Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Energy-Efficient GPU Resource Management for Generative AI on HPC Systems: A Roadmap Towards Cross-Layer Co-Design Approaches

Atha Shamas

Government College Women University Sialkot

M. Usman Ashraf

usman.ashraf@gcwus.edu.pk

Government College Women University Sialkot

Research Article

Keywords:

Posted Date: November 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9666060/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Energy-Efficient GPU Resource Management for Generative AI on HPC Systems: A Roadmap Towards Cross-Layer Co-Design Approaches

Atha Shamas[1] and M. Usman Ashraf[1*]

[1]Department of Computer Science, Government College Women University Sialkot, Sialkot, Pakistan.

*Corresponding author(s). E-mail(s): usman.ashraf@gcwus.edu.pk;
Contributing authors: athashamas.ms@gmail.com;

Abstract

Generative AI (GenAI) workloads are emerging as a major source of resource demand in high-performance computing (HPC) systems, creating new challenges for energy-efficient GPU management because of their extreme scale, memory and communication intensity, and highly dynamic training and inference behavior. Existing approaches remain fragmented across individual system layers, including hardware power control, runtime adaptation, and cluster scheduling, limiting their ability to deliver system-wide energy efficiency for large-scale GenAI deployment. This paper presents a structured review of 46 representative studies on energy-aware GPU resource management for GenAI on HPC platforms. The literature is synthesized through a cross-layer taxonomy spanning system layer, workload phase, decision type, evaluation scale, and energy-efficiency technique. The analysis shows that prior work has advanced telemetry-driven measurement, energy-performance prediction, scheduling and allocation, runtime power control, and architectural specialization, with several studies reporting substantial energy savings under specific workload and system conditions. At the same time, persistent gaps remain, including limited cross-layer coordination, weak support for token-level and phase-aware workload dynamics, and insufficient integration of carbon-aware objectives into resource management. To address these limitations, the paper proposes GenAI-XEnergy, a conceptual framework that unifies telemetry, prediction, orchestration, runtime control, and heterogeneous accelerator support within a closed-loop, cross-layer architecture. The review contributes a focused taxonomy, a comparative synthesis

of existing techniques, an explicit analysis of open challenges, and a practical foundation for future research on sustainable, energy-proportional GenAI execution in HPC environments.

Keywords: Generative AI, High-performance computing, GPU scheduling, Energy efficiency, Resource management, Cross-layer co-design.

1 Introduction

Generative AI (GenAI) workloads have imposed significant energy demands in HPC systems. GPT-4 training is estimated to consume approximately 9,450 MWh, while daily inference energy is more than 500 MWh ($\approx$ 2.5 Wh per prompt) [15]. Computational and energy requirements have increased proportionally with the rapid increase in GPT-3 (175 B parameters) to GPT-4 (1.76 T parameters) [15]. Recent studies have documented that training and deployment of large Generative AI models have unprecedented computational and energy demands on HPC systems. Their large parameter sizes and memory requirements mean that foundation models need to be trained on thousands of GPUs, which can consume whole systems over long durations and pose serious resource allocation and energy consumption problems [7]. The overall AI resource consumption is dominated by inference workloads, which cause sustained energy consumption in production settings [5], [16]. Although accelerator architecture can be enhanced to achieve better performance, it does not remove the inefficiencies that are caused by the lack of dynamic scheduling, workload awareness, and optimal use of resources. These trends highlight the need for energy-efficient GPU resource-management strategies tailored to large-scale Generative AI workloads.

Energy efficiency has been studied across multiple layers of the HPC stack, including hardware-level power management (DVFS [2], power capping [17]), scheduling policies (STS [3], CE-DLA [8], InEPS [20]), OS-level runtime optimization (LithOS [10]), data-center- and macro-level optimization [6], [35], and carbon-aware resource allocation [4], [11], [13], [16]. However, these solutions typically focus on a single layer in isolation, which limits their effectiveness for large-scale Generative AI workloads. The defining characteristics of Generative AI, including extreme model scale, high memory demand, and dynamic inference behavior, further challenge layer-specific approaches, because static or single-layer methods cannot adequately capture workload variability and system complexity [5], [15], [16].

Although interest in energy-efficient GPU management for AI has increased, existing methods remain fragmented and largely performance-oriented, which limits their applicability to Generative AI workloads. Most approaches optimize individual system layers, such as hardware, runtime, or scheduling, without fully accounting for the dynamic, memory-intensive, and inference-heavy characteristics of modern GenAI workloads.As a result, GPUs are often underutilized, and energy efficiency remains difficult to achieve at scale. Accordingly, this review analyzes energy-aware GPU resource-management techniques for Generative AI on HPC systems, classifies them by system layer, workload phase, decision type, and energy-efficiency technique, and

identifies gaps in measurement integration and system-level coordination that hinder cross-layer optimization and adaptive energy management.

Our review is dedicated to energy-efficient strategies of managing the resources of the GPU, with a focus on scheduling and cluster-level resource allocation decisions [3], [8], [20]. Architectural innovations are discussed as enablers, providing hardware capabilities such as memory hierarchies and processing-in-memory that allow resource managers to optimize execution dynamically [9], [14]. OS-level runtime mechanisms are not treated as standalone optimization targets but are examined where they interact with or enable scheduler-level energy-aware decisions. This review intentionally excludes full architectural redesign, concentrating instead on approaches that integrate scheduling intelligence with existing and emerging hardware capabilities to reduce energy consumption and operational costs.

The contributions of this review are as follows:

- A GenAI-centric review of energy-aware GPU resource management for HPC systems, synthesizing prior work across energy measurement, energy-performance prediction, scheduling, runtime control, and architecture-aware execution for both training and inference workloads.
- A cross-layer taxonomy and comparative analysis of 46 representative studies, organized by system layer, workload phase, decision type, evaluation scale, and energy-efficiency technique, enabling consistent comparison across otherwise fragmented lines of work.
- A critical analysis of existing approaches across hardware, runtime, scheduler, and cluster levels, showing persistent fragmentation, limited coordination, and insufficient support for token-level, phase-aware, and carbon-aware resource management.
- A focused synthesis of architectural enablers, including advanced memory hierarchies, processing-in-memory, wafer-scale accelerators, and software-hardware co-design, with emphasis on how they expose scheduler-visible control opportunities for energy-proportional execution.
- A literature-grounded conceptual framework, GenAI-XEnergy, together with open challenges and research directions for telemetry-driven, closed-loop, and sustainable energy management of large-scale GenAI workloads on future HPC platforms.

The remainder of this review is organized as follows. Section II describes the review methodology. Section III outlines key characteristics of Generative AI workloads and their implications for energy efficiency. Section IV surveys layer-specific energy optimization techniques, focusing on measurement, prediction, and scheduling. Section V examines architectural enablers for scheduler- and cluster-level optimization. Section VI analyzes cross-layer coordination challenges and gaps, Section VII outlines future research directions, and Section VIII concludes the review.

2 Review Methodology and Taxonomy

2.1 Scope and Objectives

This review examines energy-aware GPU resource-management techniques for Generative AI workloads on HPC systems, with particular emphasis on the system

mechanisms that influence cluster-scale efficiency rather than on model-architecture design alone. The scope includes training and inference workloads, but gives special attention to large-scale and inference-intensive deployment settings in which GPU utilization, memory pressure, communication overhead, and scheduling decisions strongly affect energy consumption. The review considers methods spanning measurement, energy-performance prediction, scheduling, runtime control, and architecture-aware execution, with the goal of identifying how these mechanisms interact and where cross-layer coordination remains inadequate.

The objective is threefold. First, the paper consolidates the recent literature into a structured and GenAI-specific view of energy-aware GPU management. Second, it uses a multidimensional taxonomy to compare otherwise heterogeneous studies on a common basis. Third, it identifies the technical gaps that continue to prevent energy-proportional, workload-aware, and sustainable execution of large-scale GenAI workloads on future HPC platforms.

2.2 Literature Selection Strategy

The review follows a focused and transparent literature-curation process rather than a fully exhaustive systematic literature review (SLR). This choice is intentional. The goal of the paper is not to enumerate every publication touching AI energy efficiency, but to synthesize the most relevant work for cross-layer GPU resource management in the context of modern Generative AI and HPC systems.

The literature search was conducted across major digital libraries and archival sources, including IEEE Xplore, the ACM Digital Library, SpringerLink, Elsevier journals, and selected arXiv preprints. The primary time window was 2022–2025 so as to capture the rapid expansion of GenAI workloads, large language model deployment, and recent advances in GPU-centered energy management. Search combinations included terms such as "Generative AI," "LLM," "GPU scheduling," "energy-aware," "HPC," "power capping," "DVFS," "GPU resource management," and "co-design."

Candidate papers were screened according to relevance, technical depth, and usefulness for cross-layer synthesis. Priority was given to peer-reviewed journal articles and top conference papers that explicitly address GPU energy, power, scheduling, runtime adaptation, or architecture-aware resource management in AI or HPC settings. High-quality preprints were retained when they introduced timely systems ideas, emerging accelerator directions, or practically validated cross-layer techniques that were directly relevant to GenAI execution.

For each retained paper, we recorded the publication venue, system layer, targeted workload phase, decision type, methodological nature (empirical, simulation-based, conceptual, or measurement-focused), evaluation scale, and primary energy-efficiency mechanism. This metadata was used to build the taxonomy and support comparative analysis across studies. The final review set contains 46 representative papers selected for their direct relevance to energy-aware GPU management, methodological clarity, and value for identifying cross-layer design patterns.

This review is intentionally curated rather than fully exhaustive. We therefore do not claim PRISMA-style completeness. Instead, the methodology is designed to

provide a reproducible analytical structure for comparing representative work in a fast-moving research area where system designs, workloads, and accelerator capabilities are evolving rapidly.

The selection criteria were as follows:

Included if:

- The paper explicitly addresses GPU or GPU-cluster energy, power, scheduling, runtime control, or resource management with clear implications for system-level efficiency.
- It proposes or evaluates measurement, prediction, scheduling, orchestration, runtime adaptation, or architectural enablers relevant to energy-efficient GenAI or AI-HPC execution.
- It includes empirical evaluation, realistic simulation, benchmark-driven analysis, or a sufficiently concrete systems framework to support comparative synthesis.

Excluded if:

- The work focuses only on model compression, algorithmic accuracy improvement, or architecture design without meaningful system-level energy or resource-management implications.
- It is purely a high-level position piece without concrete models, mechanisms, or evaluation evidence, except where a conceptual framework contributes directly to cross-layer systems reasoning.
- It is duplicative, incremental, or too limited in evaluative content to support reliable comparison within the taxonomy.

2.3 Research Questions

The review is organized around the following research questions, which are intended to connect descriptive literature coverage with critical cross-layer synthesis:

1. How have GPU resource-management techniques evolved to improve the energy efficiency of Generative AI workloads on HPC systems?
2. How is GPU energy measured or modeled in current work, and what limitations remain for GenAI-oriented resource management?
3. What job-level power or energy prediction techniques have been proposed, and how effectively can they inform scheduling and allocation decisions?
4. Which GPU scheduling and runtime strategies are used for training and inference workloads, and how do they affect energy efficiency across different HPC scales?
5. How do emerging architectural enablers, such as processing-in-memory and wafer-scale accelerators, influence energy-aware scheduling and orchestration?
6. What cross-layer coordination mechanisms are needed to jointly optimize hardware, runtime, scheduler, and cluster-level control?

Key observations in Table I are supported by prior works [1, 3, 8, 10, 12, 16, 18-20, 26, 27, 39, 41, 44, 45, 46].

Table 1 Mapping of Research Questions to Sections and Key Points

RQ	Section(s)	Key Points
RQ1	IV & V	Static → dynamic cross-layer frameworks
		Node: DVFS, power capping, GPU partitioning
		Scheduler: STS, InEPS, HARMONIC
		Runtime: LithOS, adaptive control
		Cross-layer co-design for energy efficiency
RQ2	IV.A	Multi-GPU H100 nodes, benchmarks (LLaMA2, ResNet)
		DVFS-aware modeling, ML/forecasting
		Limitations: general HPC/AI, not GenAI-specific
RQ3	IV.A & IV.B	ML predictors and analytical forecasting models
		Enables elastic allocation and dynamic scheduling
		Mostly offline/static, limited runtime integration
RQ4	IV.B & IV.C	Training: throughput-oriented (DVFS, backfill)
		Inference: latency-sensitive (DynamoLLM, CE-DLA)
		Cluster consolidation achieving ∼40% savings
RQ5	V	PIM reduces memory movement (10–70×)
		Wafer-scale: up to 16× efficiency, 200× throughput
		CGLA: fine-grained kernel placement
		Power gating enables idle shutdown
RQ6	VI	Fragmented single-layer approaches
		Coordination: DVFS, power capping, PIM, wafer-scale
		Telemetry → Prediction → Scheduler → Runtime
		Energy-aware and carbon-aware HPC systems

2.4 Taxonomy Design

To support consistent comparison across heterogeneous studies, we organize the literature using a compact multidimensional taxonomy. The taxonomy is designed not only to classify prior work, but also to expose where existing methods are concentrated and where important cross-layer gaps remain.

1. **System Layer**: Studies are grouped by the main layer at which energy or resource-management decisions are applied. The *Node / Accelerator* layer includes hardware-level mechanisms such as DVFS, GPU frequency scaling, power capping, and device-local control. The *Runtime / OS* layer includes kernel scheduling, resource isolation, power-aware runtime behavior, and execution-time adaptation. The *Scheduler / Cluster* layer covers job scheduling, GPU allocation, model placement, batching, consolidation, and multi-node coordination. The *System / Data-Center* layer includes broader placement, infrastructure-level optimization, and energy- or carbon-aware coordination across larger operational scopes. *Cross-layer* studies are those that intentionally connect decisions across multiple layers.

2. **Workload Phase**: Each paper is categorized according to the workload phase it primarily addresses: *Training*, *Inference*, *Both*, or *None* for work that is measurement-oriented, conceptual, or otherwise not tied to a single execution phase.
3. **Decision Type**: We distinguish among *Static* methods, which rely on offline rules, profiling, or fixed policies; *Dynamic* methods, which adapt during execution using system or workload feedback; and *None*, for studies that contribute measurement infrastructure, conceptual framing, or descriptive characterization without an explicit control policy.
4. **Evaluation Scale**: Because reported impact depends strongly on deployment scale, we record whether evidence is derived from microbenchmarks, single-node experiments, simulator studies, cluster-scale evaluation, or broader system/data-center settings.
5. **Energy-Efficiency Technique**: Each study is also classified by its primary mechanism, including energy measurement, prediction, DVFS and power capping, scheduling, GPU sharing, placement and consolidation, dynamic provisioning, runtime control, or architectural / hardware–software co-design.

This taxonomy reflects the hierarchical structure of GPU energy management in HPC environments while remaining practical for comparative analysis. It enables the review to connect local control mechanisms with cluster-level orchestration, distinguish training-driven and inference-driven requirements, and identify where cross-layer integration is still missing in the current literature.

2.5 Paper Mapping and Comparative Analysis Framework

The 46 chosen papers were mapped to the taxonomy and summarized in a master comparison table (Table II and Table III) that included the system layer, workload phase, decision type, evaluation scale, and key focus. This mapping allows comparative analysis to be done in a structured way, trends to be identified and gaps to be discovered throughout the literature.

Table IV indicates the allocation of energy-efficiency studies in system layers. As the analysis reveals, the most common strategies in the literature are Cluster/Scheduler-level strategies (17 papers), reflecting the emphasis on job scheduling and resource allocation for Generative AI workloads, consistent with this review's focus. System/Data Center/Macro-level studies (12 papers) consider broader infrastructure and energy-efficiency strategies. In contrast, Node/Hardware-level (9 papers), OS/Runtime-level (4 papers), and Cross-layer/Co-Design approaches (4 papers) are underrepresented, revealing gaps in fine-grained runtime control and holistic, cross-layer optimization. These gaps reveal that to enable comprehensive energy-efficient management of GenAI workloads the future research should pursue unified cross-layer frameworks, integrating hardware, OS/runtime, and scheduler layers.

Table 2 Mapping of surveyed studies across system layer, workload, decision type, evaluation scale and key focus (Refs. [1]–[23])

Ref.	Year	Venue	System Layer	Workload Phase	Decision Type	Evaluation Scale	Key Contribution
[1]	2025	IEEE Access	Node / Hardware	Training	Measurement	Single node (8×A100)	Empirical GPU power characterization for GenAI workloads
[2]	2025	Sustainable Computing (Elsevier)	OS / Runtime	Training	Dynamic	18 clusters / 9 benchmarks	DVFS-aware power–performance modeling
[3]	2024	IEEE TCC	Cluster / Scheduler	Training	Dynamic	100 nodes / 400 jobs	Energy-aware deep learning job scheduling
[4]	2025	Energy Informatics	System / Data Center	None	Measurement	Large trace (9PM100 dataset)	Job-level carbon efficiency analysis
[5]	2025	Information (MDPI)	Data Center / Macro	Both	Measurement	Multi-hardware platforms	Empirical AI energy benchmarking
[6]	2024	IEEE Micro	System / Data Center	Both	Conceptual	Meta global infrastructure	AI carbon lifecycle analysis
[7]	2025	IEEE IC3	System / Macro	None	Conceptual	Global HPC trends	HPC-AI convergence vision
[8]	2022	Energies (MDPI)	Cluster / Scheduler	Both	Dynamic	GPU trace-driven evaluation	Cost-aware GPU job allocation
[9]	2024	SC	Cross-layer / Co-design	Training	Static	10,000 A100 GPUs	AI–HPC co-design framework
[10]	2025	arXiv	OS / Runtime	Both	Dynamic	Multi-GPU systems	GPU OS-level resource management
[11]	2025	IEEE TSUSC	System / Data Center	Inference	Dynamic	Geo-distributed data centers	Carbon-aware inference routing
[12]	2025	IEEE TCC	System / Data Center	None	Dynamic	CloudAIBus / AWS	Carbon-neutral cloud resource management
[13]	2025	ACM FSES	System / Data Center	Inference	Dynamic	Forecast-based grid data	Carbon-aware quality adaptation
[14]	2025	IEEE JETCAS	Node / Hardware	Inference	Static	3D-PIM simulator	Processing-in-memory for LLM efficiency
[15]	2025	IEEE Access	System / Data Center	Both	Measurement	System-wide estimates	Energy comparison of AI and blockchain
[16]	2025	HPCA	Cluster / Scheduler	Inference	Dynamic	Microsoft Azure trace	LLM inference energy management
[17]	2023	SoCC	Node / Hardware	Training	Static	MIT supercomputing scale	GPU power capping techniques
[18]	2025	SC Workshops	Cluster / Scheduler	Both	Dynamic	NREL solar–HPC site	LLM-based power prediction
[19]	2025	IEEE TPDS	Cluster / Scheduler	None	Dynamic	Simulation + testbed	Multi-objective resource management
[20]	2025	Journal of Supercomputing	Cluster / Scheduler	None	Dynamic	Simulation	DRL-based energy-aware scheduling
[21]	2025	SC Workshops	OS / Runtime	Both	Dynamic	Multi-application workloads	Runtime energy optimization
[22]	2025	SCA	OS / Runtime	Both	Dynamic	Simulation	DRL-based power on/off control
[23]	2024	ISCA	Cluster / Scheduler	Inference	Dynamic	Large GPU clusters	Phase-split LLM inference

3 Characteristics of Generative AI Workloads on HPC

The execution and energy-consumption characteristics of Generative AI workloads differ fundamentally from those of traditional AI and conventional HPC applications, requiring specialized approaches to GPU resource management.

Table 3 Mapping of surveyed studies across system layer, workload, decision type, evaluation scale and key focus (Refs. [24]–[46])

Ref.	Year	Venue	System Layer	Workload Phase	Decision Type	Evaluation Scale	Key Contribution
[24]	2024	HotCarbon	Cluster / Scheduler	Inference	Static	Heterogeneous systems	Energy modeling for LLMs
[25]	2023	HotCarbon	System / Data Center	Inference	Static	Multi-region deployment	Carbon-aware request routing
[26]	2025	IEEE JETCAS	Cross-layer / Co-design	Both	Conceptual	Survey	GenAI CAS co-design survey
[27]	2025	IEEE TPDS	Cluster / Scheduler	None	Dynamic	Large HPC logs	Power-aware job scheduling
[28]	2025	arXiv	Cluster / Scheduler	Both	Dynamic	Tens of thousands of GPUs	Unified AI cluster scheduling
[29]	2025	FGCS	Cluster / Scheduler	None	Static	Simulation	Energy–time workload optimization
[30]	2025	arXiv	Cluster / Scheduler	None	Dynamic	Simulation	RL-based power-aware scheduling
[31]	2025	arXiv	Scheduler / Cluster	Inference	Dynamic	Production workload simulation	Carbon- and energy-aware LLM scheduling
[32]	2025	arXiv	Scheduler / Cluster	Both	Dynamic	64 GPUs / 8 nodes	Improving GPU cluster utilization
[33]	2025	arXiv	Scheduler / Cluster	Both	Dynamic	Trace-based multi-center simulation	Electricity cost minimization
[34]	2025	arXiv	Macro / System	Both	Conceptual	Analytics + micro-simulation	Carbon accounting and climate risk
[35]	2025	ACM	Macro / System	Inference	Conceptual	Cloud vs. edge case study	Environmental footprint comparison
[36]	2025	CJE	Scheduler / Cluster	Training	Conceptual	Survey / literature	Job scheduling survey
[37]	2025	arXiv	Node / Hardware	Both	Static	Blackwell B200 GPUs	Energy efficiency via power profiling
[38]	2025	arXiv	Node / Hardware	Inference	Dynamic	Microbenchmarks + simulation	GPU partitioning for energy efficiency
[39]	2025	MICRO	Node / Hardware	Both	Dynamic	Simulator implementation	Hardware power-gating in NPUs
[40]	2023	Energies (MDPI)	Scheduler / Cluster	Both	Conceptual	Survey / literature	Energy-aware scheduling methods survey
[41]	2025	JISEM	Cross-layer / Co-design	Both	Conceptual	Literature analysis	Energy optimization techniques
[42]	2023	HPEC	Node / Hardware	Inference	Measurement	32-GPU multi-node (V100/A100)	Benchmarking LLM inference energy
[43]	2025	INCOFT	Node / Hardware	Both	Dynamic	Experimental evaluation	Adaptive GPU power/performance management
[44]	2025	IEEE Access	Node / Hardware	Inference	Static	FPGA prototype + ASIC projection	Kernel mapping on CGRA
[45]	2025	Device (Elsevier)	Macro / System	Both	Conceptual	Comparative analysis	Accelerator type energy comparison
[46]	2025	arXiv	Cross-layer / Co-design	Inference	Static	Accelerator vs. GPU cluster	Wafer-scale LLM inference and mapping

3.1 Extreme Model Scale

The scaling from GPT-3 (175B parameters) to GPT-4 (1.76T parameters) represents a substantial increase in computational demand, memory footprint, and system energy

Table 4 Surveyed Studies Distribution Across System Layers

System Layer	# Papers	References
Node / Hardware Level	9	[1], [14], [17], [37], [38], [39], [42], [43], [44]
OS / Runtime Level	4	[2], [10], [21], [22]
Cluster / Scheduler Level	17	[3], [8], [16], [18], [19], [20], [23], [24], [27], [28], [29], [30], [31], [32], [33], [36], [40]
System / Data Center / Macro Level	12	[4], [5], [6], [7], [11], [12], [13], [15], [25], [34], [35], [45]
Cross-layer / Co-design Level	4	[9], [26], [41], [46]

consumption [15]. Unlike classical deep learning workloads, GenAI's attention mechanisms induce irregular memory access, variable compute intensity, and highly dynamic GPU utilization.

3.2 Memory and Communication-Dominated Energy

Large models need to be trained with high-frequency synchronization and continuous use of high-bandwidth memory (HBM) and interconnects, and memory and communication can often be the primary energy consumers [1]. Architectural studies suggest that large language models are energy-bottlenecked by excessive data movement rather than computation [14]. For example, recent processing-in-memory and near-memory accelerator studies report order-of-magnitude (up to tens-fold) energy improvements when data movement is reduced, indicating that memory and interconnect activity constitutes a dominant share of total energy in large transformer workloads [14].

3.3 Training versus Inference Workloads

Training is throughput-oriented and long-running, whereas inference is latency-sensitive, bursty, and persistent (Fig. 1). Inference-only (13 studies) and mixed workloads (19 studies) dominate the literature, while training-only workloads remain underexplored (6 studies), reflecting a research focus on latency-sensitive GPU management. Scaling inference across multiple GPUs reduces latency but increases energy non-linearly. This reveals a critical energy–latency trade-off in distributed HPC deployments [42].

3.4 Workload Representation in Energy-Aware Studies

Fig. 2 demonstrates that 18 studies cover general HPC workloads, 16 are dedicated to Generative AI, and 12 are both. Studies focused specifically on GenAI remain underrepresented relative to general HPC workloads, revealing a gap in workload-specific energy-optimization frameworks and motivating further cross-layer research.

3.5 Implications for Energy-Aware GPU Management

Extreme model scale, memory-dominated execution, and persistent inference demand render static GPU allocation and performance-only scheduling inefficient. Traditional HPC resource management fails to account for workload variability, utilization imbalance, and energy per query efficiency, motivating energy-aware GPU resource management that explicitly integrates workload behavior, energy measurement, and adaptive scheduling [1], [5], [42].

3.6 Motivation for Cross-Layer Energy Optimization

These challenges motivate cross-layer co-design and adaptive approaches that leverage hardware capabilities, runtime insights, and scheduler-level decisions to achieve energy-efficient operation. Section IV examines scheduling strategies that exploit these workload characteristics to improve efficiency, whereas Section V discusses architectural enablers that support such integration.

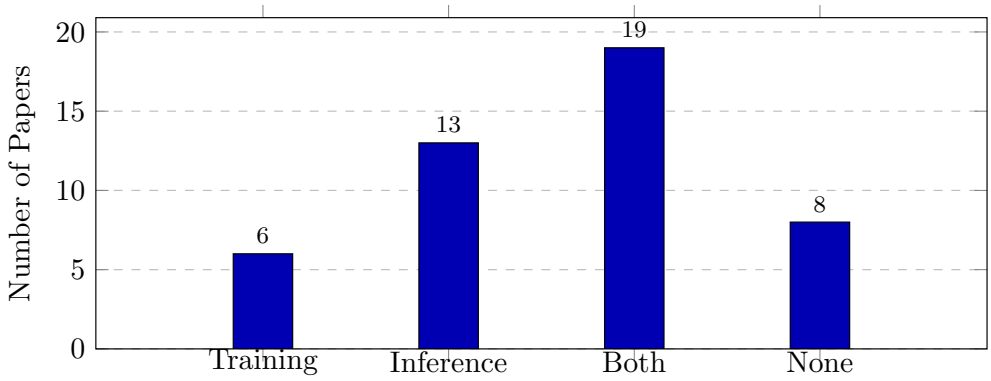


Fig. 1 Workload phase distribution of surveyed studies

Fig.1 illustrate surveyed studies by workload phase: inference (13) and Both (training and inference) (19) dominate, while training only (6) is underrepresented, highlighting emphasis on latency sensitive and mixed HPC workloads.

Fig.2 illustrate surveyed studies by AI workload type: general HPC (18), Generative AI (16), and both (12); purely Generative AI-focused energy optimization remains underrepresented.

4 Energy-Aware Resource Management Techniques

Building on the workload characteristics and energy challenges discussed in Section III, this section organizes energy-aware GPU resource-management techniques into a hierarchical framework. Accurate energy measurement forms the foundation of this framework, enabling predictive models that inform intelligent scheduling and, in turn, guide runtime and hardware-level control. This perspective reflects the principle that

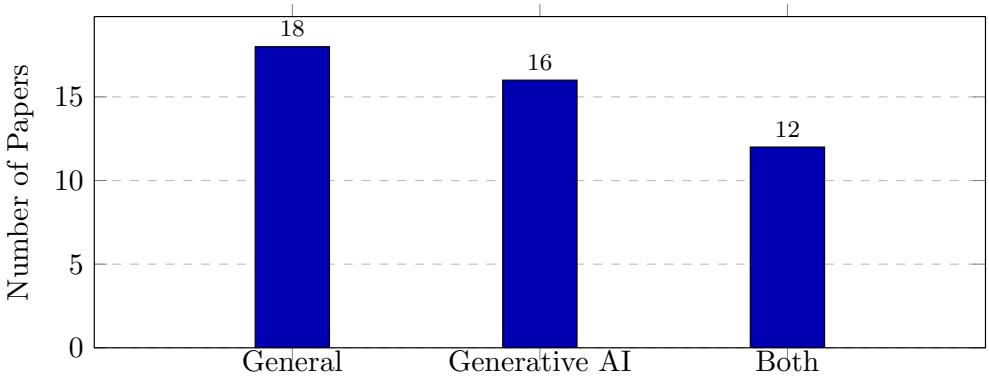


Fig. 2 Distribution of surveyed studies by workload type

"you cannot manage what you cannot measure": measurement, prediction, and scheduling are interdependent, and effective energy-aware management of Generative AI workloads requires coordination across these layers. The following subsections examine each stage in detail, highlighting relevant techniques, limitations, and adaptations for handling the extreme scale, memory intensity, and dynamic inference behavior of modern Generative AI workloads.

4.1 Energy Measurement and Prediction

Measurement and Ground Truth: Accurate energy measurement forms the foundation for energy-aware management of Generative AI workloads. Experimental results of 8-GPU NVIDIA H100 HGX nodes indicate that power consumption and overall energy consumption can change dramatically with the training stage, batch size, and GPU utilization, as demonstrated by typical workloads, such as ResNet and LLaMA2-13B [1]. These measurements provide ground-truth data that can be used to model and predict energy and subsequently make intelligent scheduling decisions.

Energy and Power Modeling: Analytical approaches, particularly DVFS-aware frameworks such as the Fluid model, capture the effects of frequency scaling, workload characteristics, and node heterogeneity on performance, power, and total energy consumption [2]. Machine-learning-based predictors use historical execution traces, workload characteristics, and runtime indicators to estimate per-job energy consumption without additional instrumentation [18], [27]. The energy-time modeling framework in [29] provides precise predictions of application execution time and energy consumption during dynamic workload allocation in heterogeneous HPC clusters, enabling more effective scheduling and resource selection. Sophisticated uncertainty-aware and multi-objective models also take into account trade-offs between energy, performance and system stability under dynamic execution conditions [19]. Frameworks like AINet0 that rely on forecasting can be used to predict future resource demands to proactively adjust provisioning, indirectly reducing energy waste [12]. Current analytical, machine-learning, and forecasting-based energy models evaluated on

traditional HPC or general AI workloads provide useful insight into power and runtime behavior [2], [12], [18], [19], [27], [29]. However, none of these models are designed specifically for Generative AI workloads, whose extreme scale, phase heterogeneity, and persistent, token-driven inference fundamentally change energy performance relationships, necessitating adaptive and Generative AI-aware prediction mechanisms.

Challenges for Generative AI: Generative AI workloads challenge classical energy prediction due to extreme variability, phase-dependent execution, and bursty inference behavior [5], [15]. Inference can frequently dominate the total energy consumption over a model lifetime [15], [25], resulting in very non-linear energy-performance relationships that are not well represented by either a static or offline model. These characteristics necessitate adaptive, workload-aware, and scheduler-integrated energy prediction mechanisms tailored specifically to Generative AI workloads.

4.2 Scheduling and Resource Allocation

Scheduling Types and Strategies: Scheduling and resource allocation are central to the energy efficiency of Generative AI workloads on HPC systems because these workloads are heterogeneous, placement-sensitive, and massively parallel [3], [36]. Static or naïve schedulers often leave GPUs underutilized, wasting energy. Stochastic and backfill-inspired strategies improve utilization by accounting for job variability and resource availability [3], while surveys highlight that static policies limit elastic allocation, GPU sharing, and heterogeneous placement [36]. Adaptive, workload-aware scheduling is therefore necessary to reduce idle resources and minimize energy consumption.

GPU Placement and Elastic Allocation: GPU placement and partitioning have a significant impact on performance and energy, especially when using large-scale LLMs. In sharded deployments across multiple GPUs, 75-80% of GPU memory may remain underutilized because of partitioning and communication overheads [42]. Placement strategies that are communication-aware reduce underutilization and energy waste [31]. Moreover, model-specific profiling enables finer-grained GPU allocation based on workload characteristics [23]. Together, these approaches highlight the potential for energy savings via co-location, sharing, and intelligent scheduling rather than indiscriminate scaling of GPU counts.

To mitigate these inefficiencies, a variety of heuristic- and learning-based schedulers have been proposed. The Stochastic Training Scheduler (STS) improves energy efficiency by leveraging workload variability and cluster conditions [3], while InEPS explicitly models job–resource compatibility in heterogeneous GPU clusters to guide energy-aware placement [20]. More recent multi-objective frameworks, such as HARMONIC, incorporate uncertainty-aware optimization to balance energy consumption, performance, and system stability under dynamic workload conditions [19]. Dynamic multi-objective scheduling heuristics improve GPU utilization, throughput, and fairness while reducing starvation, indirectly lowering energy waste [32].

Beyond placement, right-sized and elastic GPU allocation is also effective for minimizing energy loss due to over-provisioning. CE-DLA dynamically allocates and places jobs on GPUs according to electricity price and workload demand, reducing

cluster energy consumption by preventing idle or underutilized GPUs [8]. Similarly, DynamoLLM shows that dynamically reconfiguring inference clusters according to workload intensity can reduce energy consumption and operational cost while satisfying latency SLOs [16]. These results further indicate that inference energy efficiency is highly sensitive to the number of GPUs, batch size, and utilization.

Power-Aware and Multi-Tenant Scheduling: Power control mechanisms complement scheduling strategies as additional levers for energy optimization. DVFS-aware schedulers exploit analytical power–performance models to achieve meaningful energy savings with minimal performance loss [2], and GPU power capping has been shown to reduce energy consumption and thermal stress without significant throughput penalties when coordinated with placement decisions [17]. These findings highlight that runtime power management becomes most effective when integrated with workload-aware scheduling.

Schedulers in multi-tenant HPC environments also need to consider constraints of QoS, fairness, and sustainability. Latency targets, job prioritization, and service-level goals are becoming a common feature of energy-conscious scheduling models to ensure that energy savings do not affect performance [5], [16]. Carbon-conscious strategies also align job execution with cleaner energy availability or prioritize energy-efficient workload placement, thereby reducing carbon emissions with limited impact on execution time [4], [11], [13].

In general, the existing scheduling and resource allocation methods demonstrate the trade-offs between energy efficiency, latency, and system utilization. Workload-aware, adaptive schedulers can substantially reduce energy consumption and GPU underutilization, but their effectiveness depends on accurate workload characterization and coordination across placement, allocation, and power-control layers. These findings highlight the importance of integrated scheduling models that jointly optimize GPU usage, energy consumption, and scalability.

4.3 Runtime and Power Control Mechanisms

Runtime mechanisms offer finer-grained control over GPU energy usage in addition to scheduler-level decisions. Dynamic Voltage and Frequency Scaling (DVFS) is a technology that varies the frequency of the GPU and CPU based on the nature of the workload, allowing energy savings with little performance degradation [2], [21]. GPU power capping constrains peak power draw, mitigating energy waste from underutilized resources while maintaining thermal limits, with studies reporting up to 15% energy savings when coordinated with scheduler policies [17], [37].

Operating-system–level support is critical for effective runtime control. Frameworks such as LithOS [10] abstract low-level hardware interfaces, enabling dynamic frequency tuning, power capping, and intra-GPU scheduling without modifying applications. OS-level intelligence facilitates accurate monitoring of resource utilization, execution phases, and memory activity, providing the foundation for adaptive in-job energy management.

In-job adaptation, including runtime frequency tuning and reinforcement learning–based adjustments, exploits phase-dependent and bursty behaviors of Generative AI workloads, dynamically optimizing energy while preserving performance [21], [22].

Integrating these runtime controls with OS-level support and scheduler-aware placement ensures energy-efficient execution for large-scale training and inference tasks while maintaining throughput and latency guarantees.

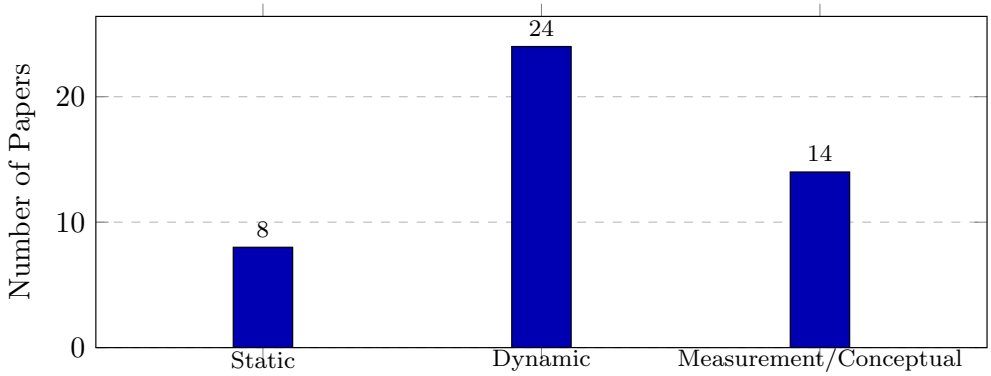


Fig. 3 Decision type distribution of surveyed studies

Figure 4 summarizes the surveyed energy-aware resource-management techniques by primary energy-reduction mechanism. Measurement and conceptual frameworks dominate the literature, while dynamic provisioning, hardware-software co-design, and runtime controls complement scheduling- and DVFS-based strategies.

[htbp] Despite the breadth of techniques, gaps remain in GenAI-specific prediction, joint optimization across scheduling and runtime layers, and cross-layer integration, motivating architectural enablers discussed in Section V.

5 Architectural Enablers for Energy-Efficient Generative AI on HPC

Although this review primarily focuses on GPU scheduling and cluster-level resource management, hardware architecture offers critical levers that OS- and scheduler-level frameworks can exploit to achieve better energy efficiency for Generative AI workloads. Modern GPUs and accelerators provide runtime energy controls, including dynamic voltage and frequency scaling (DVFS) [2] and power capping [17], allowing fine-grained power–performance adaptation without modifying applications. These mechanisms allow higher layers to enforce energy budgets and optimize execution across variable workload phases.

5.1 GPU and Accelerator Architecture

Modern GPU and accelerator architectures provide the building blocks for energy-efficient Generative AI. Mechanisms such as DVFS [2], fine-grained power gating [39], and reconfigurable execution units [44] enable dynamic adaptation to workload intensity while reducing both static and dynamic energy consumption. Real-time adaptive

Table 5 Summary of Key Studies by Subsection

Subsection	Representative Papers	Key Insights Captured
A. Energy Measurement and Prediction	[1], [2], [12], [18], [19], [27], [29]	GPU power/energy measurements; DVFS-aware analytical models; ML and forecasting-based energy prediction; uncertainty/multi-objective models; highlights phase variability, bursty inference, and non-linear energy–performance behavior.
B. Scheduling and Resource Allocation	[2], [3], [4], [5], [8], [11], [13], [16], [17], [19], [20], [23], [31], [32], [36], [42]	Heuristic- and learning-based schedulers (STS, InEPS, HARMONIC); elastic/right-sized GPU allocation (CE-DLA, DynamoLLM); GPU placement and partitioning to reduce underutilization; multi-tenant, QoS- and phase-aware inference scheduling; integration of carbon- and energy-aware policies; highlights energy–latency–utilization trade-offs and benefits of adaptive, workload-aware scheduling.
C. Runtime and Power Control Mechanisms	[2], [10], [17], [21], [22], [37]	DVFS- and power-capping-based runtime control; OS-level frameworks for adaptive scheduling and monitoring (LithOS); in-job phase-aware and RL-based adaptation; highlights coordination of runtime, OS, and scheduler mechanisms for energy-efficient GenAI execution while preserving performance and latency.

frameworks leverage workload monitoring, core allocation, and thermal management to improve performance-to-power ratios by approximately 50% [43].

Wafer-scale accelerators further improve energy efficiency by integrating massive compute and memory resources, reducing inter-chip communication, and delivering high throughput per watt for extreme-scale AI models [45]. Together, these innovations enable energy-proportional operation by combining fine-grained control, dynamic adaptation, and high-density compute.

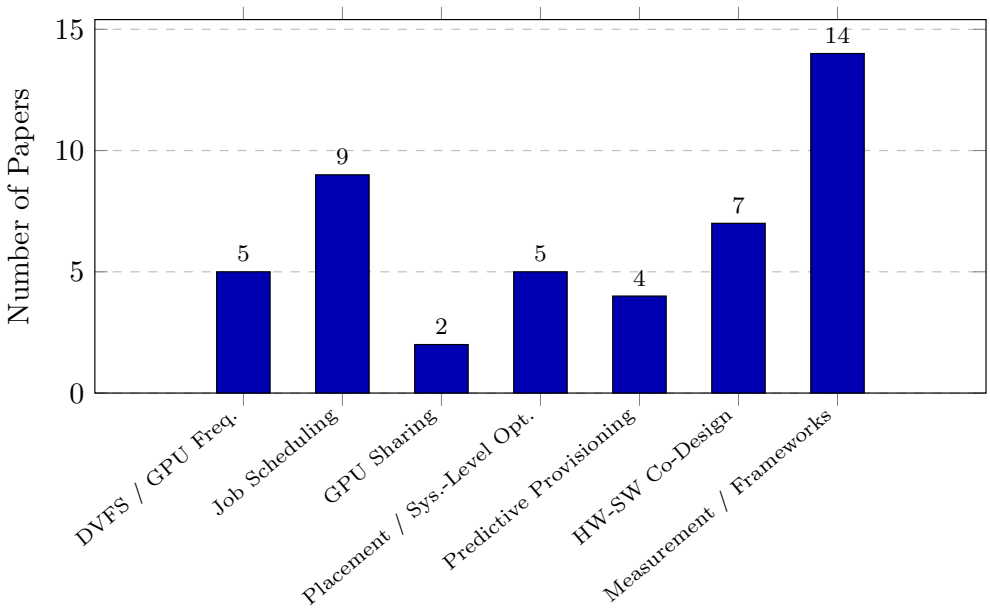


Fig. 4 Energy-efficiency techniques used in surveyed studies. DVFS = dynamic voltage and frequency scaling; HW-SW = hardware-software co-design.

5.2 Memory and Communication Optimization

Memory subsystems are a dominant source of energy consumption in large-scale LLMs. High-bandwidth memory (HBM) provides the necessary throughput but incurs substantial static power, particularly under low utilization, as observed in single-node 8×H100 training [1] and multi-GPU LLM inference [42].

Architectural strategies that minimize data movement and maximize on-chip locality have emerged as key enablers of energy-efficient GenAI. Processing-in-Memory (PIM) designs, such as CIDAN-3D, embed low-power neuron processing elements directly inside DRAM, performing computation in situ to reduce off-chip memory traffic and improve energy efficiency by up to 71× for large decoder-only LLMs [14]. Likewise, coarse-grained linear array (CGLA) accelerators optimize kernel placement and host–accelerator communication, achieving 44× PDP and 11× EDP improvements over traditional GPU inference [44].

Wafer-scale LLM inference systems such as WaferLLM take advantage of mesh-aware operator placement, distributed on-chip memory, and parallel execution strategies to maximize utilization and minimize communication overhead, yielding 16× better energy efficiency and up to 200× higher hardware utilization than A100 GPU clusters [46]. Together, these approaches demonstrate that careful memory–compute co-location, communication-aware mapping, and architecture-specific optimizations are critical for energy-proportional operation in high-throughput Generative AI workloads.

Table 6 Classification of energy-efficiency and resource-management techniques across 46 surveyed studies

Energy-Efficiency / RM Technique	Description	Example Papers
DVFS / GPU Frequency Scaling	Adjusts CPU/GPU voltage and frequency at runtime to reduce power while meeting performance targets	[2], [17], [21], [37], [43]
Job Scheduling	Prioritizes and orders jobs to improve utilization, minimize idle energy, and balance workloads	[3], [8], [19], [20], [27], [28], [30], [31], [32]
GPU / Resource Sharing	Shares GPU or CPU resources across multiple jobs to reduce underutilization and idle energy	[10], [38]
Workload Placement / System-Level Optimization	Maps workloads to energy-efficient nodes or regions and dynamically optimizes resource usage	[11], [13], [23], [25], [33]
Predictive / Dynamic Provisioning	Forecasts workload demand or scales resources in real time to avoid over-provisioning and wasted energy	[12], [16], [18], [22]
Architecture / HW–SW Co-Design	Integrates hardware and software via architectural mapping, PIM, or design-time optimization to minimize energy	[9], [14], [26], [39], [44], [45], [46]
Measurement / Conceptual Frameworks	Provides empirical energy measurement, profiling, or conceptual models without active runtime control	[1], [4], [5], [6], [7], [15], [24], [29], [34], [35], [36], [40], [41], [42]

5.3 Emerging Accelerator Paradigms: PIM, Wafer-Scale, CGLA, and Power Gating

Emerging accelerator architectures offer substantial energy gains for Generative AI workloads by minimizing data movement, increasing compute density, and enabling dynamic resource adaptation.

Processing-in-Memory (PIM) designs, such as CIDAN-3D [14], embed low-power neuron processing elements (NPEs) directly inside 3D-stacked DRAM. This

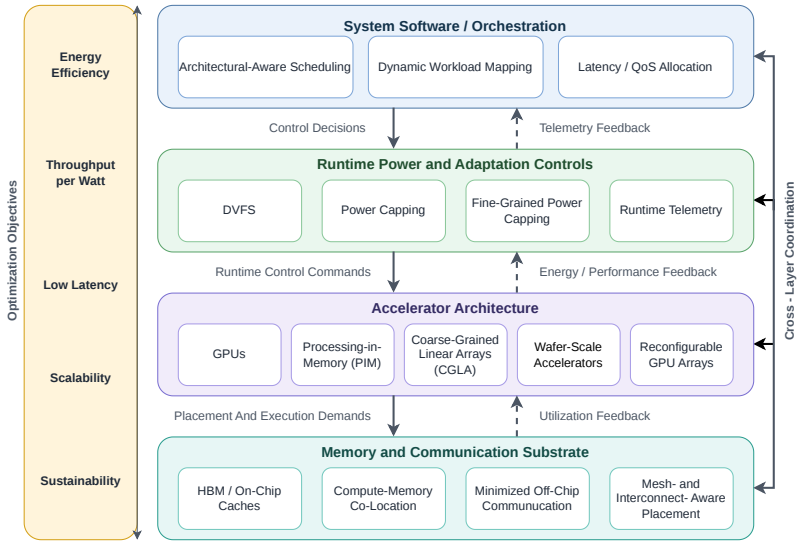


Fig. 5 Cross-layer architectural enablers for energy-efficient Generative AI on HPC, linking memory and communication efficiency, accelerator specialization, runtime adaptation, and system-level orchestration.

in-situ computation reduces conventional DRAM-GPU transfers, significantly lowering off-chip memory energy while increasing parallelism and throughput for LLM inference.

Wafer-scale accelerators consolidate massive compute and memory resources on a single die, removing inter-chip communication overheads [45]. Specific implementations, such as WaferLLM [46], achieve up to 200× higher hardware utilization and 16× better energy efficiency than multi-GPU clusters. Despite their fabrication complexity and cost, wafer-scale designs are particularly attractive for ultra-large LLMs and extreme-scale HPC inference.

Coarse-Grained Linear Arrays (CGLA) [44] provide programmable, energy-efficient execution for LLM inference by mapping kernels to reconfigurable arrays. By localizing computation and minimizing host–accelerator transfers, CGLAs improve power-delay and energy-delay metrics by an order of magnitude relative to conventional GPUs, highlighting the benefits of specialized accelerator topologies.

Power gating and reconfigurable GPU arrays (ReGate) [39] allow the selective shutdown of idle functional units, minimizing both static and dynamic power without affecting performance. Combined with resource elasticity across GPUs or accelerator tiles, these mechanisms enable active compute and memory resources to be resized dynamically, complementing higher-level scheduler policies and runtime adaptation frameworks.

When combined with real-time adaptive frameworks for workload-aware GPU management [43], these hardware mechanisms further amplify energy savings by aligning active compute and memory resources with workload intensity.

There are a number of common themes between GPU and accelerator architectures that support energy-efficient Generative AI on HPC systems:

1. *Locality of computation and memory*: Across PIM, CGLA, and wafer-scale architectures, minimizing off-chip data transfers reduces both static and dynamic energy use while improving throughput. This highlights the importance of co-locating compute units and high-speed memory and of mapping kernels to the most appropriate cores.
2. *Dynamic adaptability*: Techniques such as DVFS [2], fine-grained power gating [39], and elastic resource allocation across GPUs or accelerator tiles [43] enable hardware to adjust performance and power consumption in response to changing workload intensity.
3. *Compute density and interconnect optimization*: Wafer-scale engines [45], [46] and CGLA arrays [44] show that high integration density combined with low-latency on-chip communication can reduce the energy overheads associated with multi-GPU execution while sustaining predictable large-scale performance.
4. *Hardware–software co-design potential*: Architectural innovations achieve their full potential when combined with adaptive scheduling, workload-aware task placement, and runtime- or OS-level energy management. For example, power-gating signals and workload telemetry can feed real-time frameworks [43] that dynamically redistribute tasks and align active compute resources without reducing throughput.

These observations point to a synergistic design approach: maximizing throughput per watt for Generative AI workloads requires tightly coordinated strategies that integrate compute–memory locality, dynamic adaptation, high-density integration, and intelligent task orchestration. These principles offer practical guidance for future HPC systems, informing both accelerator design and scheduler-level energy management for sustainable, high-performance AI computation.

6 Cross-layer Coordination and Open Challenges

6.1 The Imperative for Cross-Layer Coordination

Energy inefficiency in Generative AI workloads on HPC systems does not arise from a single bottleneck; rather, it emerges from the misalignment of system layers. Hardware exposes power- and performance-control knobs, runtimes observe fine-grained execution behavior, and schedulers make coarse-grained allocation decisions, yet these layers often operate independently and therefore produce suboptimal global outcomes. For Generative AI workloads characterized by extreme model scale, memory-dominated execution, and persistent bursty inference, such isolation fundamentally limits achievable energy savings.

Recent studies increasingly demonstrate that cross-layer coordination, rather than isolated optimization, is required to achieve energy-proportional operation at scale. At the scheduler level, frameworks such as DynamoLLM dynamically reconfigure inference clusters based on workload intensity, reducing energy consumption while meeting latency objectives [16]. At the software-hardware interface, Fire-Flyer jointly accounts for architectural execution pathways and cluster-level scheduling choices, reporting

40% energy savings without performance degradation [9]. These results suggest that scheduler decisions are more effective when they are informed by architectural and runtime properties.

Memory-centric architectural innovations further reinforce this need. Processing-in-Memory (PIM) architectures such as CIDAN-3D co-locate computation and memory, reducing data-movement costs and improving the energy efficiency of attention-intensive LLM inference by an order of magnitude [14]. Wafer-scale systems such as WaferLLM minimize inter-chip communication and achieve substantially higher throughput per watt than conventional multi-GPU clusters [46]. Nonetheless, without scheduler awareness of these architectural trade-offs, their benefits remain local rather than system-wide.

At the hardware-control layer, DVFS [2], GPU power capping [17], and fine-grained power gating [39] are useful for reducing energy usage, but they are most effective when combined with workload-adaptive scheduling and placement. OS-level systems such as LithOS show that low-level control and telemetry exposed to system software can enable fine-grained and phase-aware energy management [10], implying that node-level intelligence must be coupled with cluster-scale scheduling policies.

Collectively, these studies support one central conclusion: hardware capabilities, runtime controls, and architectural innovations are enablers rather than complete solutions. Their impact on Generative AI energy efficiency remains fundamentally limited unless they are coordinated through cross-layer information flow and scheduler-visible control policies.

6.2 Persistent Gaps and Technical Challenges

Despite progress, several challenges still prevent holistic energy optimization:

1. Fragmentation Across System Layers: Most of the current methods consider one layer at a time, including energy prediction models [18], [27], [29], DVFS-aware scheduling [2], or architectural optimizations [14], [44]. While effective locally, these techniques rarely account for interactions across nodes, workloads, or system layers. As a result, localized energy savings often fail to translate into system-wide improvements, particularly under dynamic, multi-tenant workloads.
2. Inadequate Abstractions for Dynamic Generative AI Workloads: Generative AI workloads have rapid scaling characteristics, phase-sensitive execution, and extremely bursty inference requirements [1], [5], [15], [42]. The existing schedulers are usually based on either the fixed resource requests or coarse-grained profiling, which is not able to reflect the token-level variability, memory-pressure variation, or non-linear energy-latency trade-offs. This drawback results in the conservative provisioning of GPUs, inefficient over-parallelization, or underutilization, particularly in inference [36], [42].
3. Lack of Unified Frameworks: Despite the fact that a number of studies have integrated energy- or carbon-conscious objectives into the scheduling decision-making process [4], [5], [11], [16], there is no current framework that offers an end-to-end solution that integrates energy measurement, prediction, scheduling, runtime control, and architectural enablers across nodes and clusters. The absence of

such unified frameworks limits the ability to simultaneously optimize energy efficiency, performance, scalability, and sustainability for large-scale Generative AI deployments.

4. Limited Feedback and Control Loops: Current frameworks rarely leverage real-time telemetry (memory bandwidth, GPU power draw, inference queue depth) to dynamically adjust scheduling or hardware knobs. In the absence of predictive feedback loops, energy optimization is reactive rather than proactive.

These gaps highlight that achieving energy-efficient Generative AI execution on HPC systems requires more than incremental improvements it necessitates new abstractions and coordination mechanisms that span hardware, runtime, OS, and scheduler layers.

6.3 Toward a Unified Cross-Layer Energy Management Framework

Generative AI workloads in HPC systems are highly energy intensive, yet existing optimization methods remain fragmented. Most prior approaches target a single layer-hardware (DVFS, power gating), runtime (LithOS), scheduler (DynamoLLM, CE-DLA), or architecture (PIM, wafer-scale accelerators)-without explicit cross-layer coordination. As a result, energy management remains largely reactive, static, or post hoc, while phase-dependent and bursty AI workloads remain poorly matched to available resources. Carbon-awareness and sustainability objectives are also often treated as secondary concerns.

6.3.1 Motivation

These gaps motivate a unified framework that is telemetry-driven, predictive, cross-layer, and carbon-aware. Such a framework can bridge measurement, prediction, scheduling, runtime control, and hardware specialization, thereby enabling energy-efficient, sustainable, and high-throughput execution of Generative AI workloads on HPC clusters.

6.3.2 Proposed Framework: GenAI-XEnergy

This review therefore proposes *GenAI-XEnergy*, a conceptual cross-layer framework intended to enable energy-proportional, phase-aware, and sustainable execution of large-scale Generative AI workloads on extreme-scale HPC systems. The novelty of the framework does not lie in claiming that each individual component is new; rather, it lies in integrating existing but isolated mechanisms into a unified architecture that links telemetry, prediction, scheduling, runtime adaptation, and architecture-aware resource control.

6.3.3 Core Principles

- Cross-layer integration: Hardware, runtime/OS, scheduler, cluster, and architecture-specific accelerators are coordinated rather than optimized in isolation.
- Telemetry-driven prediction: Token-level and phase-level energy and performance prediction informs decision making before inefficiencies accumulate.

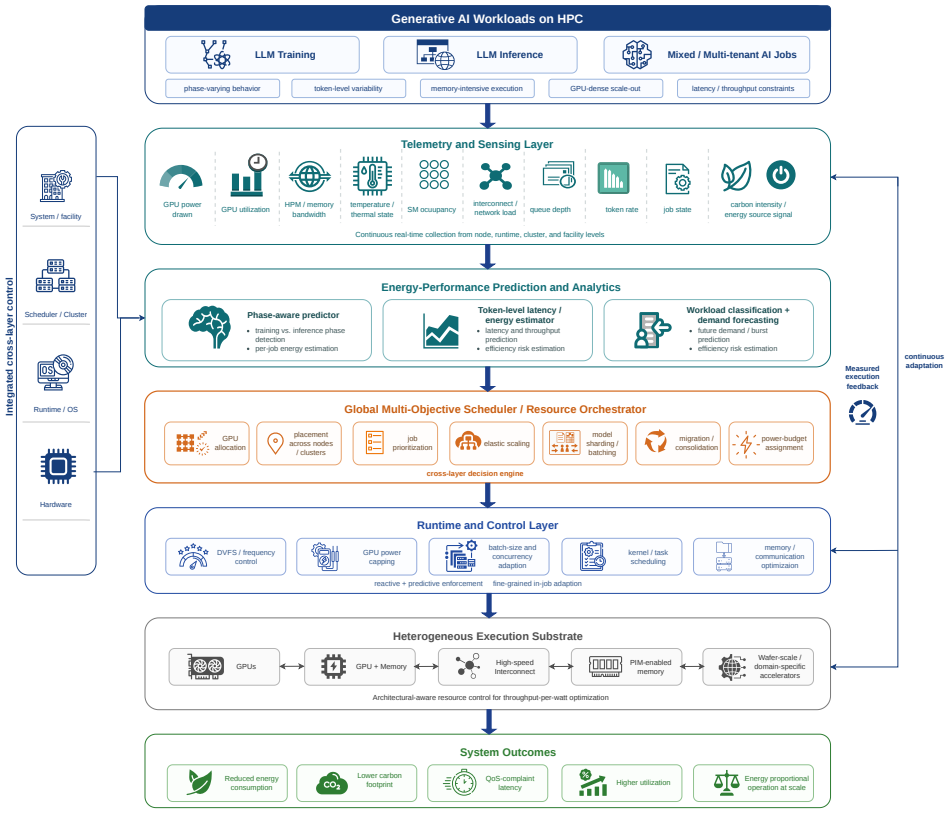


Fig. 6 Proposed GenAI-XEnergy framework for cross-layer energy management of Generative AI workloads on HPC systems, integrating telemetry, predictive modeling, scheduler orchestration, runtime control, and architecture-aware acceleration for phase-aware and energy-efficient execution.

- Dynamic workload-aware scheduling: Elastic GPU allocation, placement, and runtime control are guided by predicted workload behavior.
- Energy-proportional operation: Active resources are matched to workload intensity to reduce idle and mismatch-induced energy waste.
- Carbon-aware optimization: Scheduling decisions incorporate energy source quality and sustainability objectives in addition to performance constraints.
- Closed-loop adaptation: Predictive and reactive control are continuously combined through feedback across layers.

6.3.4 Intended Contributions

The proposed framework makes five conceptual contributions to the review. First, it provides a unifying systems view that connects hardware, runtime, scheduler, and architecture-specific mechanisms within a single control loop. Second, it elevates phase-level and token-level prediction as first-class inputs to resource management for both

Table 7 Detailed alignment of the GenAI-XEnergy framework in Fig. 6, showing its workload context, cross-layer control layers, execution substrate, feedback paths, and expected system outcomes

Fig. 6 Block	**Main Elements**	**Role in the Framework**
Workload Context	LLM training, LLM inference, and mixed / multi-tenant AI jobs, with phase-varying behavior, token-level variability, memory-intensive execution, GPU-dense scale-out, and latency / throughput constraints	Defines the workload-side demands that drive all downstream energy and resource-management decisions
Telemetry and Sensing Layer	GPU power draw, GPU utilization, HBM / memory bandwidth, temperature / thermal state, SM occupancy, interconnect / network load, queue depth, token rate, job state, and carbon intensity / energy source signal	Continuously observes system, runtime, cluster, and facility signals in real time
Energy–Performance Prediction and Analytics	Phase-aware predictor, token-level latency / energy estimator, and workload classification plus demand forecasting	Converts raw telemetry into forward-looking estimates of energy, latency, demand, and efficiency risk
Global Multi-Objective Scheduler / Resource Orchestrator	GPU allocation, placement across nodes / clusters, job prioritization, elastic scaling, model sharding / batching, migration / consolidation, and power-budget assignment	Performs cluster-level orchestration using predicted workload behavior and system state
Runtime and Control Layer	DVFS / frequency control, GPU power capping, batch-size and concurrency adaptation, kernel / task scheduling, and memory / communication optimization	Enforces fine-grained node-level and job-level actions during execution
Heterogeneous Execution Substrate	GPUs, GPU plus memory coupling, high-speed interconnect, PIM-enabled memory, and wafer-scale / domain-specific accelerators	Supplies the hardware execution platform and architecture-specific optimization opportunities
Integrated Cross-Layer Control Plane and Feedback Path	Coordination across hardware, runtime / OS, scheduler / cluster, and system / facility levels, together with measured execution feedback and continuous adaptation loops	Connects the vertical stack so telemetry, prediction, orchestration, and control remain synchronized over time
System Outcomes	Reduced energy consumption, lower carbon footprint, QoS-compliant latency, higher utilization, and energy-proportional operation at scale	Represents the target operating goals of GenAI-XEnergy

inference and mixed training workloads. Third, it brings carbon-aware scheduling into the same optimization space as energy, latency, and utilization. Fourth, it emphasizes adaptive control by coupling predictive models with real-time telemetry. Fifth, it is designed to remain applicable to future extreme-scale environments, including heterogeneous clusters that may incorporate wafer-scale and PIM-based accelerators.

6.3.5 Description of GenAI-XEnergy

In GenAI-XEnergy, a telemetry layer continuously collects real-time GPU, memory, thermal, and workload signals. An energy–performance predictor then estimates per-job energy consumption and latency, including token-level and phase-level dynamics. These predictions are exposed to the scheduler, which optimizes placement, scaling, and resource allocation across nodes while balancing energy, latency, utilization, and carbon objectives. A runtime controller enforces lower-level actions such as DVFS, power capping, and in-job adaptation. Finally, the hardware and architectural layer provides specialization opportunities, including PIM, wafer-scale execution, and reconfigurable accelerator resources. The framework closes the loop by feeding execution outcomes back into telemetry and prediction, thereby enabling continuous energy-aware adaptation rather than one-time static optimization.

6.4 Open Challenges for Implementation

- Predictive models for extreme-scale LLMs: Token-level, phase-aware prediction models remain underexplored.
- Dynamic, scalable orchestration: Elastic GPU allocation and placement policies must operate efficiently across thousands of GPUs without excessive overhead.
- Hardware-software co-design at scale: Integrating PIM, WaferLLM, or CGLA with runtime scheduling requires robust abstractions and APIs.
- Carbon-aware cross-layer control: Energy optimization should incorporate grid-aware carbon metrics alongside energy efficiency, latency, and throughput objectives.

7 Research Directions

The reviewed literature indicates several realistic opportunities for advancing energy-efficient resource management of Generative AI workloads on HPC systems. The following directions are grounded in current trends, emerging system capabilities, and practical feasibility over the next three to five years.

1. Enhanced energy prediction for Generative AI workloads: Machine-learning-based prediction models [12], [18], including workload-aware and model-aware approaches, can improve the accuracy of energy and performance estimates. Future work should integrate multi-granularity signals such as GPU utilization, memory traffic, and inference request patterns to support predictive and adaptive scheduling that minimizes wasted energy without compromising performance.
2. Intelligent energy- and QoS-aware scheduling frameworks: Future schedulers must dynamically trade off energy, latency, and resource usage in heterogeneous GPU clusters by combining heuristic methods with deep reinforcement learning (DRL)

approaches such as STS [3], InEPS [20], and HARMONIC [19]. Incorporating real-time monitoring and model-specific characteristics will allow schedulers to make finer-grained allocation decisions for large-scale LLM training and inference.

3. Lightweight software–hardware co-design strategies: Co-design efforts such as Fire-Flyer [9] and CIDAN-3D [14] show that even modest integration of software scheduling and hardware capabilities can produce substantial energy savings without requiring a complete architectural redesign. Extending these ideas to DVFS, power capping, and memory-aware placement could yield more robust, scalable, and energy-conscious HPC systems.
4. Carbon- and energy-aware optimization: Scheduler policies can incorporate both carbon and energy cost objectives [4], [5], [16] to enable more sustainable HPC operation. Practical implementations may include workload routing based on energy intensity, multi-horizon forecasting of grid carbon profiles, and adaptive quality-of-service control, enabling HPC clusters to reduce operational carbon emissions while maintaining throughput and latency objectives.

These directions collectively reinforce the central message of this review: future progress depends on cross-layer integration that combines predictive models, scheduling intelligence, runtime adaptability, and hardware capabilities to achieve substantial energy and carbon efficiency.

8 Conclusion

Generative AI workloads are redefining the energy-management problem in HPC systems. Their extreme scale, high memory and communication demand, long training horizons, and bursty inference behavior expose the limitations of resource-management strategies that optimize only one layer of the system stack. Across the 46 studies synthesized in this review, a consistent conclusion emerges: local improvements in hardware control, runtime adaptation, or scheduler design can reduce energy consumption in specific settings, but they are rarely sufficient to achieve robust, system-wide, energy-proportional operation for large-scale GenAI execution.

This review has shown that the most promising direction lies in coordinated cross-layer control. The surveyed literature demonstrates meaningful progress in telemetry-driven measurement, energy-performance prediction, adaptive scheduling, runtime power control, and architectural specialization, including processing-in-memory and wafer-scale accelerators. At the same time, the analysis reveals persistent weaknesses that continue to limit practical deployment: fragmentation across hardware, runtime, scheduler, and cluster layers; limited support for token-level and phase-aware workload dynamics; and weak integration of carbon-aware objectives into mainstream resource-management policies.

To address these gaps, this paper synthesizes the literature into GenAI-XEnergy, a conceptual framework for telemetry-driven, closed-loop, and cross-layer energy management of GenAI workloads on HPC systems. The framework is not intended as an isolated architectural proposal, but as a unifying systems view derived from the evidence reviewed in this paper. It connects measurement, prediction, orchestration, runtime enforcement, and heterogeneous accelerator support within a single control

loop, thereby clarifying how future systems can move from reactive energy reduction to proactive and sustainable resource optimization.

The broader implication of this review is that energy-efficient GenAI execution will depend on tighter integration between system observability, predictive intelligence, scheduling policy, and hardware capability. Future HPC platforms will need to optimize not only performance and utilization, but also energy proportionality and, increasingly, carbon impact. We therefore hope this review serves as both a structured reference for the current state of the field and a practical foundation for future research on sustainable AI-HPC co-design.

References

[1] Latif, A.C., Carbone, M.R., Munir, A., Lin, Y., Koomey, J., Yu, X., Dong, Z.: Single-node power demand during ai training: Measurements on an 8-gpu nvidia h100 system. IEEE Access **13**, 61740–61747 (2024) https://doi.org/10.1109/ACCESS.2025.3554728

[2] Costa, G.D.: Hardware and application aware performance, power and energy models for modern hpc servers with dvfs. Sustainable Computing: Informatics and Systems **20**, 100950 (2025) https://doi.org/10.1016/j.suscom.2024.100950

[3] Filippini, F., Anselmi, J., Ardagna, D., Gaujal, B.: A stochastic approach for scheduling ai training jobs in gpu-based systems. IEEE Transactions on Cloud Computing (2024) https://doi.org/10.1109/TCC.2024.3361234 . early access

[4] Shim, H.: Quantifying job level carbon efficiency in hpc: an empirical study based on the pm100 dataset. Energy Informatics **8**, 136 (2025) https://doi.org/10.1186/s42162-025-00586-6

[5] Mompó, A.S., Mavromatis, I., Li, P., Katsaros, K., Khan, A.: Green mlops to green genops: An empirical study of energy consumption in discriminative and generative ai operations. Information **16**(4), 281 (2025) https://doi.org/10.3390/info16040281

[6] Wu, C.-J., Acun, B., Raghavendra, R., Hazelwood, K.: Beyond efficiency: Scaling ai sustainably. IEEE Micro **44**(5), 37–46 (2024) https://doi.org/10.1109/MM.2024.3409275

[7] Resch, M.M., Gebert, J., Hoppe, D.: The future of hpc and ai. In: Proc. 2025 International Conference on Intelligent Control, Computing and Communications (IC3), Mathura, India, pp. 897–904 (2025). https://doi.org/10.1109/IC363308.2025.10956516

[8] Kang, D.K., Lee, K.B., Kim, Y.C.: Cost efficient gpu cluster management for training and inference of deep learning. Energies **15**(2), 474 (2022) https://doi.org/10.3390/en15020474

[9] An, W., Bi, X., Chen, G., Chen, S., Deng, C., Ding, H., Dong, K., Du, Q., Gao, W., Guan, K., Guo, J., Guo, Y., Fu, Z., He, Y., Huang, P., Li, J., Liang, W., Liu, X., Liu, X., Liu, Y., Liu, Y., Lu, S., Lu, X., Nie, X., Pei, T., Qiu, J., Qu, H., Ren, Z., Sha, Z., Su, X., Sun, X., Tan, Y., Tang, M., Wang, S., Wang, Y., Wang, Y., Xie, Z., Xiong, Y., Xu, Y., Ye, S., Yu, S., Zha, Y., Zhang, L., Zhang, H., Zhang, M., Zhang, W., Zhang, Y., Zhao, C., Zhao, Y., Zhou, S., Zou, Y.: Fire flyer ai hpc: A cost effective software hardware co design for deep learning. In: Proc. Int. Conf. High Performance Computing, Networking, Storage and Analysis (SC) (2024). https://doi.org/10.1109/SC41406.2024.00089

[10] Coppock, P.H., Zhang, B., Solomon, E.H., Kypriotis, V., Yang, L., Sharma, B., Schatzberg, D., Mowry, T.C., Skarlatos, D.: Lithos: An operating system for efficient machine learning on gpus. In: Proc. 31st ACM Symposium on Operating Systems Principles (SOSP '25), Seoul, Republic of Korea, pp. 1–17 (2025). https://doi.org/10.1145/3731569.3764818

[11] Hogade, N., Pasricha, S.: Game theoretic deep reinforcement learning to minimize carbon emissions and energy costs for ai inference workloads in geo distributed data centers. IEEE Trans. Sustainable Comput. **10**(4), 628–641 (2025) https://doi.org/10.1109/TSUSC.2024.3520969

[12] Wang, H., Kannan, S., Gill, S.S., Uhlig, S.: Ainet0: Ai forecasting based carbon neutral cloud resource management for net zero targets. Trans. Emerging Telecommun. Technol. **36**(5), 70166 (2025) https://doi.org/10.1002/ett.70166

[13] Wiesner, P., Grinwald, D., Weiß, P.: Carbon aware quality adaptation for energy intensive services. In: Proc. ACM Int. Conf. Future and Sustainable Energy Systems (e-Energy) (2025). to be published

[14] Singh, G., Vrudhula, S.: A scalable and energy efficient processing in memory architecture for gen ai. IEEE Journal on Emerging and Selected Topics in Circuits and Systems (2025). to be published

[15] Kaneko, Y.: Comparing electricity consumption per use of blockchain and generative ai. IEEE Access **13**, 93775–93785 (2025) https://doi.org/10.1109/ACCESS.2025.3573722

[16] Stojkovic, J., Zhang, C., Goiri, Í., Torrellas, J., Choukse, E.: Dynamollm: Designing llm inference clusters for performance and energy efficiency. arXiv preprint arXiv:2408.00741 (2024). Available: https://arxiv.org/abs/2408.00741

[17] Zhao, D., Li, B., Samsi, S.S., Bestor, D., Tiwari, D., McDonald, J., Jones, M., Gadepally, V.: Sustainable supercomputing for ai: Gpu power capping at hpc scale. In: Proceedings of the 2023 ACM Symposium on Cloud Computing (SoCC '23), pp. 588–601. ACM, Santa Cruz, CA, USA (2023). https://doi.org/10.1145/3620678.3624793 . https://doi.org/10.1145/3620678.3624793

[18] Menear, K., Wilkinson, A., Dykes, T., Haus, U.-U., Duplyakin, D.: Energy-aware hpc scheduling with llm-based power prediction. In: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC Workshops '25), pp. 16–25. ACM, St. Louis, MO, USA (2025). https://doi.org/10.1145/3731599.3767563 . https://doi.org/10.1145/3731599.3767563

[19] Adimora, K.C., Sun, H.: Harmonic: Uncertainty-aware multi-objective optimization for energy-efficient hpc resource management. IEEE Transactions on Parallel and Distributed Systems **36**(11), 1–15 (2025) https://doi.org/10.1109/TPDS.2025.3610354

[20] López, M., Stafford, E., Bosque, J.L.: Intelligent energy pairing scheduler (ineps) for heterogeneous hpc clusters. The Journal of Supercomputing **81**, 427–449 (2025) https://doi.org/10.1007/s11227-024-06907-y

[21] Hautreux, G., Gamatié, A., Sassatelli, G.: Run-time energy-efficiency optimization for ai and hpc workloads. In: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC Workshops '25), pp. 1–10. ACM, St. Louis, MO, USA (2025). https://doi.org/10.1145/3731599.3767560 . https://doi.org/10.1145/3731599.3767560

[22] Budiarjo, T., Pradata, S.Y., Santiyuda, K.G., Amrizal, M.A., Pulungan, R., Takizawa, H.: Improving the efficiency of a deep reinforcement learning-based power management system for hpc clusters using curriculum learning. In: Proceedings of the 2025 Supercomputing Asia Conference (SCA '25), pp. 1–13. ACM, ??? (2025). https://doi.org/10.1145/3718350.3718359 . https://doi.org/10.1145/3718350.3718359

[23] Patel, P., Choukse, E., Zhang, C., Shah, A., Goiri, Í., Maleki, S.: Splitwise: Efficient generative llm inference using phase splitting. In: Proceedings of the 2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA), pp. 1–15. IEEE, Buenos Aires, Argentina (2024). https://doi.org/10.1109/ISCA59077.2024.00019 . https://doi.org/10.1109/ISCA59077.2024.00019

[24] Wilkins, G., Keshav, S., Mortier, R.: Offline energy-optimal llm serving: Workload-based energy models for llm inference on heterogeneous systems. ACM SIGENERGY Energy Informatics Review **4**(5), 113–119 (2025) https://doi.org/10.1145/3727200.3727217

[25] Chien, A.A., Lin, L., Nguyen, H.D., Rao, V., Sharma, T., Wijayawardana, R.: Reducing the carbon impact of generative ai inference (today and in 2035). In: Proceedings of the 2nd Workshop on Sustainable Computer Systems (HotCarbon '23), pp. 1–7. ACM, Boston, MA, USA (2023). https://doi.org/10.1145/3604930.3605705 . https://doi.org/10.1145/3604930.3605705

[26] Zhang, C., You, Y., Wang, N., Zhang, L., Park, J.: Generative ai through cas lens: An integrated overview of algorithmic optimizations, architectural advances,

and automated designs. IEEE Journal on Emerging and Selected Topics in Circuits and Systems **15**(2), 149–150 (2025) https://doi.org/10.1109/JETCAS.2025.3575272

[27] Carastan-Santos, D., Costa, G.D., Nardin, I.F., Rzadca, K., Stolf, P., Poquet, M., Trystram, D.: Scheduling with lightweight predictions in power-constrained hpc platforms. IEEE Transactions on Parallel and Distributed Systems **36**(12), 2609–2610 (2025) https://doi.org/10.1109/TPDS.2025.3586723

[28] Zeng, L., Zhang, G., Peng, J., Xu, X., Xu, Y., Ma, L.: Kant: An efficient unified scheduling system for large-scale ai clusters. arXiv preprint arXiv:2510.01256 (2025)

[29] Escobar, J.J., Sánchez-Cuevas, P., Prieto, B., Kızıltepe, R.S., Díaz-del-Río, F., Kimovski, D.: Energy–time modelling of distributed multi-population genetic algorithms with dynamic workload in hpc clusters. Future Generation Computer Systems **167**, 107753–107768 (2025) https://doi.org/10.1016/j.future.2025.107753

[30] Amrizal, M.A., Prasasta, R.S., Pradata, S.Y., Santiyuda, K.G., Pulungan, R., Takizawa, H.: Spars: A reinforcement learning-enabled simulator for power management in hpc job scheduling. arXiv preprint arXiv:2512.13268 (2025)

[31] He, X., Fang, Z., Lian, J., Tsang, D.H.K., Zhang, B., Chen, Y.: Freesh: Fair, resource- and energy-efficient scheduling for llm serving on heterogeneous gpus. arXiv Preprint **arXiv:2511.00807** (2025). Available: https://arxiv.org/abs/2511.00807

[32] Mamirov, A.: Reducing fragmentation and starvation in gpu clusters through dynamic multi-objective scheduling. arXiv Preprint **arXiv:2512.10980** (2025). Available: https://arxiv.org/abs/2512.10980

[33] Hossain, A., Abdurahman, A., Islam, M.A., Ahmed, K.: Power-aware scheduling for multi-center hpc electricity cost optimization. arXiv Preprint **arXiv:2503.11011** (2025). Available: https://arxiv.org/abs/2503.11011

[34] Kausar, Z., Latif, S., Shahzad, R.K., Fatima, M.: Quantifying the climate risk of generative ai: Region-aware carbon accounting with g-trace and the ai sustainability pyramid. arXiv Preprint **arXiv:2511.04776** (2025). Available: https://arxiv.org/abs/2511.04776

[35] Li, P., Islam, M.J., Ren, S.: A case study of environmental footprints for generative ai inference: Cloud versus edge. ACM SIGMETRICS Performance Evaluation Review **53**(2), 21–26 (2025) https://doi.org/10.1145/3764944.3764950

[36] Fu, T., Yang, Z., Ye, Z., Ma, C., Han, Y., Luo, Y., Wang, X., Wang, Z.: A survey on the scheduling of dl and llm training jobs in gpu clusters. Chinese Journal of

Electronics **34**(3), 881–905 (2025) https://doi.org/10.23919/cje.2024.00.070

[37] Narayanaswamy, S., Gupta, A., Patel, P., Saripalli, S., Karlin, I., Guo, J.: Datacenter energy optimized power profiles. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis Workshops, Santa Clara, CA, USA (2024). NVIDIA Corporation. Blackwell B200 GPU power management and Mission Control power profiles

[38] Quach, R., Wang, Y., Jahanshahi, A., Wong, D., Kim, H.: Eclip: Energy-efficient and practical co-location of ml inference on spatially partitioned gpus. arXiv preprint arXiv:2506.12598 (2025). eess.SY

[39] Xue, Y., Huang, J.: Regate: Enabling power gating in neural processing units. In: Proceedings of the 58th IEEE/ACM International Symposium on Microarchitecture (MICRO '25), pp. 1–18. ACM, Seoul, Republic of Korea (2025). https://doi.org/10.1145/3725843.3756038 . Open Access. CC BY 4.0 License. https://doi.org/10.1145/3725843.3756038

[40] Kocot, B., Czarnul, P., Proficz, J.: Energy-aware scheduling for high-performance computing systems: A survey. Energies **16**(2), 890 (2023) https://doi.org/10.3390/en16020890

[41] Gholap, P., Pawar, K., Patil, V.: Comparative analysis on developed optimization techniques for reducing energy consumption in AI training and inference. Journal of Information Systems Engineering and Management **10**(14s) (2025)

[42] Samsi, S., Zhao, D., McDonald, J., Li, B., Michaleas, A., Jones, M., Bergeron, W., Kepner, J., Tiwari, D., Gadepally, V.: From words to watts: Benchmarking the energy costs of large language model inference. In: 2023 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–10. IEEE, ??? (2023). https://doi.org/10.1109/HPEC58863.2023.10363447

[43] Udkar, S., Vaithianathan, M., Reddy, M., Gupta, V.: Balancing performance and power efficiency in modern gpu architecture. In: Proceedings of the 3rd International Conference on Futuristic Technology (INCOFT 2025), Volume 2, pp. 105–112. SCITEPRESS – Science and Technology Publications, Lda., Lisbon, Portugal (2025). https://doi.org/10.5220/0013587900004664 . Paper published under CC license (CC BY-NC-ND 4.0)

[44] Ando, T., Eto, Y., Takeuchi, A., Nakashima, Y.: Efficient kernel mapping and comprehensive system evaluation of llm acceleration on a cgla. IEEE Access **13** (2025) https://doi.org/10.1109/ACCESS.2025.3636266 . Received 8 October 2025, accepted 19 November 2025. arXiv:2512.00335v1

[45] Ozkan, M., Gao, L., Pompa, L., Iqbal, M.S.B., Gonzalez, S.H., Chan, Y., Morales, D., Chen, Z.: Performance, efficiency, and cost analysis of wafer-scale ai accelerators vs. single-chip gpus. Device **3**, 100834 (2025) https://doi.org/10.1016/

j.device.2025.100834 . Published October 17, 2025. Open access under CC BY license

[46] He, C., Huang, Y., Mu, P., Miao, Z., Xue, J., Ma, L., Yang, F., Mai, L.: Waferllm: Large language model inference at wafer scale. arXiv preprint arXiv:2502.04563v3 (2025). Available: https://arxiv.org/abs/2502.04563