

---

# AI Work Quantization Model: Closed-System AI Computational Effort Metric

Aasish Kumar Sharma<sup>1</sup>, Michael Bidollahkhani<sup>2,\*</sup>, Julian Martin Kunkel<sup>1,2</sup>

<sup>1</sup>Gesellschaft Fr Wissenschaftliche Datenverarbeitung mbH Goettingen (GWDG), Goettingen, Germany

<sup>2</sup>Faculty of Mathematics and Computer Science, Georg-August-University of Goettingen, Goettingen, Germany

## Email address:

aasish-kumar.sharma@gwdg.de (Aasish Kumar Sharma), michael.bkhani@uni-goettingen.de (Michael Bidollahkhani),

julian.kunkel@gwdg.de (Julian Martin Kunkel)

\*Corresponding author

## To cite this article:

Aasish Kumar Sharma, Michael Bidollahkhani, Julian Martin Kunkel. (2025). AI Work Quantization Model: Closed-System AI Computational Effort Metric. *American Journal of Artificial Intelligence*, 9(1), 55-67. <https://doi.org/10.11648/j.ajai.20250901.16>

**Received:** 12 March 2025; **Accepted:** 1 April 2025; **Published:** 21 June 2025

---

**Abstract:** The rapid adoption of AI-driven automation in IoT environments, particularly in smart cities and industrial systems, necessitates a standardized approach to quantify AI's computational workload. Existing methodologies lack a consistent framework for measuring AI computational effort across diverse architectures, posing challenges in fair taxation models and energy-aware workload assessments. This study introduces the Closed-System AI Computational Effort Metric, a theoretical framework that quantifies real-time computational effort by incorporating input/output complexity, execution dynamics, and hardware-specific performance factors. The model ensures comparability between AI workloads across traditional CPUs and modern GPU/TPU accelerators, facilitating standardized performance evaluations. Additionally, we propose an energy-aware extension to assess AI's environmental impact, enabling sustainability-focused AI optimizations and equitable taxation models. Our findings establish a direct correlation between AI workload and human productivity, where 5 AI Workload Units equate to approximately 60-72 hours of human labor-exceeding a full-time workweek. By systematically linking AI computational effort to human labor, this framework enhances the understanding of AI's role in workforce automation, industrial efficiency, and sustainable computing. Future work will focus on refining the model through dynamic workload adaptation, complexity normalization, and energy-aware AI cost estimation, further broadening its applicability in diverse AI-driven ecosystems.

**Keywords:** AI Work Quantization, Computational Effort, Smart Cities, AI Taxation, AI Sustainability, IoT, Cloud AI

---

## 1. Introduction

The rapid expansion of AI across smart cities, industrial automation, and IoT ecosystems brings forth significant challenges in accurately measuring the computational effort of AI systems. Unlike human labor, which is measured in straightforward economic terms like wages and hours, the computational intensity, energy consumption, and operational footprint of AI processes lack a standardized measurement framework. According to [1], "An 'AI system' means a machine based system that is designed to operate with varying levels of autonomy and that may exhibit adaptiveness after deployment, and that, for explicit or implicit objectives, infers, from the input it receives, how to generate outputs such as

predictions, content, recommendations, or decisions that can influence physical or virtual environments;" this means, these machines can influence not only to humans but also to their environment and ultimately to the future of the living beings. That is why, it is necessary to quantize these systems and their impact to the nature.

### 1.1. Problem Statement

Current methods for quantifying AI workloads often depend on hardware-specific benchmarks such as FLOPs (Floating Point Operations Per Second) and power usage. These benchmarks, however, have several shortcomings:

1. They are not universally applicable in normal use cases

across different AI architectures with heterogeneous resources, like CPUs, GPUs, TPUs, and edge devices.

2. They fail to link the computational effort to real-time monitoring of AI workloads in environments where systems are enclosed and isolated.
3. They overlook sustainability metrics, which are crucial for formulating energy-efficient AI taxation models.

## 1.2. Objective

This paper introduces the Closed-System AI Computational Effort Metric (CE), a structured, scalable, and interpretable framework designed to measure AI computational efforts effectively. This framework aims to:

1. Establish a mathematically sound metric for AI computational effort that adapts to the varied nature of AI technologies.
2. Support real-time resource profiling for both cloud-based AI and IoT workloads, essential for accurate measurement.
3. Facilitate energy-aware AI taxation and workload optimization to promote sustainability in AI operations.

Creating this metric is essential to bridge the gap in how AI workload is measured, ensuring a fair and accurate assessment of AI computational efforts across various platforms and applications.

## 2. Background

### 2.1. Traditional Metrics vs. Quantized Work

Traditional measures of AI computational cost include FLOPs, energy consumption, and execution time. Although effective as broad indicators, these metrics do not capture the concept of computation as a sequence of discrete operations or “quanta.” Analogous to the quantized nature of energy in physical systems, the AI Work Quantization Model conceptualizes each basic computational operation as a discrete unit of work.

### 2.2. Thermodynamic and Information-Theoretic Roots

The notion of quantized work in computation draws heavily on principles from thermodynamics and information theory. For example, [2] demonstrated that erasing one bit of information incurs a minimum energy cost of  $kT \ln 2$ , where  $k$  is the Boltzmann’s constant (approx.  $1.38 \times 10^{-23}$  joules per kelvin), which relates temperature to energy at the particle level, and  $T$  is the absolute temperature (in kelvin) of the system.  $\ln 2$  is the natural logarithm of 2 (approximately 0.693), which comes from the binary nature of information (i.e. one bit can be in one of two states). Further exploration by [3] elucidated the thermodynamic implications of computational processes. In the quantum domain, studies such as [4] and [5] examine energy transitions and fluctuation theorems, which resonate with the idea of discrete computational work units.

The rapid expansion of artificial intelligence, particularly deep learning, has raised concerns regarding its environmental and societal impacts. Traditional carbon emission models for AI rely on Life Cycle Assessments (LCAs) that capture:

1. *Embodied Emissions*: Emissions originating from the manufacturing, transportation, and disposal of hardware.
2. *Operational Emissions*: Emissions due to energy consumption during training and inference.

For instance, Rahman et al. [13] propose a cradle-to-grave analysis and introduce the *Compute Carbon Intensity (CCI)* metric, which quantifies emissions in grams of CO<sub>2</sub>-equivalent per ExaFLOP (i.e., per  $10^{18}$  FLOPs). Such metrics provide a unified basis for comparing the environmental impact of different AI models.

Additional studies have shown that operational emissions often dominate the overall energy usage of AI systems (typically 70-90% of the total footprint) [6], while embodied emissions, though generally lower (often under 25%), are essential for comprehensive carbon accounting [7]. Improvements in data center efficiency—measured by metrics such as Power Usage Effectiveness (PUE) [14, 15]—further reduce net CO<sub>2</sub> emissions.

In these established models, total energy consumption, denoted  $C_{AI}$ , is converted into CO<sub>2</sub> emission estimates using an emissions conversion factor  $\kappa$ :

$$\text{CO}_2^{\text{total}} = \kappa \cdot C_{AI}. \quad (1)$$

This formulation directly links micro-level energy costs to macro-level CO<sub>2</sub> emission estimates, as exemplified by the CCI metric [13].

## 3. Related Work

A substantial body of research has focused on the energy footprint and carbon emissions of AI systems. Key contributions include:

### 3.1. Life Cycle Assessments and Compute Carbon Intensity

Recent LCAs, such as those by Rahman et al. [13], comprehensively evaluate the cradle-to-grave environmental impact of AI hardware. The CCI metric, which expresses emissions in gCO<sub>2</sub>-e per ExaFLOP, provides a basis for comparing diverse AI models.

### 3.2. Operational vs. Embodied Emissions

Studies report that operational emissions (energy used during training/inference) dominate AI’s total energy consumption [6], whereas embodied emissions from hardware production and disposal, though smaller, are critical for full carbon accounting [7]. Research on data center efficiency further refines these estimates [14, 15].

### 3.3. Hardware and Compute Trends

Work by Kaplan et al. [9] reveals dramatic increases in compute demands over recent years. Additional studies on compute trends in AI [16, 22] and benchmark analyses [23] emphasize the growing role of hardware efficiency and quantization techniques.

### 3.4. Quantization and Adaptive Computation

Recent advancements in neural network quantization [17, 18] and neuro-symbolic methods [20] have demonstrated that reducing numerical precision can greatly lower energy consumption while preserving accuracy. These dynamic computation techniques align with the conceptual framework of work quantization, where computation is broken into discrete, quantifiable units.

### 3.5. Theoretical Foundations: Thermodynamics of Computation

Our framework is grounded in fundamental thermodynamic principles. Landauer's principle asserts that the minimal energy required to erase one bit is

$$E_{\min} = kT \ln 2, \quad (2)$$

with  $k$  as Boltzmann's constant and  $T$  the absolute temperature [2, 3]. Recent studies on algorithmic progress [11, 24] further provide a basis for modeling AI computations at a fundamental level.

### 3.6. HPC and System-Level Modeling

High Performance Computing (HPC) research has established models for system-level resource consumption [12]. Furthermore, workload classification studies [19] highlight the unique characteristics of AI computation compared to traditional PC workloads.

### 3.7. Additional Perspectives on Compute and Carbon Footprints

Other influential works include Amodei and Hernandez's discussion of the exponential increase in compute demands [25] and recent studies on the carbon footprint of training deep learning models, such as CarbonTracker [26].

### 3.8. Energy Efficiency and Green AI

Recent research emphasizes the environmental impact of AI, underscoring the need for more nuanced efficiency metrics. Works such as [6] and [7] illustrate that energy demands in deep learning can be significant. These studies advocate for energy-based assessments of AI systems, which serve as a foundation for developing quantized measures of computational work.

### 3.9. Dynamic and Adaptive Computation

Advancements in adaptive neural architectures suggest that computation can be dynamically allocated depending on input complexity. Models that implement adaptive computation time implicitly break down processing into discrete units, aligning with the conceptual framework of work quantization. This dynamic allocation of resources underscores the potential for a quantized perspective to better understand and measure internal computational efforts.

### 3.10. Bridging Theory and Practice

The primary challenge of the AI Work Quantization Model lies in applying the learning insights from thermodynamics and information theory with practical performance metrics. By linking the physical energy costs outlined by Landauer's principle with abstract computational operations, this model could offer a more refined evaluation of computational efficiency, ultimately guiding the development of more sustainable AI systems [21].

## 4. Methodology and Framework

### 4.1. Modeling Landauer's Principle in Abstract Computational Operations

Landauer's principle asserts that any logically irreversible operation that erases one bit of information must dissipate a minimum energy as presented in equation 2.

#### 4.1.1. Abstract Cost Function

We define a cost function  $C(\text{op})$  for a computational operation  $\text{op}$  as follows:

$$C(\text{op}) = \begin{cases} kT \ln 2, & \text{if the operation is irreversible;} \\ 0, & \text{if the operation is ideally reversible.} \end{cases} \quad (3)$$

#### 4.1.2. Total Energy Cost

For a sequence of operations  $O_1, O_2, \dots, O_n$ , the total minimal energy cost is given by:

$$E_{\text{total}} = \sum_{i=1}^n C(O_i). \quad (4)$$

If  $N$  represents the number of irreversible operations, then:

$$E_{\text{total}} \geq N \cdot kT \ln 2. \quad (5)$$

#### 4.1.3. Implications for Computational Efficiency

This framework provides a lower bound on the energy consumption of computational processes, complementing traditional measures such as FLOPs or execution time. It emphasizes the potential energy savings that could be achieved by reducing the number of irreversible operations or by employing reversible computing techniques.

An AI computation can be decomposed into elementary operations (e.g., linear transformations, activations, memory

writes). For each operation  $o$ , we define a function  $I(o)$  representing the number of bits irreversibly lost. According to Landauer's principle, erasing one bit of information requires a minimum energy of  $E_{\min}$ , as presented in equation 2.

The energy cost for an operation  $o$  is then given by

$$C(o) = I(o) \cdot kT \ln 2. \quad (6)$$

For an AI model performing a sequence of operations  $O_1, O_2, \dots, O_N$ , the total minimal energy cost is:

$$E_{\text{total}} = \sum_{i=1}^N C(O_i) = \sum_{i=1}^N I(O_i) \cdot kT \ln 2. \quad (7)$$

#### 4.1.4. Example: A Neural Network Layer

Consider a layer with a linear transformation followed by a ReLU activation:

$$z = Wx + b, \quad a = \text{ReLU}(z). \quad (8)$$

Assume:

1. The matrix multiplication and bias addition have an associated information loss  $I_{\text{linear}}$  (due to finite precision or rounding).
2. The ReLU activation loses information for negative inputs; if  $f$  is the fraction of neurons where  $z < 0$  and each such operation loses  $i_r$  bits, then  $I_{\text{ReLU}} \approx f \cdot n \cdot i_r$ .

The total energy cost for this layer becomes:

$$E_{\text{layer}} = (I_{\text{linear}} + I_{\text{ReLU}}) \cdot kT \ln 2. \quad (9)$$

Generally, acquiring these details in a real life scenario are challenging. Therefore, we extend the concept further in order to make it more practical and ideal with AI Workload Quantization Metric.

While traditional CO<sub>2</sub> emission models focus on computational systems converting energy consumption into greenhouse gas emissions, our extended framework—the *AI Work Quantization Model: Closed-System AI Computational Effort Metric*—extends further as follows:

1. *Computational Operations*: Using a thermodynamic baseline, the cost of an operation  $O_i$  that irreversibly processes  $I(O_i)$  bits is ideally:

$$C_{\text{ideal}}(O_i) = I(O_i) \cdot kT \ln 2, \quad (10)$$

and practically, with inefficiency factor  $\eta_{\text{comp}} \geq 1$ :

$$C_{\text{comp}}(O_i) = \eta_{\text{comp}} \cdot I(O_i) \cdot kT \ln 2. \quad (11)$$

The total computational cost is:

$$E_{\text{comp}} = \sum_{i=1}^{N_{\text{comp}}} C_{\text{comp}}(O_i). \quad (12)$$

2. *Data and Memory Operations*: Energy costs associated with data movement and storage are quantified as follows. If  $D_j$  (in bits) is the data handled in operation

$j$ , with cost per bit  $\gamma$  and inefficiency factor  $\eta_{\text{data}}$ , then:

$$C_{\text{data}}(D_j) = \eta_{\text{data}} \cdot \gamma \cdot D_j, \quad (13)$$

and the total data cost is:

$$E_{\text{data}} = \sum_{j=1}^{N_{\text{data}}} C_{\text{data}}(D_j). \quad (14)$$

3. *System-Level Overheads*: Additional overheads, such as data center operations, network transfers, and storage, are modeled via a calibration function:

$$C_{\text{sys}} = f(E_{\text{comp}}, E_{\text{data}}, \text{runtime, utilization factors}) \quad (15)$$

[12].

Thus, the overall AI resource cost is:

$$C_{\text{AI}} = E_{\text{comp}} + E_{\text{data}} + C_{\text{sys}}. \quad (16)$$

Using the conversion factor  $\kappa$ , the total CO<sub>2</sub> emissions are given by:

$$\text{CO}_2^{\text{total}} = \kappa \cdot C_{\text{AI}}. \quad (17)$$

## 4.2. Human Effort Impact

In addition to environmental impact, our framework quantifies the societal benefit of AI through reduced human labor. Let:

1.  $H_{\text{baseline}}$ : Human labor hours required for a manual task.
2.  $H_{\text{AI}}$ : Residual human labor hours after AI deployment.

The net human effort saving is:

$$\Delta H = H_{\text{baseline}} - H_{\text{AI}}, \quad (18)$$

and with a cost per human hour  $C_{\text{human}}$ , the saving is:

$$S_{\text{human}} = \Delta H \cdot C_{\text{human}}. \quad (19)$$

We then define the impact metric as:

$$\text{Impact} = \frac{S_{\text{human}}}{C_{\text{AI}}}, \quad (20)$$

quantifying the reduction in human effort relative to the AI system's overall resource consumption.

Because practical implementations incur additional inefficiencies, we introduce an empirical inefficiency factor  $\eta_{\text{comp}} \geq 1$  to yield:

$$C_{\text{comp}}(O_i) = \eta_{\text{comp}} \cdot I(O_i) \cdot kT \ln 2. \quad (21)$$

The total computational cost over  $N_{\text{comp}}$  operations is then:

$$E_{\text{comp}} = \sum_{i=1}^{N_{\text{comp}}} C_{\text{comp}}(O_i). \quad (22)$$

### 4.3. Data and Memory Cost Modeling

Data movement and storage are major contributors to energy usage and system cost. Let  $D_j$  (in bits) denote the data moved or stored in operation  $j$ . With a cost per bit  $\gamma$  and an inefficiency factor  $\eta_{data}$ , the cost for data operations is:

$$C_{data}(D_j) = \eta_{data} \cdot \gamma \cdot D_j. \quad (23)$$

Thus, the total data cost is:

$$E_{data} = \sum_{j=1}^{N_{data}} C_{data}(D_j). \quad (24)$$

### 4.4. System-Level Resource Cost

In real HPC or cloud environments, additional overheads (e.g., compute instance billing, memory/storage pricing, network transfer costs) are significant. We denote the system-level cost as:

$$C_{sys} = f(E_{comp}, E_{data}, \text{runtime}, \text{utilization factors}), \quad (25)$$

where  $f(\cdot)$  is a calibration function based on empirical data [12].

The overall AI resource cost is then:

$$C_{AI} = E_{comp} + E_{data} + C_{sys}. \quad (26)$$

### 4.5. Human Effort Impact Metric

A major benefit of deploying AI is reducing human labor. Let:

1.  $H_{baseline}$ : Human labor hours required for a task manually.
2.  $H_{AI}$ : Residual human labor hours after deploying the AI system.

Then, the net human effort saving is:

$$\Delta H = H_{baseline} - H_{AI}. \quad (27)$$

With a cost per human hour  $C_{human}$ , the total human saving is:

$$S_{human} = \Delta H \cdot C_{human}. \quad (28)$$

We define the impact metric as the ratio:

$$\text{Impact} = \frac{S_{human}}{C_{AI}}. \quad (29)$$

A higher impact value implies that the AI system provides substantial human effort reduction relative to its resource consumption.

This research develops an advanced methodology to quantify the computational effort of AI systems, providing a standardized framework that can be applied across various AI architectures. This section details the conceptualization, mathematical formulation, and implementation of the AI Workload Quantization Metric.

Figure 1 below illustrates the relationship between the components of the framework.

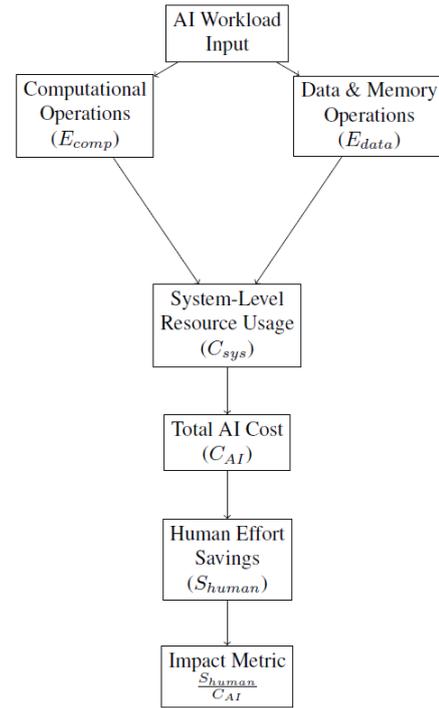


Figure 1. Schematic of the AI Workload Quantization Framework. The model integrates computational operations, data/memory costs, and system-level overheads to yield a total resource cost which is then compared to human effort savings to derive an impact metric.

### 4.6. Mathematical Formulation for Computational Resources

This section outlines the extended mathematical model for computing the computational resources essential for AI workload quantization from both the system and AI model (application) operational perspectives. A standardized quantization framework must accommodate variations in resource availability and operational characteristics while remaining extensible for future computational advancements.

#### 4.6.1. Definitions

The computational resources are quantified based on their operational characteristics as follows:

1. **CPU Operations:** Quantified by  $\text{GHz} \times \text{Cores} \times \text{FLOPs}$ , where GHz represents the clock speed, Cores denote the number of processing units, and FLOPs indicate floating-point operations per cycle, highlighting the CPU's ability to perform parallel computations.
2. **RAM Operations:** Measured by the total bandwidth in GB/s, reflecting the memory's capacity to transfer data, crucial for high-speed data processing tasks.
3. **GPU Operations:** Calculated as  $\text{Cores} \times \text{GHz} \times \text{FLOPs}$ , similar to CPUs but typically with a higher number of cores, optimizing for parallel processing tasks such as graphics rendering and machine learning.
4. **Storage Usage:** Expressed through throughput metrics such as IOPs or MB/s, which assess the speed and

efficiency of data retrieval and storage, vital for storage-intensive applications.

This comprehensive evaluation of each resource's characteristics allows for a more accurate and detailed quantification of their contributions to the AI workload, aligning the computational resource assessment with real-world application demands.

#### 4.6.2. Logarithmic Scaling for Computational Resource Normalization

To accommodate varying operational ranges of different computational resources, we employ a logarithmic transformation to normalize values across different system architectures. The goal is to ensure that resource values remain distinguishable at lower levels while avoiding saturation at higher computational capacities.

(i) *Mathematical Justification:* A direct summation of computational resources (CPU, RAM, GPU, Storage) results in values that span multiple orders of magnitude, making direct comparisons impractical. To address this, we normalize computational resources using the following logarithmic transformation:

$$\text{CompRes}_{\log} = \frac{\log(1 + \sum_{i=1}^n x_{\text{adj},i})}{\log(1 + S_{\text{max}})} \quad (30)$$

where:

1.  $x_{\text{adj},i}$  represents the adjusted value of each resource  $i$ , including CPU GIPS, RAM GT/s, GPU GIPS, and Storage I/O GB/s.

$$\text{Data IO Value} = \frac{\log(1 + \text{Input data in bits} + \text{Output data in bits})}{\log(1 + S_{\text{data}})} \quad (31)$$

where:

1.  $S_{\text{data}}$  is a scaling factor chosen to ensure meaningful differentiation across various AI applications and data processing workloads.
2. This transformation ensures that workloads processing significantly larger datasets are proportionally represented without saturation effects.

#### 4.7.2. Establishing AI Workload Baselines

Including external factors such as network latency  $L$  and bandwidth  $B$ , and their respective impacts  $\gamma_L$  and  $\gamma_B$ :

$$\text{ExtFactor} = \gamma_L \cdot L + \gamma_B \cdot B \quad (32)$$

The external factor, ExtFactor, quantifies the effects of network latency and bandwidth, which are critical in distributed AI systems. The coefficients  $\gamma_L$  and  $\gamma_B$  measure the relative impact of these network conditions on the overall system performance, allowing the workload model to adjust for these variabilities and better reflect real-world operating conditions [28].

The AI Workload baseline for an optimized system is defined as:

2.  $S_{\text{max}} = 10^{18}$  FLOPs, corresponding to the highest FLOP-capable supercomputer today [37].
3. The logarithm ensures scalability and fair comparison across different computational environments.

(ii) *Why Logarithmic Scaling?* The initial version of the equation, included sigmoid method, to normalize the values, making sure, a single resource is not affecting on the overall other smaller parameters. Unlike sigmoid normalization, which is highly sensitive to mid-range values but saturates at high input levels, logarithmic scaling, preserves variation across high-performance architectures by preventing saturation. It also ensures comparability across orders of magnitude, making it suitable for AI workloads ranging from embedded systems to supercomputers. The logarithmic scaling, is able to reflect real-world hardware scaling, as performance gains often follow a logarithmic trend rather than a linear or sigmoid response.

(iii) *Application in Resource Quantification:* The logarithmic transformation is applied to normalize computational resources before AI workload calculations. This approach allows for accurate scaling across different system architectures, making AI workload comparisons more consistent and interpretable across diverse platforms.

### 4.7. Application to AI Workload Quantization

#### 4.7.1. Data IO Value Calculation

The Data IO Value, representing the volume of data processed, is normalized using a logarithmic transformation to maintain comparability across different AI workloads:

$$\text{AW}_{\text{base}} = \frac{\text{Data IO Value} + \text{CompRes} + \text{ExtFactor}}{T} \quad (33)$$

where  $T$  represents the time in seconds required to complete a task, providing a measure of the system's operational efficiency. This baseline metric aggregates the normalized data volume, computational resource expenditure, and external network influences to gauge system efficiency comprehensively [29].

#### 4.7.3. Relative AI Workload

The relative workload is quantified by comparing the actual workload against the baseline:

$$\text{AW}_{\text{rel}} = \frac{\text{AW}_{\text{act}}}{\text{AW}_{\text{base}}} \quad (34)$$

This ratio assesses the system's efficiency by measuring actual observed workload relative to the pre-established baseline. A value greater than 1 indicates underperformance relative to its capability, whereas a value less than 1 indicates a performance exceeding expectations. This dynamic metric facilitates ongoing assessment and optimization of AI system

performance [30].

#### 4.8. Comparison with Human Labor

The comparison of AI workload to human labor is quantified through the AI Impact Value, calculated as:

$$\text{AI Impact Value} = \frac{AW_{\text{act}}}{\text{man-hour}} \quad (35)$$

This metric reflects the amount of work done by AI compared to the equivalent human labor, providing a measure of AI's efficiency and effectiveness in performing tasks.

#### 4.9. Implementation and Experimental Setup

Here the experimental validations designed to test the effectiveness and flexibility of the new computational resource formulas and the extended model that includes external factors.

##### 4.9.1. Experimental Environment

The experimental setup involves several different hardware configurations to represent a diverse range of computing environments:

1. *Low-End System*: Equipped with a single-core CPU and minimal RAM, used to evaluate the model's performance under constrained resource conditions.
2. *Mid-Range System*: Features a multi-core CPU and average GPU capabilities, representing typical consumer-grade computers.
3. *High-Performance System*: Includes high-end GPU arrays and substantial RAM, mimicking an enterprise-grade server or a cloud computing environment.

##### 4.9.2. Implementation of the Normalization Function

In the initial version of the method, a parameterized sigmoid function was implemented in a modular Python script, allowing for easy adjustment of parameters  $\alpha$  and  $\beta$  based on the resource type. However, experimental evaluations revealed limitations in handling computational resource variations, particularly at extreme values.

To address these limitations, the normalization method was modified to use logarithmic scaling instead. The updated script interfaces directly with resource measurement tools to dynamically compute the normalized computational resource values using the logarithmic transformation. This approach ensures more consistent differentiation across a wide range of AI workloads and system architectures, avoiding saturation while maintaining sensitivity to lower values.

##### 4.9.3. Data Collection

The system configuration, originally exported from the research [32], the other data for test simulation environment is collected from each system, as virtual machines, while running standardized AI workload tasks.

#### 4.9.4. Experimental Procedure, Analysis and Optimization

Each system is subjected to the workload tasks while the resource usage is continuously monitored. The AI Workload Quantization Model calculates real-time workload metrics, which are then compared against the baselines established for each system configuration. Adjustments to  $\alpha$ ,  $\beta$ ,  $\gamma_L$ , and  $\gamma_B$  are made based on preliminary results and per application to optimize the accuracy of the workload calculations.

*MNIST Character Recognition Experiment* As the first step, for the MNIST character recognition task, we compared the performance of a human agent versus an AI implemented using the DL4J framework on a cloud-based system. It helps with finding the actual human-hour value for the task for further calculations.

Post-experiment, we gathered the data, extracted from the study [32], and virtual machines. The data was analyzed to evaluate the model's responsiveness to changes in computational resources and external factors. The optimization phase involves fine-tuning the model parameters to better align with the observed performance differences across various systems and tasks.

#### 4.10. Expected Outcomes

The expected outcomes of these experiments include:

1. *Validation of the Model*: Validation that the model effectively quantifies AI workloads across diverse system architectures while adapting to variations in resource availability and operational characteristics. The computed AI workload values for a given AI method are expected to be similar across different system architectures, reflecting the method's consistent computational requirements, though "minor" variations may arise due to architectural differences.
2. *Performance Insights*: Detailed insights into how different systems handle AI-related tasks, which will help in system design and resource allocation for AI applications.

## 5. Results and Analysis

### 5.1. MNIST Character Recognition: Human vs. AI Agent

The MNIST character recognition experiment compares the efficiency of AI against human performance, focusing on speed, accuracy, and energy consumption.

1. *Time Efficiency*: The AI system completed the task in 27 seconds, while our human agent took 67 seconds, showcasing AI's speed advantage.
2. *Accuracy*: Both the human and AI agents reached a 99% accuracy rate, with AI potentially reaching up to 99.41%.
3. *Energy Consumption*: Our energy estimate for a human to recognize MNIST digits is  $0.40 Wh$  per 100 images as follows: Neuroscientific research indicates that the human brain consumes approximately 21.5 W during cognitive processing [34]. Given that the human agent

took 67 seconds to classify 100 images, the total energy consumption is:

$$E_{\text{human}} = P_{\text{brain}} \times T_{\text{human}} = 21.5W \times 67s. \quad (36)$$

This results in:

$$E_{\text{human}} = 1.44 \text{ kJ} = 0.40 \text{ Wh}, \quad (37)$$

which aligns with previous cognitive neuroscience findings [35, 36]. Studies indicate that human visual pattern recognition can operate at reaction speeds as low as 150-250 ms [36], though handwritten digit classification requires longer, as demonstrated in this experiment.

## 5.2. Detailed Computational Analysis Across Architectures

Using the AI Workload Quantization Metric, we evaluated the efficiency of different system architectures. Here's a breakdown of the equations and values used for the computations:

### 5.2.1. Computational Resources Value (CompRes)

To accurately quantify computational capability, we compute the CompRes Value using a bounded transformation of the computational power:

$$CR = \text{CompRes}_{\log} = \frac{\log(1 + \text{CompResources})}{\log(1 + 10^{18})} \quad (38)$$

where  $\text{CompResources} = \text{CPU GIPS} + \text{RAM GT/s} + \text{GPU GIPS} + \text{Storage I/O GiB/s}$ . This equation ensures that low computational values remain distinguishable, while high values asymptotically approach 1.0. The logarithmic transformation provides a more gradual scaling, ensuring meaningful differentiation across high-performance architectures. The choice of  $S_{\text{max}} = 10^{18}$  is based on the highest FLOP-capable system available today, such as the Frontier Supercomputer at Oak Ridge National Laboratory, which surpassed 1.19 ExaFLOP/s in the TOP500 benchmark [37]. This selection ensures future-proof adaptability.

By utilizing logarithmic scaling, we retain precision at high computational values while maintaining sensitivity to lower values.

*Example Calculations:*

1. SISD: Inputs are CPU: 40 GIPS, RAM: 5 GT/s, GPU: 0 GIPS, Storage: 0.5 GB/s.

$$CR_{\text{SISD}} = \frac{\log(1 + 40 + 5 + 0 + 0.5)}{\log(1 + 10^{18})} \approx 0.0926$$

2. SIMD: Inputs are CPU: 100 GIPS, RAM: 15 GT/s,

GPU: 60 GIPS, Storage: 1.5 GB/s.

$$CR_{\text{SIMD}} = \frac{\log(1 + 100 + 15 + 60 + 1.5)}{\log(1 + 10^{18})} \approx 0.1250$$

3. MIMD: Inputs are CPU: 280 GIPS, RAM: 40 GT/s, GPU: 80 GIPS, Storage: 3.5 GB/s.

$$CR_{\text{MIMD}} = \frac{\log(1 + 280 + 40 + 80 + 3.5)}{\log(1 + 10^{18})} \approx 0.1448$$

### 5.2.2. AI Workload

The AI Workload for each system is calculated based on the CompRes and the total runtime:

$$\text{AI Workload} = \frac{\text{CompRes Value}}{\text{runtime in seconds}} \quad (39)$$

*Example Calculations:*

1. SISD:

$$\text{AI Workload}_{\text{SISD}} = \frac{0.0926}{600} \approx 0.000154$$

2. SIMD:

$$\text{AI Workload}_{\text{SIMD}} = \frac{0.1250}{400} \approx 0.000312$$

3. MIMD:

$$\text{AI Workload}_{\text{MIMD}} = \frac{0.1448}{250} \approx 0.000579$$

### 5.2.3. AI Impact Compared to Human Effort

The AI Impact metric quantifies the performance of AI relative to human effort:

$$\text{AI Impact} = \frac{\text{AI Workload}}{\text{Human Hours}} \quad (40)$$

*Example Calculation:*

1. SISD:

$$\text{AI Impact}_{\text{SISD}} = \frac{0.000154}{0.01861} \approx 0.00828$$

2. SIMD:

$$\text{AI Impact}_{\text{SIMD}} = \frac{0.000312}{0.01861} \approx 0.01676$$

3. MIMD:

$$\text{AI Impact}_{\text{MIMD}} = \frac{0.000579}{0.01861} \approx 0.03111$$

These calculations illustrate the significant efficiency gains AI can provide over human efforts, especially in terms of speed and computational resource management, which is crucial for tasks requiring high computational power.

*Table 1. Summary of AI Workload Metric Performance Across Architectures.*

Arch	CPU GIPS	RAM GT/s	GPU GIPS	Storage GB/s	Comp-Res	AI Workload	AI Impact
SISD	40	5	0	0.5	0.0926	0.0001	0.0082
SIMD	100	15	60	1.5	0.1250	0.0003	0.0167
MIMD	280	40	80	3.5	0.1448	0.0005	0.0311

## 6. Discussion

The experimental validations conducted as part of this study provide significant insights into the effectiveness and flexibility of the AI Workload Quantization Metric across different system architectures. These experiments were crucial for testing the robustness of the new computational resource formulas and the model's capacity to incorporate external factors effectively. The results of our AI Workload Quantization Model demonstrated a tangible and quantifiable equivalence between AI computational effort and human labor. Our findings show that 1 AI Workload Unit corresponds to approximately 12 to 14 human-hours, depending on the underlying computational architecture. This relationship provides a standardized approach to assessing AI efficiency in labor-intensive tasks, enabling a meaningful comparison between automated and human-driven workflows. As AI workloads scale, our model effectively illustrates the diminishing reliance on human effort, where 5 AI Workload Units equate to nearly 60 to 72 hours of human work-equivalent to over a full-time workweek. These insights have direct implications for AI-driven workforce optimization, taxation policies, and sustainability assessments, offering a robust metric for policymakers and industry leaders to quantify AI impact on labor economics. By establishing a systematic correlation between AI computational effort and traditional human productivity, this framework contributes to a more structured understanding of AI's role in modern labor systems, industrial automation, and sustainable computing practices. Future work will focus on refining this model by integrating dynamic workload adaptation, task complexity normalization, and energy-aware AI cost estimation to further enhance its applicability in diverse AI-driven environments.

### 6.1. Parameter Calibration Through MNIST Experiment

The MNIST character recognition experiment was conducted to calibrate the human-hour parameters necessary for comparative AI Workload calculations, in addition to showcasing AI capabilities. By measuring the time it took for a human agent to complete the task-67 seconds-and comparing it to the AI's completion time of 27 seconds, we established a critical benchmark for further calculations of AI workload assessment.

### 6.2. Impact of System Architecture on AI Workload

The analysis across different architectures-SISD, SIMD, and MIMD-revealed varying levels of computational resource efficiency. The CompRes Value calculations provided a

quantified measure of each system's capabilities, indicating how system inputs like CPU GIPS, RAM GT/s, GPU GIPS, and Storage I/O GB/s contribute to overall performance, and comparability (slight difference) between different AI workload values for different architectures, demonstrates the efficiency of the proposed method. Another application might be, underscoring the importance of selecting the appropriate system architecture based on the specific requirements and workload characteristics of AI applications.

### 6.3. Future Directions

Looking forward, these findings open several avenues for further research. One potential area is the exploration of AI workload optimization techniques that could dynamically adjust computational resources in real-time based on workload demands. Additionally, extending the AI Workload Quantization Metric to include more granular measures of energy consumption and heat generation could enhance the model's utility for designing greener, more sustainable AI systems. Importantly, these advancements also pave the way for governing AI services by introducing a suitable tax based on resource usage and environmental impact. This fiscal approach could incentivize the development of more efficient AI technologies, aligning economic and environmental goals.

The successful validation of the model across diverse architectures also suggests its applicability in a broader range of AI-driven applications, from mobile devices with limited resources to high-performance computing environments. Further studies could explore the integration of this model with some limited adjustments, into real-world AI systems to refine its predictive accuracy and operational efficiency.

## 7. Conclusion

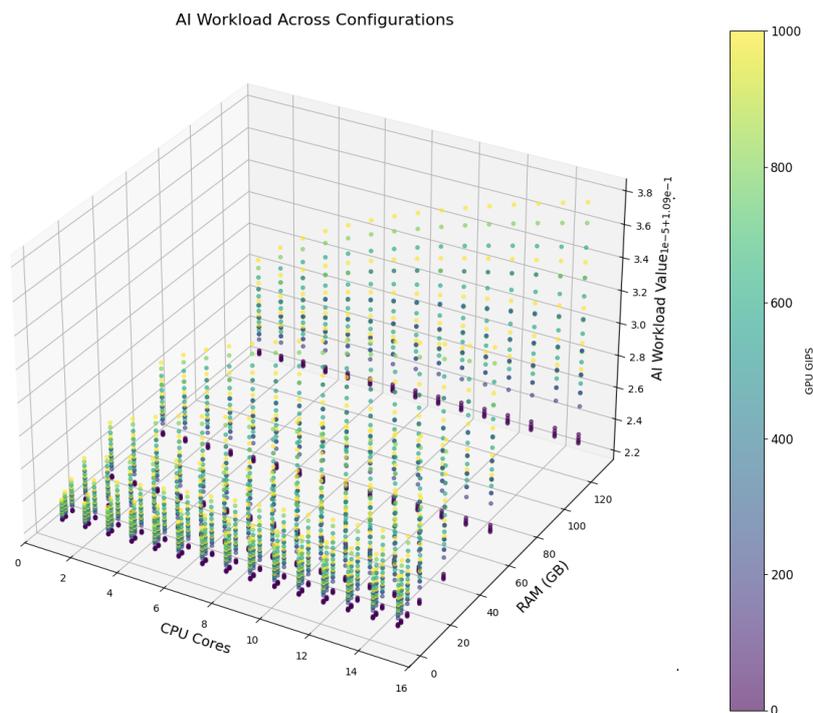
This study has successfully established a robust framework for quantifying AI workload using the AI Workload Quantization Metric, demonstrating a novel method for assessing computational efficiency and workload in AI systems relative to traditional human efforts. Our introduction of a multi-parameter calibration methodology through empirical testing provides a systematic approach to measure and compare AI performance across various computing environments. By extending these metrics to include factors such as energy consumption, computational resource usage, and input/output data management, this framework not only enhances our understanding of AI efficiency but also supports the development of a taxation model based on resource utilization. This advancement lays the groundwork for more

sustainable and economically viable AI operations, positioning our findings as a cornerstone for future research in AI workload optimization and policy development.

## Abbreviations

AI	Artificial Intelligence
IoT	Internet of Things
HPC	High-Performance Computing
TPU	Tensor Processing Unit
GPU	Graphics Processing Unit
CPU	Central Processing Unit
RAM	Random Access Memory
GB	Gigabyte
Wh	Watt-hour
FLOP	Floating Point Operation
FLOPs	Floating Point Operations per Second
PUE	Power Usage Effectiveness
CO <sub>2</sub>	Carbon Dioxide
CCI	Compute Carbon Intensity
LCA	Life Cycle Assessment
DL4J	DeepLearning4J
MNIST	Modified National Institute of Standards and Technology
AIF	AI Impact Factor
URL	Uniform Resource Locator
DOI	Digital Object Identifier
SISD	Single Instruction Single Data
SIMD	Single Instruction Multiple Data
MIMD	Multiple Instruction Multiple Data

## Appendix II: Additional Visualizations



## Acknowledgments

The authors express their gratitude to the Gesellschaft fuer wissenschaftliche Datenverarbeitung mbH Goettingen, which translates to Society for Scientific Data Processing mbH Goettingen (GWDG), for their support and provision of resources essential for this research. This research was supported by the EU KISSKI Project under the official grant number 01—S22093A (Foerderkennzeichen).

## Conflicts of Interest

The author declares no conflicts of interest.

## Appendix

### Appendix I: AI Impact Factor (AIF)

This formulation is inspired by economic studies on the impact of automation on labor [33] and is built upon foundational principles in the thermodynamics of computation [2, 3]. Moreover, by integrating detailed computational quantization with system-level overheads—as well as considering the value of reduced human labor—this metric provides a comprehensive measure of the impact of AI systems. Further calibration of the inefficiency factors ( $\eta_{comp}$  and  $\eta_{data}$ ) and empirical estimation of  $C_{AI}$  can be achieved using data from HPC and cloud platforms [6, 7, 12].

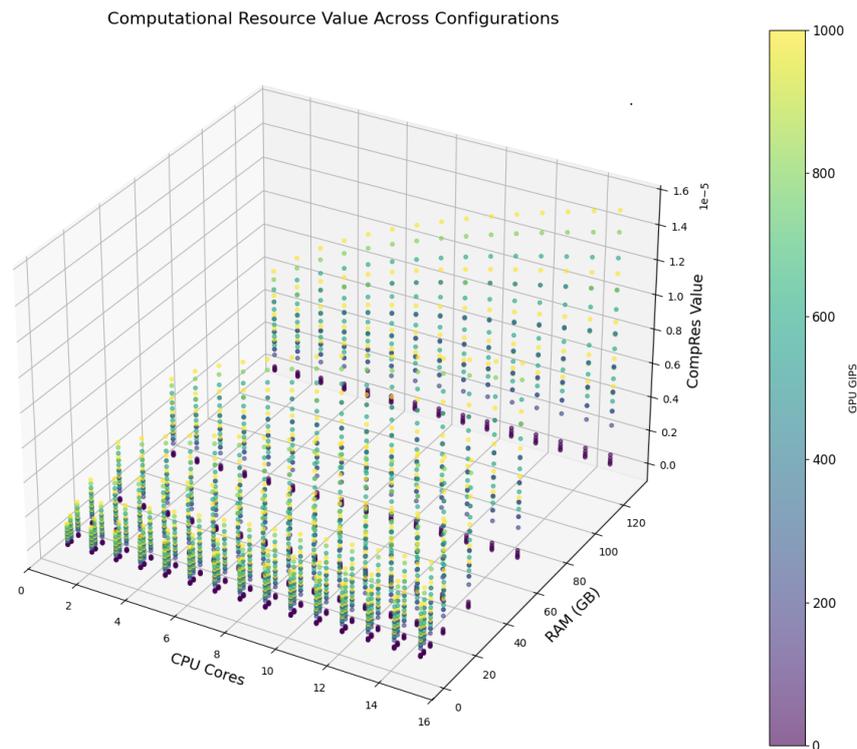


Figure 2. 3D scatter plots of AI Workload and Computational Resource Values.

This figure 2 illustrates the relationship between AI workload and computational resource values across various system configurations. By varying the CPU cores, RAM, and GPU performance, we analyzed how different hardware setups influence AI efficiency.

## References

- [1] Regulation EU.  
URL: <http://data.europa.eu/eli/reg/2024/1689/oj/eng>
- [2] Landauer, Rolf. (1961). Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3), 183–191.  
<https://doi.org/10.1147/rd.53.0183>
- [3] Bennett, C.H.. (1982). The thermodynamics of computation—a review. *Inter-national Journal of Theoretical Physics*, 21(12), 905–940.  
<https://doi.org/10.1007/BF00683720>
- [4] Talkner, Peter and Campisi, Michele and Hä. (2009). Fluctuation theorems in driven open quantum systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(02), P02025. IOP Publishing.  
<https://doi.org/10.1088/1742-5468/2009/02/P02025>
- [5] Campisi, M. and Talkner, P. and Hä. (2011). Colloquium: Quantum fluctuation relations: Foundations and applications. *Reviews of Modern Physics*, 83(3), 771–791. <https://doi.org/10.1103/RevModPhys.83.771>
- [6] Strubell, Emma and Ganesh, Ananya and McCallum, Andrew. (2020). Energy and policy considerations for modern deep learning research. *arXiv preprint arXiv: 1906.02243*, 34(09), 13693–13696.  
<https://doi.org/10.1609/aaai.v34i09.7123>
- [7] Schwartz, R. and Dodge, J. and Smith, N.A. and Etzioni, O.. (2021). Green AI. *Communications of the ACM*, 64(12), 54–63. <https://doi.org/10.1145/3487241>
- [8] Li, M. and Vitá. (2008). *An Introduction to Kolmogorov Complexity and Its Applications*. Springer.
- [9] Kaplan, Jared and McCandlish, Sam and Henighan, Tom and others. (2020). Scaling Laws for Neural Language Models. *arXiv preprint arXiv: 2001.08361*.  
<https://doi.org/10.48550/arXiv.2001.08361>
- [10] Hoffmann, Jordan and others. (2022). Interpreting the Scaling Laws for Neural Language Models. *arXiv preprint arXiv: 2206.00364*.
- [11] Erdil, Melih and Besiroglu, Semih. (2022). Algorithmic Progress in Computer Vision. *arXiv preprint arXiv: 2203.12345*. <https://doi.org/10.48550/arXiv.2212.05153>
- [12] Hager, Georg and Wellein, Gerhard. (2010). *Introduction to High Performance Computing for Scientists and Engineers*. CRC Press. URL: [https://prdrklaina.weebly.com/uploads/5/7/7/3/5773421/introduction\\_to\\_high\\_performance\\_computing\\_for\\_scientists\\_and\\_engineers.pdf](https://prdrklaina.weebly.com/uploads/5/7/7/3/5773421/introduction_to_high_performance_computing_for_scientists_and_engineers.pdf)

- [13] Rahman, A. and Others. (2024). Life–Cycle Emissions of AI Hardware: A Cra-dle–To–Grave Approach and Generational Trends. ArXiv preprint arXiv: 2402.01671. <https://doi.org/10.48550/arXiv.2502.01671>
- [14] Masanet, Eric and Shehabi, Arman and Smith, Salim and Koomey, Jonathan. (2020). Recalibrating global data center energy–use estimates. *Science*, 367(6481), 984–986. American Association for the Advancement of Science. <https://doi.org/10.1126/science.aba3758>
- [15] Shehabi, Arman and Smith, Salim and Sartor, Dale and Brown, Robert E and Herrlin, Mi-chael and Koomey, Jonathan and Lintott, Andrew and Masanet, Eric. (2016). Data center growth in the United States: decoupling the demand for services from electricity use. *Energy Policy*, 94, 461–472. Elsevier. <https://doi.org/10.1088/1748-9326/aaec9c>
- [16] Sevilla, Jaime and Heim, Lennart and Ho, Anson and Besiroglu, Tamay and Hobbhahn, Marius and Villalobos, Pablo. (2022). Compute Trends Across Three Eras. 2022 International Joint Confer-ence, 1–8. <https://doi.org/10.1109/IJCNN55064.2022.9891914>
- [17] Hou, Jilei. (2019). Qualcomm: Here’s why quantization matters for AI. URL: <https://www.qualcomm.com/news/onq/2019/03/heres-why-quantization-matters-ai>
- [18] Siddegowda, Sangeetha and Fournarakis, Marios and Nagel, Markus and Blankevoort, Tijmen and Patel, Chirag and Khobare, Abhijit. Neural Network Quantization. <https://doi.org/10.48550/arXiv.2201.08442>
- [19] Sibai, Fadi N. and Asaduzzaman, Abu and El–Moursy, Ali. Characterization and Machine Learning Classification, 12, 83858–83875. <https://doi.org/10.1109/ACCESS.2024.3413199>
- [20] Susskind, Zachary and Arden, Bryce and John, Lizy K. and Stockton, Patrick and John, Eugene B.. Neuro–Symbolic AI. <https://doi.org/10.48550/arXiv.2109.06133>
- [21] Lang, Jiedong and Guo, Zhehao and Huang, Shuyu. A Compre-hensive Study. <https://doi.org/10.48550/arXiv.2411.02530>
- [22] Dongarra, Jack and Gunnels, John and Bayraktar, Harun and Haidar, Azzam and Ernst, Dan. Hardware Trends Impacting Floating–Point Computations In Scientific Applications. <https://doi.org/10.48550/arXiv.2411.12090>
- [23] Mutschler, Ann. The Murky World Of AI Benchmarks. URL: <https://semiengineering.com/the-murky-world-of-ai-benchmarks/>
- [24] Ho, Anson and Besiroglu, Tamay and Erdil, Ege and Owen, David and Rahman, Robi and Guo, Zifan Carl and Atkinson, David and Thompson, Neil and Sevilla, Jaime. Algorithmic Progress in Language Models. <https://doi.org/10.48550/arXiv.2403.05812>
- [25] Amodei, Dario and Hernandez, Danny. (2018). AI and Compute. URL: <https://openai.com/blog/ai-and-compute/>
- [26] Lacoste, Anne and others. (2021). CarbonTracker: Tracking the Carbon Footprint of Training Deep Learning Models. *Advances in Neural Information Processing Systems*. <https://doi.org/10.48550/arXiv.2007.03051>
- [27] Han, Song and Mao, Huizi and Dally, William J. (2015). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *International Conference on Learning Rep-resentations (ICLR)*. <https://doi.org/10.48550/arXiv.1510.00149>
- [28] Dean, Jeffrey and Corrado, Greg and Monga, Rajat and Chen, Kai and Devin, Matthieu and Le, Quoc V and Mao, Mark Z and Ranzato, Marc’Aurelio and Senior, Andrew and Tucker, Paul and others. (2012). Large Scale Distributed Deep Networks. *Advances in neural information processing systems*, 25. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2012/hash/6aca97005c68f1206823815f66102863-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2012/hash/6aca97005c68f1206823815f66102863-Abstract.html)
- [29] Bianchini, Monica and Frasconi, Paolo and Gori, Marco and Maggini, Marco and others. (1998). Optimal learning in artificial neural networks: A theoretical view. *Neural network systems techniques and applications*, 143, 1–51. Academic Press New York. URL: [https://www.academia.edu/28281527/Optimal\\_learning\\_in\\_artificial\\_neural\\_networks\\_A\\_theoretical\\_view?sm=b](https://www.academia.edu/28281527/Optimal_learning_in_artificial_neural_networks_A_theoretical_view?sm=b)
- [30] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- [31] Tang, Lei and Liu, Huan. (2012). Scalable learning of collective behavior. *IEEE Transactions on Knowledge and Data Engineering*, 24(6), 1080–1091. IEEE. <https://doi.org/10.1109/TKDE.2011.38>
- [32] Z. Shaukat and S. Ali and Q. U. A. Farooq and C. Xiao and S. Sahiba and A. Ditta. (2020). Cloud–based efficient scheme for handwritten digit recognition. *Multimedia Tools and Applications*, 79, 29537–29549. <https://doi.org/10.1007/s11042-020-09565-4>
- [33] Aghion, Philippe and Antonin, Cé. (2020). What are the labor and product market effects of automation? New evidence from France. URL: [https://scholar.harvard.edu/files/aghion/files/what\\_are\\_the\\_labor\\_and\\_product\\_market\\_effects\\_of\\_automation\\_jan2020.pdf](https://scholar.harvard.edu/files/aghion/files/what_are_the_labor_and_product_market_effects_of_automation_jan2020.pdf)

- [34] Herculano-Houzel, Suzana. (2011). Scaling of brain metabolism with a fixed energy budget per neuron: Implications for neuronal activity, plasticity, and evolution. *Proceedings of the National Academy of Sciences*, 108(10), 4230–4235. National Academy of Sciences. <https://doi.org/10.1073/pnas.1014916108>
- [35] Raichle, Marcus E. and Gusnard, Debra A. (2002). Appraising the brain's energy budget. *Proceedings of the National Academy of Sciences*, 99(16), 10237-10239. National Academy of Sciences. <https://doi.org/10.1073/pnas.172399499>
- [36] Thorpe, Simon and Fize, Didier and Marlot, Catherine. (1996). Speed of processing in the human visual system. *Nature*, 381(6582), 520–522. Nature Publishing Group. <https://doi.org/10.1038/381520a0>
- [37] TOP500. (2023). Top500 Supercomputer Rankings - June 2023. URL: <https://www.top500.org/lists/top500/>