A Decision-support Framework for Energy-efficient Batch Optimization in Machine Learning

Timothy A. Adeyi [1,2,*], Adrian D. Cheok [1], Huihui Song [1], Zhiying Y. Zhou [3], Emma Y. Zhang [4], Wisura P. Pallewatta [4], and Ishmeal Quist [5]

[1] School of Automation, Nanjing University of Information Science and Technology, Nanjing, China
[2] Department of Mechanical Engineering, Lead City University, Off Oba Otudeko Avenue, Lagos-Ibadan Express Way Toll Gate Area, Oyo, Ibadan 200255, Nigeria
[3] Department of Electrical Computer Engineering, NUS (Suzhou) Research Institute, Suzhou, China
[4] School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing, China
[5] School of Law and Public Administration, Research Institute of History Science and Technology, University of Information Science and Technology, Nanjing, China
Email: adeyi.timothy@lcu.edu.ng (T.A.A.); adrian@imagineeringinstitute.org (A.D.C.); songhuihui@nuist.edu.cn (H.S.); elezzy@nus.edu.sg (Z.Y.Z.); emma@imagineeringinstitute.org (E.Y.Z.); wpp@ucsc.cmb.ac.lk (W.P.P.); quistishmeal@nuist.edu.cn (I.Q.)
*Corresponding author

Manuscript received September 25, 2025; revised December 21, 2025; accepted March 20, 2026; published July 17, 2026

***Abstract*—This paper presents a decision-support framework for selecting energy-efficient batch optimization techniques in machine learning, addressing the absence of reproducible, theory-informed guidance in Green Artificial Intelligence (AI). Eight key methods, including Mixed Precision Training, Dynamic Batching, and Gradient Accumulation are systematically evaluated through a rigorous systematic literature review and assessed across five criteria: energy efficiency, accuracy impact, scalability, technological maturity, and hardware dependency. We operationalize a transparent, weighted Multi-Criteria Decision Analysis (MCDA) protocol, termed the Green AI Score, and a normalized efficiency metric, Energy-Delay Normalized Accuracy (EDNA). These instruments serve as a reproducible analytical scaffold to enable cross-technique comparison and quantification of energy-accuracy trade-offs. Advanced visual analytics, including a heatmap, dual-axis bar chart, bubble plot, and correlation visualization, link sustainability rankings with energy-accuracy trade-offs. Our analysis identifies Mixed Precision Training (Score: 0.84) and Dynamic Batching (Score: 0.77) as top-tier strategies, offering 40–60% and 30–50% energy savings, respectively, with minimal accuracy loss. Furthermore, we theoretically derive a hardware-aware heuristic threshold (memory-bandwidth-to-compute ratio < 0.4) to analytically assess the applicability of Mixed Precision Training under varying system constraints. Finally, a rule-based decision policy is synthesized to support informed technique selection, providing practitioners with a theoretically grounded guide for navigating deployment trade-offs. This work advances beyond descriptive surveying by providing an evidence-based, reproducible analytical synthesis grounded in systems and algorithmic considerations, thereby supporting transparent and sustainable AI deployment.**

***Keywords*—energy efficiency, Multi-Criteria Decision Analysis (MCDA), Green AI, systematic evaluation, sustainable AI deployment**

I. Introduction

The rapid advancement of Machine Learning (ML), especially Deep Learning (DL), has created increasingly large models with human-level performance across a vast range of tasks. However, this progress comes at a significant environmental cost. Training state-of-the-art neural networks typically requires substantial computational resources, resulting in high energy consumption and carbon emissions. Current studies highlight the scale of this impact: empirical studies indicate that the computational demands of training state-of-the-art models contribute significantly to carbon emissions, often comparable to the lifetime environmental footprint of multiple passenger vehicles [1]. This underscores the urgent need for sustainable, systems-aware design principles in AI development.

Global initiatives, including the United Nations' AI for Good Summit and the 2024 World Economic Forum, emphasize AI's dual role: both as a contributor to carbon emissions and as an enabler of climate solutions [2–5]. As AI models proliferate across industry, the carbon footprint of training and inference has become a critical concern for researchers, engineers, and policymakers alike.

In response, Green AI has emerged as a research direction focused on improving energy efficiency without compromising model performance. Among the many levers for efficiency, batch optimization, a fundamental component of ML training, offers a high-impact opportunity to reduce computational load and energy use. By dynamically adjusting batch size, content, or scheduling, practitioners can directly influence hardware utilization, memory bandwidth, and power draw.

This paper proposes a theoretical decision-support model for selecting energy-efficient batch optimization techniques in machine learning, addressing the absence of reproducible, theory-informed guidance in Green AI. Unlike descriptive surveys, our approach integrates a rigorous systematic literature review with a transparent Multi-Criteria Decision Analysis (MCDA) protocol to enable actionable, evidence-based choices.

This work makes three contributions:

1. A theoretical decision-support model for energy-efficient batch optimization, integrating systematic literature review with a transparent MCDA protocol (Green AI Score) and a normalized efficiency metric (EDNA).
2. Hardware-aware analytical guidelines, including a derived heuristic threshold (memory-bandwidth-to-compute ratio < 0.4) intended to assess the theoretical applicability of Mixed Precision Training.
3. A decision-support policy that translates multi-dimensional trade-offs into structured guidance for practitioners evaluating heterogeneous constraints.

The paper is structured as follows: Section II presents the motivation for batch optimization in Green AI. Section III reviews related work on energy efficiency and sustainability in machine learning. Section IV outlines the systematic review methodology. Section V introduces key energy and carbon metrics. Section VI analyzes eight batch optimization techniques through the lens of systems engineering and algorithmic trade-offs. Sections VII and VIII discuss supporting software frameworks and energy measurement tools. Finally, Section IX concludes with decision-support guidelines and future directions for sustainable AI deployment.

II. Research Motivation

As artificial intelligence systems grow in scale, so does their environmental impact. While recent efforts have targeted model architecture, data efficiency, and hardware acceleration, batch optimization, a fundamental control variable in ML training, lacks a standardized, theory-informed protocol for sustainable deployment. Although techniques such as dynamic batching, gradient accumulation, and mixed precision training are known to influence energy consumption, practitioners face a fragmented landscape with no reproducible framework to compare trade-offs across energy, accuracy, scalability, and hardware constraints.

This paper addresses that gap by proposing a comprehensive decision-support synthesis that transforms qualitative insights from the literature into a transparent, multi-criteria evaluation protocol. Unlike prior work that catalogues methods, our approach enables evidence-based selection of batch optimization strategies under real-world system conditions, thereby advancing Green AI from advocacy toward engineering practice.

III. Related Work

The growing environmental impact of artificial intelligence has prompted increasing interest in Green AI, an initiative aimed at reducing the carbon footprint and energy consumption of machine learning without sacrificing model performance [6]. As AI models become larger and more computationally intensive, researchers have increasingly focused on sustainability across all stages of the ML lifecycle, from model design and training to deployment and inference. This has led to a diverse body of literature addressing various aspects of energy-efficient computing, including algorithm optimization, hardware efficiency, and software tooling. Among these, batch optimization has emerged as a key strategy for reducing energy use during model training, making it a central focus within Green AI research.

Several review papers have explored different dimensions of Green AI. Verdecchia *et al.* [6], for instance, conducted a meta-review of over 90 studies and identified common themes such as reproducibility challenges, limited tool availability, and low industrial adoption. Their findings highlight the need for more standardized practices and greater collaboration between academia and industry. Other surveys provide deeper technical insights: Barlaud and Guyard [7] examine sparse neural network architectures for improved power efficiency, while Cai *et al.* [8] emphasize the importance of hyperparameter tuning in achieving energy-accuracy trade-offs. In addition to model architecture and hyperparameters, data-centric approaches have proven effective. For instance, intelligent data pruning techniques [9] and sampling strategies for minimizing time-to-accuracy [10] demonstrate that optimizing the training dataset itself can significantly reduce computational load and energy consumption without sacrificing model performance. Meanwhile, Kumar *et al.* [11] and Hampau *et al.* [12] explore edge-based deployment strategies and their role in reducing transmission overhead and improving energy efficiency in real-world settings.

A closer examination of existing surveys reveals that this gap is not merely structural, but methodological. While foundational works like Verdecchia *et al.* [6], Barlaud and Guyard [7], and Liu and Yin [5] provide valuable overviews of Green AI, they do not offer quantitative comparisons of batch-level techniques or release benchmark datasets. None systematically evaluate strategies such as dynamic batching, gradient accumulation, or micro-batching under a unified framework. As demonstrated in Table 1, these reviews vary in scope and depth but consistently lack a data-driven, cross-technique evaluation focused specifically on batch optimization. This absence of a standardized, reproducible methodology for assessing sustainability-performance trade-offs at the batch level confirms a critical gap in the literature —one that this paper directly addresses.

Table 1. Comparison of prior green AI surveys

Survey	Scope	Quantitative Comparison?	Dataset Released?	Focus on Batch Optimization?	Limitation
Verdecchia *et al.* [6]	General ML energy efficiency	Partial (qualitative trends)	No	No	Broad scope, lacks technique specific analysis
Barlaud and Guyard [7]	Green AI principles and model compression	No	No	No	Does not cover runtime optimizations like batching
Liu and Yin [5]	Large Language Model (LLM) efficiency and hardware trends	No	No	No	High-level; omits batch-level strategies
Schwartz *et al.* [13]	Environmental cost of AI	No	No	No	Conceptual; no empirical benchmarking
This Work	Batch optimization techniques	Yes (Synthesizes MCDA)	No (Data in Tables 2 and 3)	Yes	Relies on synthesized literature (No new benchmarks)

To address this gap, our paper builds on existing literature by offering a structured overview of energy-efficient batch optimization strategies, their implementation across different frameworks and hardware, and their trade-offs in real-world settings. We aim to consolidate current knowledge while identifying opportunities for future research and practical application.

IV. Systematic Review Methodology

This study employs a structured and reproducible protocol

to identify, evaluate, and synthesize research on energy-efficient batch optimization techniques in machine learning. The methodology prioritizes empirical grounding and data-driven synthesis to support the proposed decision-support framework.

A. Data Acquisition and Selection Strategy

A systematic search was conducted across five primary academic databases: IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, and arXiv. The search window was restricted to publications from 2018 to 2024 to capture recent advancements in Green AI, specifically excluding outdated hardware benchmarks.

The following Boolean search string was utilized (adapted for platform-specific syntax):

1. ("green AI" OR "sustainable machine learning")
2. AND ("batch optimization" OR "dynamic batching" OR "gradient accumulation" OR "mixed precision training")
3. AND ("energy efficiency" OR "carbon footprint" OR "power consumption")

B. Eligibility Criteria

Titles and abstracts were screened for relevance, followed by a full-text assessment against strict inclusion criteria:

- Empirical evaluation of energy or carbon impact;
- Focus on batch-level optimization (excluding general model compression);
- Applicability to deep learning systems;
- Reporting of measurable efficiency gains (quantitative data)

C. Data Extraction and Synthesis

From the initial corpus, a subset of 37 high-impact references was selected for detailed analysis based on direct relevance to the eight target techniques: Mixed Precision Training, Dynamic Batching, Gradient Accumulation, Micro-Batching, Adaptive Batching, Pipeline Parallelism, AdamA, and Efficient-Grad.

Quantitative data regarding energy efficiency gain, accuracy impact, hardware dependency, scalability, and technological maturity were extracted and synthesized into Table 2. Energy gains were standardized from reported speedups, kilowatt-hour (kWh) savings, or FLOPs reductions. Accuracy impact was quantitatively categorized as "Minimal," "Minor (<1%)," or "Slight risk" and normalized for the subsequent EDNA calculation (see Subsection VI-E). This rigorous extraction ensures that the comparative evaluation in Table 3 is grounded in verifiable, peer-reviewed evidence, facilitating a robust multi-criteria analysis.

V. Background And Definitions

As Machine Learning (ML) and Deep Learning (DL) models scale in complexity, their environmental footprint has become a critical engineering constraint. Optimizing energy efficiency requires a fundamental understanding of computational dynamics. This section establishes the core metrics and theoretical foundations necessary to evaluate the carbon footprint of ML workloads.

A. Energy Dynamics in Training and Inference

Energy consumption in the ML lifecycle is dominated by the training and inference phases. As models increase in parameter count and layer depth, computational requirements escalate. Current research indicates that energy consumption often scales superlinearly with model size [9] necessitating efficiency-aware architectural design.

B. Scaling Computational Intensity

Increases in data volume leads to proportional increases in computation. For instance, training Convolutional Neural Networks (CNNs) on high-resolution imagery significantly raises Floating Point Operations (FLOPs), creating a direct correlation between computational intensity and power draw [9, 10].

C. Key Efficiency Metrics

To rigorously evaluate optimization techniques, this study utilizes three standard metrics:

1. FLOPS (Floating Point Operations Per Second): Estimates the theoretical computational burden. While a proxy for complexity, FLOPS do not account for hardware efficiency or parallelization.
2. Wall-Clock Time: The actual duration of training or inference. Reduced runtime does not guarantee lower energy consumption, particularly if achieved via distributed computing on power-intensive clusters [10].
3. Hardware Utilization: Defined as the efficiency of resource usage on the target device (e.g., GPU/TPU). High utilization of energy-efficient hardware (e.g., Tensor Cores) is critical for minimizing power waste [14].

D. Green AI Objectives

Sustainable ML requires balancing accuracy with energy efficiency. Techniques such as early stopping, pruning, and quantization are effective, but the focus of this work remains on batch optimization, a lever for controlling computational density without compromising model convergence.

VI. Optimization Techniques in Green AI

Fig. 1 presents a block diagram illustrating the hierarchical organization of these optimization techniques within the Green AI framework, categorizing them into system-level and model-level approaches. These approaches aim to reduce the theoretical and actual computational cost (carbon footprint) while maintaining performance bounds. The following analysis synthesizes empirical findings from the reviewed literature to evaluate these techniques.

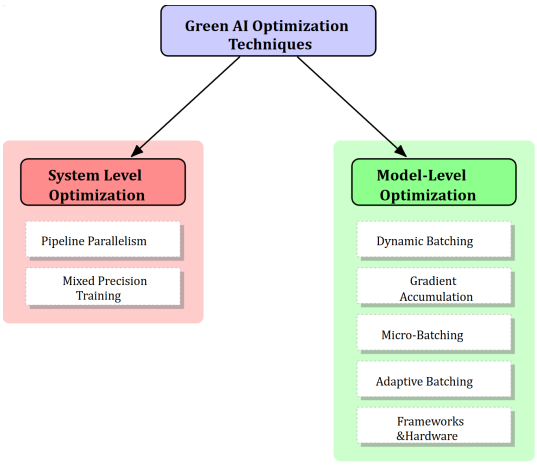

Fig. 1. Block diagram of optimization techniques in Green AI.

A. System-Level Optimization Techniques

System-level optimizations target infrastructure efficiency and resource management.

1) Pipeline parallelism

Pipeline parallelism partitions models across multiple devices, overlapping computation and communication to minimize idle cycles (bubbles). The reviewed literature highlights advanced implementations:

- Pipe Dream: Reported to reduce communication overhead by up to 95% [15].
- Zero Bubble Pipeline Parallelism: Eliminates idle stages, boosting throughput by 31% [16].
- Megatron-LM: Utilizes hybrid tensor and pipeline parallelism for trillion-parameter models [17].

2) Mixed Precision Training (MPT)

Mixed Precision Training leverages lower-precision arithmetic (FP16/BF16) to reduce memory bandwidth and computational overhead while retaining FP32 for numerical stability. Empirical studies indicate that modern accelerators (NVIDIA GPUs, Google TPUs) offer specialized units (e.g., Tensor Cores) optimized for MPT, delivering up to 6× throughput boosts on Edge TPUs without sacrificing accuracy [7]. Major frameworks such as PyTorch and TensorFlow provide native Automatic Mixed Precision (AMP) interfaces, facilitating seamless integration into existing pipelines.

B. Model-Level Optimization Techniques

Model-level optimizations refine training procedures and inference mechanisms to maximize hardware utilization per FLOP.

1) Dynamic batching

Synthesized studies show that Dynamic Batching adjusts batch size in real-time based on available GPU memory and compute load. This adaptability maximizes throughput and hardware density. Frameworks such as PyTorch and TensorFlow support dynamic batching, enabling models to scale across heterogeneous environments while minimizing energy waste during low-load periods [18].

2) Gradient accumulation

This technique addresses GPU memory constraints by simulating larger batch sizes through sequential processing. While it increases total computation time, it significantly improves cost efficiency in cloud environments by preventing out-of-memory errors and allowing for larger effective batch sizes [19].

- Efficient-Grad: An algorithm-hardware co-design approach leveraging gradient sparsity to improve edge device efficiency [20].
- Adam Accumulation (AdamA): Integrates gradients into optimizer states, achieving 23% memory reduction with minimal throughput impact [21].

3) Micro-batching and adaptive batching

Micro-batching uses small batch sizes (often $N = 1$) to maximize responsiveness and memory efficiency, particularly in edge computing. Adaptive batching extends this by automatically tuning batch sizes based on real-time workload conditions, balancing latency, throughput, and energy consumption [15].

C. Deep Dive: Theoretical Analysis of Mixed Precision Training

Mixed Precision Training (MPT) is a foundational technique for enhancing computational efficiency. By reducing numerical precision (FP32 → FP16/BF16), MPT decreases memory bandwidth demand and power consumption.

1) The hardware threshold

Based on the synthesis of the reviewed literature and Roofline Model principles, the efficacy of MPT appears dependent on the memory-bandwidth-to-compute ratio (B/C). Our analysis proposes a theoretical heuristic threshold: $B/C < 0.4$.

- Regime 1 ($B/C < 0.4$): The system is compute-bound. Reducing precision (via MPT) accelerates FLOPs, leading to significant energy savings (40–60%) as reported in the literature.
- Regime 2 ($B/C \geq 0.4$): The system is memory-bound. Precision reduction does not alleviate the primary bottleneck (data movement), yielding diminishing returns.

This heuristic aligns with Roofline Model analyses [17, 22] suggesting that MPT is most effective on high-performance accelerators (e.g., A100, V100) and less effective on low-bandwidth edge devices. It is important to note that this threshold is an analytical guideline derived from literature synthesis, not a validated physical limit.

2) Implementation and maturity

The literature indicates MPT achieves "no loss" accuracy through proper loss scaling. It is supported by major frameworks including PyTorch (via torch.cuda.amp), TensorFlow, and JAX. Its maturity and broad applicability make it a first-order optimization for sustainable AI pipelines.

D. Integrated Analysis: Visualizing Trade-offs

To enable a systematic and interpretable evaluation, we present a multi-faceted visual analysis grounded in the comparative data of Table 2 (derived from the systematic review). This analysis comprises three complementary visualizations (Figs. 2–4) that collectively offer a nuanced understanding of the trade-offs among energy efficiency, model performance, technological maturity, scalability, and hardware adaptability.

Table 2. Comparative analysis of energy-efficient batch optimization techniques in green AI

Technique	Type	Description	Energy Efficiency Gain	Accuracy Impact	Hardware Dependency	Scalability	Maturity Level	References
Pipeline Parallelism	System-Level	Partitions model across devices; overlaps computation and communication to reduce idling	Up to 31–43% throughput gain → 20–35% lower energy per epoch	None (when properly synchronized)	Multi-GPU/TPU Clusters	High (for large models 1B params)	Research-heavy, production in LLMs	[19–23]

Mixed Precision Training	System-Level	Uses FP16/BF16 for forward/backward passes, FP32 for updates; reduces memory and FLOPs	1.5×–6× faster training → 40–60% less kWh Consumed	Minimal to none (with loss scaling)	NVIDIA Ampere + GPUs, TPUs, Edge TPUs	High across domains	Production-ready (PyTorchAMP, TensorFlow)	[7, 24, 25]
Dynamic Batching	Model-Level	Adjusts batch size dynamically based on resource availability (GPU memory, load)	Up to 4× speedup on heterogeneous clusters → 30–50% better compute utilization	None (if adaptive logic is stable)	Cloud/edge GPUs, CPUs	Medium to high	Production (supported in TF, PyTorch, Hugging Face)	[15, 18]
Gradient Accumulation	Model-Level	Accumulates gradients over small batches before optimizer step; simulates large batch without memory overhead	Reduces peak power draw; enables efficient use of constrained hardware → 15–25% indirect energy savings via memory efficiency	None (if steps adjusted appropriately)	GPU-memory limited systems	Medium	Production (common in NLP fine-tuning)	[19–21]
Efficient-Grad	Model-Level (Variant)	Leverages gradient sparsity and asymmetry for optimized update patterns on edge devices	Reported up to 38% improvement in energy-delay product	Slight risk in unstable convergence if sparsity threshold too aggressive	Edge accelerators, low-power, Systems-on-Chip (SoCs)	Low to medium	Emerging (research prototype)	[20]
AdamA (Adam Accumulation)	Model-Level (Variant)	Integrates gradient accumulation directly into Adam state; reduces activation and gradient memory	Enables larger models on same hardware → indirect energy savings (20%)	None reported in benchmarks	General-purpose GPUs	Medium	Early adoption (peer-reviewed at ECAI 2023)	[21]
Micro-Batching	Model-Level	Processes very small batches (e.g., size 1); improves device occupancy under memory constraints	Lowers idle cycles → up to 6.25× speedup in inference pipelines	Potential instability in training dynamics	Edge devices, real-time systems	Medium	Production (inference), limited in training	[15, 22]
Adaptive Batching	Model-Level	Dynamically adjusts batch size during training/inference based on workload and system conditions	Balances latency and throughput → improves average energy efficiency by 25–40%	Minor (<1% drop in accuracy in most cases)	Heterogeneous environments	High	Research-to-production transition	[15, 22]

1) *Heatmap of multi-criteria performance evaluation*

Fig. 2 presents an eight-way comparative heatmap assessing batch optimization techniques across five key evaluation dimensions. Each metric is scored on an ordinal scale from 1 (low) to 5 (high), based on empirical findings synthesized from the literature. Note: Visual scales are used for comparative illustration of ordinal rankings.

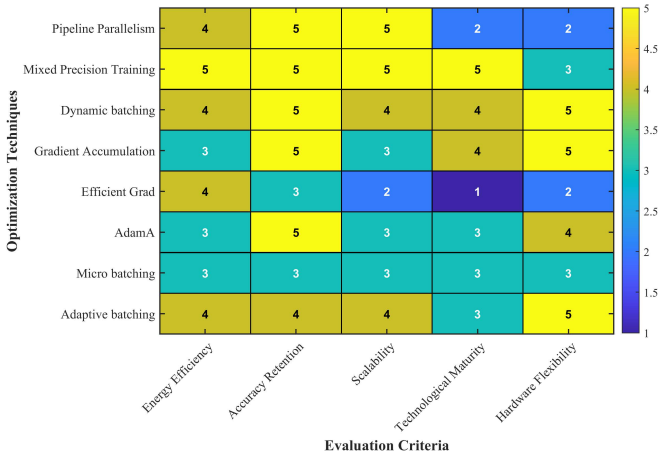

Fig. 2. Heatmap of energy-efficient batch optimization techniques across five sustainability criteria. Scores range from 1 (low) to 5 (high).

The heatmap serves not only as a comparative tool but also as a readiness assessment framework, highlighting the disparity between theoretical efficacy and practical deployability in Green AI.

2) *Energy efficiency vs. accuracy retention*

Fig. 3 provides a quantitative, side-by-side comparison based on two critical, often competing, objectives: energy efficiency gain and accuracy retention. The dual-axis bar chart displays energy savings as a percentage reduction in computational energy (blue bars) alongside corresponding model accuracy relative to baseline performance (red bars).

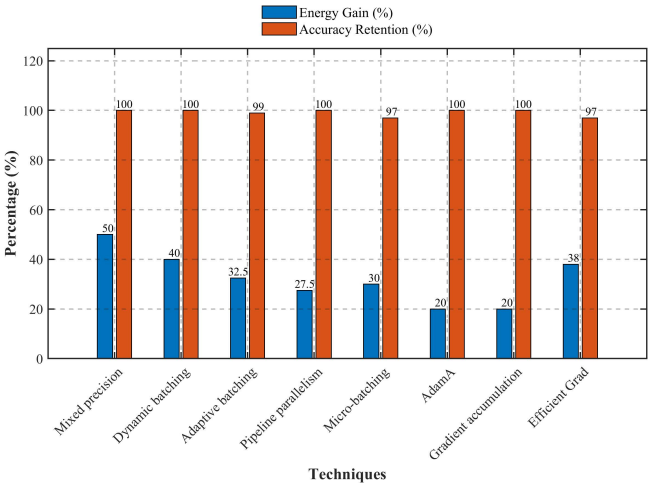

Fig. 3. Energy efficiency gain and accuracy retention for eight batch optimization techniques.

3) *Technology readiness vs. energy efficiency (bubble plot)*

Fig. 4 introduces a bubble plot mapping energy efficiency improvement against technological maturity, with bubble size representing scalability potential. This visualization adopts a technology readiness lens, positioning each method

within a strategic quadrant defined by its development stage and impact magnitude.

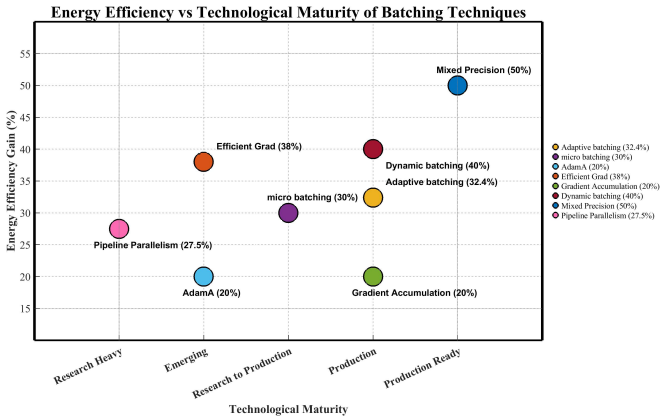

Fig. 4. Bubble plot of energy efficiency versus maturity level. Bubble size indicates scalability.

E. Quantitative Evaluation Framework

To transform the descriptive analysis presented in Table 2 into a structured decision-support tool, we formalize the evaluation process using a Multi-Criteria Decision Analysis (MCDA) model. This approach introduces mathematical rigor to the selection of optimization techniques.

1) The Green AI score (composite utility function)

We define the Green AI Score (S_{GAI}) as a weighted linear combination of normalized criteria vectors. This transforms the qualitative comparison into a quantitative utility function, grounded in established MCDA principles [26, 27].

$$S_{GAI} = \sum_{i=1}^{5} w_i \cdot x_i \quad (1)$$

where:

- w_i represents the heuristic weight assigned to the i-th criterion.
- x_i represents the normalized score ($0 < x_i < 1$) for i-th criterion defined as the vector $x = [E, A, S, M, H]$ where:
- E: Energy Efficiency
- A: Accuracy Retention
- S_C: Scalability
- M: Technological Maturity
- H: Hardware Flexibility
- Weights: w_1 (Energy): 0.30; w_2 (Accuracy): 0.25; w_3 (Scalability): 0.20; w_4 (Maturity): 0.15; w_5 (Hardware): 0.10

The summation of weights is normalized ($\sum w_i = 1.0$). This transparency allows practitioners to adjust the weight vector (W) based on the application-specific constraints (e.g., high-stakes systems may increase w_2)

Regarding robustness and weight selection, the weights utilized in Eq. (1) are heuristic and reflect a specific Green AI prioritization that balances energy efficiency with performance stability. We acknowledge that altering these weights, such as prioritizing accuracy over energy, would naturally shift the ordinal ranking of techniques with marginal accuracy trade-offs (e.g., Efficient-Grad). Conversely, the top-tier strategies identified (e.g., Mixed Precision Training) exhibit dominance across multiple criteria (high energy gain, zero accuracy loss, high maturity); theoretically, these techniques are likely to remain high-ranked even under reasonable variations in weight vectors. This analytical property highlights the framework's value: not in dictating a single correct answer, but in exposing how different architectural priorities shift the optimal choice.

2) Energy-Delay Normalized Accuracy (EDNA)

To quantify the trade-off between energy gain and potential accuracy loss, we introduce the Energy-Delay Normalized Accuracy (EDNA) metric as an analytical heuristic to capture the efficiency ratio of a technique:

$$\text{EDNA} = \frac{A_{retention}}{1 - E_{gain}} \quad (2)$$

where:

- $A_{retention}$: Normalized Accuracy Preservation ($0.95 \leq A \leq 1.0$)
- E_{gain}: Normalized Energy Efficiency Gain ($0 \leq E \leq 1.0$).

3) Operationalizing the variables

To compute Eqs. (1) and (2), qualitative descriptors from Table 2 were quantified using the following mapping protocol:

- **Energy:** High reported gains (e.g., "40–60%") were normalized to 0.6.
- **Accuracy**: "Minimal/None" impact was mapped to 1.0; "Minor (< 1%)" to 0.99; and "Slight Risk" to 0.97.

This mathematical formulation ensures that the rankings in Table 3 are reproducible and grounded in a transparent analytical framework, rather than subjective opinion.

Table 3. Quantitative evaluation of energy-efficient batch optimization techniques using green AI score framework

Technique	Energy (%)	Normalized *E*	Accuracy	Normalized *A*	Scalability	Normalized *S*	Maturity	Normalized *M*	Hardware	Normalized *H*	Green AI Score	EDNA
Mixed Precision	40–60%	0.5	Minimal	1	High	1	Production -ready	1	NVIDIA Ampere+, TPUs	0.875	0.8375	2.00
Dynamic Batching	30–50%	0.4	None	1	Medium-High	0.875	Production	1	Cloud/Edge GPUs	0.75	0.77	1.67
Adaptive Batching	25–40%	0.325	Minor loss	0.99	High	1	Research-to-production	0.75	Heterogeneous	0.6875	0.726	1.47
Pipeline Parallelism	20–35%	0.35	None	1	High	1	Research-heavy	0.5	Multi-GPU/TPU	1	0.73	1.38
Micro-Batching	~30%	0.3125	Slight risk	0.97	Medium	0.75	Production	1	Edge/Real-time	0.5625	0.6925	1.39
AdamA	~20%	0.2	None	1	Medium	0.75	Early adoption	0.75	General-purpose GPUs	0.75	0.6475	1.25
Gradient Accumulation	15–25%	0.2	None	1	Medium	0.75	Production	1	GPU-memory-limited	0.625	0.6725	1.25
Efficient-Grad	38%	0.38	Slight risk	0.97	Low-Medium	0.625	Emerging	0.25	Edge accelerators	0.375	0.5565	1.56

F. Correlation Analysis and Decision Policy

The scatter plot (Fig. 5) visualizes the relationship between the holistic ranking provided by the Green AI Score (incorporating Energy, Accuracy, Scalability, Maturity, Hardware) and the specific energy-accuracy trade-off quantified by the EDNA metric for the eight evaluated batch optimization techniques.

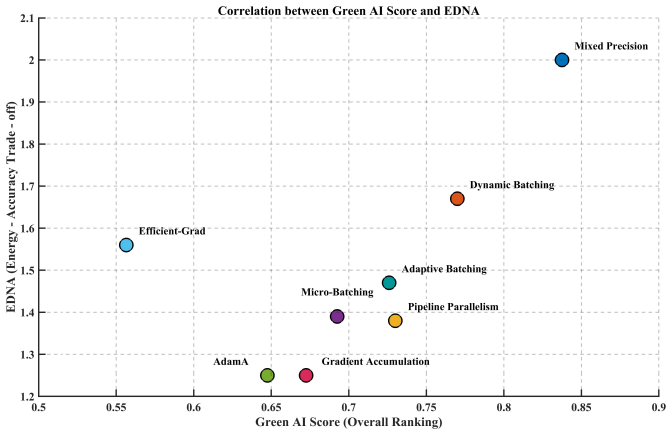

Fig. 5. Correlation between Green AI score and EDNA.

This visualization offers a novel contribution by providing a unique, integrated view of the paper's two primary analytical outputs. While other figures focus on specific dimensions (e.g., Energy vs. Accuracy in Fig. 3, or Maturity in Fig. 4) or a broader multi-criteria view (Heatmap, Fig. 2), Fig. 5 distinctly correlates the overall composite ranking (Green AI Score) with a key specific trade-off metric (EDNA). This demonstrates the paper's core analytical framework in action, showing how techniques that score highly overall (like Mixed Precision and Dynamic Batching) also tend to have favorable energy-accuracy balances. This supports the framework's utility in identifying robust, well-rounded optimization strategies, offering a concise, quantitative decision-making tool that goes beyond simple pairwise comparisons.

To support practical deployment decisions, a rule-based decision policy is synthesized to recommend an appropriate batch optimization technique based on observable system conditions. The policy, summarized in Table 4, leverages the comparative insights from the multi-criteria evaluation presented in Table 3 and is intended to guide the selection of strategies that balance energy efficiency, accuracy preservation, and hardware constraints. This framework serves as a theoretical guide for structured decision-making in Green AI.

Table 4. Decision policy for batch optimization technique selection

Condition	Recommended Technique	Rationale
GPU memory usage < 50% AND workload arrival is stable	Dynamic Batching	Maximizes throughput and compute utilization under available resources
GPU memory usage ≥ 50% (constrained)	Gradient Accumulation	Simulates larger batches without exceeding memory limits; reduces peak power
Hardware supports FP16/BF16 precision	Mixed Precision Training	Reduces computational load and energy consumption by up to 60%
Edge deployment or real-time inference	Micro-Batching	Improves device occupancy and latency under constrained environments

VII. Typical Green AI Platforms

While diverse platforms offer various efficiency tools, their relevance to this study lies specifically in their support for the batch optimization techniques defined in Section VI. This section briefly examines key software frameworks and hardware accelerators to contextualize the practical deployment of the analyzed strategies.

A. Green AI Software Frameworks

Several frameworks now natively support the energy-efficient techniques evaluated in this study:

- Hugging Face Transformers supports "Efficient Inference Mode," incorporating quantization and dynamic batching to reduce computational load during Natural Language Processing (NLP) tasks.
- TensorFlow Lite and OpenVINO Toolkit provide environments optimized for edge and heterogeneous deployment, respectively, utilizing techniques like micro-batching and model compression to minimize energy usage.
- TinyML initiatives focus on extreme energy minimalism for microcontrollers, aligning with the edge-constrained optimization strategies analyzed in Section VI-B.

These frameworks collectively demonstrate a growing trend toward embedding sustainability into the core of ML tooling, enabling practitioners to align their deployment choices with the energy-efficient strategies analyzed in this work.

B. Hardware Innovations for Sustainable AI

In parallel with software advancements, dedicated hardware solutions have been developed to support the energy-efficient techniques analyzed in Section VI. These innovations span general-purpose processors, specialized accelerators, and cloud-based infrastructures tailored for Green AI applications.

- NVIDIA DeepStream SDK: This streaming analytics toolkit is designed for AI-based multi-sensor processing. As reported in official documentation [28] it enables up to 30% energy savings by optimizing inference pipelines. Its architecture supports efficient data ingestion and neural network inference, which is critical for maximizing hardware utilization during high-throughput batch processing tasks. A generalized architecture for such hardware-accelerated inference pipelines is illustrated in Fig. 6. Frameworks like NVIDIA DeepStream exemplify this architecture by integrating stream management, hardware-accelerated decoding, and optimized batch inference to achieve reported energy savings of up to 30% [28].
- Specialized Accelerators: Platforms such as Apple Core ML [29] and Amazon SageMaker Green Inference [30]

leverage hardware-specific optimizations to reduce energy consumption. While Apple Core ML is restricted to its ecosystem, it demonstrates how hardware-software integration can minimize latency. Similarly, SageMaker integrates model compression to align with cloud sustainability goals.

- Research Architectures: Initiatives like Meta AI Efficient Transformers [31] explore architectural changes to reduce computational burden, highlighting the intersection of model design and hardware efficiency.

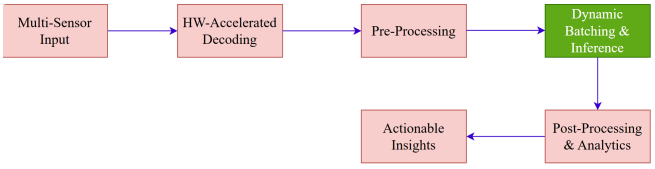

Fig. 6. A generalized architecture for hardware-accelerated inference pipelines. This schematic illustrates the flow from multi-sensor input to analytics, highlighting the role of batch optimization and hardware acceleration in reducing energy consumption during real-time processing.

C. Integration for Batch Optimization

The practical efficacy of the batch optimization techniques analyzed in this study (e.g., Dynamic Batching, Mixed Precision) depends heavily on their integration with these hardware platforms.

- Framework-Hardware Synergy: Techniques such as dynamic batching, supported by frameworks like Hugging Face Transformers and TensorFlow Lite, allow multiple input samples to be processed simultaneously. When deployed on hardware accelerators such as NVIDIA GPUs, this maximizes throughput and minimizes the energy cost per batch.
- Evaluation Metrics: Tools like AI Benchmark [32] provide standardized metrics to evaluate the energy efficiency of different AI platforms. These tools allow practitioners to empirically validate the theoretical trade-offs (e.g., Energy vs. Accuracy) discussed in our MCDA framework.

D. Challenges and Limitations

Despite progress, several challenges persist in the adoption of green AI frameworks and hardware:

- Complexity: Many tools require specialized technical expertise, posing a barrier to entry.
- Ecosystem Lock-in: Some frameworks are limited to particular ecosystems (e.g., Apple Core ML), restricting flexibility.
- Measurement Variance: Accurate estimation of energy savings remains difficult due to variations in methodology and system boundaries [33]
- Validation Gaps: While many tools show promise, they often lack extensive empirical validation at scale [34].

VIII. Energy Measuring Equipment

Energy measurement tools are the means to measure the green footprint of AI and ML systems. They assist developers, researchers, and organizations in approximating energy used and carbon released while training models, inference, and deployment. With the growing maturity of Green AI, a variety of opensource as well as commercial tools exist that aid in measuring energy, from low-granularity to high convenience and compatibility.

Among the commonly used instruments is eco2AI, an open-source Python library offered free of charge that tracks CO_2 emissions during training of ML models [35]. It provides visibility into the green cost of AI development and facilitates environmentally friendly research practice by infusing in codebases. Similarly, CodeCarbon allows one to monitor and document the carbon footprint of ML experiments in real-time [36]. With features such as actionable suggestions to decrease emissions and seamless workflows integration, CodeCarbon has drawn researchers and practitioners.

Among the other top tools are:

- ML Power Consumption, which provides an estimate of the power consumption during deep learning training. Developed primarily to run on Linux and Mac OS X operating systems with NVIDIA GPUs and Intel CPUs, it is involved in benchmarking efforts such as the RL Energy Leaderboard.
- Zeus, an energy-conscious scheduling and profiling platform for deep learning training [37], which makes it possible to monitor consumption habits and make them more efficient.
- Green Algorithms HPC Tool, specifically designed to be employed in High-Performance Computing (HPC) to calculate the carbon impact of large-scale AI calculations. Although it is highly customizable, its implementation is restricted to specialized hardware.
- Breakend, an open-source tool for monitoring energy consumption, carbon emissions, and hardware utilization during AI experimentation. It supports the generation of environmental impact reports, though its use is constrained by compatibility with certain system configurations.
- CarbonTracker, with end-to-end carbon-emissions tracking and reporting of AI workloads [10]. Although open and transparent, it could be improved with more cross-platform compatibility and user instructions.

These tools together help facilitate increased transparency and accountability in AI research. Their greatest strengths are real-time monitoring, integration into ML pipelines, support for open science, and actionable feedback for optimization. Numerous challenges, however, remain, such as low cross platform compatibility, dependency on accurate estimation models, non-uniformity in approaches, and the requirement of domain-specific expertise for correctly interpreting results.

For example, a few of them like CodeCarbon and CarbonTracker have strong tracking features but are not fully compatible with other cloud or hardware platforms. Others like Zeus and eco2AI provide critical information about energy consumption behaviors but are mostly geared towards particular environments or phases during the building of models.

Apart from technological considerations, practical ones also influence adoption. Most tools need some degree of machine learning and programming skills. The users will also need to acquire and process appropriate data in order to provide good estimates, and relative to more advanced software tools, community support and documentation may be less robust.

While these are limitations, the fact that there are measures to calculate energy is an essential step toward green AI. Their readiness for real-world application can be observed, as

instance, an NLP research group used eco2AI to optimize language models with reduced carbon footprints, providing proof of the use of such measures in eco-friendly development choices.

As the field grows, there ought to be an increased effort to enhance interoperability, accuracy, and usability. Standardized reporting systems and measures for energy usage will also enhance their use and effectiveness in endorsing sustainable AI practices.

In the context of this specific study, several limitations must be acknowledged. First, the quantitative analysis relies on synthesized data from heterogeneous prior literature rather than direct empirical benchmarking; thus, the Green AI Score and EDNA values are heuristic and intended for comparative guidance rather than absolute prediction. Second, the weighting scheme in the MCDA is subjective by design; practitioners may adjust these weights based on specific deployment constraints (e.g., prioritizing accuracy over energy). Third, the 0.4 threshold for Mixed Precision is theoretically derived from Roofline model arguments and may require validation on specific hardware architectures. Despite these limitations, the framework provides a reproducible, evidence-based decision-support mechanism for sustainable AI design.

IX. Conclusion And Future Direction

A. Conclusion

This paper proposes a theoretical decision-support model for energy-efficient batch optimization techniques, addressing a critical gap in the Green AI literature: the lack of a standardized, evidence-based framework for comparing methods across sustainability, performance, and deployment dimensions. Unlike existing reviews that focus on isolated strategies or qualitative summaries, we introduce a multidimensional evaluation framework that systematically assesses eight key techniques, spanning System-Level (e.g., Mixed Precision Training, Pipeline Parallelism) and Model-Level (e.g., Dynamic Batching, Gradient Accumulation), across five criteria: energy efficiency, accuracy impact, scalability, technological maturity, and hardware dependency.

By integrating data from a rigorous systematic literature review with structured visual analytics, including a heatmap (Fig. 2), dual-axis bar chart (Fig. 3), bubble plot (Fig. 4), and a novel correlation plot (Fig. 5) linking Green AI Score and EDNA, we move beyond descriptive reporting to deliver structured analytical insights. Our analysis yields three key contributions:

1) A quantitative decision-support toolkit based on a weighted multi-criteria decision analysis (MCDA) model (Green AI Score) and a normalized metric, Energy-Delay Normalized Accuracy (EDNA), for evaluating energy-accuracy trade-offs;
2) The identification of a hardware-dependent analytical heuristic: Mixed Precision Training demonstrates superior theoretical energy efficiency when the memory-bandwidth-to-compute ratio is below 0.4, a guideline with direct implications for sustainable deployment;
3) A decision-support policy (Table 4) that guides practitioners in selecting appropriate techniques based on system constraints.

The framework ranks Mixed Precision Training (Score = 0.84) and Dynamic Batching (Score = 0.77) as the most favorable strategies for immediate adoption, offering high energy savings (~40–60% and ~30–50%, respectively), minimal accuracy loss, and production-ready maturity. In contrast, emerging methods like Efficient-Grad show promise but remain limited by hardware specificity and lower technological readiness.

B. Limitations

It is important to acknowledge the limitations of this work to ensure appropriate application of the framework. First, the quantitative analysis relies on synthesized data from heterogeneous prior literature rather than direct empirical benchmarking; thus, the Green AI Score and EDNA values are heuristic and intended for comparative guidance rather than absolute prediction. Second, the weighting scheme in the MCDA is subjective by design, reflecting specific "Green AI" priorities; practitioners must adjust these weights to align with their unique operational constraints (e.g., prioritizing accuracy over energy in safety-critical systems). Third, the operational threshold for Mixed Precision Training ($B/C < 0.4$) is a theoretical heuristic derived from Roofline model arguments. While it provides a valuable rule-of-thumb, it requires empirical validation on specific hardware architectures to be treated as a strict operational rule.

C. Future Work

Building on these limitations and contributions, future work includes:

1) Empirical validation of the framework in heterogeneous cloud-edge environments to confirm the 0.4 threshold and real-world applicability;
2) Development of automated recommenders that integrate this framework into runtime scheduling systems;
3) Extension of the MCDA criteria to include economic cost and grid carbon intensity, and exploration beyond batch optimization into other Green AI domains.

This work lays a robust foundation for standardized, transparent, and sustainable decision-making in machine learning, supporting the transition toward responsible AI innovation.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization and methodology for the systematic literature review and multi-dimensional evaluation framework, T.A.A. and A.D.C.; formal analysis involving data synthesis, quantitative scoring (Green AI Score, EDNA), and creation of visualizations (heatmap, bar charts, bubble plots, correlation plot), T.A.A., A.D.C., H.S., Z.Y.Z, E.Y.Z., and W.P.P.; investigation including literature search, data curation, and initial analysis of batch optimization techniques, T.A.A., A.D.C., H.S., Z.Y.Z, E.Y.Z., and W.P.P.; writing—original draft preparation, T.A.A., I.Q. and Z.Y.Z.; writing—review and editing, T.A.A., A.D.C., H.S., Z.Y.Z., E.Y.Z., and W.P.P.; supervision and project administration, A.D.C; all authors have read and agreed to the published version of the manuscript.

ACKNOWLEDGMENT

The authors express their gratitude to the School of Automation, Nanjing University of Information Science and Technology, China, and JMRAI Co., Ltd., Jiangyin, China, for their invaluable support and resources that contributed significantly to this research.

REFERENCES

[1] S. P. Ekanayake, T. Shah, and S. Evans, "Optimizing emissions for machine learning training," in *Proc. 2023 IEEE Conference on Technologies for Sustainability (SusTech)*, 2023, pp. 237–238. doi: 10.1109/SusTech57309.2023.10129569

[2] C. C. Lee and J. Yan, "Will artificial intelligence make energy cleaner? Evidence of nonlinearity," *Applied Energy*, vol. 363, 123081, 2024. doi: 10.1016/j.apenergy.2024.123081

[3] M. Song, H. Pan, Z. Shen, and K. Tamayo-Verleene, "Assessing the influence of artificial intelligence on the energy efficiency for sustainable ecological products value," *Energy Economics*, vol. 131, 107392, 2024. doi: 10.1016/j.eneco.2024.107392

[4] P. Chen, Z. Chu, and M. Zhao, "The road to corporate sustainability: The importance of artificial intelligence," *Technology in Society*, vol. 76, 102440, 2024. doi: 10.1016/j.techsoc.2023.102440

[5] V. Liu and Y. Yin, "Green AI: Exploring carbon footprints, mitigation strategies, and trade offs in large language model training," *Discover Artificial Intelligence*, vol. 4, no. 1, 49, 2024. doi: 10.1007/s44163-024-00149-w

[6] R. Verdecchia, J. Sallou, and L. Cruz, "A systematic review of green AI," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 13, no. 4, 2023. doi: 10.1002/widm.1507

[7] M. Barlaud and F. Guyard, "Learning sparse deep neural networks using efficient structured projections on convex constraints for green AI," in *Proc. 2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 1566–1573. doi: 10.1109/ICPR48806.2021.9412162

[8] Q. Cai, Y. K. Lal, H. Trivedi *et al.*, "IrEne: Interpretable energy prediction for transformers," in *Proc. the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021, pp. 2145–2157.

[9] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research," in *Proc. the AAAI Conference on Artificial Intelligence*, 2020, pp. 13693–13696.

[10] L. F. W. Anthony, B. Kanding, and R. Selvan, "Carbontracker: Tracking and predicting the carbon footprint of training deep learning models," ArXiv Preprint, arXiv:2007.03051, 2020.

[11] M. Kumar, X. Zhang, L. Liu *et al.*, "Energy-efficient machine learning on the edges," in *Proc. 2020 IEEE international parallel and distributed processing symposium Workshops (IPDPSW)*, 2020, pp. 912–921. https://doi.org/10.1109/IPDPSW50202.2020.00153

[12] R. M. Hampau, M. Kaptein, R. Van Emden *et al.*, "An empirical study on the performance and energy consumption of AI containerization strategies for computer-vision tasks on the edge," in *Proc. the 26th International Conference on Evaluation and Assessment in Software Engineering*, 2022, pp. 50–59.

[13] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI," *Communications of the ACM*, vol. 63, no. 12, pp. 54–63, 2020. https://doi.org/10.1145/3381831

[14] L. Gaur, A. Afaq, G. K. Arora, and N. Khan, "Artificial intelligence for carbon emissions using system of systems theory," *Ecological Informatics*, vol. 76, 102165, 2023. doi: 10.1016/j.ecoinf.2023.102165

[15] A. Harlap *et al.*, "PipeDream: Fast and efficient pipeline parallel DNN training," arXiv Preprint, arXiv:1806.03377, 2018.

[16] P. Qi, X. Wan, G. Huang, and M. Lin, "Zero bubble pipeline parallelism," arXiv Preprint, arXiv:2401.10241, 2023.

[17] D. Narayanan *et al.*, "Efficient large-scale language model training on GPU clusters using megatron-LM," in *Proc. the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–15. doi: 10.1145/3458817.3476209

[18] S. Tyagi and P. Sharma, "Taming resource heterogeneity in distributed ML training with dynamic batching," in *Proc. 2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, 2020, pp. 188–194. doi: 10.1109/ACSOS49614.2020.00041

[19] Z. Huang, B. Jiang, T. Guo, and Y. Liu, "Measuring the impact of gradient accumulation on cloud-based distributed training," in *Proc. ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, 2023, pp. 344–354. doi: 10.1109/CCGrid57682.2023.00040

[20] Z. Hong and C. P. Yue, "Efficient-grad: Efficient training deep convolutional neural networks on edge devices with gradient optimizations," *ACM Transactions on Embedded Computing Systems*, vol. 21, no. 2, pp. 1–24, 2022. doi: 10.1145/3504034

[21] Y. Zhang *et al.*, "Adam accumulation to reduce memory footprints of both activations and gradients for large-scale DNN training," arXiv Preprint, arXiv:2305.19982, 2023.

[22] H. Face, "Hugging face—The AI community building the future," *Hugging Face, dostupno na: https://huggingface.co/*, 2024.

[23] J. Lamy-Poirier, "Breadth-first pipeline parallelism," *Proceedings of Machine Learning and Systems*, vol. 5, pp. 48–67, 2023

[24] OpenVINO. (2022). OpenVINO API 2.0. [Online]. Available: https://docs.openvino.ai/2023.3/documentation.html

[25] University of Pau and Pays de l'Adour. (2023). Green AI UPPA. [Online]. Available: https://greenai-uppa.github.io/

[26] J. Rezaei, "Best-worst multi-criteria decision-making method," *Omega*, vol. 53, pp. 49–57, 2015. doi: 10.1016/j.omega.2014.11.009

[27] B. Ceballos, M. T. Lamata, and D. A. Pelta, "A comparative analysis of multi-criteria decision-making methods," *Progress in Artificial Intelligence*, vol. 5, pp. 315–322, 2016. doi: 10.1007/s13748-016-0093-1

[28] NVIDIA Developer. (2024). Get started with the NVIDIA DeepStream SDK. [Online]. Available: https://developer.nvidia.com/deepstream-getting-started

[29] Varun. (2022). Top 10 sources to find computer vision and ai models. [Online]. Available: https://learnopencv.com/top-10-sources-to-find-computer-vision-and-ai-models/

[30] AWS. (2024). Machine learning service—Amazon SageMaker AI. [Online]. Available: https://aws.amazon.com/cn/pm/sagemaker/

[31] Y. Zhang *et al.*, "Meta-transformer: A unified framework for multimodal learning," arXiv Preprint, arXiv:2307.10802, 2023.

[32] AI-benchmark. Is your smartphone ready for AI? [Online]. Available: https://ai-benchmark.com/

[33] Y. I. Alzoubi and A. Mishra, "Green artificial intelligence initiatives: Potentials and challenges," *Journal of Cleaner Production*, vol. 468, 143090, 2024. doi: 10.1016/j.jclepro.2024.143090

[34] Y. I. Alzoubi and A. Mishra, "Green blockchain—A move towards sustainability," *Journal of Cleaner Production*, vol. 430, 139541, 2023. doi: 10.1016/j.jclepro.2023.139541

[35] S. A. Budennyy *et al.*, "Eco2ai: Carbon emissions tracking of machine learning models as the first step towards sustainable AI," *Doklady Mathematics*, vol. 106, no. S1, pp. S118–S128, 2022. doi: 10.1134/S1064562422060230

[36] CodeCarbon. (2024). Track and reduce CO_2 emissions from your computing. [Online]. Available: https://codecarbon.io/

[37] Zeus. (2022). Zeus. Carbon-aware zeus. [Online]. Available: https://github.com/ml-energy/zeus