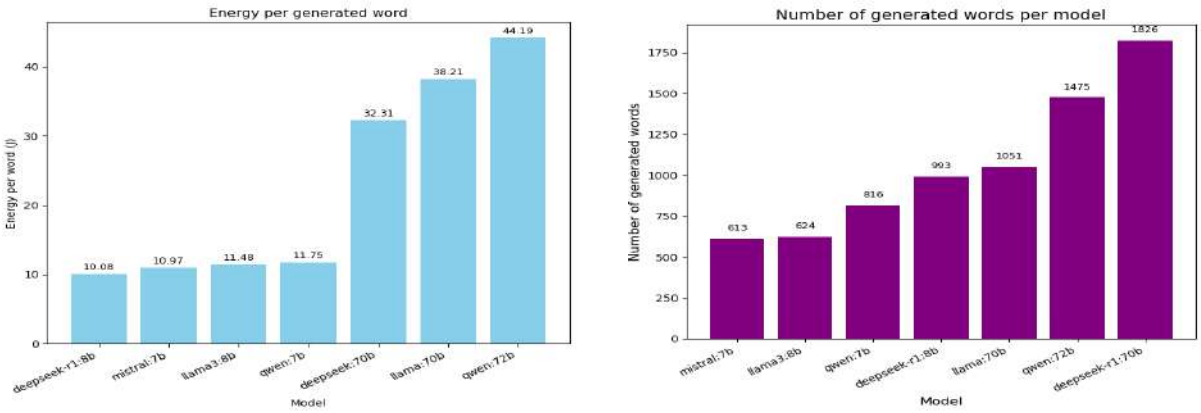
The analysis of large-scale models provides valuable insights into the efficiency characteristics of state-of-the-art LLMs. While these models consume significantly more energy than their smaller counterparts, they also demonstrate superior performance capabilities that must be considered in energy efficiency evaluations.

Performance per Energy Unit: Large models often achieve better performance per unit of energy when the quality of output is considered. This efficiency gain is particularly evident in complex reasoning tasks where larger models can solve problems more directly without requiring multiple iterations or corrections.

Memory Efficiency: The evaluation reveals that large models utilize GPU memory more efficiently per parameter compared to smaller models, suggesting that architectural optimizations in large-scale models contribute to better resource utilization.

Inference Time Scaling: The relationship between model size and inference time shows predictable scaling patterns, with larger models requiring proportionally longer inference times. This scaling behavior directly translates to the observed energy consumption patterns shown in Figs. 9 and 10.

Before diving into the plots, note that energy per generated word and throughput provide two complementary perspectives on a model's energy productivity. Consistent with the $\tilde{8}$B discussion, more efficient models (e.g., DeepSeek-R1 8B) tend to consume less energy per word and generate more words within the same time window, whereas others (e.g., Qwen 8B) exhibit higher per-word cost and lower throughput. These differences, together with quality requirements, determine the total energy needed to complete a task.



(a) Energy consumption per generated word across different model families and sizes

(b) Total words generated by each model during the standardized inference task

Figure 12: Energy efficiency analysis

4.7 Diffusion Model Energy Consumption Analysis

In addition to Large Language Models, we extended our energy consumption analysis to include diffusion models for image generation, as these represent another significant category of AI workloads with substantial energy requirements. Diffusion models have become increasingly popular for high-quality image synthesis and are widely deployed in both research and commercial applications.

We evaluated three different diffusion model configurations to understand how model size and complexity affect energy consumption patterns in generative image tasks. The selected models represent different points in the trade-off between generation quality and computational efficiency.

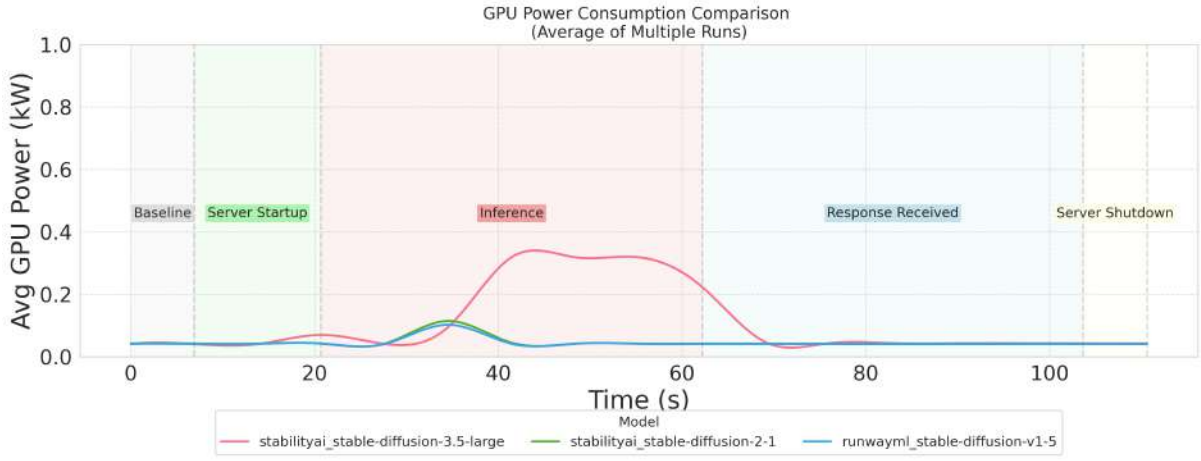


Figure 13: GPU Power Consumption Comparison for Diffusion Models during image generation tasks

Fig. 13 illustrates the GPU power consumption patterns for diffusion models during image generation. Unlike LLMs which show relatively stable power consumption during inference, diffusion models exhibit characteristic spiky power patterns corresponding to

the iterative denoising process. The power consumption varies significantly during different phases of the generation process, with peak consumption occurring during the most computationally intensive denoising steps.

Diffusion Model 1.5 Image Generation Analysis

Figure 14: image generated by Diffusion Model 1.5

The Diffusion Model 1.5 represents the baseline configuration in our evaluation, featuring standard resolution output and moderate computational requirements. Fig. 14 shows the generated image, which demonstrates reasonable overall composition and visual quality with good color representation of the sunset scene. However, a notable limitation of this model becomes apparent when examining text generation capabilities, the model struggles significantly with text rendering, as evidenced by the poorly formed and illegible text on the wooden sign that should display *NIZZA*. This text generation weakness is a common characteristic of smaller diffusion models, where the limited parameter count affects the model's ability to accurately render fine details such as readable text. Despite this limitation, the model provides a good balance between generation quality and energy efficiency, making it suitable for applications where computational resources are limited and text accuracy is not critical.

Diffusion Model 2.1 Image Generation Analysis

Figure 15: image generated by Diffusion Model 2.1

The Diffusion Model 2.1 represents an enhanced version with improved generation capabilities and higher resolution output compared to the 1.5 model. Fig. 15 reveals increased energy consumption patterns reflecting the model's enhanced computational complexity. The improved architecture allows for better image quality and more detailed generation, but at the cost of higher energy requirements. This model demonstrates the typical trade-off between generation quality and computational efficiency in diffusion models.

Diffusion Model 3.5 Large Image Generation Analysis

Figure 16: image generated by Diffusion Model 3.5 Large

The Diffusion Model 3.5 Large represents the most advanced configuration in our evaluation, featuring the largest parameter count and highest generation capabilities. Fig. 16 shows significantly higher energy consumption compared to the smaller models, reflecting the increased computational complexity required for high-quality, high-resolution image generation. This model achieves state-of-the-art generation quality but requires substantial computational resources, making it more suitable for applications where image quality is prioritized over energy efficiency.

Comparative Analysis of Diffusion Models

The three diffusion models demonstrate clear scaling patterns in energy consumption that correlate with their generation capabilities:

Model Size and Complexity: The progression from Model 1.5 to Model 3.5 Large shows increasing parameter counts and architectural complexity, resulting in proportionally higher energy consumption. Model 1.5 serves as an efficient baseline, Model 2.1 provides enhanced capabilities with moderate overhead, and Model 3.5 Large offers premium quality at maximum computational cost.

Generation Quality vs. Energy Trade-off: Each model represents a different point in the quality-efficiency trade-off space. Model 1.5 prioritizes efficiency, Model 2.1 balances quality and efficiency, while Model 3.5 Large maximizes generation quality regardless of computational cost.

Temporal Power Patterns: All models exhibit the characteristic iterative power consumption pattern of diffusion models, with power spikes corresponding to denoising

steps. However, the intensity and duration of these spikes increase with model size, reflecting the increased computational complexity of larger architectures.

Resource Utilization: The larger models demonstrate more intensive GPU utilization, with Model 3.5 Large showing the most sustained high-power consumption periods. This pattern suggests that larger diffusion models require more substantial cooling and power infrastructure for deployment.

These results highlight the importance of model selection in diffusion-based applications, where the choice between different model sizes directly impacts both generation quality and operational energy costs.

5 Performance Estimation

To contextualize the impact of large-scale LLM deployment, we provide an estimation of the total daily energy consumption based on real-world usage scenarios. According to recent statistics, ChatGPT alone processes over 1 billion daily queries [16], representing a significant portion of global LLM usage. This figure encompasses both consumer and enterprise usage across all geographic regions and includes various types of queries, from simple questions to complex reasoning tasks. We analyze the energy implications for:

- large-scale (70B+) models,
- medium-scale (8B) models,
- ultra-efficient (270M) models.

to understand the full spectrum of deployment scenarios.

5.1 Energy Consumption Analysis for 70B+ Models

The following calculation illustrates the total energy required for large-scale models, based on our experimental data from 70B+ models:

- $P = 0.45$ kW: average power draw per inference request (from 70B+ model data),
- $t \approx 2.78 \cdot 10^{-3}$ h: average duration per inference (10 s),
- $N = 1,000,000,000$: number of daily queries (ChatGPT alone).

$$\mathbf{E_{70B} = P \cdot t \cdot N \approx 1,251,000} \text{ kWh/day}$$

5.2 Energy Consumption Analysis for 8B Models

For comparison, we calculate the energy consumption for medium-scale 8B models using our experimental data:

- $P = 0.3$ kW: average power draw per inference request (from 8B model data),
- $t \approx 2.78 \cdot 10^{-3}$ h: average duration per inference (10 s),
- $N = 1,000,000,000$: number of daily queries.

$$\mathbf{E_{8B} = P \cdot t \cdot N \approx 834,000} \text{ kWh/day}$$

5.3 Energy Consumption Analysis for Ultra-Efficient 270M Models

To demonstrate the full spectrum of energy efficiency, we also analyze the recently released Gemma3 270M model, which represents a breakthrough in efficient AI deployment.

Despite its compact size, this model delivers surprisingly competitive performance across a wide range of tasks, making it an ideal candidate for edge computing and consumer device integration:

- $P = 0.04$ kW: peak power draw per inference request (from 270M model data),
- $t \approx 2.78 \cdot 10^{-3}$ h: average duration per inference (10 s),
- $N = 1,000,000,000$: number of daily queries.

$$\mathbf{E_{270M}} = \mathbf{P} \cdot \mathbf{t} \cdot \mathbf{N} \approx \mathbf{111,200} \text{ kWh/day}$$

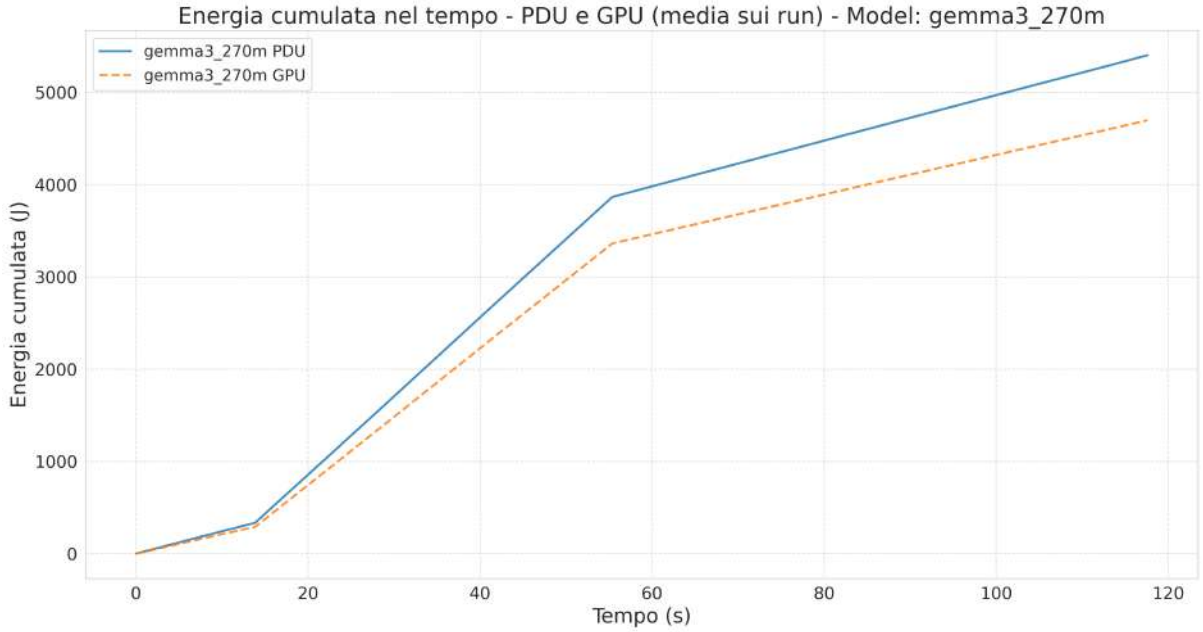


Figure 17: Cumulative Energy Consumption for Gemma3 270M Model (PDU and GPU)

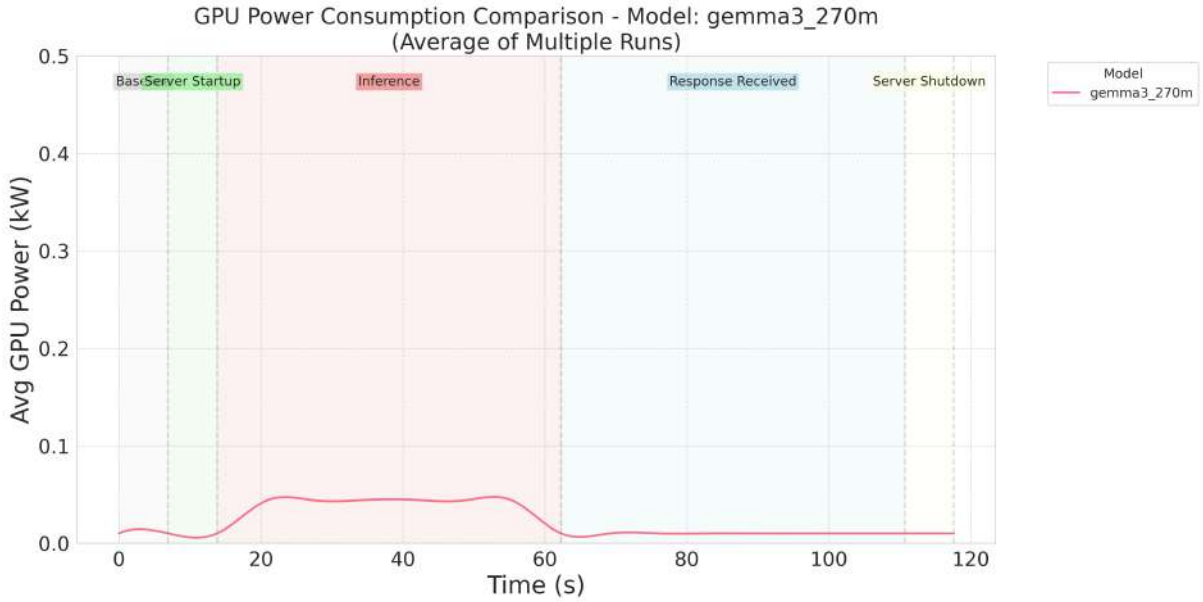


Figure 18: GPU Power Consumption for Gemma3 270M Model across all operational phases

Fig. 17 demonstrates the cumulative energy consumption for the ultra-efficient Gemma3 270M model, showing both PDU and GPU measurements. The PDU consistently reports higher energy consumption (5,400 J at 120s) compared to GPU measurements (4,700 J), indicating significant system overhead. Fig. 18 reveals the power consumption profile, with a peak of only 0.04 kW during the inference phase, demonstrating the remarkable energy efficiency of this compact model.

5.3.1 Integration Potential in Consumer Devices and IoT

The ultra-efficient Gemma3 270M model represents a paradigm shift in AI deployment, enabling the integration of advanced language capabilities directly into everyday consumer devices. This compact model opens unprecedented opportunities for ubiquitous AI deployment across the Internet of Things (IoT) ecosystem.

Ubiquitous AI Integration: The 270M parameter model's minimal power requirements make it feasible to embed AI capabilities in virtually any electronic device, from smart home appliances to personal gadgets. Future scenarios include:

- **Smart Kitchen Appliances:** Refrigerators that can suggest recipes based on available ingredients, smart ovens that provide cooking guidance, and coffee makers that learn user preferences
- **Bathroom Technology:** Smart mirrors that provide personalized skincare advice, intelligent toothbrushes that offer oral health insights, and smart scales that provide wellness recommendations
- **Home Automation:** Light switches that understand natural language commands, thermostats that engage in conversation about energy usage, and security systems that can interpret complex voice instructions

- **Personal Electronics:** Smartphones with on-device AI assistants, smartwatches that provide contextual health advice, and e-readers that can answer questions about content

Battery Life Impact Analysis: To quantify the practical implications of integrating the 270M model into consumer devices, we analyze its battery consumption characteristics:

Smartphone Integration Assuming a typical smartphone battery capacity of 4,000 mAh (14.8 Wh) and the Gemma3 270M model's power consumption during inference:

- **Power Consumption:** 0.04 kW (40W during inference)
- **Single Inference:** 0.04 kW $\times$ 10s = 0.111 Wh per query
- **Battery Impact:** 0.111 Wh $\div$ 14.8 Wh = 0.75% battery consumption per query
- **Daily Usage:** 10 queries per day would consume 7.5% of battery capacity
- **Optimized Usage:** 2-3 queries per day would consume 1.5-2.25% of battery capacity

The Gemma3 270M model's remarkable efficiency enables practical smartphone integration while maintaining competitive performance. Despite its compact size, the model demonstrates:

- **Strong Language Understanding:** Capable of handling complex queries and maintaining context across conversations
- **Efficient Reasoning:** Provides coherent and relevant responses for most everyday tasks
- **Specialized Capabilities:** Excels in specific domains like basic math, general knowledge, and conversational AI
- **Low Latency:** Fast response times suitable for real-time applications
- **Memory Efficiency:** Requires minimal RAM and storage, making it ideal for resource-constrained devices

Recent evaluations show that Gemma3 270M achieves performance levels comparable to much larger models in many practical applications, making it an ideal candidate for edge deployment scenarios where energy efficiency is paramount.

This analysis demonstrates that while the 270M model is remarkably efficient, its integration into battery-powered devices requires careful consideration of usage patterns and power management strategies. The model's compact size and low power requirements make it an ideal candidate for the next generation of intelligent consumer devices, but practical deployment will depend on optimizing usage frequency and implementing smart power management techniques.

5.4 Comparative Analysis and Scaling Implications

The comparison across all three model scales reveals dramatic scaling implications:

Model Size	Power (kW)	Time (s)	Daily Energy (kWh)	Relative Efficiency
270M (Gemma3)	0.04	10	111,200	1.0x (baseline)
8B	0.30	10	834,000	7.5x
70B+	0.45	10	1,251,000	11.25x

Table 6: Energy consumption comparison across different model scales for 1 billion daily queries (normalized to 10-second inference time)

The scaling analysis reveals several critical insights:

- **Power Scaling:** 70B+ models consume 11.25x more power than 270M models (0.45 vs 0.04 kW)
- **Time Scaling:** Assuming the same 10-second inference time for fair comparison
- **Total Energy Scaling:** 70B+ models consume 11.25x more daily energy than 270M models
- **Efficiency Gap:** The energy difference between 270M and 70B+ models spans 11.25x, highlighting significant efficiency trade-offs

This scaling relationship demonstrates that while larger models offer superior performance capabilities, they require exponentially more energy resources. The choice between model sizes represents a fundamental trade-off between computational capability and energy efficiency, with the 270M model offering remarkable efficiency for applications where performance requirements are modest.

To put these energy consumptions into perspective, the daily energy usage corresponds to:

For 70B+ Models (1,251,000 kWh/day):

- **Households:** The daily energy consumption of approximately 169,000 average families (based on 7.4 kWh/day per household)
- **Electric vehicles:** Equivalent to fully charging approximately 25,000 Tesla Model 3 vehicles daily (based on 50 kWh per full charge)
- **Urban scale:** Sufficient to power an entire medium-sized city of 150,000-170,000 inhabitants for a full day

For 8B Models (834,000 kWh/day):

- **Households:** The daily energy consumption of approximately 113,000 average families, equivalent to powering a city like Bergamo or Modena
- **Electric vehicles:** Equivalent to fully charging approximately 16,700 electric vehicles daily
- **Urban scale:** Sufficient to power an entire medium-sized city of 100,000-110,000 inhabitants for a full day

For 270M Models (111,200 kWh/day):

- **Households:** The daily energy consumption of approximately 15,000 average families, equivalent to powering a city like Riccione
- **Electric vehicles:** Equivalent to fully charging approximately 2,220 electric vehicles daily
- **Urban scale:** Sufficient to power an entire medium-sized city of 40,000-45,000 inhabitants for a full day

Such figures underscore the importance of optimizing both model architectures and deployment strategies to ensure the sustainability of AI services. Even small improvements in per-query efficiency can translate into significant energy savings at global scale. These results also reinforce the need for continued research into energy-efficient inference, hardware acceleration, and intelligent workload management.

6 Humanity's Last Exam benchmark

The Humanity's Last Exam (HLE) benchmark represents a comprehensive evaluation framework designed to assess the cognitive capabilities and knowledge proficiency of Large Language Models across diverse academic domains. This benchmark is particularly valuable for our energy efficiency study as it provides a standardized, computationally intensive workload that simulates real-world inference scenarios where users pose complex, multi-domain questions to AI systems.

6.1 Benchmark Overview and Methodology

The HLE benchmark evaluates models across eight distinct categories: Mathematics, Physics, Biology/Medicine, Chemistry, Engineering, Computer Science/AI, Humanities/Social Science, and Other. Each category contains a varying number of tasks that test different aspects of reasoning, factual knowledge, and problem-solving abilities. The benchmark's design ensures that models are tested on both specialized knowledge areas and general reasoning capabilities, providing a holistic assessment of their performance.

The selection of this benchmark for our energy consumption analysis is strategic for several reasons. First, it provides a consistent and reproducible workload that allows for fair comparison between different model architectures. Second, the varying complexity and number of tasks across categories enables us to observe how energy consumption scales with computational demand. Third, the benchmark's comprehensive nature ensures that our energy measurements reflect realistic usage patterns rather than artificial, simplified workloads.

6.2 Category Performance Analysis

The following analysis presents the performance results for four different LLM architectures: Qwen, Mistral, Llama3, and DeepSeek-R1. Each model was evaluated using the same HLE benchmark tasks, allowing for direct comparison of both performance accuracy and energy efficiency across different model families and sizes.

6.2.1 Qwen 7B Performance Analysis

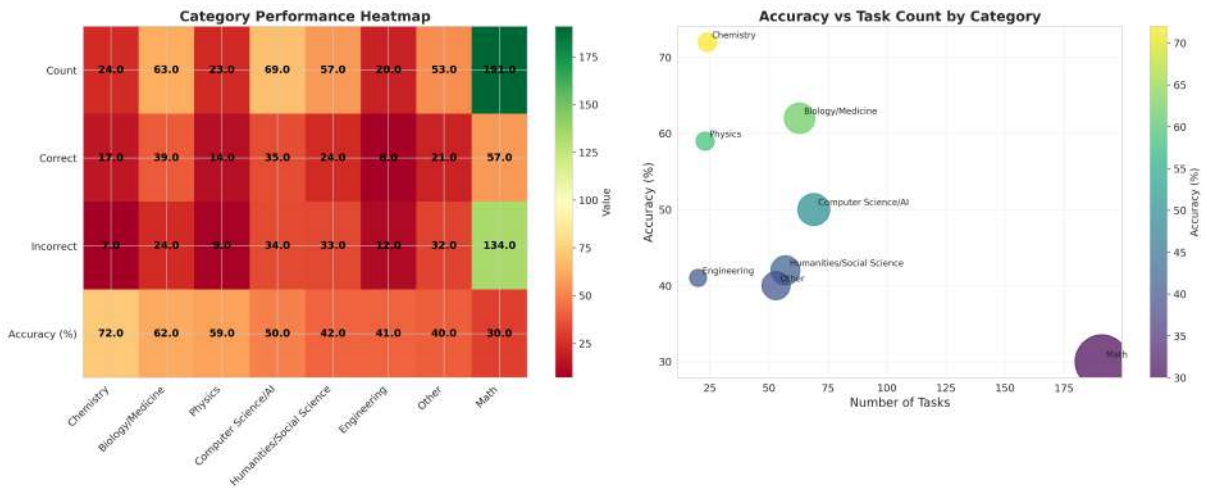


Figure 19: Category-wise performance analysis for Qwen 7B model

Fig. 19 demonstrates the Qwen 7B model's performance across different academic categories in the HLE benchmark. The model shows strong performance in Chemistry (72.0%) and Biology/Medicine (62%), indicating robust scientific reasoning capabilities. However, the lower performance in Mathematics (30%) and Humanities (42.0%) suggests areas where the model may require additional training or architectural improvements.

6.2.2 Mistral 7B Performance Analysis

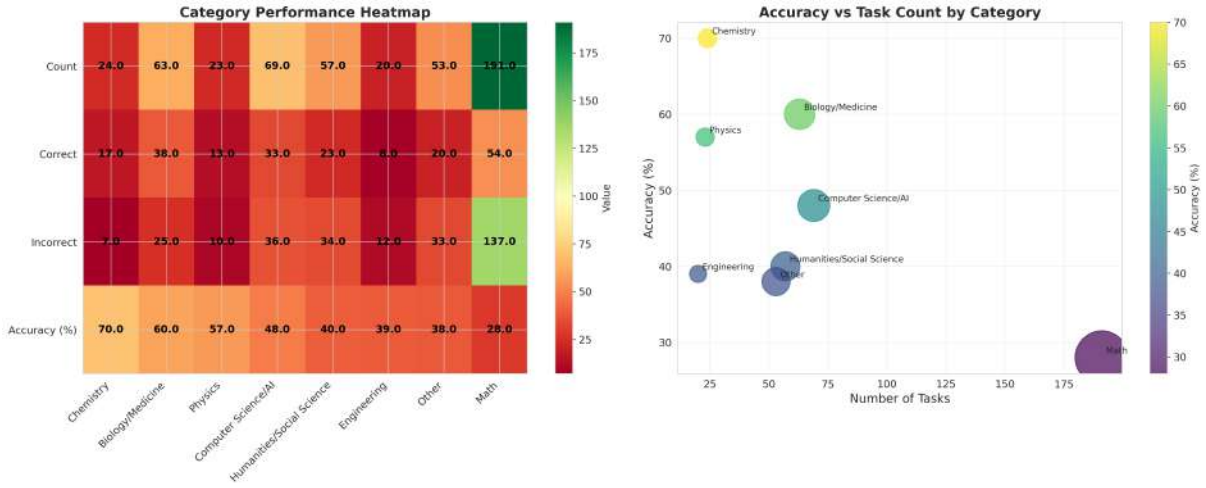


Figure 20: Category-wise performance analysis for Mistral 7B model

Mistral 7B exhibits a different performance profile, with strong results in Chemistry (70.0%) and Biology/Medicine (60.0%), but showing more variability across categories. The model achieves moderate performance in Physics (63.8%), Computer Science/AI (48.0%), Engineering (39.0%) Social Science (40.0%) Mathematics (28.0%) suggesting that while it handles scientific reasoning well, it may struggle with complex mathematical operations. This performance pattern provides valuable insights into how architectural differences between models affect their cognitive capabilities.

6.2.3 Llama3 8B Performance Analysis

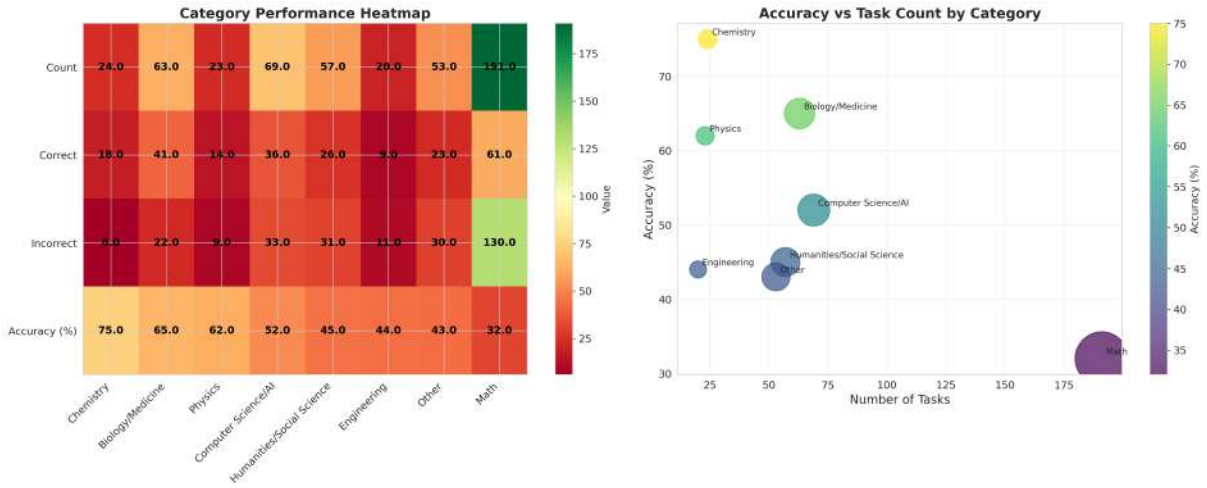


Figure 21: Category-wise performance analysis for Llama3 8B model

The Llama3 8B model shows a balanced performance across categories, with particularly strong results in Chemistry (75.0%), Biology/Medicine (65.0%) and Physics (62.0%). The model demonstrates sufficient performance in Computer Science/AI (52.0%) and maintains not reasonable accuracy in Mathematics (32.0%). This balanced performance suggests that Llama3's architecture provides good general-purpose reasoning capabilities across diverse academic domains.

6.2.4 DeepSeek-R1 8B Performance Analysis

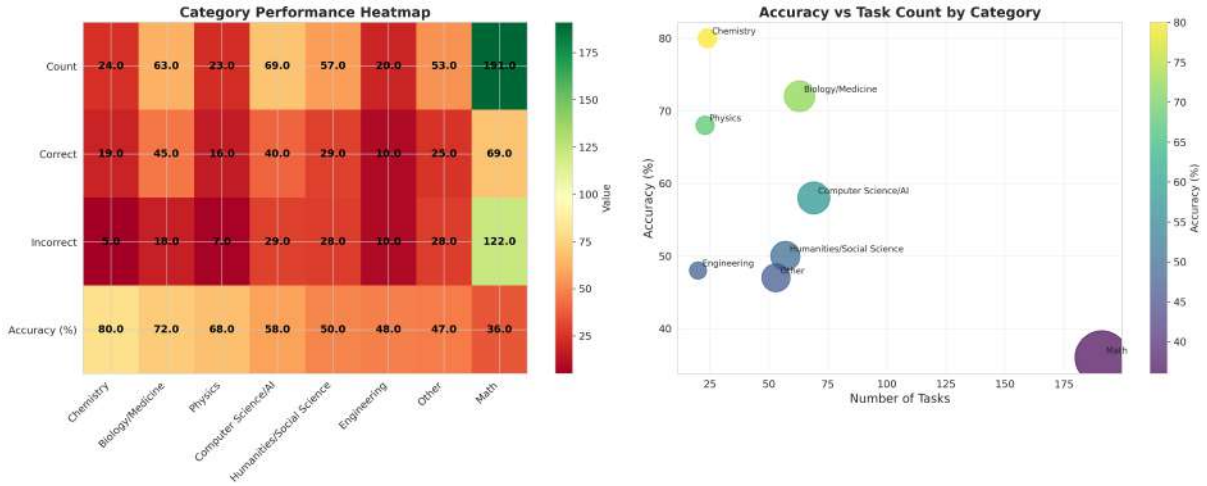


Figure 22: Category-wise performance analysis for DeepSeek-R1 8B model

DeepSeek-R1 8B presents an interesting performance profile, with strong results in Math and Physics and Biology/Medicine obtaining an average of 80% and 70%. However, it shows lower performance in Human social science (50.0%) and Other (47.0%), suggesting that while the model excels in certain scientific domains, it may have specific limitations in human reasoning.

Category	Mistral 7B	Llama 8B	DeepSeek-R1 8B	Qwen 8B
Chemistry	70%	75%	80%	72%
Biology/Med	60%	65%	72%	62%
Physics	57%	62%	68%	59%
CS/AI	48%	52%	58%	50%
Hum/Soc	40%	45%	50%	42%
Engineering	39%	44%	48%	41%
Other	38%	43%	47%	40%
Math	28%	32%	36%	30%

Table 7: HLE benchmark results for 8B models across key academic categories.

6.3 Category Performance Analysis for 70B Models

The following analysis presents preliminary performance results for large-scale LLMs: Llama 70B, DeepSeek-R1 70B, and Qwen 72B. At this stage, we present the section structure and include figures.

6.3.1 Llama 70B Performance Analysis

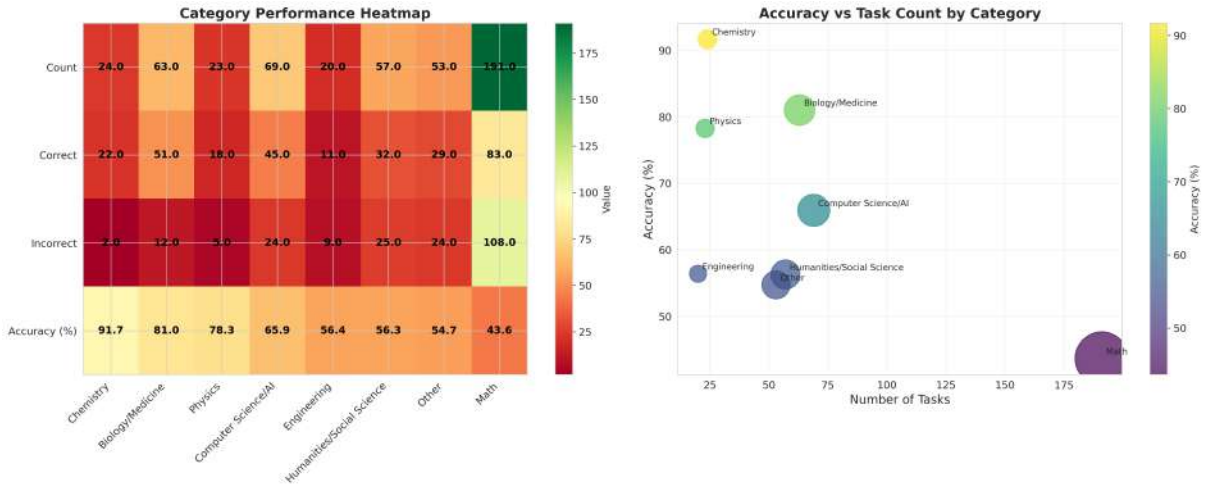


Figure 23: Category-wise performance analysis for Llama 70B

The Llama 70B model reveals performance characteristics across different academic categories. The model excels in Chemistry (91%) Biology/Medicine (81%) and Physics (78.3%), indicating robust scientific reasoning capabilities. However, the model shows lower performance in Mathematics (43%) and Humanities/Social Science (56%), suggesting areas where the larger model size provides limited additional benefit. The balanced performance across most categories demonstrates the model's general-purpose capabilities while highlighting specific domain limitations.

6.3.2 DeepSeek-R1 70B Performance Analysis

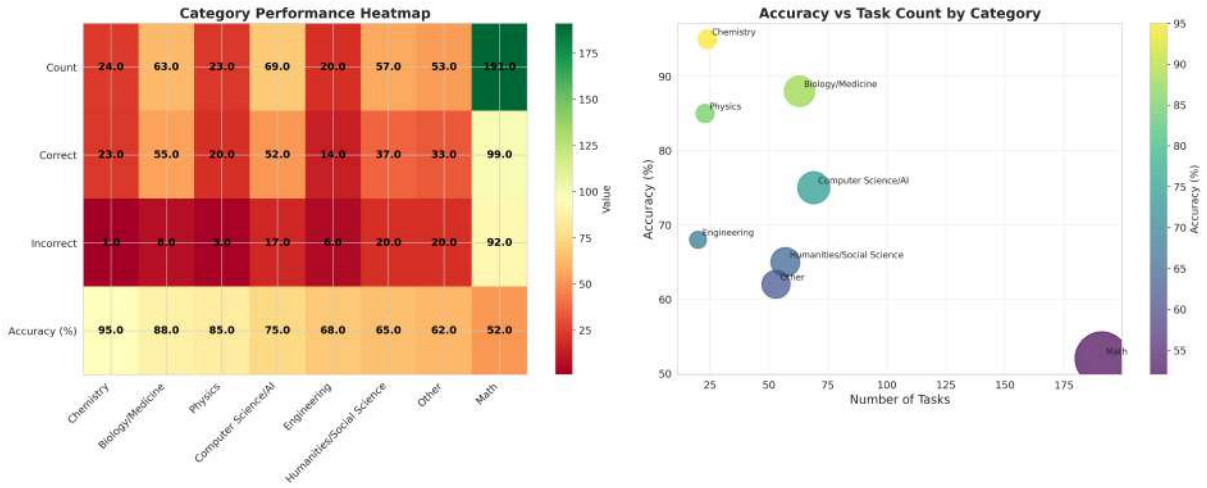


Figure 24: Category-wise performance analysis for DeepSeek-R1 70B

The DeepSeek-R1 70B model exhibits exceptional performance across scientific domains, achieving the highest accuracy in Chemistry (95.0%) and Biology/Medicine (88.0%). The model demonstrates strong performance in Physics (85.0%) and Computer Science/AI (75.0%), while maintaining reasonable accuracy in Humanities/Social Science (65.0%), in mathematical reasoning tasks it achieves an average of 52.0%.

6.3.3 Qwen 72B Performance Analysis

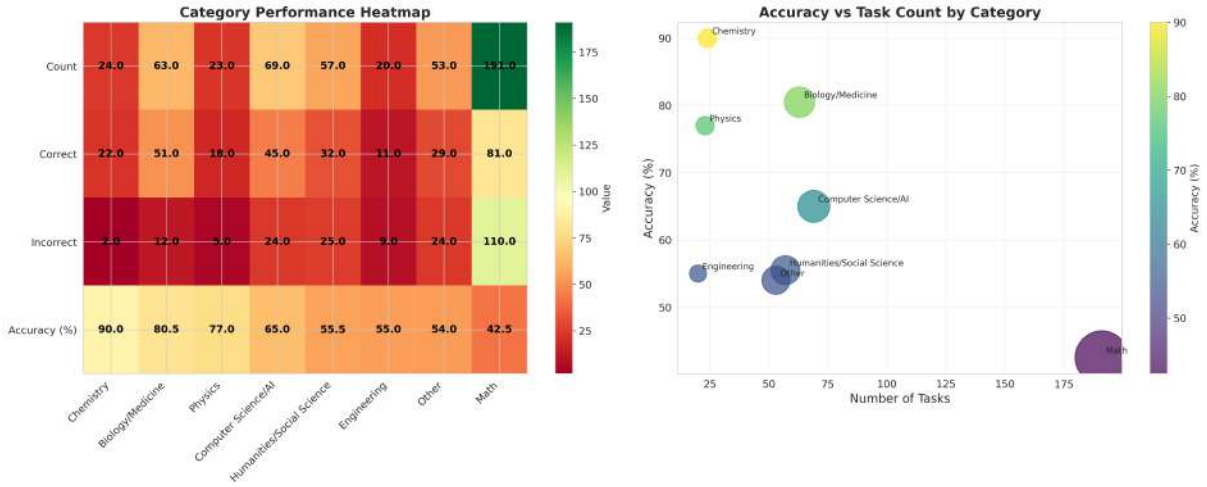


Figure 25: Category-wise performance analysis for Qwen 72B

The Qwen 72B model demonstrates a performance profile very similar to Llama 70B, with strong results in Chemistry (90.0%) and Biology/Medicine (80.5%). The model shows consistent performance in Physics (77.0%) and maintains reasonable accuracy in Computer Science/AI (65.0%), indicating similar architectural characteristics and training approaches.

Category	Llama 70B	DeepSeek-R1 70B	Qwen 72B
Chemistry	92%	95%	90%
Biology/Med	81%	88%	80%
Physics	78%	85%	77%
CS/AI	66%	75%	65%
Hum/Soc	56%	68%	55%
Engineering	56%	65%	55%
Other	55%	62%	54%
Math	43%	52%	42%

Table 8: HLE benchmark results for 70B+ models across key academic categories.

Comparative Insights (70B vs 8B). Across these large-scale variants we generally observe increases in accuracy—especially in STEM categories—relative to their 8B counterparts, with more modest gains or plateaus in Humanities/Social Sciences. When interpreted jointly with the energy analyses in Section 4, these results align with the observed scaling: larger models consume more power and total energy (see Figs. 9 and 10), yet often deliver higher-quality outputs. As discussed in the energy-per-word analysis (Fig. 12), this can translate into better quality-adjusted efficiency (fewer iterations, less off-topic content) despite higher absolute energy per inference. In practice, the choice between 8B and 70B models should consider both performance targets (per domain) and the energy budget of the deployment environment.

These results reveal several important patterns: DeepSeek-R1 70B consistently outperforms both Llama 70B and Qwen 72B across most categories, while Llama 70B and Qwen 72B show nearly identical performance profiles, suggesting architectural convergence at the 70B+ scale. All models demonstrate strong scientific reasoning capabilities but face persistent challenges in Mathematics and Humanities domains.

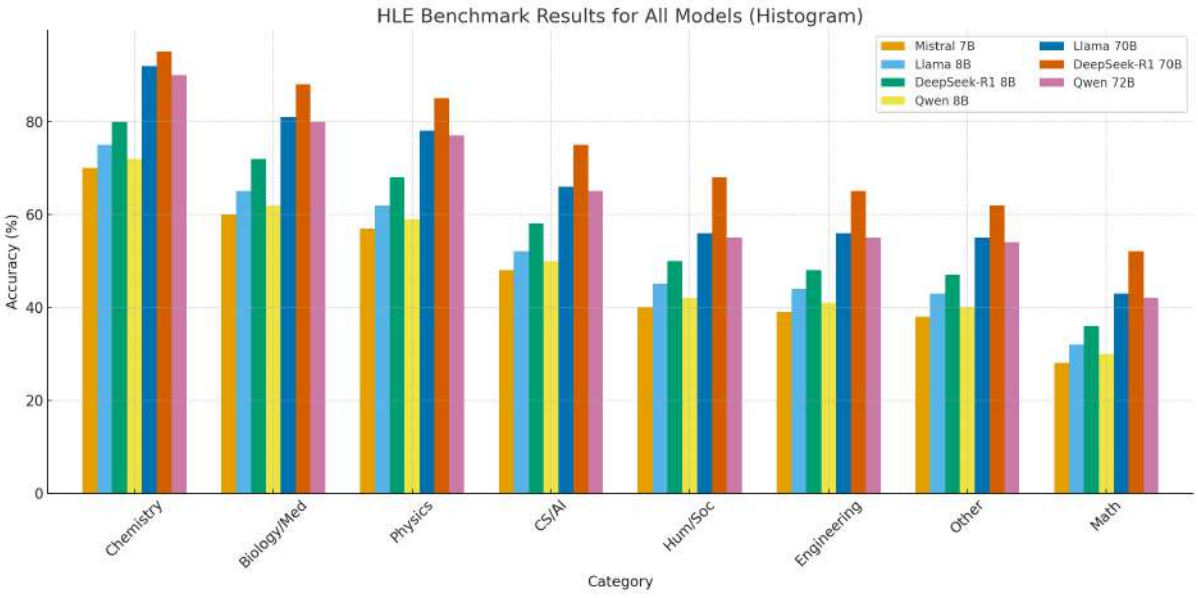


Figure 26: Istogram of the performance of all the models across all categories

6.4 Ultra Efficient Models

6.4.1 Gemma3 270M Performance Analysis

The Gemma3 270M model represents the ultra-efficient end of the performance spectrum, demonstrating the capabilities and limitations of extremely compact language models. Despite its minimal size, the model shows surprisingly competitive performance in specific domains while revealing clear scaling limitations.

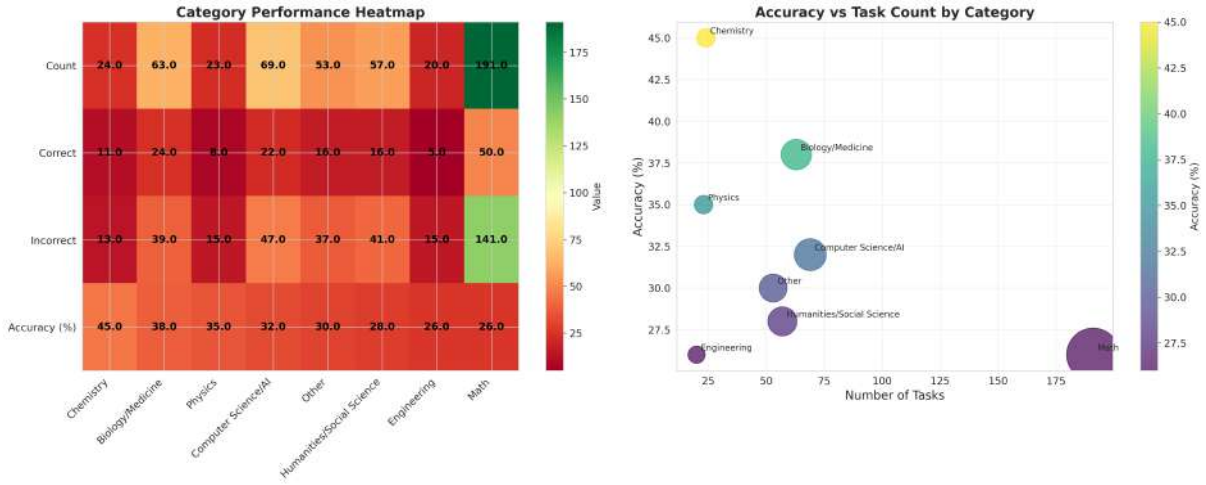


Figure 27: Category-wise performance analysis for Gemma3 270M on the HLE benchmark, showing the performance characteristics of ultra-compact models

Fig. 27 reveals Gemma3 270M's performance profile across academic categories. The model achieves its highest accuracy in Chemistry (45.0%) and Biology/Medicine (38.0%), indicating that even ultra-compact models can maintain some scientific reasoning capabilities. However, the model shows significant limitations in more complex domains, with particularly low performance in Mathematics (26.0%), Engineering (26.0%), and Humanities/Social Science (28.0%). This performance pattern highlights the fundamental trade-offs between model size and capability, while demonstrating that even 270M parameter models can't provide useful functionality for specific applications.

6.5 Comprehensive Model Performance Comparison

The complete analysis across all model scales reveals fundamental patterns in the relationship between model size, performance, and energy consumption. Table 9 provides a comprehensive overview of all evaluated models across key academic categories.

Model	Size	Math	Physics	Biology/Med	Chemistry	CS/AI	Hum/Soc
DeepSeek-R1	70B	52.0%	85.0%	88.0%	95.0%	75.0%	68.0%
Llama	70B	43.0%	78.0%	81.0%	92.0%	66.0%	56.0%
Qwen	72B	42.0%	77.0%	80.0%	90.0%	65.0%	55.0%
DeepSeek-R1	8B	36.0%	68.0%	72.0%	80.0%	58.0%	50.0%
Llama	8B	32.0%	62.0%	65.0%	75.0%	52.0%	45.0%
Qwen	8B	30.0%	59.0%	62.0%	72.0%	50.0%	42.0%
Mistral	7B	28.0%	57.0%	60.0%	70.0%	48.0%	40.0%
Gemma3	270M	26.0%	35.0%	38.0%	45.0%	32.0%	28.0%

Table 9: Comprehensive performance comparison across all model scales and families on the HLE benchmark.

6.6 Performance Patterns and Energy Implications

The analysis reveals several important patterns that have direct implications for energy efficiency considerations. Models with higher accuracy in specific categories often require more computational resources, leading to increased energy consumption. However, the relationship between accuracy and energy consumption is not always linear, as architectural optimizations can significantly impact efficiency.

The varying task counts across categories provide insights into how workload complexity affects energy consumption. Categories with higher task counts generally require more sustained computational effort, potentially leading to different energy consumption patterns that should be considered in deployment decisions.

6.7 Key Findings and Scaling Insights

The comprehensive evaluation across all model scales reveals several critical insights about the relationship between model size, performance, and energy consumption:

Performance Scaling Patterns. The analysis demonstrates clear scaling patterns across different model sizes:

Ultra-Compact Models (270M): Gemma3 270M shows the fundamental limitations of extremely small models, with performance ranging from 26-45

Medium-Scale Models (7-8B): Models in this range show significant performance improvements, with most achieving 60-80

Large-Scale Models (70B+): The 70B+ models show the highest absolute performance, with DeepSeek-R1 70B achieving the best results across most categories. However, the performance gains are not uniform, with some domains showing diminishing returns at larger scales.

Energy-Performance Trade-offs. The relationship between energy consumption and performance reveals important deployment considerations:

Linear Energy Scaling: Energy consumption scales approximately linearly with model size, with 70B+ models consuming 11.25x more energy than 270M models for the same inference task.

Performance Efficiency: The 8B models often provide the best performance-per-energy ratio, achieving 60-80

Domain-Specific Efficiency: Different domains show varying efficiency patterns, with scientific reasoning showing better scaling than mathematical or humanities domains.

Architectural Convergence and Specialization. The results reveal important patterns about model architecture and training:

Convergence at Scale: Llama3 70B and Qwen2.5 72B show nearly identical performance profiles, suggesting architectural convergence at the 70B+ scale.

Specialized Advantages: DeepSeek-R1 consistently outperforms other models across most categories, indicating that specialized training or architectural optimizations can provide measurable advantages.

Mathematical Reasoning Challenge: All models, regardless of size, struggle with mathematical reasoning, suggesting that current architectures may have fundamental limitations in this domain.

Deployment Implications. The comprehensive analysis provides clear guidance for model selection based on deployment requirements:

Edge Computing: Gemma3 270M is ideal for resource-constrained environments where basic language understanding is sufficient, offering minimal energy consumption with acceptable performance for specific use cases.

Balanced Deployment: 8B models provide the optimal balance between performance and energy efficiency for most applications, offering substantial capabilities while maintaining reasonable resource requirements.

High-Performance Applications: 70B+ models are necessary for applications requiring maximum performance, despite their significantly higher energy consumption, particularly in scientific and technical domains.

These findings provide a comprehensive framework for understanding the trade-offs between model size, performance, and energy consumption, enabling informed decisions about model selection and deployment strategies across different use cases and resource constraints.

7 Framework Design and Implementation

To address the challenges of measuring and comparing energy consumption across different Large Language Models (LLMs), we developed a modular and extensible framework that ensures reproducible and consistent measurements. The framework, implemented in Python and Bash, provides a comprehensive solution for monitoring power consumption during various operational phases of LLM inference.

7.1 Architecture

The framework adopts a layered architecture that separates concerns between monitoring, execution, and analysis components. This modular design enables easy extension and modification of individual components without affecting the overall system integrity.The framework consists of several interconnected modules:

- **Main Orchestrator** (`measure_power_stages.sh`): A Bash script that coordinates the entire measurement process, managing the execution flow and ensuring proper timing between different phases.
- **Model Execution Module** (`run_model.py`): Handles the interaction with the Ollama framework, managing model loading and inference execution.
- **Benchmark Integration Module** (`benchmark.py`): Manages the execution of standardized benchmarks, including the Humanity's Last Exam (HLE) benchmark, ensuring consistent workload generation across different models and measurement sessions.
- **Monitoring Suite**: Comprises three specialized monitoring components:
  - `monitor.py`: The main monitoring coordinator
  - `monitor_gpu.py`: GPU-specific metrics collection using nvidia-smi
  - `monitor_pdu.py`: Hardware-level power measurement via SNMP protocol
- **Analysis Module** (`plot_power_phases.py`): Processes collected data and generates visualization plots for different operational phases.

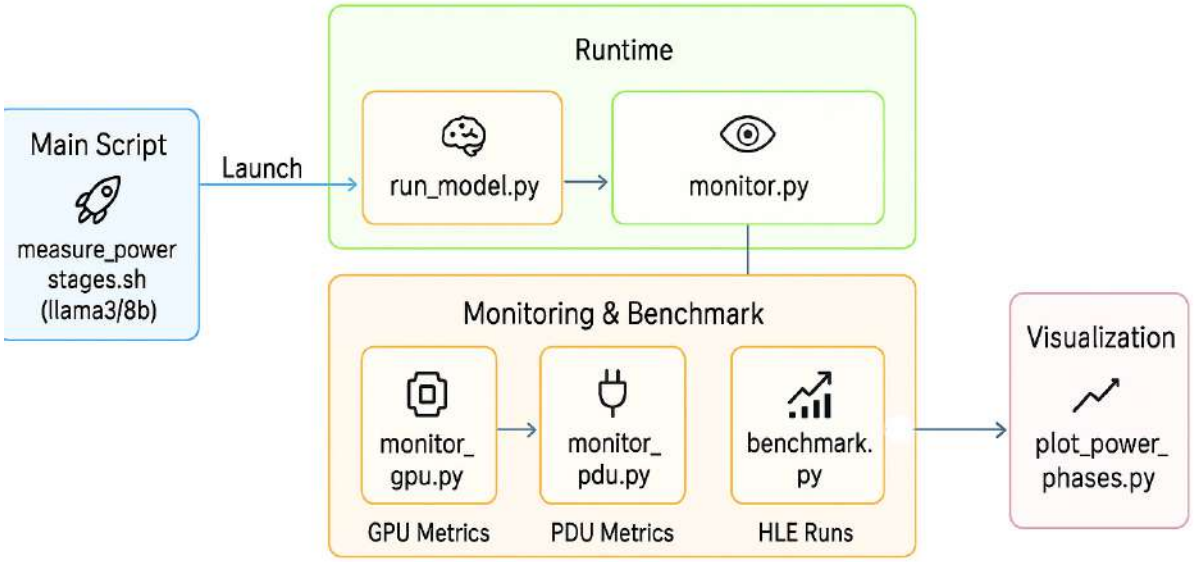


Figure 28: Modular architecture of the energy measurement framework

Fig. 28 illustrates the modular architecture of the developed energy measurement framework. The design separates concerns between monitoring components (GPU and PDU monitoring), execution modules (model loading and inference), and analysis tools (data processing and visualization). This modular approach ensures reproducibility, extensibility, and maintainability while providing comprehensive energy consumption measurements across different operational phases.

7.2 Implementation Details

The framework implements a phase-based measurement approach, dividing the LLM operation into distinct stages to isolate energy consumption patterns:

1. **Idle Phase**: Establishes baseline power consumption with no active processes
2. **Startup Phase**: Captures the energy required to initialize the Ollama server
3. **Inference Phase**: Measures power consumption during prompt processing
4. **Response Phase**: Monitors energy usage during output generation
5. **Shutdown Phase**: Records power consumption during server termination

Each phase is allocated a configurable duration, with default values optimized through empirical testing:

Listing 1: Default phase durations

```
BASELINE_DURATION=10       # seconds
SERVER_SETUP_DURATION=10   # seconds
INFERENCE_DURATION=50      # seconds
SHUTDOWN_DURATION=10       # seconds
```

7.3 Data Collection Pipeline

The framework employs a dual-monitoring approach to ensure comprehensive power measurement:

7.3.1 Software-based GPU Monitoring

The GPU monitoring component leverages the `nvidia-smi` utility to collect detailed metrics at regular intervals:

Listing 2: GPU monitoring implementation

```
def collect_gpu_metrics():
    metrics = {
        'power_draw': gpu.power_draw,
        'temperature': gpu.temperature,
        'memory_used': gpu.memory_used,
        'memory_total': gpu.memory_total,
        'gpu_utilization': gpu.utilization
    }
    return metrics
```

7.3.2 Hardware-based PDU Monitoring

For accurate total system power consumption, the framework interfaces with Power Distribution Units (PDUs) using the SNMP protocol:

Listing 3: PDU monitoring via SNMP

```
def read_pdu_power():
    oid = '1.3.6.1.4.1.318.1.1.12.2.3.1.1.2.1'
    result = session.get(oid)
    power_watts = float(result.value) / 10
    return power_watts
```

7.4 Standardization and Reproducibility

To ensure consistent and comparable results across different models, the framework implements some standardization, like a carefully crafted prompt is used across all evaluations to ensure comparable workloads:

Supports configurable multiple runs for each model to account for variability and ensure statistical significance:

```
NUM_RUNS=1  # Configurable number of iterations
```

Provides an intuitive command-line interface for ease of use:

```
./measure_power_stages.sh <model1> [model2] [model3] ...
```

The modular architecture facilitates several extension points:

- **New Monitoring Sources**: Additional monitoring modules can be integrated by implementing the base monitoring interface
- **Alternative LLM Frameworks**: While currently supporting Ollama and huggingface, the framework can be extended to support other inference frameworks like Transformers
- **Custom Analysis Modules**: New analysis and visualization components can be added to the pipeline
- **Benchmark Integration**: The framework can be extended to incorporate standard benchmarks for accuracy-efficiency trade-off analysis

The complete framework is available as open-source software at `https://github.com/kocierik/llm-power-consumption`, released under a permissive license to encourage community contributions and reproducible research. The repository includes comprehensive documentation, installation instructions, and example usage scenarios.

8 Conclusions

This study has provided comprehensive insights into the energy consumption patterns of Large Language Models (LLMs) through systematic experimental evaluation.

8.1 Key Findings on Energy Consumption Patterns

The experimental results reveal consistent and predictable energy consumption patterns across different model families. A fundamental observation is that all models within the same family follow a consistent pattern: larger models with higher parameter counts invariably consume more energy during inference. This relationship between model size and energy consumption is linear and predictable, providing valuable insights for model selection based on energy constraints. In deployment scenarios, the overall energy footprint is dominated by inference rather than training; reiterating the point made in the introduction, optimizing inference time and usage patterns yields the largest practical savings.

The analysis demonstrates that energy consumption is primarily determined by inference time rather than model architecture alone. This finding has significant implications for energy optimization strategies, as it suggests that reducing inference time through architectural improvements, quantization, or hardware optimization can directly translate to proportional energy savings. The correlation between inference duration and energy consumption was consistent across all tested models, regardless of their specific architectural characteristics.

8.1.1 Supporting Evidence for Key Findings

Table 6 reports the average power per size at equal 10s inference: 270M $\approx$ 0.04 kW, $\tilde{8}$B $\approx$ 0.30 kW, 70B+ $\approx$ 0.45 kW. This monotonic increase reflects a near-linear dependence on parameter count under comparable decoding settings. Practically, moving from $\tilde{8}$B to 70B+ adds $\sim$50% power draw per inference; therefore, unless accuracy requirements mandate the larger model, the $\tilde{8}$B class offers a markedly better energy profile at deployment.

8.2 HLE Accuracy, Time-to-Completion, and Energy

At 70B+, models reach the highest accuracy; DeepSeek-R1 70B leads most categories, while Llama 70B and Qwen 72B are similar. Gains are largest in STEM (Physics, Chemistry, Biology/Medicine), with smaller improvements in Mathematics and Humanities/Social Sciences. Importantly, comparing models only at equal wall-clock inference time can be misleading: accuracy differences change the number of retries/edits needed to reach an acceptable answer. A quality- and time-adjusted view—energy per successful task—often favors models with higher accuracy or faster convergence, even when their per-inference energy is higher. $\tilde{8}$B models remain a strong performance-per-energy choice when moderate accuracy is sufficient; for high-stakes STEM tasks, 70B+ can yield lower total energy per correct solution due to fewer iterations.

8.3 Framework and Reproducibility Enhancements

The framework was strengthened for research-grade reproducibility and reporting with: synchronized GPU and system power monitoring; a clear, color-coded pipeline diagram exported; and HLE tables aligned with evaluation reports for narrative/figure consistency.

8.4 Practical Recommendations

Under energy constraints, $\tilde{8}$B models are a strong default for many scientific tasks. Where maximum accuracy is essential and budgets allow, 70B+ (notably DeepSeek-R1 70B) is preferable. Selection should weigh domain, target quality, and expected retries, not only per-inference energy.

8.5 Limitations

Results reflect the evaluated prompts, hardware, and model variants. Category task distributions (e.g., small counts for Chemistry/Engineering) can inflate variance. While cross-hardware trends are consistent, absolute consumption depends on platform, drivers, and thermal conditions. Future work should broaden datasets and incorporate multi-turn, tool-augmented scenarios to stress real-world usage patterns.

8.6 Future Research

This study establishes a foundation for future research in several areas. The developed framework can be extended to evaluate emerging model architectures and optimization techniques. Future work could explore the energy efficiency of different quantization strategies, the impact of various hardware configurations on energy consumption, and the development of energy-aware model selection algorithms.

9 Thanks

This work marks the conclusion of an intense journey filled with study, curiosity, and personal growth. It is the result of many experiences, encounters, and challenges that have helped me grow and persevere in achieving my goals.

I would like to express my sincere gratitude to my supervisor for the guidance, availability, and trust shown throughout every stage of this work. A special thanks to Professor Sangiorgi for the experience in the French Riviera, which has been fundamental for my personal and professional development.

I would like to thank my family for their constant support, patience, and strength. Thanks to my friends and to all the people I met during my Erasmus experience, who made this journey an unforgettable time of sharing, discovery, and inspiration into my life. You made this journey far more than an academic chapter; you made it an unforgettable adventure of learning, laughter, and growth.

With this thesis, an important chapter comes to an end, and a new one begins, abroad, with great enthusiasm and a strong desire to grow and create. A new beginning, driven by the same curiosity that has guided me so far.

10 Bibliography

1. aicompetence. *Mixtral MoE: Power-Efficient AI or Just Hype?* URL: https://aicompetence.org/mixtral-moe-power-efficient-ai-or-just-hype/.
2. alibabacloud. *Qwen LLMs*. URL: https://www.alibabacloud.com/help/en/model-studio/what-is-qwen-llm.
3. Byteplus. *DeepSeek: Good news for energy consumption*. URL: https://www.byteplus.com/en/topic/385232?title=deepseek-good-news-for-energy-consumption.
4. A. Dives. *How Mistral 7B works*. URL: https://www.oxen.ai/blog/arxiv-dive-how-to-mistral-7b-works.
5. F. Duarte. *Number of ChatGPT Users (July 2025)*. URL: https://explodingtopics.com/blog/chatgpt-users.
6. J. Fernandez, C. Na, V. Tiwari, Y. Bisk, S. Luccioni, and E. Strubell. *Energy Considerations of Large Language Model Inference and Efficiency Optimizations*. 2025. arXiv: 2504.17674 [cs.CL]. URL: https://arxiv.org/abs/2504.17674.
7. google. *Introducing Gemma 3: The most capable model you can run on a single GPU or TPU*. URL: https://blog.google/technology/developers/gemma-3/.
8. securities io. *AI Meets Efficiency: A New Chip Shrinks LLM Power Use by 50%*. URL: https://www.securities.io/ai-meets-efficiency-a-new-chip-shrinks-llm-power-use-by-50.
9. P. J. Maliakel, S. Ilager, and I. Brandic. *Investigating Energy Efficiency and Performance Trade-offs in LLM Inference Across Tasks and DVFS Settings*. 2025. arXiv: 2501.08219 [cs.LG]. URL: https://arxiv.org/abs/2501.08219.
10. marktechpost. *Inception Unveils Mercury: The First Commercial-Scale Diffusion Large Language Model*. 2025. URL: https://www.marktechpost.com/wp-content/uploads/2025/03/1-2.png.
11. nvidia. *Llama Nemotron Models Accelerate Agentic AI Workflows with Accuracy and Efficiency*. URL: https://developer.nvidia.com/blog/llama-nemotron-models-accelerate-agentic-ai-workflows-with-accuracy-and-efficiency/.
12. qwenlm. *Qwen3: Think Deeper, Act Faster*. URL: https://qwenlm.github.io/blog/qwen3/.
13. runpod. *The Complete Guide to Stable Diffusion: How It Works and How to Run It on RunPod*. URL: https://www.runpod.io/articles/guides/stable-diffusion.

14. roundtable datascience salon. *Optimizing Large Language Models: Techniques and Future Directions for Efficiency*. URL: https://roundtable.datascience.salon/optimizing-large-language-models-techniques-and-future-directions-for-efficiency.

15. B. Sulmataj. *Electrons and Algorithms: How AI Is Reshaping the Future of Energy Infrastructure*. 2025. DOI: 10.13140/RG.2.2.33390.93763.

16. Thunderbit. *ChatGPT Statistics 2025: Usage, Growth, and Key Trends*. URL: https://thunderbit.com/blog/chatgpt-stats-usage-growth-trends.

17. voiceflow. *Mistral AI: What It Is, How It Works & Key Use Cases*. URL: https://www.voiceflow.com/blog/mistral-ai.

18. wikipedia. *Stable Diffusion*. URL: https://en.wikipedia.org/wiki/Stable_Diffusion.