# The Transition Institute 1.5

L'ambition d'une véritable transition

## WORKING PAPER

#5 - 11/2025

# From Observation to Optimization: A Systematic Methodology for Engineering Sustainable AI Systems

ROBLEX NANA TCHAKOUTE

Mines Paris, PSL University, Centre de Recherche en Informatique (CRI), 77305 Fontainebleau, France.

Mines Paris, PSL University, Center for Mathematical Morphology (CMM), 77305 Fontainebleau, France.

Mines Paris, PSL University, Centre for Materials Forming (CEMEF), UMR CNRS 7635, 06904 Sophia Antipolis, France

*roblex.nana_tchakoute@minesparis.psl.eu*



DOI : 10.23646/ygfy-0111

MINES PARIS | PSL | TTI.5 The Transition Institute

# TABLE OF CONTENTS

# Abstract


The rapid and ongoing expansion of Artificial Intelligence, particularly large-scale Deep Learning models, has positioned computational power as a key driver of modern innovation. However, this progress is shadowed by a serious consequence: an unsustainable trajectory of energy consumption. The electrical power required to train and operate these complex models now represents a first-order economic and environmental constraint, posing a critical challenge to the long-term viability and societal acceptance of AI. To date, efforts to improve the energy efficiency of AI systems have often been fragmented, addressing specific components or algorithmic techniques in an ad-hoc manner. This approach lacks a unified and systematic engineering process that integrates the full lifecycle from initial system diagnosis to the deployment of verifiable optimizations.

This paper addresses this methodological gap. We propose and detail a comprehensive, four-pillar framework, "Measure, Understand, Model, and Optimize," that structures the pursuit of energy efficiency as a virtuous iterative cycle. By synthesizing a broad review of the state-of-the-art in energy metrology, system characterization, predictive modeling, and resource management, our framework provides a coherent and actionable workflow. It transforms energy optimization from a specialized art into a rigorous engineering discipline. This work provides a practical roadmap for researchers, developers, and infrastructure managers to systematically analyze, predict, and improve the energy footprint of their systems. In doing so, it aims to foster the necessary engineering principles for a truly Sustainable AI, ensuring that its development and deployment are not only powerful but also responsible.



***Keywords:*** runtime prediction, power modeling, energy efficiency, analytical framework, HPC-AI, DL training, GPU modeling.

# I. Introduction: The Need for an Engineering Discipline in Sustainable AI

The 21st century is increasingly defined by two concurrent narratives: the urgent global imperative for sustainable development and the exponential rise of Artificial Intelligence. AI models have become powerful tools for tackling complex challenges, from climate modeling and medical diagnostics to resource optimization. Yet, this promise is predicated on a computational paradigm whose energy footprint is growing at a whopping rate [1]. The training of a single large language model can now consume several Gigawatt-hours of electricity, corresponding to a carbon footprint of hundreds of tons of CO2 equivalent [2], [3]. As these models proliferate, data centers that host them are becoming active drivers of global electricity demand, raising serious concerns about the long-term sustainability of the entire digital ecosystem [4], [5].

If Artificial Intelligence is to be a net positive for sustainable development, it cannot be built on an energetically unsustainable foundation. Addressing this challenge requires moving beyond basic incremental improvements. The current approach to energy efficiency in AI is often fragmented. Hardware architects design more efficient chips [6]; system software researchers develop low-power runtimes [7], [8]; and machine learning scientists design more compact algorithms [9], [10], [11]. Although each of these contributions is valuable, they often occur in silos, lacking a unifying framework that connects the physical consumption of electricity with the abstract logic of an AI model through all the intermediate layers of the system stack. What is missing is a formal **engineering discipline for Sustainable AI** (a systematic process that enables any practitioner to diagnose, analyze, and optimize their systems in a holistic and reproducible manner).

This paper proposes the aforementioned discipline. We formalize the process of energy-aware computing into a virtuous and iterative cycle built upon four fundamental pillars:

1. **Measure:** To make the invisible visible by accurately quantifying energy consumption across all relevant system components.
2. **Understand & Characterize:** To move from raw data to insight by analyzing the causal links between an application's behavior and its energy signature.
3. **Model:** To abstract this understanding into predictive models that enable a form of operational prospective analysis (looking into the immediate future to evaluate the impact of different configurations without running costly real-world scenarios).

4. **Optimize:** To translate predictive insight into action by deploying intelligent strategies that improve the energy-performance trade-off.

By structuring our analysis around this cycle, we reviewed the state-of-the-art tools and techniques available for each stage, drawing on the latest advances in computer architecture, systems software, and high-performance computing. We demonstrate how these individual pieces can be assembled into a coherent workflow to provide a tangible roadmap for the community. The ultimate goal of this work is to equip researchers, developers, and infrastructure managers with methodological tools to systematically integrate energy efficiency as a first-order design constraint, thereby transforming the development of AI from a pursuit of pure performance into a more balanced and sustainable engineering endeavor.

# II. The Virtuous Cycle of Sustainable AI Engineering: A Methodological Framework

The optimization of a complex system is not a straightforward one-shot action, but a continuous process of refinement. To formalize this process for the domain of energy-aware AI, we structure our methodology as an iterative, four-pillar cycle. This **virtuous cycle of energy optimization**, illustrated in Figure 1, provides a systematic operational methodology for transforming energy-inefficient systems into highly optimized ones. Each phase logically builds upon the previous one and provides feedback that initiates new rounds of improvement. The process is iterative, with the validation of optimizations providing new data that feeds back into the measurement and understanding phases for continuous improvement.

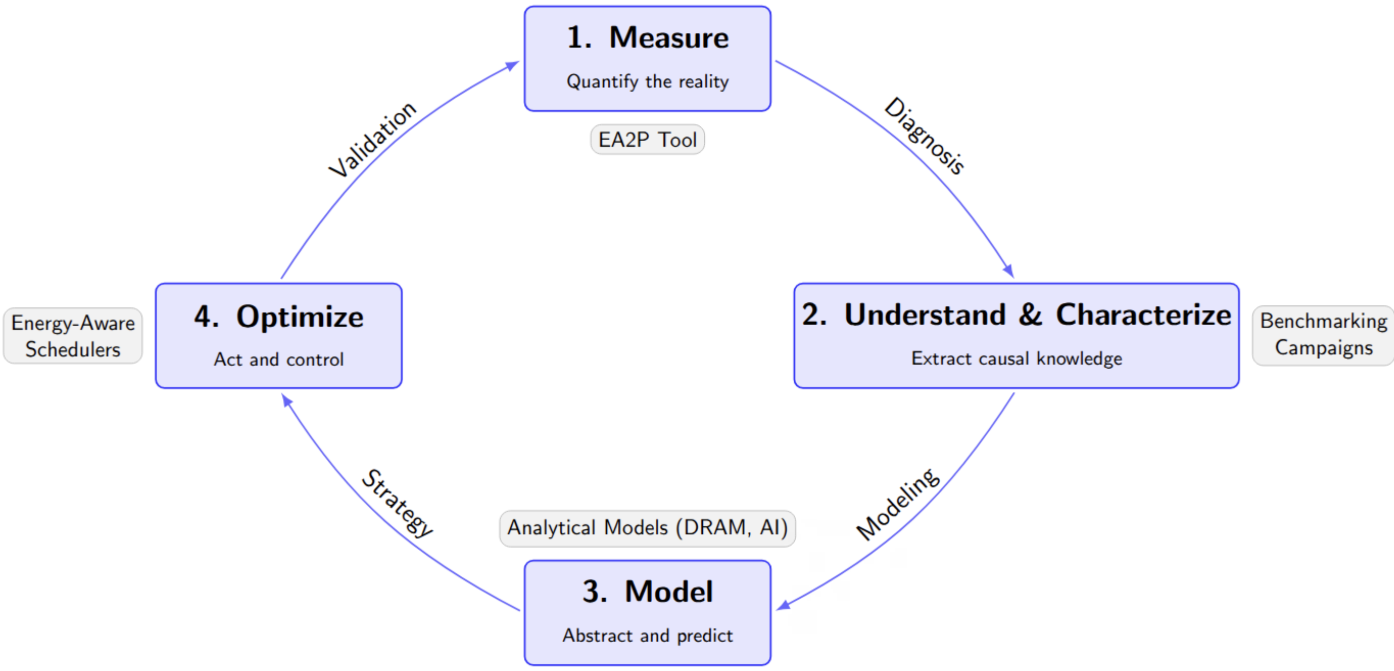


*Figure 1: The virtuous cycle of energy optimization, illustrating the four fundamental pillars.*

The four pillars of this cycle are:

- **Pillar 1: Measure.** This is the empirical foundation. In line with the engineering adage: ***«If you cannot measure it, you cannot improve it»,*** this phase transforms an abstract problem (e.g., *«the cluster is inefficient»*) into a set of hard, quantifiable data points. The goal is to make the invisible flow of energy visible and to establish a precise, multi-faceted baseline against which all future improvements will be judged.
- **Pillar 2: Understand & Characterize.** This pillar moves beyond raw data to extract causal

knowledge. It is a phase of diagnostic analysis, where the objective is to correlate the measured energy consumption with specific software behaviors and hardware states. This phase answers the critical question: *«Why does my system consume energy in this specific way under this specific workload?»* Targeted benchmarking and trade-off analysis are the core activities here.

- **Pillar 3: Model.** This is the phase of abstraction and generalization. The understanding gained in the previous step is encapsulated into predictive mathematical models. The goal of modeling, in this context, is to enable operational prospective analysis: the ability to ask «what-if» questions and predict the energy and performance outcomes of countless potential configurations (e.g., different hardware, different model hyperparameters) without the prohibitive cost of running each one empirically. These models form the core intelligence of any advanced optimization strategy
- **Pillar 4: Optimize.** This is the culmination of the cycle: turning insight and prediction into action. Based on the intelligence provided by the models, this phase involves designing, implementing, and deploying control strategies (such as advanced resource scheduling or runtime adaptations) that actively steer the system towards a more energy-efficient operating point. The validation of the gains achieved in this phase provides new, refined measurements that feed back into the first pillar, thus closing the loop and enabling continuous improvement.

The remainder of this paper will delve into the state-of-the-art techniques and the core challenges associated with each of the four previous pillars, thus providing a comprehensive guide to the practical implementation of our methodology.

## 1. Pillar 1: The Foundation of Measurement – A review of the literature

The entire edifice of energy optimization relies on the ability to perform accurate, reliable, and fine-grained measurements. Without a well-grounded empirical foundation, any attempt at optimization is merely guesswork. This section reviews the state-of-the-art energy metrology for high-performance computing (HPC) and AI systems, covering the fundamental approaches, the tools of the trade, and the inherent challenges that must be overcome to establish a meaningful baseline.

### 1.1. The Metrology Spectrum: In-Band vs. Out-of-Band Measurement

The acquisition of energy-related data from a computer system can be broadly categorized into two distinct approaches (In-band and Out-of-band), each with a fundamental trade-off between accuracy, granularity, and practicality. Figure 2 provides an overview of both approaches.

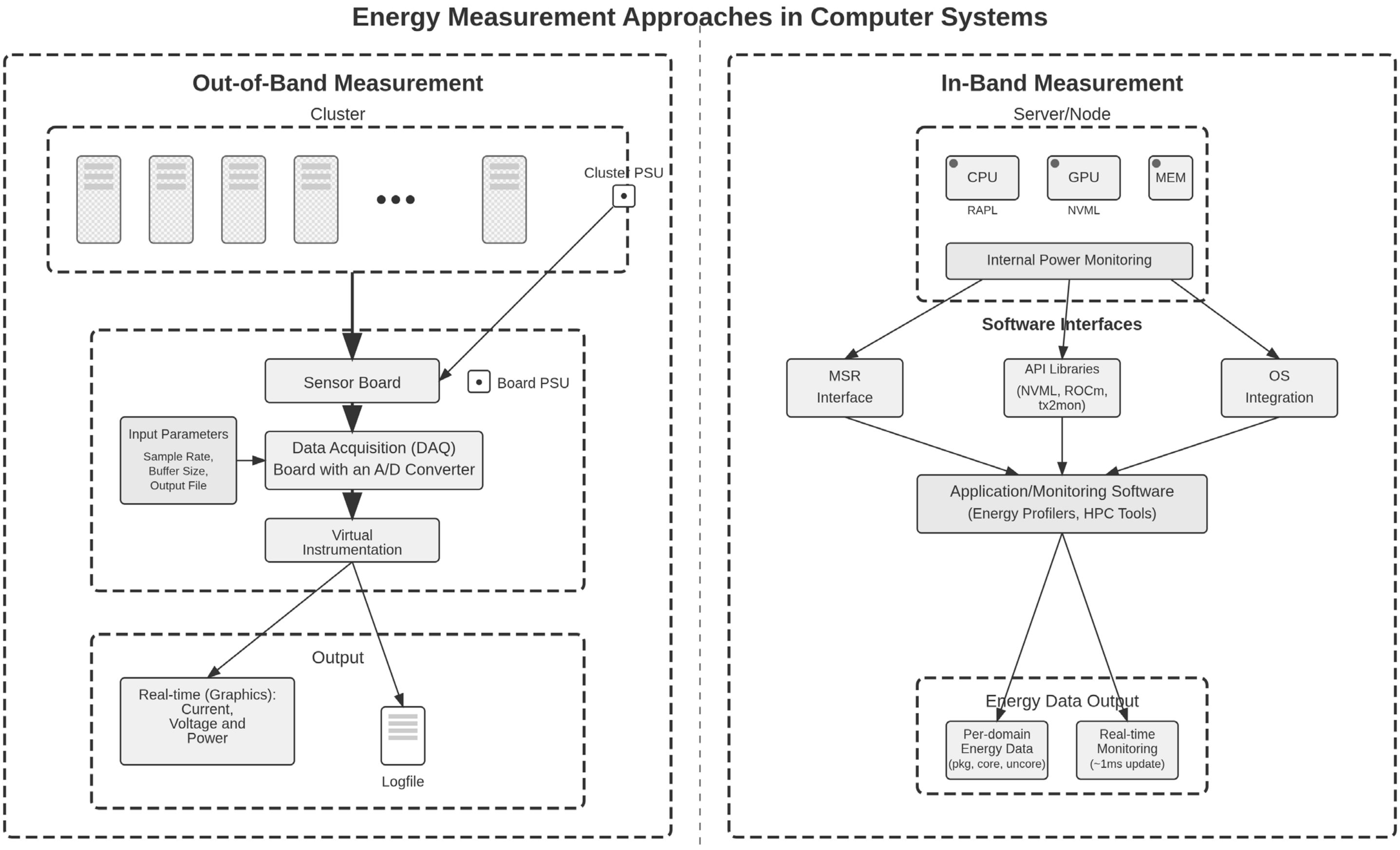


*Figure 2: The two major energy measurement paradigms*

### *Out-of-Band Measurement.*

This approach utilizes external, physically separate measurement hardware, such as digital wattmeters or data acquisition (DAQ) probes, inserted into the power delivery circuits of the system.

**Strengths:** Out-of-band measurement is the *«ground truth».* It provides direct, physical readings of current and voltage, offering the highest potential for absolute accuracy. It is also non-intrusive to the software execution, as the measurement process runs on a separate device, thus avoiding the *«observer effect».*

**Weaknesses:** Its primary weakness is the difficulty of **synchronization**. Correlating a high-frequency power trace from an external device with the specific software function that was running at that exact instant is a non-trivial instrumentation and data analysis challenge. Furthermore, it is often costly and complex to set up, and it typically offers limited spatial granularity (e.g., measuring the whole server power from the outlet, but not the individual CPU vs. GPU power without invasive hardware modifications).

### *In-Band Measurement.*

This approach relies on sensors and power models embedded directly within the silicon of the processors themselves. These internal estimates are then exposed to the software through specialized interfaces.

**Strengths:** The primary advantage is ease of access and excellent **granularity**. Interfaces like Intel's Running Average Power Limit (RAPL) [12] can provide distinct power estimates for different parts of the CPU (e.g., the 'pkg' for the whole socket, 'core' for the compute cores, and sometimes 'dram'). Similarly, the NVIDIA Management Library (NVML) [13] provides power readings for GPUs. Crucially, these measurements are easily synchronized with the software, as they can be queried from within the application code itself.

**Weaknesses:** These are often not direct measurements but sophisticated *hardware models*. Their absolute accuracy is not always guaranteed and can be subject to calibration variance. More importantly, *their coverage is often incomplete*. Many in-band tools fail to report the power consumption of critical components like the main DRAM, storage, or networking hardware, creating a significant "blind spot" in the system's total energy profile.

## 1.2. Existing Tools for Energy Profiling

A rich ecosystem of tools has been developed to provide access to these measurement interfaces.

**System-Level Tools:** The Linux kernel's standard 'perf' toolset is a powerful and versatile profiler that can access both CPU hardware performance counters and RAPL energy events ('perf stat -e power/energy-pkg/ ...'). For diagnostics, tools like 'PowerTOP' [14] are effective at identifying sources of idle power consumption.

**HPC-Centric Libraries:** In the HPC domain, libraries like PAPI (Performance Application Programming Interface) [15] and LIKWID [16] provide a portable and programmatic interface to a wide range of hardware counters, including RAPL events, making them indispensable for instrumenting scientific applications written in C or Fortran.

**AI and Python Ecosystem Tools:** The prevalence of Python in AI has led to the need for Python-friendly tools. While some early tools focused on coarse-grained CO2 estimation from TDP, more recent tools provide direct energy measurement. Our EA2P tool [17] is an example of an effort to address this specific need, providing a flexible Python API that unifies multi-vendor CPU and GPU monitoring while also tackling the DRAM measurement gap with a validated analytical model.

## 1.3. Methodological Guideline for Pillar 1

The review of the state-of-the-art leads to a clear methodological guideline for the first pillar of our framework: *Any rigorous energy optimization process must begin by establishing a comprehensive baseline using a toolchain that is capable of providing* *synchronized multi-component energy data.* An ideal approach combines the strengths of multiple techniques:

1. Use **in-band tools** like RAPL and NVML as the primary source of data due to their excellent granularity and ease of software synchronization.
2. Where possible, **validate or calibrate** these in-band readings against an out-of-band power meter to establish confidence in their absolute accuracy.
3. Acknowledge and **address any coverage gaps**. For components like DRAM where direct in-band measurement is often unavailable, an estimation based on a validated model (as described in the EA2P paper [17]) is methodologically preferable to simply ignoring a potentially significant portion of the system's energy consumption.

Only with this complete and validated baseline, engineers or researchers proceed with confidence to the next phase of the cycle: understanding the root causes of the now-visible energy consumption.

## 2. Pillar 2: Understand & Characterize – From Data to Insight

Possessing accurate energy-related data, as established in the previous pillar, is a necessary but insufficient condition for effective optimization. A time series of power consumption is merely a symptom; the goal of this second pillar is to diagnose the underlying causes. The **Understand & Characterize** phase is a process of systematic investigation aimed at building a causal link between the dynamic behavior of an application and its measured energy footprint [18]. This involves moving beyond global metrics to dissect the workload, identifying its fundamental characteristics, and quantifying how it interacts with the system's hardware and its available power management levers [19], [20].

### 2.1. Beyond Utilization: The Importance of Workload Characterization

Early, simplistic models of energy consumption often relied on high-level utilization metrics, such as CPU utilization percentage. However, this is an overly coarse abstraction. Two different programs, both utilizing 50% of a CPU, can have vastly different power draws. One might be executing a stream of simple integer instructions that are highly cache-resident, while another might be stalled, waiting for data from memory after a series of cache misses. These two states have profoundly different micro-architectural activity levels and, consequently, different energy signatures.

An insightful characterization therefore requires classifying a workload based on its primary performance bottleneck. The most fundamental distinction in high-performance computing is between:

**Compute-Bound Workloads:** These are applications whose performance is limited by the speed of the processor's arithmetic units. They typically exhibit a high Instructions Per Cycle (IPC) rate, high utilization of floating-point units (FPUs) or Tensor Cores, and a high hit rate in the caches. The core components of many AI models, such as large, dense matrix multiplications (GEMM), are prime examples of compute-bound kernels.

**Memory-Bound (or Bandwidth-Bound) Workloads:** These are applications whose performance is dominated by the overhead of data accesses between the main memory (DRAM/HBM) and the processor. They are characterized by a low IPC, a high rate of cache misses, and high utilization of the memory bus. Operations like streaming large vectors (e.g., the TRIAD benchmark) or gathering data from sparse data structures fall into this category.

Identifying which category a workload (or even a specific phase within a workload) falls into is the first crucial step in understanding its energy consumption and in selecting the appropriate optimization strategy.

## 2.2. An Overview of Benchmarking and Characterization

To perform workload characterization, the community relies on systematic benchmarking using standardized or representative kernels.

**Standardized Benchmark Suites:** For decades, suites like SPEC (Standard Performance Evaluation Corporation) [21] for CPUs or LINPACK [22] for HPC have provided a means to compare hardware performance. In the AI domain, suites like MLPerf [23] have emerged to standardize the benchmarking of training and inference performance across different hardware and software stacks.

**Micro-architectural Benchmarking:** At a lower level, targeted micro-benchmarks are used to isolate and measure specific aspects of the hardware, such as peak memory bandwidth (e.g., STREAM benchmark [24]) or the performance of specific instruction types.

By using a diverse set of kernels (from the compute-bound GEMM to the memory-bound TRIAD) and executing them under various parallel programming paradigms (SIMD, OpenMP, GPU frameworks), we were able to create a rich dataset that characterizes how these different workloads stress the underlying hardware.

## 2.3. Methodological Guideline for Pillar 2

The guideline for this pillar is to move from passive, system-wide monitoring to **active, targeted experimentation**. Once a baseline has been established (Pillar 1), the engineer must formulate hypotheses about the primary drivers of energy consumption and test them in a supervised manner. The quintessential methodology for this phase is the **trade-off analysis**.

A trade-off analysis involves systematically varying a single system parameter (a «lever») and observing its impact on two competing metrics: performance (typically execution time) and energy consumption. The results are plotted to reveal the **Energy-Performance Pareto Frontier**, which represents the set of all optimal operating points.

An excellent example of this methodology is the in-depth characterization of power management techniques from our own work. To understand the potential of Power Capping, for instance, we did not simply turn it on; we systematically executed a representative workload under a range of power caps, from very restrictive to the nominal TDP. For each point, we used EA2P to measure the resulting execution time and total energy consumed. An example of a trade-off analysis plot, generated by characterizing a single workload under varying levels of a power management technique, is shown in Figure 3. This analysis reveals that the point of maximum performance (The baseline) is often not the most energy-efficient. There is usually a «sweet spot» (Min EDP points) that minimizes the Energy-Delay Product (EDP), offering substantial energy savings for a negligible performance cost. This confirms the existence of an optimization potential.

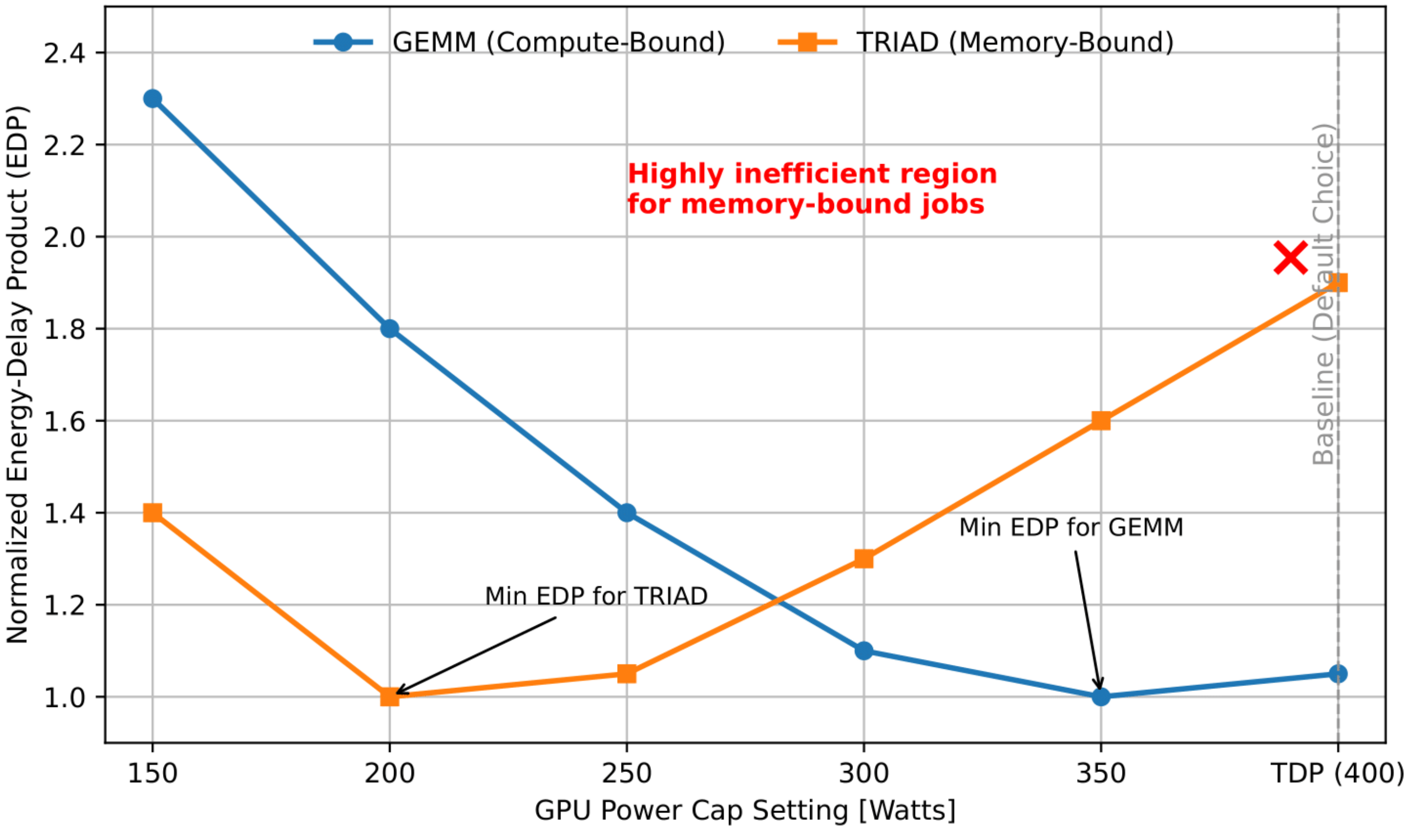


*Figure 3: Targeted characterization of kernels under Power Capping on an Nvidia A100 GPU.*

The outcome of such an analysis, as conceptually illustrated in Figure 3, provides profound insight.

1. **It Quantifies the Potential:** The plot clearly shows that there is a more efficient operating point (has a lower EDP) than the default maximum-performance point (the baseline at full TDP). This moves from a vague notion that «energy could be saved» to a quantitative statement: *«for this workload, a 15% energy reduction is achievable with a mere 3% performance degradation.»*

2. **It informs the Optimization Strategy:** The shape of the curve is itself informative. A steep curve for a compute-bound workload indicates that performance is susceptible to the active power limit. A flatter curve for a memory-bound workload shows that power can be reduced significantly before performance is substantially impacted.

This phase of understanding and characterization is the critical bridge between observation and modeling. By performing targeted experiments and quantifying these fundamental trade-offs, we validate our initial hypotheses and gather the essential insights needed to construct predictive models that can generalize this knowledge to the entire system.

## 3. Pillar 3: Modeling for Analysis

The first two pillars of our methodology equip us with the ability to measure a system's current state and figure out the root causes of its behavior. However, a purely reactive approach (running an experiment for every possible system configuration) is intractable. To move towards proactive and strategic optimization, we need the power of abstraction and generalization. The third pillar addresses this need. It is the process of encapsulating the complex, empirically-derived understanding of the system into a set of mathematical relationships that can be used to predict performance and energy consumption for configurations that have not yet been run [25].

This is where our methodology explicitly aligns with the philosophy of **predictive modeling**. In the context of AI systems engineering, modeling serves as a tool for operational prospective analysis: it allows us to look into the near future and ask crucial *«what-if»* questions. *«What will be the energy cost of training this new model architecture?», «How would the cluster's throughput change if we upgraded our GPUs?», «What is the optimal number of GPUs to allocate to this job to balance performance and energy?»* Providing reliable answers to these questions *a priori* is the key to making informed, resource-efficient, and sustainable decisions [26].

## 3.1. The Spectrum of Modeling Approaches

The state-of-the-art in energy modeling for computer systems is rich and diverse, spanning a spectrum from purely theoretical to purely empirical approaches [27]. The choice of modeling paradigm involves a fundamental trade-off between accuracy, generality, and interpretability.

*First-Principles Models.*

At one end of the spectrum, there are models based on fundamental physical or architectural principles. A classic example is the **Roofline model** [28], which provides an insightful upper bound on the performance of a kernel by comparing its arithmetic intensity (FLOPs/byte) to the hardware's peak performance and memory bandwidth. While immensely valuable for identifying high-level bottlenecks, such models are often too coarse to provide precise, actionable predictions for complex AI workloads on modern GPUs [29].

*Black-Box Empirical Models.*

At the other end, there are black-box models that rely on statistical techniques or machine learning. A system is subjected to a wide variety of workloads, and a large dataset of system metrics (e.g., CPU utilization, Hardware Performance Counters (HPCs)) and corresponding power measurements is collected [30], [31]. A model (e.g., linear regression, a support vector machine (SVM), or a neural network) is then trained to learn the mapping from system state to power consumption.

**Strengths:** For a given, stable hardware platform, these models can achieve very high predictive accuracy within the range of workloads they were trained on. They can capture complex and non-linear interactions without requiring a deep a priori understanding of the hardware.

**Weaknesses:** Their primary drawback is a lack of interpretability and generalizability. As «black boxes,» they do not provide insight into *why* a given configuration is energy-efficient. Furthermore, a model trained on one specific GPU architecture (e.g., NVIDIA A100) is unlikely to generalize well to a different one (e.g., NVIDIA H100) without being completely retrained.

*White-Box Analytical Models.*

Positioned between these two extremes, white-box analytical models seek to combine the rigor of first principles with the accuracy of empirical calibration. These models are built upon a structural decomposition of the hardware and of the software workload.

Their equations and parameters are designed to have a physical or architectural meaning [32]. The performance of a complex workload is predicted by composing the predicted performance of its elementary operations, modulated by an understanding of the system's architectural constraints (e.g., saturation of execution units, cache behavior). Our sophisticated analytical framework ADEPT [33], is a prime example of this approach. The general workflow of our analytical framework is shown in Figure 4.

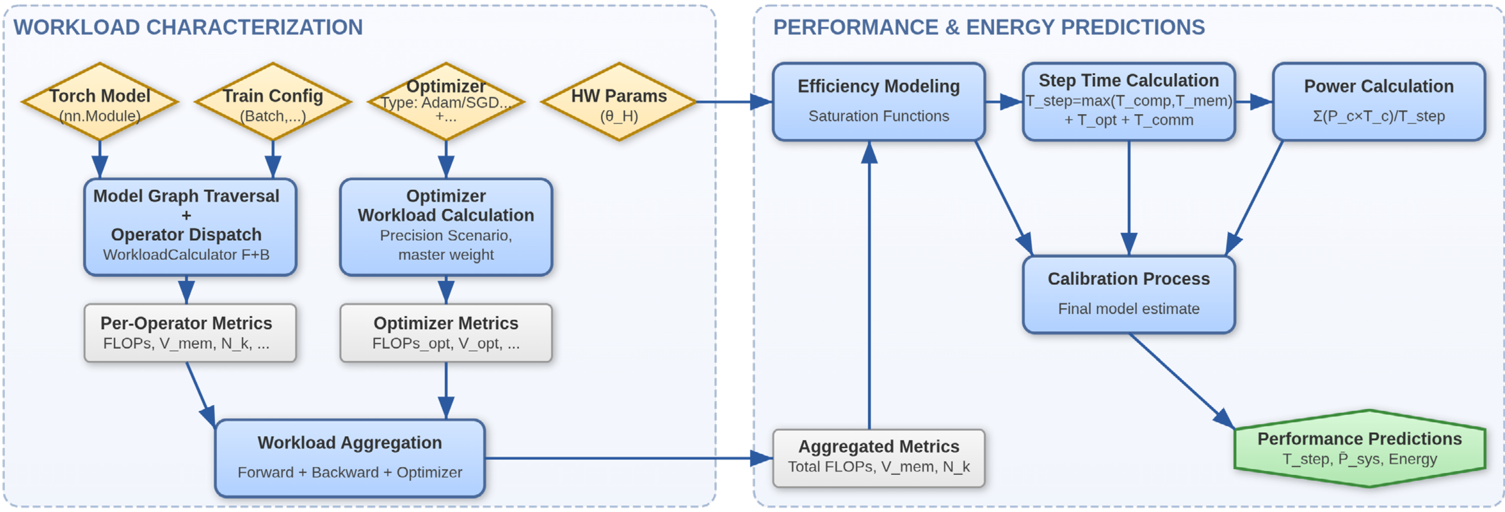


*Figure 4: Conceptual flow of the ADEPT analytical prediction framework*

## 3.2. Methodological Guideline for Pillar 3

The guideline for this pillar is to invest in interpretable, white-box models wherever the problem's complexity demands deep insight and robust generalization. While black-box models can be sufficient for simple power monitoring on a fixed system, they are inadequate for the strategic and prospective goals of sustainable engineering. A true prospective model must not only provide a prediction, but also an *explanation*.

The benefits of the white-box approach are multifold:

- **Interpretability and Insight:** When our analytical model [33] predicts that an ALBERT model is less energy-efficient than a DistilBERT model of a similar size [34], it can also spot the cause: a higher-than-average penalty on memory access time due to its specific parameter-sharing architecture. This insight is actionable for both software developers and hardware designers.
- **Extensibility and Generalizability:** Because the model's parameters correspond to physical hardware characteristics (e.g., peak performance of Tensor Cores, size of the L2 cache, memory bandwidth), it is likely to generalize to new hardware. Porting the model to a future GPU architecture would involve recalibrating a known set of physical parameters rather than a complete, blind retraining.

- **A Foundation for True Co-Design:** White-box models are the essential enabler for hardware-software co-design. An AI researcher can use the model to predict the energy cost of a novel neural network layer on existing hardware. Conversely, a GPU architect can use the model to predict the performance impact of doubling the L2 cache on a representative suite of AI workloads. This predictive capability allows for a vast design space to be explored in simulation, a core tenet of prospective modeling.

Implementing this pillar requires a significant upfront investment in understanding and characterizing the system (as per Pillar 2). However, the payoff is a powerful prospective tool that moves beyond simple forecasting to enable genuine understanding and informed, strategic decision-making. It is this modeling capability that unlocks the final phase of the cycle: systematic optimization.

## 4. Pillar 4: Optimize – From Prediction to Action

The first three pillars of our methodology—Measure, Understand, and Model—provide us with a deep, predictive knowledge of our system. This final pillar is where this knowledge is translated into concrete action. Optimization, in this context, is the process of designing and deploying control strategies that actively steer the system towards a more efficient operating point, using the predictive models as their guide. This is the stage where potential energy savings, identified through characterization and quantified by modeling, are realized. This section reviews the state-of-the-art in system-level optimization and demonstrates how our methodological approach culminates in the design of sophisticated, multi-objective resource management strategies.

### 4.1. The Landscape of System-Level Optimization Techniques

Optimization strategies can be applied at various levels of the system stack, from low-level hardware control to high-level cluster management.

*Runtime Power Management.*

The most direct form of optimization involves the dynamic control of hardware power management levers. As characterized in our benchmarks Section, hardware mechanisms like **DVFS** (Dynamic Voltage and Frequency Scaling) and **Power Capping** offer powerful ways to trade performance for energy. The state-of-the-art in this area involves moving beyond generic OS-level policies (e.g., standard CPU governors) towards more workload-aware control [19]. Advanced techniques involve runtime phase detection, where a program's execution is monitored (often using Hardware Performance Counters) to identify whether it is in a compute-bound or memory-bound phase, and the DVFS settings are adjusted accordingly to save energy with minimal performance impact [35].

*Cluster-Level Resource Management.*

While runtime management optimizes a single node, the greatest potential for energy savings in a large-scale AI facility lies in the intelligent management of the entire cluster. This is the domain of the **job scheduler**. A modern scheduler makes two critical decisions:

1. **Placement:** Which of the potentially heterogeneous resources (e.g., which type of GPU) should a job be allocated to?

2. **Sequencing and Configuration:** When should a job run, and with what configuration (e.g., how many GPUs, what batch size)?

As our state-of-the-art review has shown, traditional schedulers for AI clusters, such as Pollux [36] or Themis [37], were designed to optimize for single objectives like performance or fairness, largely ignoring the energy dimension.

### 4.2. Methodological Guideline for Pillar 4: Model-Driven, Multi-Objective Optimization

The guideline for the optimization pillar is to leverage the predictive models from Pillar 3 to build control strategies that are explicitly **multi-objective** [38] and **proactive** [39]. Instead of making greedy, short-sighted decisions, an advanced optimizer uses its prospective model to evaluate the long-term consequences of its decisions on the entire system's efficiency.

The first example of this methodology is the design of an **energy-aware job scheduler**. Such a scheduler embodies the full cycle:

- It uses a **model** to predict the time ($T$) and power ($P$) for every possible placement configuration of every pending job.
- It uses this predictive insight to make an "optimal decision" based not on a single metric, but on a composite heuristic that captures the desired multi-objective trade-off.

*The Energy-Delay Product (EDP) as a Robust Heuristic.*

Our research has identified the **Energy-Delay Product** [40] as a particularly effective metric for efficient energy-aware scheduling heuristics. As we demonstrated, selecting the job configuration that minimizes EDP acts as a natural regularizer against greedy behavior. It inherently balances the desire for fast execution (low $T$) with the need for power efficiency (low $P$), often leading to configurations that use fewer, more appropriate resources.

*Designing a Suite of Multi-Objective Schedulers.*

By systematically injecting the EDP-based heuristic into the core logic of state-of-the-art schedulers, we developed a suite of new policies, each targeting a different point in the complex trade-off space:

- **Zeus**: Co-optimizes for energy and performance. Zeus is our energy-aware counterpart to Pollux, as it replaces the "**minimum Time**" objective with a "**minimum EDP**", directly optimizing for the best balance of energy and performance. The algorithm retains Pollux's exhaustive search over malleable configurations but uses EDP as its selection criterion.
- **Hades**: Hades applies the same transformation to Themis, demonstrating the generality of our approach for fairness. Its scoring function firstly finds the most energy-efficient placement via EDP and then modulates this score by the user's fairness share. Its final score is proportional to *"EDP / Fairness"*, thus co-optimizing for fairness and energy.
- **AuraChronos**: Co-optimizes for energy and Quality of Service, using preemption as a key mechanism. It is our most sophisticated policy, designed for mixed-criticality environments. Its scheduling logic is two-fold. First, it implements a proactive preemption loop that constantly checks if any urgent job is in danger of missing its deadline, freeing resources by preempting non-urgent tasks as necessary. Second, its scoring function uses a tiered-priority system that ranks urgent jobs by deadline and backfills non-urgent jobs using Zeus's energy-efficient EDP heuristic.
- **Charon**: Optimizes for performance under a strict energy/power budget. Charon explores a different paradigm: constraint-based scheduling. It uses Zeus's EDP heuristic for prioritization but adds a hard constraint to its packing logic. It will not dispatch a job and, if doing so, it would violate a pre-defined cluster-wide power budget, a critical capability for power-provisioned datacenters.

## 4.3. Validation through High-Fidelity Simulation

Evaluating such complex control strategies directly on a production cluster is too difficult. This is where the modeling pillar provides its final, crucial contribution: powering a high-fidelity simulator. Our **EAS-Sim framework** (https://github.com/HPC-CRI/EAS-Sim), using the analytical model as its predictive oracle, provides the necessary virtual testbed to safely and rapidly validate these optimization policies at scale. The general view of the framework is shown in Figure 5.

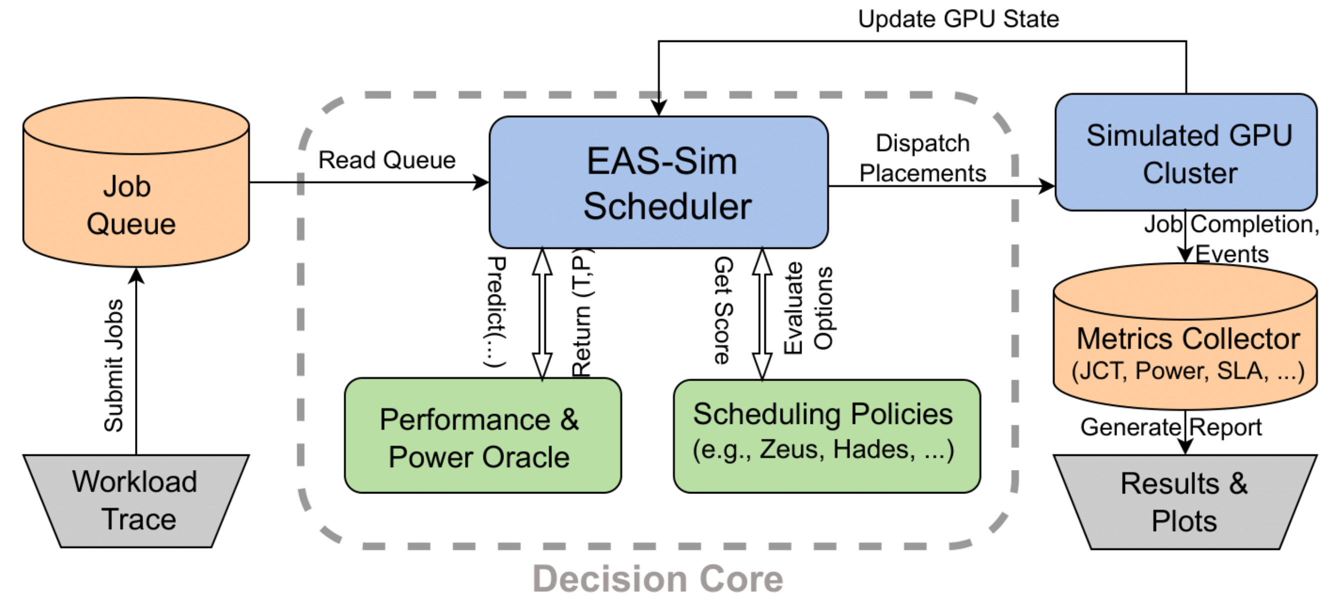


*Figure 6: The EAS-Sim workflow, where our analytical performance & power oracle (ADEPT) feeds the scheduling policies for a large-scale comparative simulation.*

Our experimental results provide a clear validation of this entire methodological pipeline. The simulations confirmed that the **Zeus** policy, built upon our model-driven, EDP-based approach, was able to reduce total cluster energy consumption by approximately 10.5% with no loss in overall throughput compared to a purely performance-oriented policy.

As the decision tree in Figure 6 illustrates, if the objectives had been different (e.g., guaranteeing strict SLAs), the simulation would have pointed us toward another policy (AuraChronos), but in our context, Zeus is the optimal solution. The technical implementation of this strategy consists of reconfiguring the real cluster's scheduler. Production schedulers like Slurm or Kubernetes are highly configurable systems that often allow for the implementation of custom priority or scoring plugins.

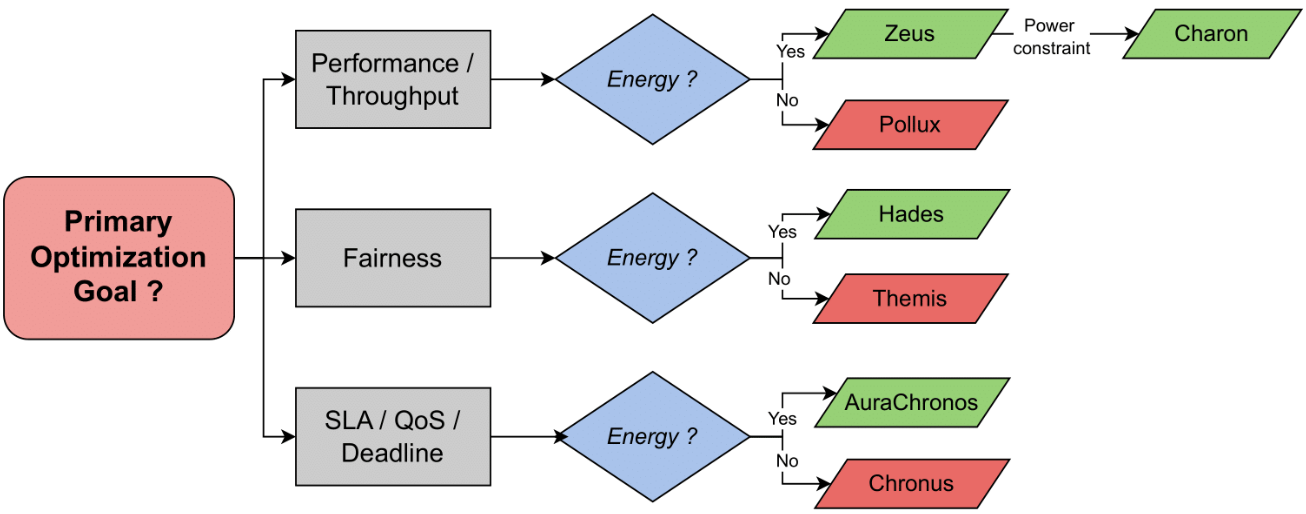


*Figure 6: The decision tree for selecting the optimal scheduling strategy*

This final pillar thus demonstrates the power of a fully integrated methodology. By building upon a foundation of measurement and understanding, and leveraging the foresight provided by an interpretable predictive model, we can design and validate intelligent control strategies that successfully navigate within the complex, multi-objective landscape of modern AI systems, delivering tangible and significant gains in energy efficiency without compromising the ultimate goal of high performance. This closes the loop of the virtuous cycle, providing a robust and repeatable process for engineering a more sustainable AI infrastructure.

# III. Discussion: Towards a Culture of Digital Sobriety in AI

The systematic methodology presented in this paper (built upon the pillars of —Measure, Understand, Model, and Optimize—) offers more than a set of technical procedures. It provides the foundation for a much-needed cultural shift in the field of Artificial Intelligence: a move from a paradigm of «*performance at all costs*» to a culture of **digital sobriety** and sustainable engineering. The implications of adopting such a framework extend beyond individual systems to the broader AI ecosystem.

## 1. Redefining Performance in the Era of Sustainable AI

For decades, the primary metric of progress in computer capability has been raw speed. Our work argues that this standpoint is no longer sufficient when it comes to evaluating a given computing system. The experimental validation of our scheduling policies revealed an important insight: the strategy that purely optimized for the shortest job completion time (Pollux) did not yield the highest system throughput and was demonstrably less stable under high load than the policy that co-optimized for energy (Zeus).

This suggests that the **Energy-Delay Product (EDP)** is not merely an "eco-metric» but rather a robust heuristic for overall system efficiency. By penalizing energy-profligate configurations, an EDP-driven strategy acts as a natural regulator, preventing a «greedy» allocation of resources that might lead to system fragmentation and instability. This has the following profound implication: energy efficiency should no longer be seen as a tax on performance, but rather as a critical component of achieving robust, high-throughput, and scalable performance. The community should move towards embracing composite metrics like EDP as a standard for evaluating system performance.

## 2. The Role of Predictive Modeling in Responsible Innovation

A central tenet of our methodology is the use of predictive modeling for analysis. This stands in contrast to the current, largely reactive, approach to AI development, where the true energy cost of a new model is only known *after* vast computational resources have been expended on its training. This is a fundamentally unsustainable model for innovation.

The energy-aware analytical framework we have developed represents a step towards a more responsible paradigm. By providing tools to estimate the cost and performance of an AI model *a priori*, we enable a new form of eco-design. This capability is transformative for multiple stakeholders:

- **AI Researchers** can get an «energy sticker price» for a new model architecture on their drawing board, allowing them to iterate on more frugal designs before committing to costly training runs.
- **Infrastructure Planners** [41] can conduct quantitative «what-if» analyses for future hardware choices, ensuring that their investments are optimally aligned with their projected workloads and sustainability goals.
- **Policymakers and Standards Bodies** can use such validated modeling frameworks as the basis for creating standardized «energy labels» for AI models, bringing a much-needed level of transparency and accountability to the field.

## 3. Systematizing the Practice of Energy-Awareness

The key contribution of this work relies on advocating for a systematic approach. Energy efficiency will no longer be a second-order concern and will motivate standard practice only when it is embedded in the daily workflows of engineers and researchers. The four-pillar cycle is a template for this integration. We envision a future where:

- Energy profiling (Pillar 1) is a standard, automated step in Continuous Integration / Continuous Deployment (CI/CD) pipelines [42], considering energy regressions with the same urgency as functional bugs.
- Benchmarking tools (Pillar 2) are not only used for hero-run performance numbers but for systematically mapping out the energy-performance trade-offs of critical workloads.
- Predictive models (Pillar 3) are an integrated service within MLOps platforms [43], providing cost estimates before a user can launch a large-scale training job.
- Cluster schedulers (Pillar 4) are multi-objective by default, allowing administrators to dynamically balance the institution's performance needs with its energy budget and sustainability commitments [44].

Achieving the previously described vision requires a collective effort from the community to build, share, and standardize the tools and practices that make this methodology accessible to all.

# IV. Conclusion

This working paper has addressed the critical and growing challenge of energy consumption in large-scale Artificial Intelligence systems. Dealing with a problem with noticeable technical complexity and significant societal implications, we have argued that the path forward requires moving beyond ad-hoc optimizations towards a formal "*engineering discipline for sustainable AI*".

We have proposed a comprehensive and systematic methodology, structured as a virtuous cycle of four pillars: **Measure, Understand, Model, and Optimize**. This framework provides a coherent roadmap for any organization seeking to rigorously analyze and improve its energy footprint. By synthesizing an extensive review of the state-of-the-art for each pillar, this work serves as both a survey of current techniques and a practical guide for their integrated application.

We demonstrated how this methodology transforms theoretical goals into concrete engineering actions. It begins by making energy consumption tangible through fine-grained measurement. It builds a deep, causal understanding through targeted characterization and trade-off analysis. It leverages this understanding to build powerful, open analytical models that enable proactive, prospective analysis. Finally, it uses these models to drive intelligent, multi-objective optimization strategies, such as the energy-aware schedulers we have designed. Our work has shown that such a principled approach can yield significant energy savings (in our case, over 10%) without sacrificing system performance.

The tools and strategies developed in our research (EA2P for measurement, the analytical framework for modeling, and EAS-Sim with its suite of schedulers) are concrete implementations of this philosophy. They represent a tangible contribution towards a future where energy efficiency is no longer an optional extra, but a core component of how AI systems are designed, deployed, and managed. Ultimately, the long-term success and societal benefit of Artificial Intelligence will depend on our ability to align its incredible power with the principles of sustainability. We believe the engineering discipline outlined in this paper offers a clear and actionable path towards achieving that essential goal.

# V. Acknowledgements

Claude TADONKI, Mines Paris, PSL University, Centre de Recherche en Informatique (CRI), Fontainebleau, France, Petr DOKLADAL, Mines Paris, PSL University, Center for Mathematical Morphology (CMM), Fontainebleau, France, and Youssef MESRI, Mines Paris, PSL University, Centre for Materials Forming (CEMEF), Sophia Antipolis, France, are gratefully acknowledged for their support of writing as doctoral thesis supervisors.

# References

[1] J. McDonald, B. Li, N. Frey, D. Tiwari, V. Gadepally, and S. Samsi, "Great power, great responsibility: Recommendations for reducing energy for training language models," in Findings of the Association for Computational Linguistics: NAACL 2022. Association for Computational Linguistics, 2022. [Online]. Available: http://dx.doi.org/10.18653/v1/2022.findings-naacl.151

[2] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 09, pp. 13 693–13 696, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/7123

[3] S. Luccioni, Y. Jernite, and E. Strubell, "Power hungry processing: Watts driving the cost of ai deployment?" in The 2024 ACM Conference on Fairness, Accountability, and Transparency, ser. FAccT '24. ACM, Jun. 2024, p. 85–99. [Online]. Available: http://dx.doi.org/10.1145/3630106.3658542

[4] C. Freitag, M. Berners-Lee, K. Widdicks, B. Knowles, G. S. Blair, and A. Friday, "The real climate and transformative impact of ict: A critique of estimates, trends, and regulations," Patterns, vol. 2, no. 9, p. 100340, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666389921001884

[5] C.-J. Wu, R. Raghavendra, U. Gupta, B. Acun, N. Ardalani, K. Maeng, G. Chang, F. A. Behram, J. Huang, C. Bai, M. K. Gschwind, A. Gupta, M. Ott, A. Melnikov, S. Candido, D. Brooks, G. Chauhan, B. Lee, H.-H. S. Lee, B. Akyildiz, M. Balandat, J. Spisak, R. K. Jain, M. G. Rabbat, and K. M. Hazelwood, "Sustainable ai: Environmental implications, challenges and opportunities," ArXiv, vol. abs/2111.00364, 2021.

[6] J.-H. Syu, J. C.-W. Lin, G. Srivastava, and K. Yu, "A comprehensive survey on artificial intelligence empowered edge computing on consumer electronics," IEEE Transactions on Consumer Electronics, vol. 69, no. 4, pp. 1023–1034, 2023.

[7] J. You, J.-W. Chung, and M. Chowdhury, "Zeus: Understanding and optimizing GPU energy consumption of DNN training," in 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23). Boston, MA: USENIX Association, Apr. 2023, pp. 119–139. [Online]. Available: https://www.usenix.org/conference/nsdi23/presentation/you

[8] S. Bhalachandra, B. Austin, and N. J. Wright, “Understanding power variation and its implications on performance optimization on the cori supercomputer,” in 2021 International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS), 2021, pp. 51–62.

[9] R. Desislavov, F. Martínez-Plumed, and J. Hernández-Orallo, “Trends in ai inference energy consumption: Beyond the performance-vs-parameter laws of deep learning,” Sustainable Computing: Informatics and Systems, vol. 38, p. 100857, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210537923000124

[10] S. Nasrin, A. Shylendra, N. Darabi, T. Tulabandhula, W. Gomes, A. Chakrabarty, and A. R. Trivedi, “Enos: Energy-aware network operator search in deep neural networks,” IEEE Access, vol. 10, pp. 81 447–81 457, 2022.

[11] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, “A survey of quantization methods for efficient neural network inference,” ArXiv, vol. abs/2103.13630, 2021.

[12] K. N. Khan, M. Hirki, T. Niemi, J. K. Nurminen, and Z. Ou, “Rapl in action,” ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS), vol. 3, pp. 1 – 26, 2018.

[13] NVIDIA, “nvidia-smi - nvidia system management interface program,” 2016, accessed: 2024-08-29. [Online]. Available: https://developer.download.nvidia.com/compute/DCGM/docs/nvidia-smi-367.38.pdf

[14] Intel, “Powertop,” 2020, accessed: 2025-06-25. [Online]. Available: https://github.com/fenrus75/powertop

[15] H. McCraw, J. Ralph, A. Danalis, and J. Dongarra, “Power monitoring with papi for extreme scale architectures and dataflow-based programming models,” in 2014 IEEE International Conference on Cluster Computing (CLUSTER), 2014, pp. 385–391.

[16] J. Treibig, G. Hager, and G. Wellein, “Likwid: A lightweight performance-oriented tool suite for x86 multicore environments,” in 2010 39th International Conference on Parallel Processing Workshops, 2010, pp. 207–216.

[17] R. N. Tchakoute, C. Tadonki, P. Dokladal, and Y. Mesri, “A flexible operational framework for energy profiling of programs,” in 2024 International Symposium on Computer Architecture and High Performance Computing Workshops (SBAC-PADW), 2024, pp. 12–22.

[18] Y. Wang, Q. Wang, S. Shi, X. He, Z. Tang, K. Zhao, and X. Chu, “Benchmarking the performance and energy efficiency of ai accelerators for ai training,” in 2020 20th IEEE/

ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), 2020, pp. 744–751.

[19] R. N. Tchakoute, C. Tadonki, P. Dokladal, and Y. Mesri, "Benchmark-based study of cpu/gpu power-related features through jax and tensorflow," 2025. [Online]. Available: https://arxiv.org/abs/2505.03398

[20] S. Georgiou, M. Kechagia, T. Sharma, F. Sarro, and Y. Zou, "Green ai: Do deep learning frameworks have different costs?" in 2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE), 2022, pp. 1082–1094.

[21] J. Bucek, K.-D. Lange, and J. v. Kistowski, "Spec cpu2017: Next-generation compute benchmark," in Companion of the 2018 ACM/SPEC International Conference on Performance Engineering, ser. ICPE '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 41–42. [Online]. Available: https://doi.org/10.1145/3185768.3185771

[22] J. Dongarra, P. Luszczek, and A. Petitet, "The linpack benchmark: past, present and future," Concurrency and Computation: Practice and Experience, vol. 15, pp. 803–820, 08 2003.

[23] A. Tschand, A. T. R. Rajan, S. Idgunji, A. Ghosh, J. Holleman, C. Kiraly, P. Ambalkar, R. Borkar, R. Chukka, T. Cockrell, O. Curtis, G. Fursin, M. Hodak, H. Kassa, A. Lokhmotov, D. Miskovic, Y. Pan, M. P. Manmathan, L. Raymond, T. S. John, A. Suresh, R. Taubitz, S. Zhan, S. Wasson, D. Kanter, and V. J. Reddi, "Mlperf power: Benchmarking the energy efficiency of machine learning systems from microwatts to megawatts for sustainable ai," 2025. [Online]. Available: https://arxiv.org/abs/2410.12032

[24] J. D. McCalpin, "Memory bandwidth and machine balance in current high performance computers," IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter, pp. 19–25, Dec. 1995.

[25] Y. Xin-She, "Chapter 28 - Mathematical Modeling," Engineering Mathematics with Examples and Applications, Academic Press, 2017, Pages 325-340, ISBN 9780128097304, [Online]. Available: https://doi.org/10.1016/B978-0-12-809730-4.00037-9

[26] J. C. McCullough, Y. Agarwal, J. Chandrashekar, S. Kuppuswamy, A. C. Snoeren, and R. K. Gupta, "Evaluating the effectiveness of model-based power characterization," in Proceedings of the 2011 USENIX Conference on USENIX Annual Technical Conference, ser. USENIXATC'11. USA: USENIX Association, 2011, p. 12.

[27] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 732–794, 2016.

[28] S. Williams, A. Waterman, and D. Patterson, "Roofline: an insightful visual performance model for multicore architectures," Commun. ACM, vol. 52, no. 4, p. 65–76, Apr. 2009. [Online]. Available: https://doi.org/10.1145/1498765.1498785

[29] N. Ding and S. Williams, "An instruction roofline model for gpus," in 2019 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS), 2019, pp. 7–18.

[30] R. Bertran, M. Gonzalez, X. Martorell, N. Navarro, and E. Ayguade, "Decomposable and responsive power models for multicore processors using performance counters," in Proceedings of the 24th ACM International Conference on Supercomputing, ser. ICS '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 147–158. [Online]. Available: https://doi.org/10.1145/1810085.1810108

[31] L. F. Cupertino, G. Costa, and J.-M. Pierson, "Towards a generic power estimator," Comput. Sci., vol. 30, no. 2, p. 145–153, may 2015. [Online]. Available: https://doi.org/10.1007/s00450-014-0264-x

[32] M. F. Dolz, J. Kunkel, K. Chasapis, and S. Catalán, "An analytical methodology to derive power models based on hardware and software metrics," Comput. Sci., vol. 31, no. 4, p. 165–174, nov 2016. [Online]. Available: https://doi.org/10.1007/s00450-015-0298-8

[33] HPC-CRI, "Adept (analytical deep-learning energy, performance and time-estimator)," 2025, accessed: 2025-08-14. [Online]. Available: https://github.com/HPC-CRI/ADEPT

[34] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," CoRR, vol. abs/1909.11942, 2019. [Online]. Available: http://arxiv.org/abs/1909.11942

[35] Y. Oleynik, M. Gerndt, J. Schuchart, P. G. Kjeldsberg, and W. E. Nagel, "Run-time exploitation of application dynamism for energy-efficient exascale computing (readex)," IEEE 18th International Conference on Computational Science and Engineering, pp. 347–350, 2015.

[36] A. Qiao, S. K. Choe, S. J. Subramanya, W. Neiswanger, Q. Ho, H. Zhang, G. R. Ganger, and E. P. Xing, "Pollux: Co-adaptive cluster scheduling for goodput-optimized deep learning," in 15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21). USENIX Association, Jul. 2021, pp. 1–18.

[37] K. Mahajan, A. Balasubramanian, A. Singhvi, S. Venkataraman, A. Akella, A. Phanishayee, and S. Chawla, "Themis: Fair and efficient GPU cluster scheduling," in 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20), Santa Clara, CA, 2020, pp. 289–304.

[38] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, and A. Y. Zomaya, "A multi-objective optimization scheme for job scheduling in sustainable cloud data centers," IEEE Transactions on Cloud Computing, vol. 10, no. 1, pp. 172–186, 2022.

[39] H. Shen and Z. Zhang, "Deep learning training job scheduling for proactive straggler reduction," in 2025 IEEE 25th International Symposium on Cluster, Cloud and Internet Computing (CCGrid), 2025, pp. 1–12.

[40] M. Horowitz, T. Indermaur, and R. Gonzalez, "Low-power digital design," in Proc. IEEE Symp. on Low Power Electronics, 1994.

[41] M. Gómez, D. Perales, and E. J. Torres, "An energy-aware design and reporting tool for on-demand service infrastructures," in 2009 10th IEEE/ACM International Conference on Grid Computing, 2009, pp. 209–216.

[42] A. Kruglov, G. Succi, and X. Vasuez, "Incorporating energy efficiency measurement into ci cd pipeline," in Proceedings of the 2021 European Symposium on Software Engineering, ser. ESSE '21. New York, NY, USA: Association for Computing Machinery, 2022, p. 14–20. [Online]. Available: https://doi.org/10.1145/3501774.3501777

[43] S. J. Warnett, E. Ntentos, and U. Zdun, "A model-driven, metrics-based approach to assessing support for quality aspects in mlops system architectures," Journal of Systems and Software, vol. 220, p. 112257, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121224003017

[44] F. Qiao, Y. Fang, and A. Cidon, "Energy-aware process scheduling in linux," SIGENERGY Energy Inform. Rev., vol. 4, no. 5, p. 91–97, Apr. 2025. [Online]. Available: https://doi.org/10.1145/3727200.3727214

MINES PARIS | PSL | TTI.5° The Transition Institute

Ce travail de recherche a été financé par The Transition Institute 1.5 (TTI.5).
TTI.5 est un institut de recherche et d'enseignement de
Mines Paris - PSL dédié au design de la transition bas-carbone.

*This research is funded by The Transition Institute 1.5 (TTI.5).
TTI.5 is a research and teaching institute
driven by Mines Paris - PSL and dedicated to the design of the low-carbon transition.*

**CONTACT**

the-transition-institute.minesparis.psl.eu

tti.5@minesparis.psl.eu





WORKING PAPER #5