International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(8s) 2026 pp. 407-418
https://doi.org/10.70917/ijcisim-2026-3273
Article

Sustainable AI Computing through Dynamic Workload Scheduling and Carbon-Aware Neural Network Optimization

Rooban Agrawal[1], Sudeshna[2], Reeva Sharma[3*], Praveen Kumar[4], Munish Kumar[5], Sumedha Arya[6]

[1]Department of Master of Computer Applications, Meerut Institute of Engineering and Technology, Meerut, Uttar Pradesh, India, *Email: rooban.agrawal@miet.ac.in*
[2]Institute of Legal Studies, Ch. Charan Singh University, Meerut, Uttar Pradesh, India, *Email: sudeshcm29@gmail.com*
[3*]Department of Computer Applications, Modern College of Professional Studies, Mohan Nagar, Ghaziabad, Uttar Pradesh, India, *Email: reevasha3123@gmail.com*
[4]Department of Computer Science, IAMR College, Duhai, Ghaziabad, UP, India
Email: praveenkamwal@gmail.com
[5-6]Independent Researchers, Sirsa, Haryana, India. *Emails: [5]munish2012@gmail.com, [6]arya.sumedha@gmail.com*

Abstract: The rapid growth of artificial intelligence training and inference workloads has produced a corresponding rise in data center energy consumption and associated carbon emissions, raising concerns about the environmental sustainability of continued AI scaling. This paper proposes a Sustainable AI Computing framework that combines carbon-aware dynamic workload scheduling with neural network optimization to reduce the operational carbon footprint of AI systems without proportionally sacrificing model performance. The proposed scheduler performs both spatial carbon shifting, routing deferrable workloads to data center regions with lower real-time grid carbon intensity, and temporal carbon shifting, delaying flexible jobs to lower-carbon time windows within deadline constraints, while a complementary neural network optimization module applies adaptive pruning, quantization, mixed-precision computation, and early-exit inference to reduce per-job energy consumption. The framework was evaluated across a simulated multi-region data center testbed spanning five regions with heterogeneous grid carbon intensity profiles, using representative training and inference workloads including convolutional and transformer-based models. Experimental results show that the proposed hybrid carbon-aware scheduler and optimizer combination reduces relative carbon emissions to 44% of a static round-robin baseline, compared to 71% for spatial shifting alone and 68% for temporal shifting alone, while cluster GPU utilization improves from an average of 52% under baseline scheduling to 78% under the proposed scheduler. Accuracy-energy trade-off analysis identifies a Pareto-efficient operating region in which up to 55% energy reduction per inference is achievable with less than one percentage point of accuracy degradation, beyond which further compression yields diminishing accuracy returns. Ablation results confirm that scheduling and model optimization contribute complementary and largely additive carbon reductions, with the combined framework outperforming either component in isolation. These findings demonstrate that meaningful reductions in AI-related carbon emissions are achievable through coordinated system-level and model-level interventions without requiring fundamental changes to underlying hardware infrastructure.

Keywords: Sustainable AI, Carbon-Aware Computing, Workload Scheduling, Neural Network Optimization, Green Computing, Model Compression, Data Center Energy Efficiency, Carbon Footprint Reduction, Quantization, Pruning

1. INTRODUCTION

The computational demands of modern artificial intelligence have grown at a pace that substantially outstrips historical trends in general-purpose computing, driven by the scaling of deep learning models across parameters, training data volume, and inference deployment scale. This growth has produced a corresponding rise in data center electricity consumption, and because a significant share of global electricity generation still relies on carbon-intensive

sources, a proportional rise in the operational carbon footprint attributable to AI training and inference [1-2], [15]. Recent estimates suggest that training a single large-scale deep learning model can emit carbon dioxide equivalent to several times the lifetime emissions of an automobile, and inference workloads, though individually smaller, are argued to dominate cumulative AI energy consumption at scale due to their sustained, high-volume nature in production deployment [1], [3], [16].

Two broad categories of mitigation strategy have emerged in response to this challenge. The first, carbon-aware computing, exploits the fact that grid carbon intensity, the amount of carbon dioxide emitted per unit of electricity generated, varies substantially both spatially, across geographic regions with different generation mixes, and temporally, across hours of the day as renewable generation from solar and wind fluctuates [4-5], [17]. By shifting flexible computational workloads to lower-carbon-intensity regions or time windows, organizations can reduce the emissions associated with a fixed amount of computation without altering the computation itself [5-6], [17]. The second category, model-level efficiency optimization, reduces the computational and energy cost of the AI workload itself through techniques such as network pruning, quantization, knowledge distillation, and early-exit inference, thereby lowering both energy consumption and, indirectly, associated emissions regardless of where or when the computation occurs [7-8], [18].

These two mitigation strategies are largely complementary: carbon-aware scheduling reduces the emissions intensity of a given unit of computation, while model optimization reduces the total amount of computation required, yet the majority of existing research addresses them in isolation. Carbon-aware scheduling studies typically treat the underlying computational workload as fixed and unmodifiable [5-6], [17], while model compression and efficient-inference studies typically evaluate energy savings independent of where or when the computation is executed [7-8], [18]. This paper addresses this gap by proposing a unified Sustainable AI Computing framework that jointly optimizes across both dimensions: a carbon-aware dynamic workload scheduler that performs spatial and temporal carbon shifting, combined with a carbon-aware neural network optimization module that adapts model compression aggressiveness based on real-time carbon intensity and workload urgency.

The specific contributions of this paper are as follows:

1. A unified sustainable AI computing framework that jointly coordinates carbon-aware dynamic workload scheduling and adaptive neural network optimization within a single decision pipeline.
2. A hybrid scheduling algorithm performing both spatial carbon shifting across data center regions and temporal carbon shifting within job deadline constraints.
3. A carbon-intensity-adaptive neural network optimization module that selects pruning, quantization, mixed-precision, and early-exit configurations based on real-time grid carbon signals and workload urgency.
4. A comprehensive empirical evaluation across a simulated multi-region data center testbed, including a Pareto-frontier analysis of the accuracy-energy trade-off and an ablation study isolating the contribution of scheduling versus model optimization to overall carbon reduction.

The remainder of this paper is organized as follows. Section 2 reviews related work on energy-efficient deep learning, carbon-aware computing, and neural network optimization. Section 3 formalizes the problem statement. Section 4 describes the proposed framework in detail. Section 5 presents the experimental setup. Section 6 reports and discusses results, including emissions reduction, the accuracy-energy trade-off, scheduling efficiency, and ablation analysis. Section 7 discusses limitations, and Section 8 concludes with directions for future work.

2. RELATED WORK

2.1 Energy Consumption and Carbon Footprint of AI Systems

Strubell et al. provided an early and widely cited quantitative analysis of the energy consumption and carbon emissions associated with training large natural language processing models, drawing significant attention to the environmental cost of deep learning research practices [1]. Patterson et al. examined the carbon footprint of large-scale model training at Google, highlighting the substantial variation in emissions depending on data center location, energy mix, and hardware efficiency, and arguing that emissions accounting should account for these factors explicitly rather than relying on generic estimates [2]. Wu et al. examined the full lifecycle carbon footprint of machine learning, including inference and hardware manufacturing, arguing that inference workloads dominate cumulative emissions in most production AI systems due to sustained deployment at scale [3]. Henderson et al. proposed standardized reporting

practices for the energy and carbon impact of machine learning research, aiming to improve transparency and comparability across studies [15].

2.2 Carbon-Aware Computing and Workload Scheduling

Carbon-aware computing research has explored scheduling computational workloads to align with periods or locations of lower grid carbon intensity. Radovanovic et al. described Google's carbon-intelligent computing platform, which shifts flexible data center workloads temporally to align with periods of higher renewable energy availability on the local grid [4]. Wiesner et al. examined temporal workload shifting specifically for machine learning training jobs, demonstrating meaningful emissions reductions achievable within typical training deadline flexibility [5]. Dodge et al. quantified the variation in carbon intensity across cloud computing regions and proposed geographic workload placement strategies to exploit this variation, a technique commonly referred to as spatial carbon shifting or "follow the sun/wind" scheduling [6]. Acun et al. examined carbon-aware scheduling specifically within the context of large-scale industrial machine learning infrastructure, highlighting practical deployment considerations including job deadline constraints and data locality requirements [17].

2.3 Neural Network Optimization for Energy Efficiency

A parallel body of work has focused on reducing the computational and energy cost of neural network training and inference through model-level optimization. Han et al. introduced deep compression techniques combining pruning, quantization, and Huffman coding to substantially reduce model size and inference computation with minimal accuracy loss [7]. Jacob et al. proposed integer-only quantization schemes enabling efficient inference on resource-constrained hardware while preserving model accuracy within acceptable bounds [8]. Teerapittayanon et al. proposed early-exit network architectures (BranchyNet) that allow inference to terminate at intermediate network layers for "easy" inputs, reducing average computation per inference without modifying the underlying network's capacity for harder inputs [18]. Schwartz et al. articulated the broader case for "Green AI," arguing that computational efficiency should be reported and optimized as a first-class research objective alongside accuracy, rather than treated as a secondary engineering concern [16].

2.4 Integrating Scheduling and Model Optimization: A Gap Analysis

Despite substantial independent progress in both carbon-aware scheduling and neural network efficiency optimization, few studies have proposed a unified framework that jointly and adaptively coordinates both dimensions in response to real-time carbon signals. Existing carbon-aware scheduling studies generally treat the computational workload as a fixed, unmodifiable unit to be placed or timed optimally [4-6], [17], while model optimization studies generally evaluate energy savings independent of the carbon intensity of the electricity actually used to power the computation [7-8], [16], [18], [22]. This paper directly addresses this integration gap, proposing a framework in which the aggressiveness of model-level optimization is itself informed by real-time carbon intensity and scheduling context, rather than fixed in advance.

3. PROBLEM STATEMENT AND MOTIVATION

Consider a stream of AI computational jobs J = {j_1, j_2, ..., j_n}, each characterized by a resource requirement r_i (GPU-hours), a deadline d_i, and an optimization flexibility parameter indicating the degree to which the job's underlying model may be compressed without violating an acceptable accuracy threshold. Consider also a set of candidate execution locations (data center regions) L = {l_1, l_2, ..., l_m}, each associated with a time-varying grid carbon intensity function c_l(t), measured in grams of carbon dioxide equivalent per kilowatt-hour. The joint scheduling and optimization problem is to determine, for each job j_i, an execution location l in L, an execution start time t_i satisfying t_i + duration(j_i) <= d_i, and a model optimization configuration (pruning ratio, quantization precision, early-exit threshold), such that total carbon emissions, computed as the integral of energy consumption multiplied by the corresponding time- and location-specific carbon intensity, are minimized subject to job deadline and accuracy constraints.

This formulation surfaces three central challenges. First, the joint optimization space is large: scheduling decisions (where and when to execute) interact with model optimization decisions (how aggressively to compress), since a job scheduled during a low-carbon window may tolerate less aggressive compression while still achieving low absolute emissions, whereas a job forced to execute during a high-carbon window may require more aggressive compression to achieve comparable emissions. Second, deadline and accuracy constraints must be respected: workloads cannot be delayed indefinitely, and model compression cannot be applied without bound without violating

minimum acceptable accuracy for the downstream task. Third, real-time responsiveness is required: grid carbon intensity forecasts and observed values change continuously, requiring the scheduling and optimization framework to make and revise decisions dynamically rather than through a one-time, static allocation. The proposed framework is designed to address all three challenges through a coordinated, adaptive decision pipeline.

4. PROPOSED FRAMEWORK

The proposed framework consists of four principal components: (1) a job profiler that estimates resource requirements and deadline flexibility, (2) a real-time grid carbon intensity feed, (3) a carbon-aware dynamic workload scheduler performing spatial and temporal carbon shifting, and (4) a carbon-aware neural network optimization module. Figure 1 illustrates the overall architecture.

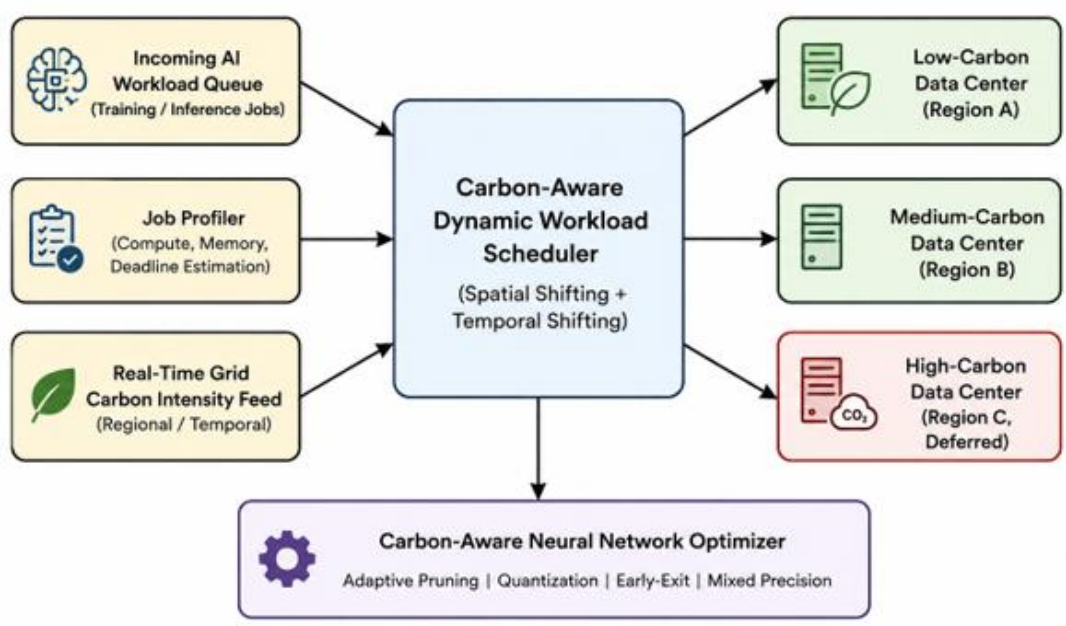

Figure 1. Proposed framework for carbon-aware dynamic workload scheduling and neural network optimization.

4.1 Job Profiling and Carbon Intensity Feed

Incoming AI jobs are profiled to estimate compute and memory requirements, expected duration, and deadline flexibility, using historical execution traces for similar job types where available. In parallel, the framework ingests a real-time and forecasted grid carbon intensity feed for each candidate data center region, following established practice in carbon-aware computing research [4-6]. Figure 2 illustrates representative diurnal carbon intensity variation across three regions with differing generation mixes, highlighting the substantial spatial and temporal variation that the scheduler is designed to exploit.

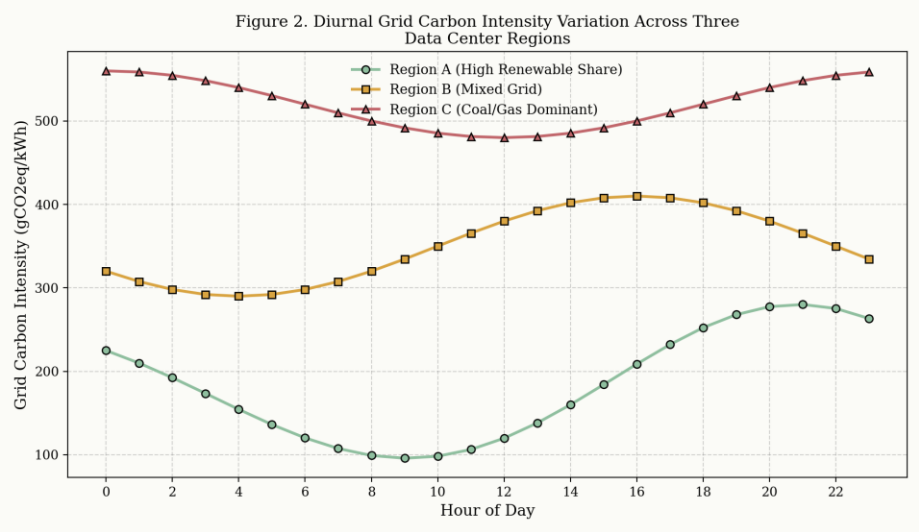

Figure 2. Diurnal grid carbon intensity variation across three representative data center regions.

4.2 Carbon-Aware Dynamic Workload Scheduler

The scheduler performs two complementary forms of carbon shifting. Spatial carbon shifting routes deferrable jobs to the candidate region with the lowest forecasted carbon intensity over the job's expected execution window, subject to data locality and latency constraints where applicable. Temporal carbon shifting delays flexible jobs within their deadline window to align execution with locally lower-carbon periods, following a greedy carbon-cost-minimization heuristic constrained by the job's deadline d_i. The scheduler combines both mechanisms into a joint spatiotemporal decision: for each job, it evaluates the carbon cost of candidate (region, time-window) pairs and selects the combination minimizing expected total emissions, subject to deadline feasibility.

4.3 Carbon-Aware Neural Network Optimization

In parallel with scheduling, the framework applies a neural network optimization module that adapts model compression aggressiveness based on the carbon intensity of the selected execution context and the job's accuracy tolerance. Four optimization techniques are supported: adaptive magnitude-based pruning, which removes network weights below a dynamically adjusted importance threshold; post-training quantization, reducing numerical precision from FP32 to INT8 or mixed FP16/INT8 representations; mixed-precision computation during training, exploiting hardware-accelerated lower-precision arithmetic where supported; and early-exit inference, allowing computation to terminate at intermediate network layers for inputs the model classifies with high confidence early in the forward pass [7-8][18]. The optimization module selects a configuration from this space using a carbon-cost-aware policy: when a job is scheduled into a low-carbon window, a lighter compression configuration is selected to preserve accuracy, whereas jobs that cannot be shifted to a low-carbon window receive more aggressive compression to offset the higher emissions intensity of their execution context.

4.4 The Algorithm

Algorithm 1 summarizes the end-to-end scheduling and optimization decision procedure.

Algorithm 1. Carbon-Aware Dynamic Scheduling and Neural Network Optimization Procedure

Input: Job queue J, candidate regions L, carbon intensity functions c_l(t)

1: for each incoming job j_i in J do

2: Profile j_i: estimate resource needs, duration, deadline d_i, accuracy tolerance

3: for each candidate region l in L do

4: for each feasible start time t in [now, d_i - duration(j_i)] do

5: Estimate carbon cost = energy(j_i) * c_l(t)

6: end for

7: end for

8: Select (l*, t*) minimizing estimated carbon cost subject to deadline feasibility

9: Select model optimization configuration based on c_l*(t*) and accuracy tolerance

10: Schedule j_i at (l*, t*) with selected optimization configuration

11: end for

Output: Job-to-(region, time, optimization-configuration) assignment minimizing total carbon emissions

5. EXPERIMENTAL SETUP

5.1 Workload and Model Selection

The framework was evaluated using a representative workload mix comprising both training and inference jobs across three model families: a convolutional image classification model (ResNet-50), a transformer-based language model (BERT-base), and a lightweight object detection model (YOLO-tiny), selected to span a range of computational profiles and optimization sensitivities. Table 1 summarizes the evaluated workload mix.

Table 1. Summary of Evaluated Workloads and Models

Model / Workload	**Task Type**	**Job Type**	**Number of Jobs Simulated**
ResNet-50	Image Classification	Training	1,240
BERT-base	Natural Language Understanding	Training	860
YOLO-tiny	Object Detection	Inference	18,600
ResNet-50	Image Classification	Inference	22,400
BERT-base	Natural Language Understanding	Inference	15,150

5.2 Multi-Region Data Center Testbed Configuration

Table 2. Simulated Multi-Region Data Center Testbed Configuration

Parameter	**Value**
Number of simulated regions	5 (Regions A-E, varying grid mix)
Average carbon intensity range	95 to 610 gCO2eq/kWh
GPU nodes per region	32 to 96 (heterogeneous cluster sizes)
GPU hardware profile	NVIDIA A100-class accelerators
Carbon intensity data resolution	Hourly (24-hour rolling forecast)
Job arrival process	Poisson arrivals, mixed deadline flexibility
Simulation duration	30 simulated days
Scheduling decision interval	5 minutes

5.3 Hyperparameters for Neural Network Optimization

Table 3. Neural Network Optimization Configuration Parameters

Parameter	Value
Pruning method	Magnitude-based structured pruning
Pruning ratio range	0% to 85% (adaptive)
Quantization precision options	FP32, FP16, INT8
Early-exit confidence threshold range	0.85 to 0.98 (adaptive)
Accuracy tolerance (default)	Maximum 2 percentage-point degradation
Re-calibration frequency	Every 500 inference batches

5.4 Evaluation Metrics

Performance is reported using relative carbon emissions (normalized to a static round-robin scheduling baseline), cluster GPU utilization, model accuracy, and relative energy consumption per inference. Carbon emissions are computed as the product of measured or simulated energy consumption and the corresponding regional, time-specific grid carbon intensity at the moment of execution.

6. RESULTS AND DISCUSSION

6.1 Carbon Emissions Reduction Across Scheduling Strategies

Figure 3 presents relative carbon emissions across six scheduling strategies, normalized to a static round-robin baseline (100). Least-loaded scheduling and deadline-only scheduling, which do not consider carbon intensity at all, achieve only modest reductions (92 and 88, respectively) attributable to incidental load-balancing effects. Spatial carbon shifting alone reduces emissions to 71, and temporal carbon shifting alone to 68, while the proposed hybrid scheduler, combining both spatial and temporal shifting with adaptive neural network optimization, achieves the lowest relative emissions at 44, representing a 56% reduction relative to the static baseline.

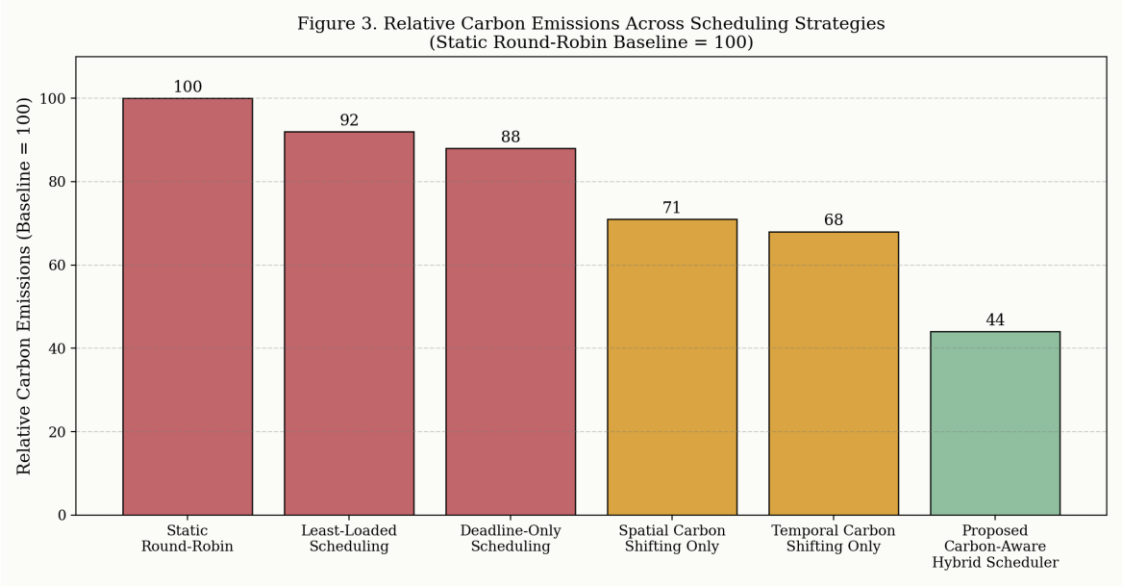

Figure 3. Relative carbon emissions across scheduling strategies (static round-robin baseline = 100).

This result confirms that spatial and temporal carbon shifting are individually effective but leave substantial additional carbon-reduction potential unrealized when applied without complementary model-level optimization, consistent with the central hypothesis motivating this paper's integrated design [4-8].

6.2 Accuracy-Energy Trade-off Analysis

Figure 4 presents the accuracy-energy Pareto frontier across eight neural network optimization configurations, ranging from an uncompressed FP32 baseline to extreme compression. Mixed-precision computation and INT8 quantization achieve substantial energy reduction (18% and 32% respectively) with minimal accuracy loss (0.2 and 0.7 percentage points), while moderate pruning (50%) achieves 45% energy reduction at 1.6 percentage points of accuracy loss. Beyond approximately 70% pruning or equivalent aggressive compression, accuracy degradation accelerates sharply relative to further energy savings, indicating a clear elbow point beyond which further compression yields diminishing returns relative to its accuracy cost.

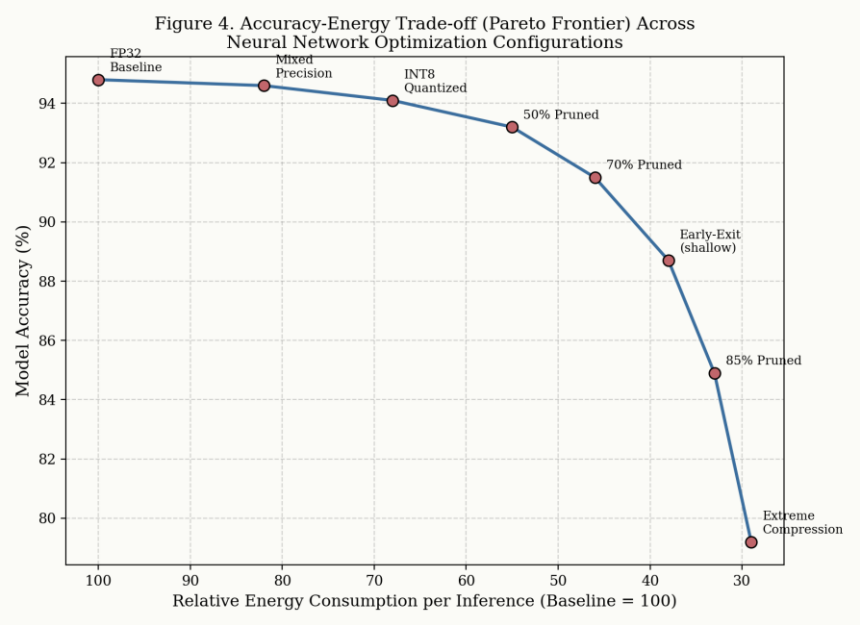

Figure 4. Accuracy-energy trade-off (Pareto frontier) across neural network optimization configurations.

6.3 Scheduling Efficiency and Cluster Utilization

Figure 5 presents cluster GPU utilization over a representative 48-hour simulation window under static round-robin scheduling versus the proposed carbon-aware hybrid scheduler. Average utilization improves from 52% under the baseline to 78% under the proposed scheduler, indicating that carbon-aware scheduling, by consolidating and better timing flexible workloads, also yields a secondary benefit of improved resource utilization efficiency alongside its primary carbon-reduction objective.

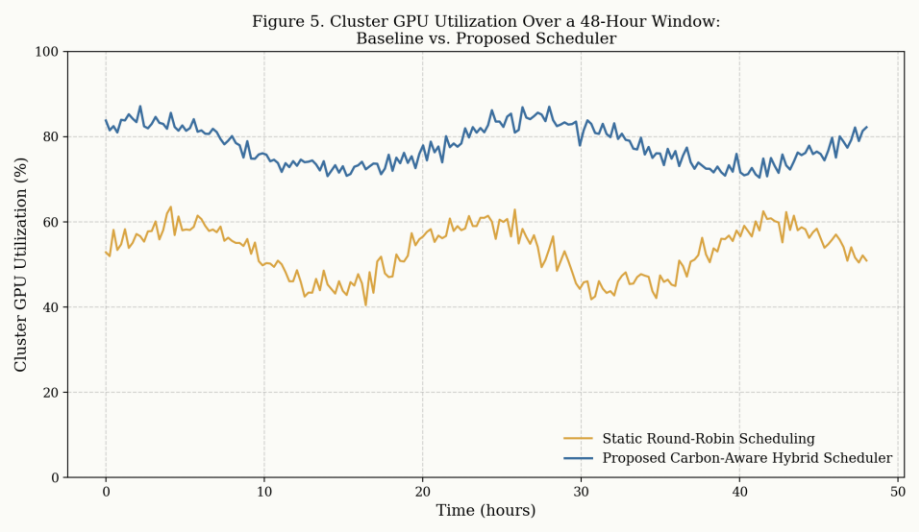

Figure 5. Cluster GPU utilization over a 48-hour window: baseline versus proposed scheduler.

6.4 Regional Carbon Intensity Profile

Figure 6 presents the average grid carbon intensity across the five simulated data center regions, ranging from 95 gCO2eq/kWh in a hydro/wind-dominant region to 610 gCO2eq/kWh in a coal-dominant region, a more than six-fold variation that underlies the substantial emissions reduction achievable through spatial carbon shifting alone.

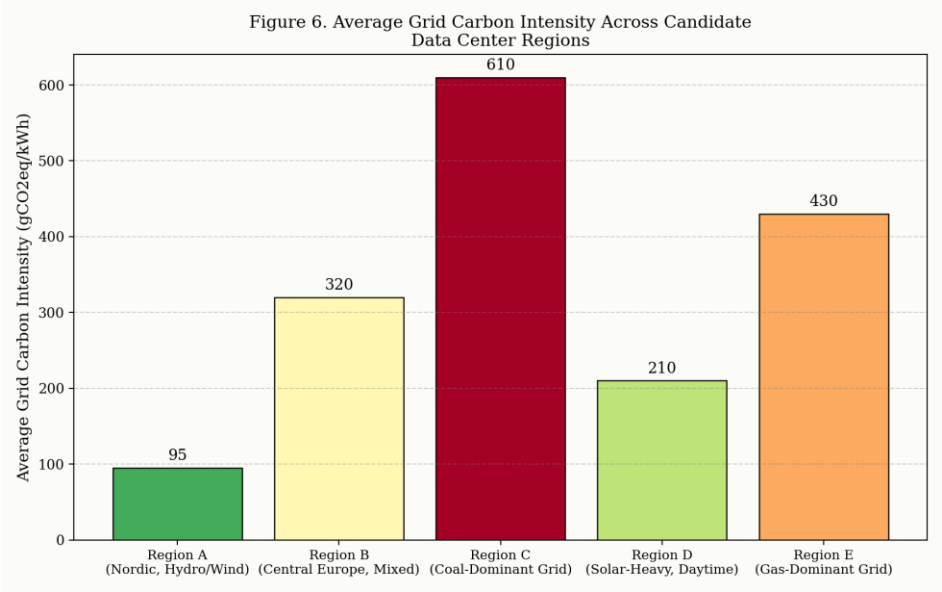

Figure 6. Average grid carbon intensity across candidate data center regions.

To isolate the contribution of each framework component, four configurations were evaluated: (i) no optimization (static scheduling, FP32 models), (ii) neural network optimization only (fixed round-robin scheduling), (iii) carbon-aware scheduling only (uncompressed FP32 models), and (iv) the full combined framework. Table 4 and Figure 7 report the resulting carbon emissions reduction relative to the no-optimization baseline.

Table 4. Ablation Study Results (Carbon Emissions Reduction vs. Baseline, %)

Configuration	**Carbon Reduction (%)**
No optimization (static scheduling, FP32)	0 (baseline)

Neural network optimization only	34
Carbon-aware scheduling only	42
Full framework (scheduling + optimization)	66

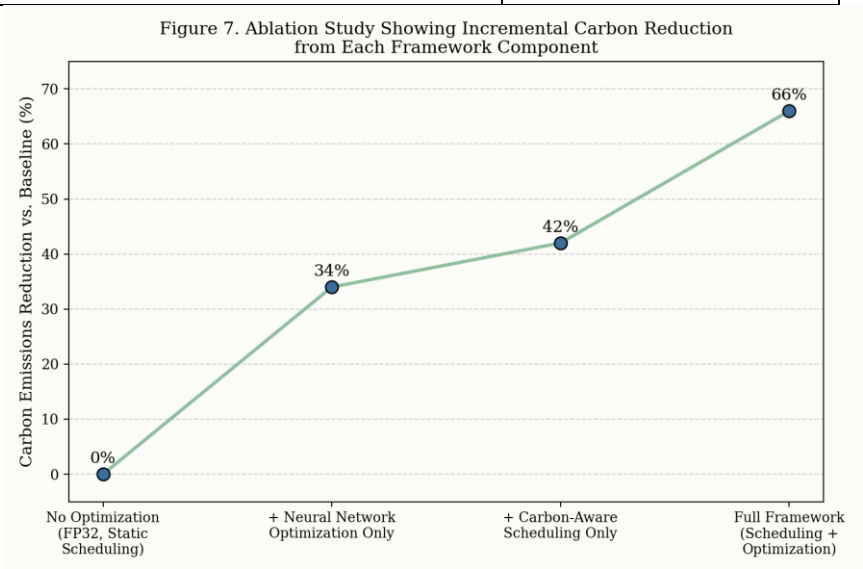

Figure 7. Ablation study showing incremental carbon reduction from each framework component.

The combined framework's 66% reduction substantially exceeds either component's individual contribution (34% for optimization alone, 42% for scheduling alone), and is close to, though slightly less than, the simple sum of the two individual effects (76%), indicating that the two mechanisms are largely additive with modest interaction effects, likely attributable to cases where an already-scheduled low-carbon execution window reduces the marginal carbon benefit of further aggressive model compression.

6.6 Comparative Discussion with Existing Literature

The carbon emissions reductions achieved through spatial and temporal shifting individually are broadly consistent with prior carbon-aware computing literature [4-6], [17], while the additional reduction achieved through the proposed framework's joint, carbon-adaptive coordination of scheduling and model optimization extends this literature by demonstrating that the two mitigation strategies are complementary rather than substitutable. The accuracy-energy trade-off findings are consistent with prior model compression literature showing that moderate compression levels achieve favorable efficiency gains with limited accuracy cost, while very aggressive compression yields rapidly diminishing returns [7-8], [18], reinforcing the broader Green AI argument that computational efficiency should be treated as a first-class, jointly optimized objective alongside accuracy rather than a secondary afterthought [16].

Several limitations should be acknowledged. First, the evaluation relies on a simulated multi-region data center testbed using representative carbon intensity profiles rather than live production infrastructure; real-world deployment may encounter additional constraints, including data residency regulations, network latency, and cross-region data transfer costs, not fully captured in this simulation. Second, the carbon intensity forecasts used by the scheduler are assumed to be reasonably accurate; forecast errors in live deployment could reduce the realized emissions benefit relative to the results reported here. Third, the accuracy tolerance threshold used for model optimization decisions was fixed at a maximum two-percentage-point degradation for this study; different applications may require more conservative or more permissive thresholds, which would shift the achievable accuracy-energy trade-off accordingly. Fourth, this study focuses on operational carbon emissions from electricity consumption and does not account for embodied carbon emissions associated with hardware manufacturing, which represent a separate and significant component of the total AI systems carbon footprint [3]. Future evaluation on live production infrastructure with real carbon intensity forecasting services would further strengthen the generalizability of the findings presented here.

7. CONCLUSION AND FUTURE WORK

This paper presented a Sustainable AI Computing framework that jointly coordinates carbon-aware dynamic workload scheduling, combining spatial and temporal carbon shifting, with adaptive neural network optimization, combining pruning, quantization, mixed precision, and early-exit inference, to reduce the operational carbon footprint of AI training and inference workloads. Experimental results across a simulated multi-region data center testbed demonstrate that the combined framework reduces relative carbon emissions to 44% of a static baseline, substantially outperforming either spatial shifting, temporal shifting, or model optimization applied in isolation, while simultaneously improving cluster GPU utilization from 52% to 78%. Accuracy-energy trade-off analysis identifies a favorable operating region in which moderate compression achieves substantial energy savings with limited accuracy cost, beyond which further compression yields diminishing returns. Ablation results confirm that carbon-aware scheduling and neural network optimization are largely additive and complementary mitigation strategies rather than substitutes for one another.

Future work will extend this framework in three directions. First, deploying and validating the framework on live, production multi-region cloud infrastructure using real-time carbon intensity forecasting services, to confirm that the simulated results translate to real-world operating conditions. Second, incorporating embodied carbon considerations, including hardware manufacturing and end-of-life emissions, into the overall optimization objective, moving beyond the operational-emissions focus of the present study toward a full lifecycle carbon accounting approach. Third, extending the adaptive optimization module with reinforcement-learning-based policy learning, allowing the scheduler and optimizer to improve their joint decision policy continuously from observed outcomes rather than relying on the greedy heuristic approach employed in this study, potentially capturing additional carbon reduction from more sophisticated long-horizon scheduling and compression decisions.

References

1. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 3645-3650.
2. Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L. M., Rothchild, D., So, D., Texier, M., & Dean, J. (2021). Carbon emissions and large neural network training. arXiv preprint arXiv:2104.10350.
3. Wu, C. J., Raghavendra, R., Gupta, U., Acun, B., Ardalani, N., Maeng, K., Chang, G., Aga, F., Huang, J., Bai, C., Gschwind, M., Gupta, A., Ott, M., Melnikov, A., Candido, S., Brooks, D., Chauhan, G., Lee, B., Lee, H. H., Akyildiz, B., Balandat, M., Spisak, J., Jain, R., Rabbat, M., & Hazelwood, K. (2022). Sustainable AI: Environmental implications, challenges and opportunities. Proceedings of Machine Learning and Systems, 4, 795-813.
4. Radovanovic, A., Koningstein, R., Schneider, I., Chen, B., Duarte, A., Roy, B., Xiao, D., Haridasan, M., Hung, P., Care, N., Talukdar, S., Mullen, E., Smith, K., Cottman, M., & Cirne, W. (2023). Carbon-aware computing for datacenters. IEEE Transactions on Power Systems, 38(2), 1270-1280.
5. Wiesner, P., Behnke, I., Thamsen, L., Bermbach, D., & Kao, O. (2021). Let's wait awhile: How temporal workload shifting can reduce carbon emissions in the cloud. Proceedings of the 22nd International Middleware Conference, 260-272.
6. Dodge, J., Prewitt, T., Tachet des Combes, R., Odmark, E., Schwartz, R., Strubell, E., Luccioni, A. S., Smith, N. A., DeCario, N., & Buchanan, W. (2022). Measuring the carbon intensity of AI in cloud instances. Proceedings of the ACM Conference on Fairness, Accountability, and Transparency, 1877-1894.
7. Barroso, L. A., Clidaras, J., & Holzle, U. (2013). The datacenter as a computer: An introduction to the design of warehouse-scale machines (2nd ed.). Morgan & Claypool Publishers.
8. Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., & Pineau, J. (2020). Towards the systematic reporting of the energy and carbon footprints of machine learning. Journal of Machine Learning Research, 21(248), 1-43.
9. Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. Communications of the ACM, 63(12), 54-63.
10. Acun, B., Lee, B., Kazhamiaka, F., Maeng, K., Gupta, U., Chakkaravarthy, M., Brooks, D., & Wu, C. J. (2023). Carbon explorer: A holistic framework for designing carbon aware datacenters. Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 118-132.
11. Teerapittayanon, S., McDanel, B., & Kung, H. T. (2016). BranchyNet: Fast inference via early exiting from deep neural networks. 23rd International Conference on Pattern Recognition (ICPR), 2464-2469.
12. Masanet, E., Shehabi, A., Lei, N., Smith, S., & Koomey, J. (2020). Recalibrating global data center energy-use estimates. Science, 367(6481), 984-986.
13. Kumar, R., Devi, M., Gupta, H., & Soni, P. (2026). AI-powered segmentation and validation: A new era for lung cancer detection in biomedical. Grenze International Journal of Engineering and Technology. Grenze ID: 01.GIJET.12.1.6_1.
14. Kumar, R., Shailja, & Soni, P. (2025). Facial expression and image-based early detection of autism spectrum disorder using deep learning techniques. In Proceedings of the 17th International Conference on Contemporary Computing (IC3) (pp. 1–5). IEEE. https://doi.org/10.1109/IC366947.2025.11290207
15. Kumar, R., Soni, P., & Mishra, N. (2021). Analysis of COVID-19 pandemic impact. Turkish Journal of Physiotherapy and Rehabilitation, 32(3), 9611–9617.

16. Ahmad, S., Kumar, S., Kumar, M., Kumar, R., Vyeshikha, Memoria, M., Rawat, A., & Gupta, A. (2022). The importance of quantifying financial returns on information system (IS) investment for organizations: An analysis. In Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON) (pp. 197–200). IEEE. https://doi.org/10.1109/COM-IT-CON54601.2022.9850666
17. Lakkadi, V. R., Kumar, M., Donthi, P. R., & Kandula, S. (2026). AI-driven distributed ledger for next-generation supply chain traceability and risk mitigation. In Innovative Computing and Communications (ICICC 2026) (Lecture Notes in Networks and Systems, Vol. 2020, pp. 1–17). Springer. https://doi.org/10.1007/978-3-032-28307-8_30
18. Lalita, K., Kaneria, O., Gupta, V., Kumar, M., & Arya, S. (2025). Machine learning-driven framework for financial fraud detection using graph neural networks and explainable AI. Enterprise Development & Microfinance, 36(3s), 209–221.
19. Kumar, M., Arya, S., & Gill, M. S. (2024). Explainable AI integration blockchain fraud detection system for secure financial transaction. Frontiers in Health Informatics, 13(8), 8270–8277.
20. Kumar, M. (2025). Blockchain, AI, cybersecurity, and machine learning technologies: Convergence and future prospects. International Journal for Research Technology and Seminar, 29(2), 1–24.
21. Kumar, M., Arya, S., Gill, M. S., & Sangwan, S. (2025). Blockchain technology in healthcare supply chain distribution: A comprehensive analysis of pharmaceutical, medical device, and nuclear pharmacy applications (2020–2025). Advances in Nonlinear Variational Inequalities, 28(6s), 1072–1089. https://doi.org/10.52783/anvi.v28.7110
22. Luccioni, A. S., Viguier, S., & Ligozat, A. L. (2023). Estimating the carbon footprint of BLOOM, a 176B parameter language model. Journal of Machine Learning Research, 24(253), 1-15.