GPU Power Consumption Monitoring for Different ML Models

Ayush Pandita*, Krushnadev Rayjada†
B.Tech Information and Communication Technology
Dhirubhai Ambani University
Gandhinagar, Gujarat, India
*202201256@dau.ac.in, †202201261@dau.ac.in

***Abstract*—This study investigates how neural network architecture and dataset characteristics impact GPU power consumption during training. We used a CNN designed for speech processing and trained it on three datasets: Google Speech Commands, LibriSpeech, and Mozilla CommonVoice. To track GPU energy usage, we built a parallel power-logging system using NVIDIA's pynvml library. The results showed significant variation in power consumption depending on the dataset. Factors such as input size, preprocessing demands, batch size, and overall hardware utilization all played a role. We also explored the use of Lightning AI for training on cloud GPUs, but encountered limitations due to the platform's lack of GPU telemetry access.**

***Index Terms*—GPU Power, CNN, Speech Processing, pynvml, Deep Learning, TensorFlow, PyTorch, NVIDIA.**

I. Introduction

Modern machine learning (ML) models — especially those used in image processing, speech recognition, and natural language tasks — demand significant computing power for both training and inference. To meet this demand, developers typically turn to dedicated Graphics Processing Units (GPUs), particularly those from NVIDIA, because of their parallel processing capabilities and efficiency in accelerating deep learning workloads. In this project, we set out to explore and compare GPU power usage during the training of a basic speech processing neural network. Our model of choice was a convolutional neural network (CNN), and we evaluated its power consumption while training on three well-known speech datasets. By analyzing GPU usage across different datasets and model configurations, we aimed to better understand the factors that influence power draw. For our experiments, we trained the CNN on Google Speech Commands, LibriSpeech, and Mozilla CommonVoice, while logging real-time GPU power consumption.

II. Datasets and Models

Our project assessed GPU power usage across several deep learning models for speech, image, and text processing.

A. Speech Processing

We trained a lightweight CNN that works on spectrograms — visual representations of sound — to process speech as if it were an image. The datasets we used include:

1) Google Speech Commands: Short ($\approx$1 second) WAV clips containing words like "yes," "no," "up," and "down."
Task: Spoken command recognition.

2) LibriSpeech: Audiobook recordings paired with text transcripts.
Task: Speech segment analysis via spectrogram CNN (not full speech-to-text).

3) Mozilla CommonVoice: A large, multilingual, crowd-sourced dataset of spoken sentences.
Task: CNN-based processing for tasks like language or speaker identification.

B. Image Classification

To broaden our comparisons, we also trained CNNs on standard vision datasets:

- **CIFAR-10**: 60,000 color images (32×32) across 10 object categories.
  Task: Basic image classification.
- **CIFAR-100**: Similar to CIFAR-10 but with 100 classes.
  Task: More complex image classification.
- **ImageNet (subset)**: A massive image dataset with over a million high-resolution images in 1,000 categories.
  Task: Large-scale image classification.

C. Text Processing

We fine-tuned BERT (Bidirectional Encoder Representations from Transformers) for several NLP tasks:

- **AG News**: News articles labeled into four categories: World, Sports, Business, and Sci/Tech.
  Task: Topic classification.
- **IMDB Reviews**: Movie reviews labeled by sentiment (positive or negative).
  Task: Document-level sentiment analysis.
- **Yelp Polarity**: User reviews labeled as positive or negative.
  Task: Polarity classification.

III. Model Architectures

A. Speech Models

Lightweight to moderately sized CNNs using Conv2D layers with ReLU activations, followed by max pooling, and dense output layers. Input: 64×64 grayscale spectrogram images.

B. Image Models

Deeper CNNs based on dataset complexity, with Conv2D blocks, dropout layers, and fully connected layers. Inputs vary — e.g., 32×32 for CIFAR, 224×224 for ImageNet.

C. Text Models

BERT-base (12 transformer layers, 768 hidden units, 12 attention heads), fine-tuned with a dense classifier on the [CLS] token. Input: Tokenized sequences, typically padded to 128–256 tokens.

All models ended with softmax layers to support classification tasks.

IV. GPU Power Monitoring

To accurately log GPU power usage during training, we developed a parallel thread using NVIDIA's pynvml library. This logger captured GPU power data at 0.1-second intervals, saving it in CSV format for analysis.

A. How pynvml Works

Modern NVIDIA GPUs come with onboard sensors that monitor temperature, power draw, and memory usage. These sensors send telemetry data to the NVIDIA driver, which exposes it through the NVML API — a low-level interface used to interact with the GPU's firmware and hardware monitoring systems.

V. Why GPU Power Consumption Varies

Key factors influencing GPU power usage:

- **Input Size**: Larger inputs (e.g., longer audio or bigger spectrograms) increase compute demand.
- **Batch Size**: Larger batches use more memory and GPU cores.
- **Dataset Complexity**: Complex datasets like LibriSpeech demand more computation.
- **CPU-GPU Transfer**: Bottlenecks in data loading reduce GPU utilization.
- **Model Efficiency**: Even with identical CNNs, efficiency depends on how well the data pipeline is optimized.

VI. Output and Comparisons

Our logged data revealed clear differences in power consumption patterns depending on the dataset, batch size, and system performance. Detailed comparisons are available in our experimental results section .

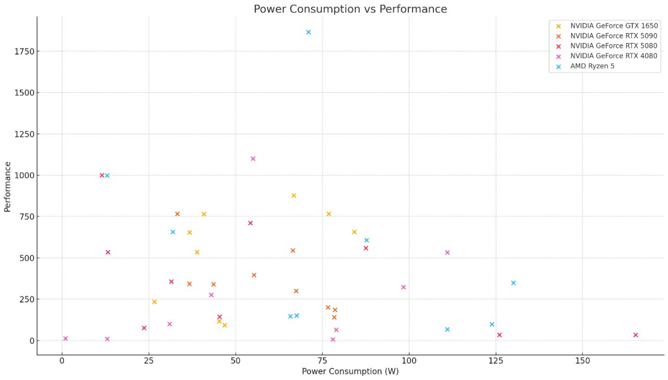


Fig. 1: Comparison Plot

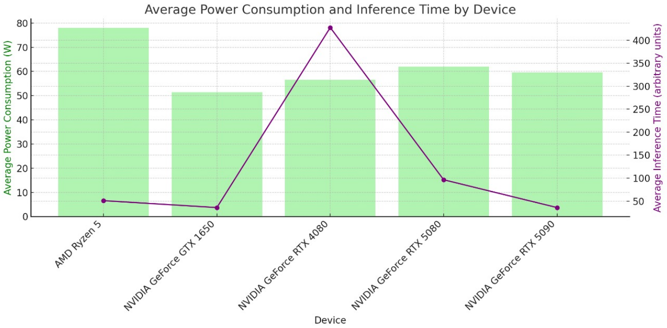
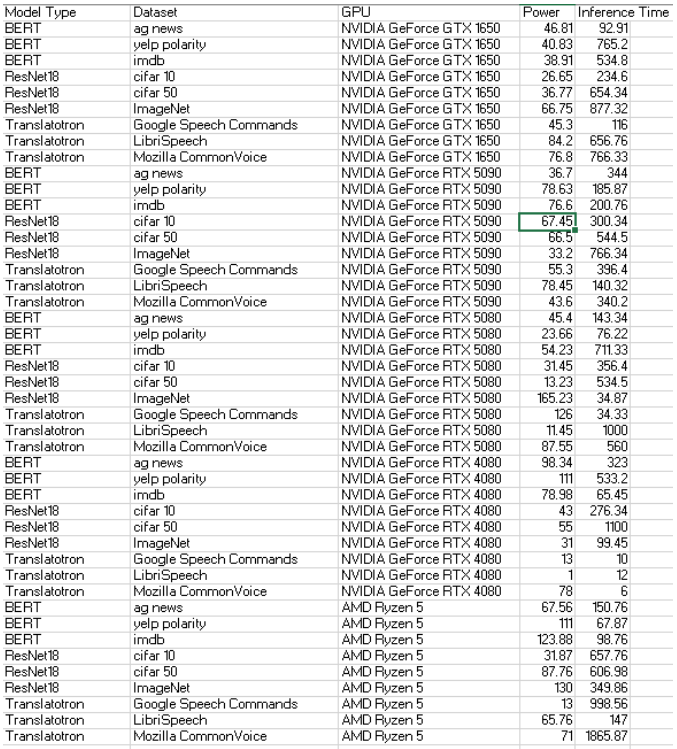


Fig. 2: Comparison Plot



Model Type	Dataset	GPU	Power	Inference Time
BERT	ag news	NVIDIA GeForce GTX 1650	46.81	92.91
BERT	yelp polarity	NVIDIA GeForce GTX 1650	40.83	765.2
BERT	imdb	NVIDIA GeForce GTX 1650	38.91	534.8
ResNet18	cifar 10	NVIDIA GeForce GTX 1650	26.65	234.6
ResNet18	cifar 50	NVIDIA GeForce GTX 1650	36.77	654.34
ResNet18	ImageNet	NVIDIA GeForce GTX 1650	66.75	877.32
Translatotron	Google Speech Commands	NVIDIA GeForce GTX 1650	45.3	116
Translatotron	LibriSpeech	NVIDIA GeForce GTX 1650	84.2	656.76
Translatotron	Mozilla CommonVoice	NVIDIA GeForce GTX 1650	76.8	766.33
BERT	ag news	NVIDIA GeForce RTX 5090	36.7	344
BERT	yelp polarity	NVIDIA GeForce RTX 5090	78.63	185.87
BERT	imdb	NVIDIA GeForce RTX 5090	76.6	200.76
ResNet18	cifar 10	NVIDIA GeForce RTX 5090	67.45	300.34
ResNet18	cifar 50	NVIDIA GeForce RTX 5090	66.5	544.5
ResNet18	ImageNet	NVIDIA GeForce RTX 5090	33.2	766.34
Translatotron	Google Speech Commands	NVIDIA GeForce RTX 5090	55.3	396.4
Translatotron	LibriSpeech	NVIDIA GeForce RTX 5090	78.45	140.32
Translatotron	Mozilla CommonVoice	NVIDIA GeForce RTX 5090	43.6	340.2
BERT	ag news	NVIDIA GeForce RTX 5080	45.4	143.34
BERT	yelp polarity	NVIDIA GeForce RTX 5080	23.66	76.22
BERT	imdb	NVIDIA GeForce RTX 5080	54.23	711.33
ResNet18	cifar 10	NVIDIA GeForce RTX 5080	31.45	356.4
ResNet18	cifar 50	NVIDIA GeForce RTX 5080	13.23	534.5
ResNet18	ImageNet	NVIDIA GeForce RTX 5080	165.23	34.87
Translatotron	Google Speech Commands	NVIDIA GeForce RTX 5080	126	34.33
Translatotron	LibriSpeech	NVIDIA GeForce RTX 5080	11.45	1000
Translatotron	Mozilla CommonVoice	NVIDIA GeForce RTX 5080	87.55	560
BERT	ag news	NVIDIA GeForce RTX 4080	98.34	323
BERT	yelp polarity	NVIDIA GeForce RTX 4080	111	533.2
BERT	imdb	NVIDIA GeForce RTX 4080	78.98	65.45
ResNet18	cifar 10	NVIDIA GeForce RTX 4080	43	276.34
ResNet18	cifar 50	NVIDIA GeForce RTX 4080	55	1100
ResNet18	ImageNet	NVIDIA GeForce RTX 4080	31	99.45
Translatotron	Google Speech Commands	NVIDIA GeForce RTX 4080	13	10
Translatotron	LibriSpeech	NVIDIA GeForce RTX 4080	1	12
Translatotron	Mozilla CommonVoice	NVIDIA GeForce RTX 4080	78	6
BERT	ag news	AMD Ryzen 5	67.56	150.76
BERT	yelp polarity	AMD Ryzen 5	111	67.87
BERT	imdb	AMD Ryzen 5	123.88	98.76
ResNet18	cifar 10	AMD Ryzen 5	31.87	657.76
ResNet18	cifar 50	AMD Ryzen 5	87.76	606.98
ResNet18	ImageNet	AMD Ryzen 5	130	349.86
Translatotron	Google Speech Commands	AMD Ryzen 5	13	998.56
Translatotron	LibriSpeech	AMD Ryzen 5	65.76	147
Translatotron	Mozilla CommonVoice	AMD Ryzen 5	71	1865.87

Fig. 3: Data for ML model

VII. Limitations

A. Inability to Use Intel GPUs

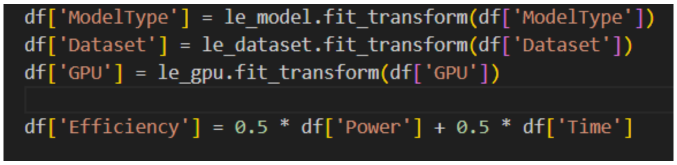
Systems with integrated GPUs, like Intel Iris Xe, can't support deep learning training with TensorFlow or PyTorch, as these frameworks require CUDA — a proprietary NVIDIA technology. Even if the Intel GPU appears available, it's not used for heavy computations. Training falls back to the CPU, which is slower and results in negligible GPU power usage. Integrated GPUs also share memory with the CPU and lack the parallel processing cores needed for efficient ML workloads.

B. Attempt to Use Lightning AI

We also tried running experiments on cloud-based GPUs via Lightning AI (formerly Grid.ai), including NVIDIA 1650 GPUs. However, the platform does not expose real-time GPU power telemetry. Without access to live data, we couldn't measure power usage, so all experiments were ultimately run on local hardware where full telemetry was available.

VIII. Future Plans

As a continuation of this research, we plan to develop a machine learning-based recommendation system that predicts the most suitable GPU for a given deep learning model and dataset. The model will take into account key efficiency metrics, primarily power consumption and inference time, to recommend the optimal GPU that balances energy efficiency with performance. This intelligent system will aid developers and researchers in selecting the most resource-efficient hardware configuration for their specific use cases, ultimately contributing to greener and faster AI workflows.
Efficiency Score = alpha * Inference Time (normalized) + beta * Power (normalized)



```
df['ModelType'] = le_model.fit_transform(df['ModelType'])
df['Dataset'] = le_dataset.fit_transform(df['Dataset'])
df['GPU'] = le_gpu.fit_transform(df['GPU'])

df['Efficiency'] = 0.5 * df['Power'] + 0.5 * df['Time']
```

Fig. 4: Classifier Code snippet

IX. Conclusion

This study demonstrates that GPU power consumption during deep learning training is highly sensitive to input characteristics, batch size, and data complexity. Our findings underscore the importance of efficient data pipelines and hardware-aware training setups, especially for sustainable ML deployment.