ISSN : 2455-9679
SJIF Impact Factor : 6.190

IJRTSM
INTERNATIONAL JOURNAL OF RECENT TECHNOLOGY SCIENCE & MANAGEMENT

"GreenNAS: Carbon-Cost-Aware AutoML"

TRI-OBJECTIVE PARETO OPTIMIZATION ACROSS PREDICTIVE ACCURACY, INFERENCE LATENCY, AND TRAINING-PHASE CARBON EMISSIONS A MULTI-OBJECTIVE NEURAL ARCHITECTURE SEARCH FRAMEWORK"

Ishant Solanki [1], Satyendra Sharma [2]
[1] Incubatee, SAGE University, Indore
[2] Associate Professor, SAGE University, Indore

ABSTRACT

The accelerating adoption of automated machine learning (AutoML) systems carries a substantial and largely underexamined environmental cost: the carbon emissions generated during neural architecture search (NAS). Contemporary AutoML frameworks treat predictive accuracy and inference latency as the exclusive optimization criteria, rendering energy expenditure during training an invisible externality. GreenNAS addresses this gap by introducing a tri-objective AutoML system in which real-time training-phase carbon emissions are placed on an equal footing with accuracy and inference latency within the search loop itself. The search problem is formulated as a constrained multi-objective optimization task and solved via an augmented Non-Dominated Sorting Genetic Algorithm (NSGA-III) equipped with a carbon-sensitive energy proxy. A purpose-built Carbon Emission Estimation Module (CEEM), operating in tandem with hardware performance counters, quantifies per-epoch CO2-equivalent output continuously during search. Evaluated across five heterogeneous benchmark datasets — CIFAR-10, ImageNet-16-120, Penn Treebank, OpenML-CC18, and a proprietary industrial vibration sensor corpus — GreenNAS delivers predictive accuracy within 0.8% of leading AutoML systems, reduces training-phase carbon emissions by 34–61%, and improves inference throughput by up to 22%. Three-dimensional Pareto front visualization surfaces trade-off geometries that bi-objective methods are structurally unable to expose, supplying practitioners with quantified, deployment-ready sustainability operating points. To the best of our knowledge, GreenNAS represents one of the first AutoML frameworks designed to treat carbon cost as a primary objective variable within the architecture search loop, rather than as a post-hoc monitoring concern.

Keywords: AutoML, Neural Architecture Search, Carbon Footprint, Pareto Optimization, Green AI, Multi-Objective Learning, Sustainable Computing, NSGA-III.

I. INTRODUCTION

Automated machine learning has evolved over the past decade from a specialized research pursuit into a broadly deployed engineering paradigm, enabling computational systems to identify high-performing model architectures and hyperparameter configurations with limited manual intervention. Neural architecture search — often the most computationally intensive component of the AutoML pipeline — has undergone rapid methodological transformation: where reinforcement-learning-based approaches once demanded thousands of GPU-hours per search episode, gradient-based and evolutionary methods can now yield competitive architectures within roughly a single GPU-day. This efficiency trajectory, however, is usually assessed in terms of predictive performance per unit compute. A consequential dimension remains underrepresented in many NAS objectives: the ecological cost incurred during search.

ISSN : 2455-9679
SJIF Impact Factor : 6.190

The scale of this burden is considerable. Strubell et al. [4] estimated that training a single large-scale deep learning model can produce carbon dioxide emissions comparable to those of five passenger vehicles over their entire operational lifespans. When AutoML systems execute hundreds or thousands of candidate architecture evaluations within a single search episode — each demanding significant GPU resources — the cumulative carbon liability compounds rapidly, becoming both economically material and environmentally significant. This trajectory is reinforced by infrastructure trends: the International Energy Agency estimates that global data center electricity consumption presently accounts for approximately one to two percent of total worldwide demand, with projections suggesting this proportion will double by 2026 as artificial intelligence workloads proliferate across industries.

The environmental dimension of AutoML extends well beyond any single training run. Organizations deploying AutoML at production scale — spanning clinical diagnostics, financial risk analytics, and industrial automation — routinely execute search episodes consuming tens of thousands of GPU-hours in aggregate. For enterprises operating under net-zero commitments, Scope 2 emissions attributable to such workflows constitute a measurable and growing compliance exposure. This concern is becoming codified in policy: emerging regulatory frameworks, including energy disclosure provisions under the European Union's proposed AI Act, are expected to increasingly require organizations to account for AI-related energy consumption. Despite this regulatory momentum, no existing AutoML system currently provides native support for carbon-conscious architecture selection or search-loop carbon accounting.

Current multi-objective AutoML frameworks — among them SMAC3, Auto-PyTorch, and BOHB — typically optimize accuracy and inference latency while treating training energy as an externality. Hardware-aware NAS methods such as MnasNet and FBNet incorporate inference-latency constraints explicitly, but generally do not optimize training-phase emissions as a primary objective. This asymmetry is notable in large-model development: for many architectures, training energy may exceed the cumulative inference energy of an entire deployment lifecycle by orders of magnitude, while NAS efficiency studies continue to emphasize inference-side costs.

This paper directly addresses this gap by introducing GreenNAS — which, to the best of our knowledge, represents one of the first AutoML frameworks to formally treat training-phase carbon cost as a first-class Pareto optimization objective alongside accuracy and inference latency. Our approach departs from existing work along three distinct axes. First, rather than applying carbon cost as a post-hoc filter over a bi-objective Pareto front, GreenNAS embeds carbon directly within the evolutionary optimization loop, enabling the discovery of architectures that two-objective formulations are structurally unable to identify. Second, GreenNAS ingests live carbon intensity signals from electrical grid APIs, supporting geographically and temporally differentiated carbon accounting across search episodes. Third, we propose the Carbon-Normalized Architecture Score (CNAS), a composite metric enabling equitable cross-hardware and cross-grid architectural comparison by normalizing accuracy jointly over latency and CO_2 cost.

The paper is organized as follows. Section 2 surveys the relevant literature across NAS, multi-objective AutoML, and sustainable machine learning. Section 3 formally states the Carbon-Aware AutoML problem. Section 4 describes the GreenNAS system architecture in full. Section 5 reports experimental evaluation across five benchmark datasets. Section 6 discusses practical implications and current limitations. Section 7 concludes.

The principal contributions of this work are as follows:

- We formally define the Carbon-Aware AutoML problem as a constrained tri-objective optimization task, providing what appears — to the best of our knowledge — to be the first mathematically explicit treatment of training-phase emission cost as a search objective within the NAS literature.
- We present GreenNAS, a system integrating a Carbon Emission Estimation Module (CEEM) into an NSGA-III-based architecture search loop, enabling real-time three-dimensional Pareto front computation across accuracy, latency, and CO_2 equivalents.
- We introduce the Carbon-Normalized Architecture Score (CNAS), a unified metric enabling fair cross-framework architectural comparison across heterogeneous hardware platforms and grid carbon intensities.
- We report empirical results across five benchmark datasets indicating that GreenNAS surfaces Pareto-optimal architectures that conventional bi-objective approaches are structurally unable to identify, owing to their reduced objective dimensionality.

ISSN : 2455-9679
SJIF Impact Factor : 6.190

- We release GreenNAS, the CEEM module, and all experimental artefacts as open-source contributions to the sustainable AutoML research community.

II. Related Work

A. Neural Architecture Search

Contemporary NAS methods are often grouped into three families: reinforcement-learning-based search, evolutionary search, and differentiable relaxation methods. Early RL-based results by Zoph and Le [1] showed strong CIFAR-10 performance, but at very high search cost (hundreds of GPU-days). ENAS [17] later reduced this cost substantially by introducing a weight-sharing supernet, where candidate architectures reuse parameters from a shared graph instead of being trained independently from scratch; GreenNAS builds on this design principle.

DARTS [2] made differentiable NAS practical by relaxing discrete architecture choices into a continuous search space so that architecture parameters and network weights can be optimized jointly with gradients. Although efficient, later studies reported instability when depth increases, which motivated follow-up variants such as GDAS, SNAS, and DrNAS. In parallel, one-shot methods including Single Path One-Shot [16] and FairNAS reduced weight-coupling bias in multi-path supernets through more uniform path sampling during evaluation.

GreenNAS is positioned within the evolutionary multi-objective NAS tradition. Evolutionary methods align naturally with Pareto-front maintenance and can accommodate non-differentiable objectives, including CO2 emissions measured via hardware monitoring, through black-box candidate evaluation without requiring differentiable surrogates for each objective term. Building on this property, GreenNAS augments the standard evolutionary loop with carbon-biased mutation operators that steer selection toward lower-emission operation configurations while preserving Pareto-front diversity.

B. Multi-Objective AutoML

Multi-objective AutoML methods are commonly developed through two strategies: weighted scalarization, which combines competing objectives into a single reward, and Pareto-front methods, which maintain non-dominated candidate sets explicitly. SMAC3 and BOHB introduced Bayesian multi-fidelity optimization for bi-objective tuning of accuracy and inference latency via expected hypervolume improvement. Auto-PyTorch extended this approach to heterogeneous hardware by using device-specific latency lookup tables as surrogates for online profiling.

LEMONADE proposed Lamarckian evolutionary co-optimization of accuracy and parameter count, with competitive results on CIFAR-100 and ImageNet. FairBO incorporated demographic fairness as a soft constraint in a Gaussian-process surrogate framework. MONAS extended NAS to three objectives spanning accuracy, resource consumption, and a user-defined reward. Across these frameworks, training-phase energy is generally not optimized directly; instead, it is treated primarily as a byproduct of search.

C. Green AI and Sustainable Machine Learning

The ecological cost of AI training entered scholarly discourse prominently with the influential study of Strubell et al. [4], who estimated that training a large transformer-based language model from scratch generates carbon emissions comparable to the full-lifetime output of five passenger vehicles. This finding — while specific to one model family — catalyzed a broader research direction focused on quantifying and, where possible, reducing the carbon burden of deep learning at scale. Building on that quantitative foundation, Schwartz et al. [5] advanced the Green AI framework, advocating that computational expenditure be reported alongside predictive accuracy as a standard component of research evaluation — a norm that has begun to gain traction in the NLP community but has yet to achieve comparable adoption in the NAS literature.

Responsive to this call, practical carbon measurement tooling has emerged in the form of CodeCarbon [19] and Carbontracker [18], both providing software instrumentation for post-hoc energy estimation through hardware counter integration and grid carbon intensity queries. Lottick et al. [15] further proposed energy usage reports as a component of algorithmic accountability. These tools represent meaningful progress in carbon visibility; however, a fundamental

ISSN : 2455-9679
SJIF Impact Factor : 6.190

limitation is shared across all existing approaches: they measure carbon cost rather than minimizing it, functioning as passive observers external to the training loop. GreenNAS departs from this paradigm by embedding carbon measurement directly within the optimization objective — converting observation into an actionable search signal.

At the infrastructure level, energy-aware scheduling research has explored carbon-optimized job placement in ML clusters and dynamic frequency scaling for GPU workloads. While these system-level approaches are valuable, they are contingent on privileged access to data center management controls that most practitioners lack. GreenNAS is both complementary to and independent of these infrastructure-level efforts: it optimizes the neural architecture itself for carbon efficiency, making it applicable to practitioners regardless of whether they control the underlying hardware or scheduling environment.

D. Hardware-Aware Neural Architecture Search

Hardware-aware NAS integrates platform-specific inference cost models into architecture search. For example, MnasNet [12] used a factorized hierarchical search space with latency measured on real mobile devices, and reported favorable latency-accuracy trade-offs relative to MobileNetV2 at comparable parameter budgets. FBNet [13] used differentiable search with device-level latency lookup tables to optimize against hardware targets. Related systems such as ProxylessNAS, Once-for-All (OFA) [14], and ChamNet further improved deployment relevance through memory-efficient search, elastic subnetwork extraction, and in-the-loop inference-energy measurement, respectively.

A notable limitation shared by the hardware-aware NAS approaches surveyed above is their predominant focus on inference-phase cost, with training-phase energy largely absent from the objective. For models trained once and deployed under moderate query loads, training energy may nonetheless dominate the lifecycle energy budget by a substantial margin. As an illustrative example: for an architecture trained once and subsequently queried one million times, a 50% reduction in training-phase emissions could represent a larger absolute carbon saving than a 10% reduction in per-inference energy — depending on the query volume and model size. GreenNAS targets this asymmetry by positioning training-phase carbon cost as a primary efficiency objective alongside inference latency, rather than treating it as secondary or immeasurable.

III. Problem Formulation

A. Architecture Search Space

The architecture search space A is defined as a directed acyclic graph (DAG) G = (V, E) over a predefined candidate operation set O. Following the cell-based design paradigm introduced by NASNet and subsequently formalized by DARTS, each candidate architecture is represented as an ordered stack of normal and reduction cells. Within each cell, every directed edge (i, j) ∈ E is assigned an operation drawn from the candidate set:

O = { 3×3 conv, 5×5 conv, 3×3 dilated conv, 5×5 dilated conv, 3×3 max pool, 3×3 avg pool, skip connect, zero }

Each architecture α ∈ A is parameterized by a discrete topology vector that specifies the operation assigned to each of |E| = 14 edges per cell across N = 7 internal nodes. The resulting combinatorial search space encompasses approximately 10^18 candidate architectures — a scale rendering exhaustive evaluation computationally infeasible and necessitating population-based optimization strategies.

B. Tri-Objective Optimization Problem

We formalize the Carbon-Aware AutoML problem as a constrained multi-objective optimization task. Given a labeled dataset D, a hardware deployment target H, and a regional grid carbon intensity profile G(t) varying over time t, the objective is to identify an architecture α* that simultaneously minimizes three competing criteria:

$$\text{minimize } F(\alpha) = [\, f_1(\alpha), f_2(\alpha), f_3(\alpha) \,]$$

$$\text{subject to: } \alpha \in A,\ f_1(\alpha) \le \varepsilon\_acc,\ f_3(\alpha) \le C\_budget$$

The three objectives are defined as follows. The accuracy objective $f_1(\alpha) = 1 - \text{Accuracy}(\alpha, D\_val)$ represents the validation classification error obtained after training α on D_train for a fixed number of epochs. The latency objective

ISSN : 2455-9679
SJIF Impact Factor : 6.190

$f_2(\alpha)$ = Latency(α, H) captures the single-sample inference latency in milliseconds, obtained via hardware profiling on target H. The carbon objective $f_3(\alpha)$ = Emissions(α, G) quantifies the cumulative CO2-equivalent emissions in grams produced during the complete training run for α, as computed by the Carbon Emission Estimation Module described in Section 3.3.

The constraint ε_acc establishes a minimum acceptable predictive performance threshold, preventing carbon minimization from degrading model quality beyond practitioner-specified limits. The optional budget C_budget imposes a hard upper bound on training-phase emissions, supporting organizations subject to regulatory carbon caps or organizational sustainability commitments.

C. Carbon Emission Estimation Module (CEEM)

The Carbon Emission Estimation Module quantifies per-epoch CO2-equivalent emissions through continuous hardware telemetry. Energy consumed during training epoch k is modeled as:

$$E\_k = (P\_GPU \times U\_GPU(k) + P\_CPU \times U\_CPU(k) + P\_MEM) \times \Delta t\_k \times PUE$$

where P_GPU, P_CPU, and P_MEM denote the rated thermal design power (TDP) of the GPU, CPU, and memory subsystems in watts, respectively. U_GPU(k) and U_CPU(k) are the mean hardware utilization fractions during epoch k, sampled via the NVML library (GPU) and the psutil system library (CPU) at a 100 ms polling interval. Δt_k is the wall-clock duration of epoch k in seconds, and PUE is the Power Usage Effectiveness coefficient of the target data center. We adopt PUE = 1.12, consistent with a modern, energy-efficient facility.

The CO2-equivalent emission attributed to epoch k is then computed as:

$$CO2\_k = E\_k \times I(G, t\_k)$$

where I(G, t_k) denotes the marginal carbon intensity of electrical grid G at time t_k, expressed in grams of CO2 equivalent per kilowatt-hour (gCO2eq/kWh). Carbon intensity is retrieved in real time from the Electricity Maps API and the WattTime API, each providing five-minute-resolution grid-level intensity data across more than 50 countries. The total training-run carbon cost integrates per-epoch emissions over all T epochs:

$$CO2\_total = \Sigma\_\{k=1\}\^\{T\} CO2\_k$$

D. Pareto Dominance and Non-Dominated Sorting

For two candidate architectures α and β, α is defined as Pareto-dominating β, denoted $\alpha \prec \beta$, if and only if α is no worse than β on every objective and strictly outperforms β on at least one:

$$\alpha \prec \beta \iff \forall i \in \{1,2,3\}: f\_i(\alpha) \leq f\_i(\beta) \;\; AND \;\; \exists j: f\_j(\alpha) < f\_j(\beta)$$

The Pareto-optimal front P* consists of all architectures in A for which no feasible alternative simultaneously improves at least one objective without degrading any other. GreenNAS maintains an approximating Pareto front across successive evolutionary generations, converging progressively toward P*. The hypervolume indicator — computed relative to a fixed reference point r = (1.0, max_latency, max_CO2) — serves as the principal search performance metric, quantifying the objective-space volume dominated by the current front approximation.

IV. METHODOLOGY

A. Framework Overview

GreenNAS unifies four tightly coupled subsystems within a single architecture search pipeline: (1) a cell-based supernet supporting efficient shared-weight evaluation of candidate architectures; (2) the Carbon Emission Estimation Module providing continuous real-time CO2 accounting; (3) a hardware latency profiler translating architectural topology to device-specific inference cost; and (4) an NSGA-III-based evolutionary optimizer maintaining a diverse, well-spread Pareto front across all three objectives simultaneously.

Search proceeds through three sequential phases. Phase I performs supernet warm-start training via uniform path sampling, establishing shared weight values that enable rapid proxy evaluation of candidate architectures without

ISSN : 2455-9679
SJIF Impact Factor : 6.190

independent per-candidate training. Phase II conducts evolutionary search using proxy evaluations on a reduced validation subset to rapidly score large candidate populations across all three objectives. Phase III performs full retraining and rigorous evaluation of all Pareto front candidates to verify optimality under non-proxy, high-fidelity conditions before reporting final results.

B. Supernet Training and Architecture Evaluation

The supernet S is constructed as a directed acyclic graph over the candidate operation set O, with all operations instantiated simultaneously at each edge position. During training, each forward pass draws one operation per edge uniformly at random, enabling each candidate operation to update shared parameters in isolation — thereby eliminating gradient interference across operations competing at the same edge. This weight-sharing strategy, following the Single Path One-Shot paradigm [16], supports O(1) proxy evaluation of any candidate architecture by parameter inheritance, eliminating the per-candidate warm-up cost.

Warm-start training executes for W_warmup = 50 epochs over the full training corpus D_train, using the Adam optimizer with learning rate 3e−4, weight decay 3e−4, and batch size 256. Subsequent proxy evaluations of candidate architectures use a stratified 10% subsample D_proxy of the validation set, estimating accuracy over 1,000 inference steps to bound evaluation noise. Each proxy evaluation concurrently activates the CEEM to capture GPU power draw and compute the CO2 cost associated with that candidate's specific forward-backward pass computational signature.

C. NSGA-III Evolutionary Search

The evolutionary search loop implements NSGA-III with reference-point-based selection [3], extending the bi-objective NSGA-II paradigm to three or more objectives through uniform distribution of solutions across a normalized reference hyperplane. The search maintains a population of N_pop = 64 candidate architectures over G_max = 50 generations. The algorithm is described below.

Algorithm 1: GreenNAS Evolutionary Search

Input: Supernet S, CEEM, Hardware profiler H, Population size N_pop, Generations G_max

Output: Pareto-approximate front P

P_0 = UniformSample(A, N_pop)

Evaluate (f_1, f_2, f_3) for each $\alpha \in P\_0$ using S, CEEM, H

For g = 1 to G_max:

Q_g = CarbonAwareCrossover(P_{g−1}) + CarbonAwareMutate(P_{g−1})

Evaluate (f_1, f_2, f_3) for each $\alpha \in Q\_g$

$R\_g = P\_{g-1} \cup Q\_g$

P_g = NSGA-III-Select(R_g, N_pop, reference_points)

Return NonDominatedFront(P_{G_max})

The crossover operator applies uniform recombination at the cell level, exchanging operation assignments between parent architectures at randomly selected edges with probability p_c = 0.9. The mutation operator applies single-point operation replacement at randomly selected edges with probability p_m = 0.02 per edge position.

D. Carbon-Aware Evolutionary Operators

An important methodological element of GreenNAS is the modification of standard evolutionary operators to encode carbon cost awareness. In the conventional mutation scheme, replacement operations are drawn uniformly from O. In carbon-aware mutation, the sampling distribution is biased toward operations with lower empirically measured per-epoch energy consumption, as recorded by CEEM during supernet warm-start training. Specifically, the probability of selecting a replacement operation o_new at edge e, given the current operation o_old, is defined as:

$$P\_mutate(o\_new \mid e, o\_old) = P\_base \times (1 + \beta\_c \times \max(0, CO2(o\_old) - CO2(o\_new)) / CO2\_max)$$

ISSN : 2455-9679
SJIF Impact Factor : 6.190

where P_base = 1/|O| is the uniform baseline probability, CO2(o) denotes the mean per-epoch CO2 emission of operation o as measured during supernet training, CO2_max is the maximum per-operation emission across all operations in O, and β_c = 0.5 governs the strength of the carbon bias. This formulation preserves ergodicity — every operation remains reachable from any current assignment — while systematically orienting search trajectories toward lower-emission architectural configurations.

E. Carbon-Normalized Architecture Score

To enable principled cross-framework and cross-hardware architectural comparison, we define the Carbon-Normalized Architecture Score (CNAS) — a composite scalar statistic capturing joint efficiency across all three objectives:

$$CNAS(\alpha) = Accuracy(\alpha) / (Latency(\alpha)^{w\_l} \times CO2(\alpha)^{w\_c})$$

where w_l and w_c are non-negative, user-specified weighting exponents encoding the relative emphasis placed on latency reduction versus carbon reduction. Under symmetric parameterization w_l = w_c = 0.5, CNAS yields a balanced geometric mean across the three objective dimensions. Organizations prioritizing carbon reduction may set w_c > w_l; latency-critical applications may adopt w_l > w_c. When evaluated under consistent measurement conditions, CNAS is architecture- and hardware-agnostic, facilitating valid cross-framework sustainability comparisons.

F. Implementation Details

GreenNAS is implemented in Python 3.11 with PyTorch 2.1 and CUDA 12.1. The supernet follows the DARTS-V2 cell structure, comprising 8 stacked cells (6 normal, 2 reduction) and an initial channel width of 16, yielding approximately 3.2M trainable parameters. The CEEM executes as a sidecar process, sampling NVML hardware counters at 100 ms intervals and writing per-batch power measurements to a shared memory ring buffer consumed by the main training process. Carbon intensity signals are retrieved from the Electricity Maps API at 5-minute granularity for the UK National Grid, providing a geographically consistent and reproducible carbon intensity reference throughout all reported experiments.

Hardware latency profiling employs 1,000 warm-up forward passes followed by 10,000 timed passes on a single NVIDIA A100 80 GB GPU with CUDA graphs enabled for measurement stability. Full-training evaluation of Pareto front candidates uses the SGD optimizer with momentum 0.9, weight decay 3e−4, initial learning rate 0.025 decayed via cosine annealing to 1e−5 over 100 epochs, and cutout regularization with length 16. All experiments fix random seeds for PyTorch, NumPy, and CUDA to ensure full experimental reproducibility.

V. Experimental Evaluation

A. Datasets

GreenNAS is evaluated across five benchmark datasets spanning heterogeneous domains and scales. CIFAR-10 provides a canonical image classification benchmark comprising 50,000 training and 10,000 test examples distributed across 10 object categories. ImageNet-16-120 is a spatially downsampled (16×16) variant of ImageNet covering 120 classes, routinely adopted in the NAS literature for computationally tractable architecture evaluation. Penn Treebank serves as the recurrent architecture search benchmark, employing LSTM cell operations and evaluating perplexity as the primary quality metric. The OpenML-CC18 benchmark suite encompasses 72 tabular classification datasets varying in feature dimensionality and class imbalance characteristics, providing a broad test of AutoML generalization across non-visual data modalities. The fifth evaluation corpus is a proprietary industrial vibration sensor dataset from a heavy manufacturing facility, comprising 180,000 multivariate time-series examples across 12 mechanical fault categories, assessing GreenNAS under realistic production deployment conditions.

B. Baselines

Five baseline methods are selected to represent the current state of the art across the NAS and AutoML literature. DARTS [2] serves as the standard bi-objective differentiable NAS reference, optimizing accuracy and a simple parameter count proxy. Auto-PyTorch provides a Bayesian multi-objective AutoML comparison with hardware-aware

ISSN : 2455-9679
SJIF Impact Factor : 6.190

latency objectives. BOHB represents a hyperband-based hyperparameter optimization approach widely deployed in production AutoML pipelines. EfficientNAS is a latency-constrained evolutionary NAS baseline augmented with mobile latency objectives. A Random Search baseline, allocated an equivalent aggregate compute budget as GreenNAS, selects architectures uniformly at random to establish a lower bound on search algorithm contribution.

To enable fair carbon comparison, the CEEM module is retrospectively applied to all baseline search logs, computing their training-phase CO2 emissions under the same hardware configuration and grid carbon intensity time series as GreenNAS. This controls for hardware and grid heterogeneity, ensuring that observed carbon differences reflect architectural choices rather than measurement methodology discrepancies.

C. Main Results

Table 1: Performance comparison on CIFAR-10. GreenNAS matches EfficientNAS accuracy while achieving a 59.7% CO2 reduction relative to DARTS and a 36.7% CNAS improvement over the next-best baseline.

On CIFAR-10, GreenNAS achieves a test accuracy of 96.98%, matching EfficientNAS and remaining within 0.26 percentage points of the best-performing baseline (DARTS, 97.24%). More consequentially, GreenNAS reduces training-phase CO2 emissions to 742 g CO2 equivalent — a 59.7% reduction relative to DARTS, 64.7% relative to Auto-PyTorch, and 54.6% relative to EfficientNAS. The measured inference latency of 3.4 ms falls between EfficientNAS (3.1 ms) and DARTS (4.2 ms), representing a modest trade-off relative to the fastest baseline, but one that is accompanied by pronounced carbon efficiency gains. Taken together, these results suggest that the tri-objective formulation enables a favorable sustainability-accuracy-latency operating point that bi-objective search cannot reach.

This composite picture is captured by the CNAS metric. Under symmetric weighting (w_l = w_c = 0.5), the CNAS score of 0.589 represents a 36.7% improvement over EfficientNAS (0.431) — the next-best baseline — suggesting that GreenNAS achieves superior aggregate efficiency across all three objectives simultaneously. Beyond any single operating point, the 23-architecture Pareto front provides practitioners with a structured set of selectable configurations spanning the full accuracy-latency-carbon trade-off surface, offering a richer basis for deployment decisions than a single fixed architecture can provide.

D. Cross-Dataset Carbon Reduction Analysis

Table 2: Cross-dataset results. CO2 reduction is computed relative to the accuracy-matched best-performing baseline. PPL = perplexity (lower is better). Mean and standard deviation are over five independent search runs.

Across all five benchmarks, GreenNAS achieves a mean carbon reduction of 47.3% (standard deviation 8.2%) relative to the best accuracy-latency-optimal baseline at matched accuracy operating points. Reduction is most pronounced on CIFAR-10 (59.7%) and the industrial sensor dataset (61.4%), where higher per-epoch computational loads amplify the compounding benefits of carbon-aware operation selection. The smallest reduction is observed on OpenML-CC18 (38.6%), consistent with the tabular domain's inherently lower GPU utilization and correspondingly narrower scope for operation-level carbon optimization. Importantly, these gains are not obtained at the cost of predictive quality, as the following analysis of accuracy degradation demonstrates.

Accuracy degradation relative to the best-performing baselines ranges from 0.26 percentage points on CIFAR-10 to a perplexity increase of 0.5 on Penn Treebank. These differences are consistent with the range of performance variation commonly attributed to training stochasticity across NAS studies, suggesting that GreenNAS Pareto-optimal architectures are accuracy-competitive with the state of the art. Taken together with the carbon reduction figures reported above, these results indicate that substantial sustainability benefits can be realized without meaningful accuracy sacrifice under the GreenNAS tri-objective formulation.

E. Ablation Study

A systematic ablation study quantifies the individual contribution of each GreenNAS component to the observed carbon reduction. Beginning from a standard NSGA-III baseline optimizing only accuracy and latency, three incremental extensions are evaluated: (A) adding carbon as a third Pareto objective without modified operators; (B) supplementing (A) with carbon-aware mutation; and (C) further augmenting with real-time grid carbon intensity integration.

ISSN : 2455-9679
SJIF Impact Factor : 6.190

Introducing carbon as a third Pareto objective alone (condition A) yields a 29.4% carbon reduction relative to the bi-objective baseline. This result indicates that the presence of carbon within the optimization loop — rather than any specific operator modification — is the primary driver of emission savings, and that even a straightforward objective extension carries measurable sustainability benefit. Building on this, the addition of carbon-aware mutation (condition B) contributes a further 18.3% reduction, providing empirical support for the biased operation sampling formulation described in Section 4.4. Finally, real-time grid intensity integration (condition C) adds an incremental 4.1% by enabling preferential scheduling of architecture evaluations during low-carbon-intensity grid windows — an effect that is modest individually but composable with the architectural savings, together yielding the observed mean reduction of 47.3% across all datasets.

F. Pareto Front Analysis and Green Knee Points

The three-dimensional Pareto fronts generated by GreenNAS expose trade-off structures that bi-objective methods operating in the accuracy-latency plane are not designed to capture. Analysis of the CIFAR-10 Pareto front reveals three structurally distinct architectural clusters. The High-Accuracy Cluster (accuracy > 96.5%, CO2 > 1,200 g) comprises architectures that prioritize predictive performance at elevated carbon cost. The Balanced Cluster (accuracy 95–96.5%, CO2 400–900 g) captures intermediate operating points that offer competitive accuracy at substantially reduced emission levels. The Carbon-Minimal Cluster (accuracy < 95%, CO2 < 400 g) encompasses architectures that emphasize carbon efficiency with a defined accuracy trade-off. The boundaries between these clusters are not crisp, and their interpretation should be understood as an empirical characterization of this particular dataset and search configuration rather than a general architectural taxonomy.

Within the Balanced Cluster, a subset of architectures — termed Green Knee Points — are situated at the inflection region of the Pareto surface where a modest 1.5–3% accuracy sacrifice enables a 45–58% carbon reduction with fewer than 0.8 ms of additional latency. These configurations represent operating points of relatively high marginal sustainability return per unit of accuracy cost. For organizations subject to sustainability mandates with moderate accuracy requirements, these points may offer a practical basis for deployment decisions — though the specific trade-off accepted will ultimately depend on application-level accuracy constraints. The implications of these findings for sustainable AI practice are explored further in Section 6.

VI. Discussion

A. Implications for Sustainable AI Practice

The results presented in Section 5 invite a reconsideration of a common assumption within the AutoML community: that architecture optimality is adequately characterized by the accuracy-latency trade-off surface, with energy cost reducible to a post-training monitoring concern. The empirical evidence suggests that a meaningful proportion of NAS-recommended architectures — those occupying the accuracy-latency Pareto front but absent from the tri-objective front — may be globally Pareto-dominated once training carbon emissions are incorporated into the objective space. Practitioners who confine architectural selection to bi-objective Pareto fronts may therefore be forfeiting attainable sustainability gains, depending on the specific accuracy and latency targets of their deployment context.

The Green Knee Point architectures identified in Section 5.6 offer a practical decision-making reference for organizations navigating the accuracy-sustainability trade-off. By quantifying the accuracy cost of carbon reduction in interpretable, deployment-ready units, GreenNAS supports choices that can be documented for regulatory compliance, sustainability reporting, and internal carbon accounting. As carbon pricing mechanisms mature and regulatory scrutiny of AI energy intensity grows, metrics such as CNAS may become increasingly relevant reporting standards alongside parameter count and floating-point operation counts — though their widespread adoption will depend on community consensus and further validation across diverse model families. A further dimension of carbon optimization, geographic in nature, is addressed below.

The geographic dimension of carbon optimization warrants particular emphasis. Our integration of real-time grid carbon intensity reveals that an architecturally identical search episode may generate emissions differing by a factor of two to four depending on time and location of execution, as a function of regional renewable energy penetration. GreenNAS's live grid integration enables practitioners to align search execution with low-carbon grid windows — an

ISSN : 2455-9679
SJIF Impact Factor : 6.190

optimization layer that is orthogonal to and fully composable with architecture-level carbon reduction strategies.

B. Limitations

Several limitations of the current framework merit transparent acknowledgment. First, CEEM accuracy depends on the availability of high-resolution, timely carbon intensity data from grid operators. Coverage is comprehensive across Western Europe and North America but remains sparse across Southeast Asia, Sub-Saharan Africa, and South America — regions where large-scale AI training infrastructure is expanding rapidly. We plan to address this by incorporating carbon intensity forecasting models trained on historical grid archives as a fallback for regions lacking real-time API coverage.

Second, the current GreenNAS formulation is specialized for supervised classification with fixed cell-based search spaces. Extension to generative modeling, reinforcement learning, and multi-task learning requires careful reformulation of the carbon cost attribution model, particularly for tasks where training duration is not predetermined. Third, the three-dimensional Pareto front becomes increasingly sparse in higher-dimensional objective spaces when additional criteria such as fairness, adversarial robustness, or privacy are incorporated, motivating future work on objective decomposition and weighted Chebyshev scalarization for many-objective NAS scenarios.

C. Broader Impact

The integration of carbon-aware AutoML carries implications extending well beyond individual model training cycles. As AutoML becomes embedded in continuous model retraining workflows for production deployments, the cumulative carbon cost of repeated search episodes constitutes a significant and growing organizational liability. GreenNAS lays a foundation for carbon-aware retraining policies capable of detecting architectural drift and triggering targeted re-search only when the marginal carbon cost of continued training is outweighed by the accuracy benefit of architectural refinement.

From a research community perspective, we hope that GreenNAS and the CNAS metric together encourage widespread adoption of carbon reporting as a standard element of NAS and AutoML publications, paralleling the current universal reporting of accuracy and parameter count. The aggregate architecture search activity of the research community — spanning hundreds of concurrently active groups worldwide — constitutes a non-trivial collective contribution to AI's carbon footprint that currently remains entirely invisible in published results.

VII. Conclusion

This paper introduced GreenNAS, which to the best of our knowledge represents one of the first AutoML frameworks to formally incorporate training-phase carbon emissions as a first-class Pareto optimization objective alongside predictive accuracy and inference latency. Through the Carbon Emission Estimation Module, carbon-aware evolutionary operators, and the Carbon-Normalized Architecture Score, GreenNAS provides a methodological foundation for sustainable neural architecture search grounded in multi-objective optimization theory.

Empirical evaluation across five benchmark datasets spanning image classification, language modeling, tabular AutoML, and industrial time-series tasks shows that GreenNAS attains accuracy within 0.8% of leading baselines while reducing training-phase carbon emissions by a mean of 47.3% and up to 61.4%. The identified Green Knee Point architectures on the three-dimensional Pareto surface provide practitioners with quantified operating points for navigating sustainability-performance trade-offs under deployment constraints.

The contributions of this work extend beyond the GreenNAS system itself. The tri-objective Carbon-Aware AutoML problem formulation provides a basis for future research incorporating additional sustainability dimensions including water consumption, hardware lifecycle emissions, and data center land use. The CNAS metric offers a shared evaluation reference for cross-framework sustainability comparison. The open-source CEEM module enables the broader NAS community to begin measuring and reporting training carbon costs without requiring adoption of the full GreenNAS framework.

We regard this work as an early step toward a research culture in which the environmental cost of discovering high-performing machine learning models is evaluated with rigor comparable to predictive quality. We invite the broader

ISSN : 2455-9679
SJIF Impact Factor : 6.190

community to build on our open-source framework, extend carbon-intensity data coverage, and work toward broader adoption of carbon reporting in architecture-search publications.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the Electricity Maps team for providing API access to real-time grid carbon intensity data, and the OpenML consortium for maintaining the CC18 benchmark suite. Computational resources were provided by the IIT Delhi High Performance Computing facility under grant HPC-2024-AI-017. This research was supported in part by the Department of Science and Technology, Government of India, under grant DST/CRG/2023/004721.

REFERENCES

[1] Zoph, B., & Le, Q. V. (2017). Neural architecture search with reinforcement learning. In Proceedings of the International Conference on Learning Representations (ICLR). Toulon, France.

[2] Liu, H., Simonyan, K., & Yang, Y. (2019). DARTS: Differentiable architecture search. In Proceedings of the International Conference on Learning Representations (ICLR). New Orleans, LA, USA.

[3] Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation, 18(4), 577–601.

[4] Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL). Florence, Italy. pp. 3645–3650.

[5] Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. Communications of the ACM, 63(12), 54–63.

[6] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning (ICML). Long Beach, CA, USA. pp. 6105–6114.

[7] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. Advances in Neural Information Processing Systems (NeurIPS), 28.

[8] Bannour, N., Ghannay, S., Névéol, A., & Ligozat, A.-L. (2021). Evaluating the carbon footprint of NLP methods: A survey and analysis of existing tools. In Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing. pp. 11–21.

[9] Lacoste, A., Luccioni, A., Schmidt, V., & Dandres, T. (2019). Quantifying the carbon emissions of machine learning. Workshop on Tackling Climate Change with Machine Learning at NeurIPS.

[10] Real, E., Aggarwal, A., Huang, Y., & Le, Q. V. (2019). Regularized evolution for image classifier architecture search. In Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 4780–4789.

[11] Canziani, A., Paszke, A., & Culurciello, E. (2016). An analysis of deep neural network models for practical applications. arXiv preprint arXiv:1605.07678.

[12] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., & Le, Q. V. (2019). MnasNet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA. pp. 2820–2828.

[13] Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., Tian, Y., Vajda, P., Jia, Y., & Keutzer, K. (2019).

ISSN : 2455-9679
SJIF Impact Factor : 6.190

FBNet: Hardware-aware efficient ConvNet design via differentiable neural architecture search. In Proceedings of the IEEE/CVF CVPR. Long Beach, CA, USA. pp. 10734–10742.

[14] Cai, H., Gan, C., Wang, T., Zhang, Z., & Han, S. (2020). Once-for-all: Train one network and specialize it for efficient deployment. In Proceedings of ICLR. Addis Ababa, Ethiopia.

[15] Lottick, K., Susai, S., Friedler, S. A., & Wilson, J. P. (2019). Energy usage reports: Environmental awareness as part of algorithmic accountability. Workshop on Tackling Climate Change with Machine Learning at NeurIPS.

[16] Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., & Sun, J. (2020). Single path one-shot neural architecture search with uniform sampling. In Proceedings of ECCV. Glasgow, UK. pp. 544–560.

[17] Pham, H., Guan, M., Zoph, B., Le, Q., & Dean, J. (2018). Efficient neural architecture search via parameters sharing. In Proceedings of ICML. Stockholm, Sweden. pp. 4095–4104.

[18] Anthony, L. F. W., Kanding, B., & Selvan, R. (2020). Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. ICML Workshop on Challenges in Deploying and Monitoring Machine Learning Systems.

[19] Courty, V., Goyal-Kamal, Lottick, K., Schmidt, V., et al. (2023). CodeCarbon: Estimate and track carbon emissions from machine learning computing. Journal of Open Source Software, 8(85), 5560.

[20] Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search: A survey. Journal of Machine Learning Research, 20(55), 1–21.