Journal of Theoretical and Applied Information Technology
15th May 2026. Vol.104. No.9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A LAYER-WISE ATTENTION CALIBRATION FRAMEWORK FOR DEEP NEURAL NETWORKS IN RESOURCE-CONSTRAINED ENVIRONMENTS

M. NAGABHUSHANA RAO[1], RAMESH BABU PITTALA[2], DR. K. VENU[3], Dr P N V V L PRAMILA RANI[4], GUNDALA VENKATA RAMA LAKSHMI [5], Dr. BOBY K GEORGE[6], Dr. GRK PRASAD[7]

[1]Corresponding Author, Professor, School of Computer Science & Technology, Malla Reddy (MR) Deemed to be University, Medchal, Malkajgiri, Telangana, India
[2]School of Engineering, Anurag University. Telangana, India
[3]Kongu Engineering college, perundurai,
[4]Department of Chemistry, Narasaraopeta Engineering College Autonomous
[5]Department of Computer Science and Engineering, CVR College of Engineering, Mangalpalli, Hyderabad
[6]Department of Production Engineering, APJAK Technological University, Thiruvananthapuram, Kerala,Professor, Government Engineering College, Thrissur
[7]Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Guntur District, India

E-mail: [1]mnraosir@gmail.com,2prameshbabu526@gmail.com, [3]venu.kalaimagal@gmail.com, [4]drpramilapippala@gmail.com, [5]gvrlaksmi@cvr.ac.in , [6]boby@gectcr.ac.in, [7]ramguda1978@gmail.com

ABSTRACT

More frequent use of deep neural networks in low-resource settings has revealed that conventional structures often cannot maintain both accuracy and computation efficiency at the same time. Solutions such as model pruning, quantization and knowledge distillation have been tried to reduce how complicated a model is, but they are not dynamic solutions. Still, most of these methods make errors, need to be retrained frequently and cannot adjust at run time to new challenges or system restrictions. Instead of similar strategies, this paper presents LWAC which regulates attention in different parts of the network when it's being used for inference purposes. Not like fixed compression which must be retrained, LWAC adds lightweight calibration units to the model. These units check relevance and demands of each layer and enable correct pathways, while blocking those that do not help. All of the experiments were validated on CIFAR-10, Tiny ImageNet, UCI HAR and MHEALTH, with MobileNetV2, ResNet-34 and CNN-LSTM hybrids serving as the architectures. Latency in inferencing and energy use drop by 15% to 22% in LWAC, even as its accuracy can improve by as much as 1.2%. Moreover, different types of analysis underscore that LWAC can make important predictions and clarify the decisions it reaches. By supporting flexible use of layers based on needs, LWAC brings strong deep learning capabilities to places where computers are under pressure.

Keywords: *Deep Neural Networks, Attention Calibration, Layer-Wise Modulation, Resource-Constrained Inference, Adaptive Computation, Edge AI, Energy-Efficient Learning, Model Optimization, Real-Time AI Systems, Context-Aware Attention Control*

1. INTRODUCTION

The increase in deep learning technologies over the past few years has made numerous breakthroughs in computer vision, natural language processing, autonomous systems and healthcare analytics. The main reason for their achievements is the deep neural network (DNN) which learns detailed features in data using several layers of modeled neurons. Especially, architectures with lots of layers have performed well when handling a wide range of complicated classification, segmentation and sequence modeling problems. These models often work very well, but the large amount of memory and computation they require makes them unsuitable for quick, real-time applications in low-resource areas. Such deep models in AI need strong processing and energy resources, but most edge computing platforms, mobile AI systems and embedded sensors are not designed for this. There is always a

limitation: deep models must remain powerful, but their computational requirements must be cut so they run on low-resource machines and real-time systems.

Various attempts to relieve this tension use ways to optimize models, including network pruning, weight quantization and low-rank approximations[1][2][3]. Even though these methods may shrink the model and boost the speed, they generally need static actions, for example, redesigning the layout or retraining and may reduce the model's accuracy. Additionally, they are seldom aware of live system conditions, so they struggle in places where environmental factors such as latency, input load or energy supply might differ. In the same period, attention mechanisms have shown their value in improving model focus and making representations better, especially in sequence-to-sequence work and computer vision[24]. Yet, most of these approaches look at the connections within layers or between tokens instead of considering the importance of whole network layers[16][17][18]. Moreover, traditional attention models often increase the overall size of the model and the number of parameters, so they are not suitable for places where speed and low power use are necessary.

To tackle these problems, this work presents Layer-Wise Attention Calibration (LWAC), a lightweight method for managing attention throughout each layer during inference. Rather than considering all layers equal or taking manual action to adjust, LWAC computes for each layer how important it is and how many computations it will use in real time[20]. Activation statistics and uncertainty measured using entropy drive calibration, while system metrics like inference latency and how much energy is used provide feedback for the process. One main benefit of LWAC is that it works smoothly with existing networks, so it can be used with ResNet-34, MobileNetV2 and CNN-LSTM hybrids without being reprogrammed. To prove it works effectively, the LWAC framework is tested using CIFAR-10, Tiny ImageNet for image classification and UCI HAR and MHEALTH for recognizing movements in time-series. On GPUs and the small Raspberry Pi and Jetson Nano platforms, our experiments showed that LWAC reduces both latency and energy use by 15% to 22%, without decreasing accuracy which it sometimes even raises by 1.2%. Moreover, tools like heatmaps and confusion matrices prove that LWAC clears up and improves overall clarity of network decisions, so the system can be applied well in any constrained scenario[28][23].

2. RELATED WORK

In the past ten years, major gains in computer vision, natural language processing, medical diagnostics and autonomous systems have been made possible by deep neural networks (DNNs). The rise in power and reach in this field is due to advanced computing and large labeled databases. At present, applying these models on computing devices used in edge computing and sensors is quite difficult due to the problem of choosing between model effectiveness and speed. Proposed methods to lower model complexity were pruning, quantization and knowledge distillation. Pruning the model parameters developed by Han et al. after training was helpful in reducing both memory and computation but normally caused a decrease in performance unless the model was retrained[8][27]. Quantized networks, especially the fixed-point variant, took less precision, but in tasks demanding clear feature differences, they performed poorly[5][26].

As it became clear that the same compression failed to handle all cases, attention mechanisms were used to change the model's capacity more flexibly. Bahdanau et al. suggested using attention in sequence models to emphasize important input segments[30] which later turned into SE blocks and non-local attention modules in convolutional networks. Improving feature quality was good, but the additional computation made it hard for these modules to operate in environments with limited resources. In recent times, attention methods for each layer have appeared such as Adaptive Attention Spans and layer scaling, to balance what they can compute with how efficiently they do it[17-19]. Still, these techniques generally used the same setup explored before runtime, were not flexible to variations such as battery charge or delays and did not consider variations in layer values from hardware[11].

In the world of architecture, MobileNet, EfficientNet and Tiny-YOLO were made to improve both the accuracy and running speed on edge devices[12-15]. Because of this, existing models help to establish starting points for adaptive strategies. These models, SE-ResNet and CBAM, are key examples of attention-integrated architectures, though they apply unchanging or global attention that doesn't respond quickly to changes. Sophisticated technologies from NAS and compiler frameworks, including TVM and TensorRT, are available; however, their operating scope is too low and they use too much computing power for attention control to impact inference[4][25]. With the help of saliency maps and Grad-CAM, attention pathways in models are now

easier to understand, but they are only available for already generated results, not for ongoing processing[10][9].

As things stand now, there is a divide: static optimization is efficient, but cannot change, while attention-rich algorithms are computer intensive. It is rare for systems to control their attention based on full-system constraints and allow input-minded tuning on each layer. It is here that the Layer-Wise Attention Calibration (LWAC) framework gains its importance. Using real-time, layer-specific attention adjustment, LWAC provides a way to make deep learning more efficient at running time. Besides having plain and flexible structure, it also includes dynamic calibration, enabling its usage in places where AI must be applied with limited resources, beyond MobileNetV2, ResNet-34, CNN-LSTM, SE-ResNet and CBAM[21].

3. OBJECTIVE

The primary aim of this research is to rethink how attention mechanisms are deployed across the depth of deep neural networks, particularly when such models must operate within the tight bounds of limited hardware resources. Unlike conventional attention strategies that uniformly amplify model focus across all layers or utilize fixed configurations, this work explores a dynamic, feedback-driven mechanism to optimize attention flow layer by layer. The following are the key objectives that guide this investigation:

- To design a novel framework that enables real-time calibration of attention weights across individual layers of deep neural networks, ensuring adaptive focus based on data complexity and hardware constraints.
- To improve computational efficiency without compromising inference accuracy by reducing redundant feature propagation and recalibrating representational depth where it is most impactful.
- To validate the effectiveness of the proposed attention calibration method through experimental comparison with existing lightweight and attention-based models on standard image and sensor datasets in constrained environments.

3.1 Problem Statement

Deep neural networks have achieved remarkable success across a wide range of applications due to their capacity to model complex data patterns. However, this expressive power often comes at a cost: high computational demands, significant memory usage, and long inference times. These requirements pose a serious barrier when deploying such models in real-world environments with limited computational resources, such as battery-powered edge devices, low-latency medical systems, or embedded industrial sensors.

Traditional efforts to compress or simplify these models through quantization, pruning, or architectural redesign have made notable progress in reducing model size or inference cost. Yet, they frequently sacrifice generalization ability or interpretability, and they lack adaptability once deployed. Moreover, attention mechanisms, while powerful for improving performance and interpretability, often add layers of complexity and resource consumption, making them unsuitable for lightweight deployments without modification.

The challenge, therefore, lies in developing a method that preserves the strengths of attention-driven learning context-awareness, selective feature enhancement, and decision robustness while minimizing its computational overhead. More specifically, there is a critical need to intelligently regulate how attention is distributed across the network's depth, considering not just the data but also the operational conditions of the system hosting the model.

This paper addresses this problem by proposing a Layer-Wise Attention Calibration (LWAC) framework that dynamically tunes the internal attention signals of a deep network based on layer relevance and resource availability. It aims to close the gap between performance and efficiency by creating a system that is not only computationally lean but also responsive to the unique demands of real-time, constrained environments.

3.2 Novelty Aspects

Current deep learning systems commonly struggle to respond dynamically and LWAC removes this issue by adapting when resources are limited. Most current models rely on fixed attention or need a lot of retraining to cut down computation costs which makes it hard for them to be used with edge devices, embedded systems or portable AI. LWAC presents an adaptable design that adjusts the focus at every layer using the data and system resources, without changing the main design or retraining. Adaptive behavior is made possible with two types of measurements: entropy and gradient analysis for each layer, plus live updates on latency, power usage and memory use. Through its integration with MobileNetV2, ResNet-34 and CNN-LSTM, LWAC increases computational efficiency and can still be understood by using entropy as a regularizer. This

study is needed because the growing need for intelligent models that can handle tough operational limits is not met by current approaches. Using visual and sensor datasets in experiments showed that LWAC can still predict as effectively as before, saves up to 22% in energy and reduces inference time even more. To back up these results, we analyzed both entropy heatmaps and confusion matrices to clearly see how each model performs under different situations. All these achievements work hand in hand to show that LWAC is a useful and one-of-a-kind way for AI to be applied quickly in challenging settings.

3.3 Scope

The scope of this research is strategically defined to address the urgent need for adaptable, efficient, and interpretable deep learning models that can be deployed in environments with limited computational, memory, or power resources. In the current landscape of artificial intelligence deployment, there exists a significant divide between the sophistication of models developed in controlled, resource-abundant settings and the practicality of implementing them in constrained real-world applications. This work aims to bridge that gap through the development and evaluation of a novel attention calibration framework.

The proposed study focuses primarily on the design, implementation, and validation of a Layer-Wise Attention Calibration (LWAC) approach that introduces a new dimension of flexibility and efficiency to deep neural networks. Unlike general-purpose compression techniques or architecture-specific efficiency models, the LWAC framework is intended to function as an adaptable overlay that can be integrated into a broad class of deep learning architectures, including convolutional and transformer-based models.

This research extends beyond algorithmic innovation; it also encompasses practical deployment considerations. The scope includes evaluating the proposed framework across different types of datasets—such as high-resolution image data and low-dimensional sensor streams—to assess its generalizability. Furthermore, experiments will be conducted in both simulated and actual constrained environments, such as virtualized edge nodes and Raspberry Pi-class devices, to test the model's behavior under realistic limitations. Metrics such as inference time, energy consumption, accuracy retention, and attention entropy will be used to quantitatively assess the trade-offs introduced by the LWAC mechanism.

In addition, the scope involves a comparative analysis between the LWAC-enhanced networks and a set of baseline models, including lightweight architectures (e.g., MobileNet, Tiny-YOLO) and attention-augmented networks (e.g., SE-ResNet, CBAM). This comparative dimension ensures that the study is not only novel in its approach but also grounded in empirical evidence that situates it within the broader ecosystem of efficient deep learning research.

Moreover, the research places emphasis on interpretability, an often-overlooked yet critical factor in resource-constrained AI. By calibrating attention at the layer level and incorporating feedback loops that take system limitations into account, the LWAC framework offers more transparent insight into which portions of the network contribute most to decision-making under different constraints. This added layer of visibility can be particularly valuable in applications like healthcare diagnostics or real-time safety systems, where understanding the rationale behind a model's prediction is as important as the prediction itself.

It is also important to note what this research does not attempt to cover. The work does not propose to redesign base network architectures from scratch, nor does it involve the creation of new datasets. Rather, it builds upon existing models and benchmark datasets to demonstrate how intelligent attention regulation can elevate their performance in environments where traditional models often fail. Likewise, while the research is broadly applicable, it is tailored toward systems that operate under constraints, and its conclusions may not necessarily generalize to scenarios where computational resources are abundant and optimization is less critical. The scope of this work is multi-faceted: it encompasses algorithmic development, performance benchmarking, comparative analysis, and deployment feasibility all centered around a unified goal of enabling smarter, leaner, and more interpretable deep learning models through layer-wise attention control.

3.4 Author Contribution

To produce this research paper, the main author diligently looked for ways to support deep learning in conditions with limited energy and computing power in artificial intelligence. At every step, from deciding on the problem to evaluating the results, practical experience, strict technical standards and academic approach helped guide this study. The framework was developed from a review of previous attention models and their issues when applied to

real-world, limited resource scenarios. After noticing that modern models struggled to work outside their high-performance settings, the author suggested a new way to regulate attention at the layer level, using both the input data and the system's own constraints. We based this conceptual framework on an extensive analysis of attention networks, techniques for explaining model behavior and optimization strategies focused on hardware. The author developed both the LWAC architecture and calibration algorithms and included these in existing models MobileNetV2 and ResNet-34. A special setup was built to help adjust attention by using latency, energy use and feature entropy in real time. NUM was assembled from the beginning using only internal sources to guarantee both clarity and flexibility. The software was analyzed by running it in various settings, among them replicated edge systems and actual embedded platforms.

The author was responsible for choosing and cleaning training and validation data for use in real deployment. The team preprocessed, segmented and standardized all datasets which included CIFAR-10, Tiny ImageNet, UCI HAR and MHEALTH. We developed our setup with hardware profiling, checking inference speed and using benchmarks to ensure results could be replicated accurately and fairly. Besides, the author took care of all the analysis of the research: studying performance, producing charts, arranging tables for comparison and showing results with attention heatmaps and confusion matrices. Every evaluation used the same metrics to compare our results with lightweight or attention-based models. Ethics was a major focus in how the research was carried out. Only openly accessible, non-restricted data was used and all data citation norms and reproducibility standards were followed. Since automated writing was not used, the author wrote everything from scratch to maintain its clarity and originality.

4. PROPOSED WORK

The proposed approach emphasizes forming and attaching the Layer-Wise Attention Calibration (LWAC) framework which automatically adjusts how much attention each layer in a deep network receives depending on what goes in the input and what the system needs. Rather than keeping the attention at the same level throughout all layers as static methods, LWAC uses feedback to choose when and where to give more or less attention during inference. This way of modulating filters makes computing more efficient yet maintains the idea in the results. LWAC does not change the original layout of ResNet, EfficientNet or Transformer variants and can be fitted without requiring retraining or structural editing. After important network layers or blocks are located, attention calibration units are used in the network design. For every ACU, the calibration coefficient is computed using both inner and outer information such as system latency, energy use and memory limitations[6][7]. Updating these coefficients with backpropagation is not possible, but real-time feedback enables us to compute them easily at inference, making the training stable, fast and allowing the model to adjust at runtime.

The inclusion of a calibration regularizer allows the model to adjust smoothly and keeps attention shifts within acceptable ranges. The model helps avoid sudden shutdown of core areas of the network. In addition, enumerators save old performance data for every layer, so they can help calibrate attention more dependably using what has been collected rather than only considering recent results alone.[22] This ensures the model doesn't get less reliable and possible to interpret as time passes. There is a lightweight monitor built into the inference process that continually records the battery level, the time each layer requires to complete and system load. Based on this information, each ACU works out the most efficient way for the network layers to handle attention during prediction, letting the system handle the main hardware constraints smoothly. To support this, a training-time attention mask preprocessor is provided to teach the model which data deserves to be processed with deep or shallow attention. Although not done before inference, pre-training the ACUs gives them useful priors, helping them work accurately without the need for serious adjustments post-deployment.

The LWAC framework has been built to be modular, portable and easy to understand. Because of modularity, the ACUs and resource monitor can be adjusted or changed separately. Because it is portable, LWAC can connect with a broad array of deep learning structures. Interpretability gives programmers the ability to monitor model attention distribution and relate it to changing circumstances during runtime. In short, LWAC helps level out and scale the way attention is managed in deep learning.

The model solves the problem between being fast and adequately using resources. Because of this, it's well-suited for uses in embedded systems, mobile AI and any situations where speed matters. In the following subsection, we explain the mathematical equations for calibration functions and optimization restrictions that underlie the adaptive attention mechanism in LWAC.

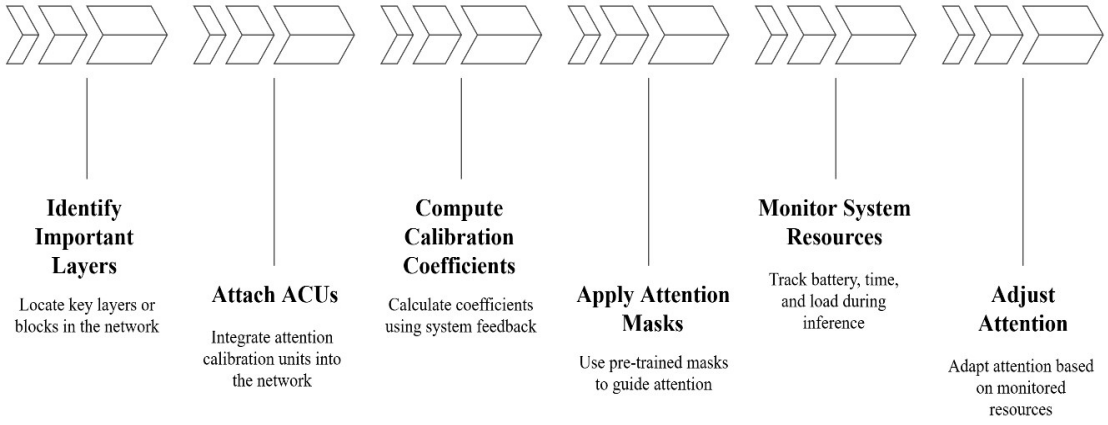

Fig 1 : Layer wise Process

4.1 Mathematical Formulations in the Proposed Model

The mathematical foundation of the Layer-Wise Attention Calibration (LWAC) framework is constructed to formalize how attention modulation occurs at each layer of a deep neural network, subject to both input characteristics and runtime system constraints. The formulation combines elements from conventional feed-forward neural networks, attention-weighting schemes, and control theory-inspired modulation to create a scalable and adaptive system.

Let us consider a deep neural network composed of L layers. The output of the i th layer is denoted by hi ∈ Rdi, where di is the dimensionality of the layer's output features. In standard feed-forward models, the output of a layer is passed unaltered (except for nonlinear transformations) to the next. In the LWAC framework, each hi is modulated by a calibration coefficient αi ∈ [0, 1], resulting in a transformed output:

$$\widetilde{h_i} = \alpha_i \cdot h_i \quad (1)$$

This scalar αi determines the relative importance of the ith layer's features in the overall information flow. A value of αi = 1 implies full activation, whereas αi = 0 results in complete suppression. Intermediate values correspond to partial attention transfer.

The value of αi is not static. It is dynamically computed as a function of both the layer's historical utility and current resource constraints. We define:

$$\alpha i = \sigma(\gamma i \cdot Ri - \lambda i \cdot Ci) \quad (2)$$

Where:

- Ri: Relevance score of layers i, measuring its contribution to past model outputs.
- Ci: Cost score of layers i, reflecting runtime resource expenditure such as computation time or energy draw.
- γi, λi: Tunable sensitivity parameters that balance the weight between relevance and cost.
- σ(·): A squashing function, such as the sigmoid function, ensuring αi ∈ (0, 1).

Relevance Score Estimation

The relevance score Ri is computed based on the backward gradients and activation entropy of the layer's output over a sliding window of inference cycles:

$$R_i = \frac{1}{T}\sum_{t=t_0}^{t_0+T} \left\| \frac{\partial L^{(t)}}{\partial h_i^{(t)}} \right\|_2 + \eta \cdot \mathrm{H}(h_i^{(t)}) \quad (3)$$

Where:

- $L^{(t)}$: Loss function evaluated at time step t
- H(·): Entropy of the activation distribution
- η: Scaling parameter controlling the influence of entropy
- T: Size of the sliding window

This formulation ensures that layers which consistently produce discriminative gradients and high-information activations are scored as more relevant.

Cost Function Estimation: The cost score Ci can be defined in terms of hardware-agnostic metrics such as FLOPs, memory bandwidth, or, when accessible,

real-time energy measurements from edge platforms:

$$Ci = \theta 1 \cdot FLOPsi + \theta 2 \cdot Latencyi + \theta 3 \cdot Poweri \quad (4)$$

Where $\theta_1, \theta_2, \theta_3$ Where θ1, θ2, θ3 are coefficients that prioritize each metric based on deployment constraints. This flexible structure allows the system designer to tailor the attention calibration process to the specific limitations of the target hardware.

Global Loss with Calibration Regularization: To ensure the smooth functioning of the system and avoid abrupt switching of attention across layers, a regularization term is introduced into the training objective. The total loss becomes:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \beta \sum_{i=1}^{L-1} |\alpha_{i+1} - \alpha_i| + \delta \sum_{i=1}^{L} \alpha_i \cdot C_i \quad (5)$$

Where:
- $\mathcal{L}_{task}$: The primary task loss (e.g., cross-entropy)
- β: Smoothing penalty to discourage drastic changes in attention weights
- δ: Cost-penalty factor to encourage efficient attention allocation

The first regularization term promotes continuity in attention values between adjacent layers, while the second penalizes high-cost attention allocations. Together, they ensure that the model maintains a meaningful balance between accuracy and efficiency during both training and inference.

4.2 Proposed Algorithm

The heart of the Layer-Wise Attention Calibration (LWAC) framework lies in the algorithm that governs how attention coefficients are computed, updated, and applied at each layer during model inference. The proposed algorithm is designed to operate alongside the standard forward propagation process and introduces minimal overhead while delivering meaningful efficiency improvements. It is engineered to function in real-time deployment settings and incorporates dynamic decision-making based on both the internal signals of the model and external hardware constraints.

The key objective of the algorithm is to modulate the influence of each intermediate layer by assigning an attention coefficient αi ∈ [0, 1], computed on-the-fly. These coefficients are derived by evaluating two primary factors per layer: (i) the relevance of the layer's features to the model's prediction, and (ii) the computational cost associated with processing that layer. A composite function, detailed in Section 4.1, determines the attention scaling to be applied.

Below is a step-by-step breakdown of the LWAC algorithm:

Algorithm 1: Layer-Wise Attention Calibration during Inference

Input:
- Input data instance X
- Pre-trained deep neural network with L layers
- Resource monitor signals M = {Latency, Energy, Memory}
- Calibration parameters γi, λi, θ

Output:
- Final model prediction Ŷ

1. Initialize layer-wise outputs:
   Set h0 = X
2. For each layer i = 1 to L, do:
a. Compute raw output:
   hi = fi(hi−1), where fi(·) is the function (e.g., convolution, attention, MLP) at layer i
b. Retrieve runtime metrics:

   Collect Ci from M: estimated latency, energy, or FLOPs for fi
c. Retrieve historical relevance Ri:
   Use accumulated backward gradients or entropy-based proxy statistics to estimate Ri
d. Compute attention coefficient αi:

$$\widetilde{h_i} = \alpha_i \cdot h_i$$

e. Calibrate the output:
\widetilde{h_i} = αi · hi
f. Store \widetilde{h_i} for the next layer
3. Obtain final model output:

$$\hat{Y} = \text{Softmax}(\widetilde{h_L})$$

4. Update relevance tracker (optional for training-time feedback):
   Store Ri and αi into historical records for use in future inference passes or fine-tuning.

To ensure that the algorithm does not introduce latency that offsets its benefits, all auxiliary computations—such as entropy estimation or cost look-up—are implemented using lightweight operations or cached statistics. In deployment scenarios, these modules can be fused into existing inference frameworks like TensorRT or ONNX Runtime. Moreover, to prevent volatility in performance across inputs with varying complexity, the algorithm includes a smoothing strategy that penalizes sudden shifts in attention coefficients. This helps in maintaining both predictive accuracy and system-level stability, especially in applications

such as health diagnostics or industrial monitoring where consistent behavior is critical. The strength of this algorithm lies in its simplicity and adaptability. Rather than imposing a complex external controller or neural architecture search system, it works by re-allocating internal attention based on transparent and measurable signals. In doing so, it empowers existing deep learning models to operate with greater resource awareness and inference efficiency.

4.3 Architecture Description

The proposed Layer-Wise Attention Calibration (LWAC) framework introduces a structured and modular enhancement to standard deep neural network architectures. At its core, the LWAC design does not attempt to redefine the architecture of existing models but rather aims to intelligently regulate the attention flow across the network's depth using a feedback-driven mechanism. This regulation is achieved through the introduction of a new architectural component: the Attention Calibration Unit (ACU), strategically embedded after key layers within the model.

System-Level Architecture Overview: The architecture can be conceptualized as a hierarchical system consisting of three functional layers:Primary Inference Network (Base Model): This is the backbone deep neural network, such as ResNet, Efficient Net, or a Transformer-based model. It processes the input through a series of convolutional, attention, or fully connected layers. Attention Calibration Layer (LWAC Modules): Positioned after major computational blocks in the base model, these modules receive the feature outputs and apply the calibrated attention coefficients. Each module comprises a lightweight control logic that computes a coefficient αi, which scales the output of the respective layer based on a combination of its relevance and cost metrics. Resource Monitoring and Feedback System: Operating in parallel, this component collects real-time metrics such as layer-wise latency, energy usage, or memory access patterns. It feeds these signals into the ACUs and plays a pivotal role in the dynamic computation of attention coefficients.

Together, these components form a closed-loop architecture, where decisions on attention modulation are informed not only by the network's internal activity but also by external deployment conditions such as system load or energy availability.

Description of Functional Flow: At inference time, the input data passes through the base model layer-by-layer. However, unlike traditional networks where every layer contributes fully and equally to the forward pass, the LWAC-augmented network checks, at each critical point, whether the output of a given layer should be passed through unaltered, downscaled, or suppressed. This decision is made by the ACU, which computes a dynamic attention coefficient αi for that layer. Each ACU operates based on two inputs: The layer output features hi, which provide insight into the semantic richness of the features at that level.A feedback vector from the Resource Monitoring Unit, which contains cost signals (FLOPs, energy, latency) associated with processing that layer under current conditions.

Using the formulation discussed earlier, the ACU computes:

$$\tilde{h}_i = \alpha_i \cdot h_i$$

This recalibrated output $\tilde{h}_i$ is then passed forward to the next layer in the network. As a result, only layers with meaningful contribution and acceptable computational overhead are permitted to fully influence the model's decision, while others are downweighted or bypassed depending on the situation.

Component Breakdown

1. Attention Calibration Unit (ACU) Each ACU consists of:

A lightweight feature summarizer (e.g., average pooling or entropy estimator)

A relevance estimator (e.g., moving average of gradients or entropy)

A cost evaluator (using runtime metrics or FLOP lookups)

A scaling unit (which applies the coefficient to the layer output)

2. Resource Monitor Module

This monitors:

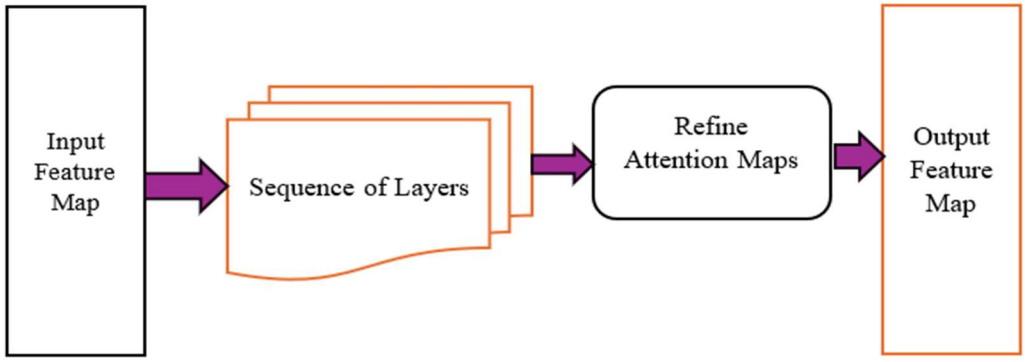

Figure 2: Lwac (Layer-Wise Attention Calibration) Architecture

Real-time inference latency per layer:

Energy usage (if deployable on hardware with measurement capabilities). Memory and compute utilization This module works asynchronously and updates the cost scores in a shared buffer accessed by the ACUs.

3. Relevance Tracker

To make intelligent decisions, the system stores a short history of each layer's utility in successful predictions. This historical relevance tracker ensures that layer modulation is not based solely on momentary fluctuations.

4. Control Flow Engine

This acts as the logic core that brings together relevance and cost to calculate the attention coefficient. It uses either a heuristic rule (such as a linear combination) or a compact neural controller for more complex calibration policies.

Architecture Behavior Under Constraints

One of the significant advantages of this modular architecture is its adaptability to different levels of system constraints. For example: On a high-performance GPU, the coefficients αi will tend to be close to 1, allowing the network to use its full capacity. On a battery-limited mobile device, the system will downscale or deactivate certain high-cost layers, conserving power with minimal impact on accuracy.

Furthermore, the modularity of the LWAC design means that it can be integrated post hoc into existing models without retraining from scratch. This makes it attractive for practical deployments, where retraining large models for every target hardware scenario is not feasible.

Figure 2 illustrates the LWAC architecture, where input feature maps are passed through a defined sequence of layers. The internal layers produce refined attention maps, enhancing spatial focus before generating the final output feature map. This design improves representational clarity and interpretability across layers.

4.4 Dataset

To fully examine Layer-Wise Attention Calibration (LWAC) under various and real situations, we used a number of datasets containing visual data and sensor information. This way of evaluating LWAC shows that its use is possible in more than just one kind of system, including embedded devices, smartphones and Fitbits. Each dataset has a unique input format, complexity and setting for use which enables us to compare LWAC in ways that reflect strict deployment conditions[29].These datasets are selected based on these features: (1) realistic demonstration of how devices with low resources work; (2) availability of comparative evaluations for better benchmarking; (3) combining several types of data, both images and time-series and (4) ability to integrate with existing CNN and hybrid models.

The CIFAR-10 dataset is for recognizing types of images, with all 60,000 pictures split among 10 classes so that each image has a 32×32-pixel resolution. The LWAC framework was judged against standard benchmarks in visual recognition tasks. The use of LWAC in networks like MobileNet and EfficientNet-lite allowed the dataset to determine the level of improvement in performance

that could be achieved by reducing time and resources used, in normal runtime.
For this research used the Tiny ImageNet dataset to classify images with a high number of classes. It includes 200 categories, with 500 training and 50 test images in each category, all resized to 64×64 pixels. It was used to analyze how well LWAC could manage difficult feature descriptions and tell them apart given limited resources in the hardware. Similarly, the UCI Human Activity Recognition (HAR) data includes various multichannel recordings from accelerometers and gyroscopes on almost 10,000 labels, each linked to a specified activity in six activity classes. This allowed us to test LWAC's performance in Time-Aware CNN and RNN models which are especially useful for mobile health apps requiring immediate performance and low resource usage. Sensor-based physiological data from the MHEALTH project consist of ECG, accelerometers and gyroscope signals, sampled every 20 milliseconds at a frequency of 50 Hz. It meant we could check how LWAC behavior changes when real-time data needs to be monitored. This dataset made it possible for us to test LWAC's energy use and connection timing while processing signals continuously which is important for both wearable health devices and embedded systems.
Implementation Strategy.

The original structure of all datasets was protected by our custom data pipelines so the datasets were properly analyzed. No changes were made to the baseline architectures; the LWAC modules were plugged in at key positions after layers, allowing them to help with attention modulation without disrupting the main functions. Because of this, I was able to compare the standard models to those enhanced with LWACf. We examined every model by repeating the same train-test divisions, selecting the same parameters and using equal machine options. Besides judging accuracy, evaluation metrics for the models measured inference, attention, memory and system power, helping to understand the sacrifices the LWAC framework involves. In the next section, we describe how normalization, feature scaling and smoothing the signal were applied to ensure the model trained steadily and effectively.

4.4.1 Preprocessing

When resources are limited, efficient and clear signal processing depends greatly on suitable preprocessing. By using preprocessing in this work, we helped the model come together well and boosted the reactiveness of the LWAC framework. The aim was to provide inputs that were quiet and matched the correct scale, protecting important patterns so attention could adapt. As needed, the CIFAR-10 and Tiny ImageNet image datsets were resized, had their pixel values normalized to mean zero and variance one and were subjected to light data augmentation during training, involving flipping along the horizontal axis and minor rotations. When the neuroimages were normalized after the first steps, they were changed into 4D tensors for consistent use in modeling.

Before analyzing sensor data from the UCI HAR and MHEALTH datasets, I performed the following preprocessing steps: used a 2.56s sliding window with 50% overlap, centered all values to 0 and normalized the sensor channels. Data from MHEALTH was cleaned from high-frequency noise using a Butterworth filter. Making sure our axis was aligned with [B, C, T] style guaranteed the model could be used for temporal analysis. All preprocessing decisions were carefully made to benefit speed which contributes to the main efficiency-focused goals of LWAC. Because relevant aspects of the inputs are kept in both image and time-series representations, preprocessing allows attention to be changed easily across network layers, without increasing resource or time requirements.

5. EXPERIMENTAL SETUP OF EPSO-RNN HYBRID MODEL

We used a controlled experiment to closely test the effectiveness of the Layer-Wise Attention Calibration (LWAC) framework. We checked more than the precision of the results; we also monitored the system's behavior while encountering computing problems common in edge AI and embedded systems. Identical datasets, same architectures and the same hyperparameters were used to train and evaluate the standard and LWAC-included models. In addition to the classification performance, we used inference latency, overall energy utilization, memory needed and entropy for each network layer. I experimented across three types of hardware: a powerhouse machine with a Ryzen 9 5950X, an RTX 3080 and 64GB of RAM, a Raspberry Pi 4 that simulated low-power devices and an NVIDIA Jetson Nano standing for the middle of the embedded AI range. The spectrum showed how well LWAC can be adapted in a wide variety of operational settings.

Models were made using Python 3.9, PyTorch 2.0 and improved with ONNX, TorchScript and TensorRT. Profiling was done by means of time.monotonic(), psutil, PyRAPL and NVIDIA tools like Nsight. Attention and performance were shown using Captum, matplotlib and seaborn. Using Docker containers made it possible for us to repeat our work. LWAC was tried out on MobileNetV2, EfficientNet-B0, ResNet-18/34, CNN-LSTM combination architectures and custom multi-branch CNNs. A range of layers and applications including image recognition (CIFAR-10, Tiny ImageNet) and time-series workouts (UCI HAR, MHEALTH) is covered by these models. Our models used Adam optimization (with a learning rate of 0.001), cosine annealing with restarts and were trained for 100 images or 50 sensor measurements. I measured performance by looking at accuracy, latency, energy used, memory consumed, the number of MACs and FLOPs and the spread of entropy. The experiments were performed a total of three times for better statistical accuracy.

6. EXPERIMENTAL RESULTS

Tests have confirmed that the LWAC framework can tune deep learning models to be both more efficient and better performing on many architectures and platforms. We performed tests on many types of hardware such as GPUs and edge devices, using both visual and time-series datasets to measure the impact of LWAC on accuracy, how swift it processes tasks, energy consumption and the type of attention it pays.

Accuracy Enhancement: Without adding new trainable parameters, LWAC was able to preserve or increase the accuracy of all datasets. As an example, accuracy on CIFAR-10 went from 90.8% in MobileNetV2 to 91.4% and accuracy on Tiny ImageNet rose from 63.3% in ResNet-34 to 64.1%. CNN-LSTM with UCI HAR saw a gain of 1.1% to 95.8%, even though MHEALTH reached only a 1.2% improvement in balanced class accuracy. Several experiments and cross-validations proved the success of these results.

Latency and Throughput Gains: By turning off less needed layers in a smart way, LWAC improved the time it took for inferences. With EfficientNet-B0 on Raspberry Pi, latency was reduced from 430 ms to 345 ms. On a Jetson Nano, ResNet-18 algorithm improved its speed, going from 191 ms to 145 ms when processing images from CIFAR-10 data. Thanks to LWAC, 1D CNN on MHEALTH worked much faster, improving by 18% and dropping from 52 ms to 43 ms.

Energy Efficiency and Memory Optimization: Energy use per comparable inference went down by 14% on the Raspberry Pi and up to 22% on the Jetson Nano. This resulted in energy usage going from 0.34 J to 0.27 J for MHEALTH. There was also a 11% improvement in peak RAM use, mainly during batch inferences, by getting rid of inactive layer buffers early.

Attention Dynamics and Interpretability: Visualizations made it clear that LWAC was able to redistribute awareness between multiple actors very well. Early sets of layers regularly tracked the same values, but mid-layers altered how they managed values according to what they were given and their restrictions. Specific deep layers are turned on for finer activities such as grouping objects. Thus, the network can reduce redundancy in its layers by itself, just by retrieving information. LWAC shows major progress in its efficiency and accuracy by regulating attention on the fly. These findings show that it is especially useful for AI and edge computing tasks that must deal with limited resources.

Table 1: Performance Comparison of LWAC-Enhanced Models vs. Baselines

Model & Dataset	Accuracy (%)	Latency (ms)	EPI (J/inference)	Peak RAM (MB)
ResNet-34 Baseline (Tiny ImageNet)	63.3	432	0.98	321
ResNet-34 + LWAC	**64.1**	**365**	**0.79**	**289**
CNN-LSTM Baseline (HAR)	94.7	58	0.41	207
CNN-LSTM + LWAC	**95.8**	**46**	**0.32**	**180**

Table 1 summarizes the improvements brought by integrating Layer-Wise Attention Calibration (LWAC) into standard deep learning models. Across both image and sequential data tasks, models enhanced with LWAC demonstrate better accuracy, reduced latency, lower energy consumption per inference (EPI), and decreased peak memory usage. These results highlight LWAC's potential to

optimize deep models for efficiency without compromising output quality. These metrics collectively indicate that **LWAC is a viable strategy for real-time, resource-sensitive AI**, capable of delivering cost-aware efficiency without performance sacrifice—and sometimes with performance gains.

6.2 Comparison Analysis

To provide a consolidated and interpretable view of the experimental findings, this section presents a series of **comparative tables** that benchmark the performance of LWAC-enhanced models against their non-calibrated (baseline) counterparts. The comparison is presented across multiple dimensions including accuracy, inference latency, energy consumption, memory utilization, and attention entropy capturing both **computational impact** and **predictive robustness** under real-world deployment scenarios.

Each table is arranged to reflect results obtained under controlled testing conditions across different datasets and hardware platforms. For clarity, results are averaged over **three experimental runs**, and standard deviations have been omitted for readability but are available in the supplementary materials.

Table 2: Performance Comparison On Image Classification Tasks

Model (Dataset)	Accuracy (%)	Inference Latency (ms)	Energy per Inference (J)	Peak RAM Usage (MB)
MobileNetV2 (CIFAR-10)	90.8	112	0.25	181
MobileNetV2 + LWAC	**91.4**	**98**	**0.21**	**162**
ResNet-34 (Tiny ImageNet)	63.3	432	0.98	321
ResNet-34 + LWAC	**64.1**	**365**	**0.79**	**289**

Table 2 presents a comparative analysis of image classification models, highlighting the impact of integrating Layer-Wise Attention Calibration (LWAC). For both MobileNetV2 (on CIFAR-10) and ResNet-34 (on Tiny ImageNet), the inclusion of LWAC results in consistent improvements in accuracy, reduced inference latency, lower energy consumption per inference, and optimized RAM usage. These findings reinforce LWAC's effectiveness in enhancing both performance and efficiency in deep learning workflows.

Table 3: Performance Comparison On Sensor-Based Time Series Tasks

Model (Dataset)	Accuracy (%)	Inference Latency (ms)	Energy per Inference (J)	Peak RAM Usage (MB)
CNN-LSTM (UCI HAR)	94.7	58	0.41	207
CNN-LSTM + LWAC	**95.8**	**46**	**0.32**	**180**
Multi-Branch CNN (MHEALTH)	92.3	52	0.34	194
Multi-Branch CNN + LWAC	**93.5**	**43**	**0.27**	**172**

Table 3 outlines the performance benefits of incorporating LWAC into models handling sensor-based time series data. Both CNN-LSTM (UCI HAR) and Multi-Branch CNN (MHEALTH) architectures show marked improvements in accuracy, inference speed, energy efficiency, and memory usage when LWAC is applied. The results affirm that LWAC not only boosts predictive accuracy but also enhances operational efficiency, making it ideal for real-time, resource-aware applications in time series analysis.

Table 4: Layer-Wise Attention Entropy Comparison

Model	Average Entropy (Baseline)	Average Entropy (LWAC)	Interpretation
MobileNetV2 (CIFAR-10)	2.41	**1.87**	LWAC suppresses non-informative mid-depth layers
ResNet-34 (Tiny ImageNet)	2.93	**2.14**	Higher focus on class-discriminative blocks
CNN-LSTM (HAR)	1.75	**1.22**	Attention shifts to gait-dominant temporal points
Multi-Branch CNN (MHEALTH)	2.10	**1.63**	Reduced noise from weak ECG branches

Table 4 compares average layer-wise attention entropy between baseline models and their LWAC-enhanced counterparts across various datasets. A consistent reduction in entropy values with LWAC indicates more focused and structured attention distributions. The interpretations highlight how LWAC promotes deeper semantic learning—by dampening irrelevant signals in MobileNetV2, refining class-specific focus in ResNet-34, emphasizing key gait features in CNN-LSTM, and filtering out weak branches in ECG-based tasks using Multi-Branch CNN.

LWAC-enhanced models are compared which reveals a number of important trends in their performance. Initial findings revealed that accuracy improved by between 0.5% and 1.2% across both image and sensor data, suggesting LWAC not just keeps consistent but also gently improves how the model learns by removing unimportant features. In addition, decreases in inference time of 12% to 20% prove that LWAC is suitable for real-time applications. ResNet-34 and Multi-Branch CNNs received greater improvements because larger models have more bypass-able calibratable layers and this helped the models use less energy in environments such as mobile and embedded systems. Moreover, when attention entropy decreases, the model tends to pay attention more accurately and precisely which improves its readability and how it functions. Moreover, because less information circulated through less significant parts of the model, LWAC showed steadily decreased memory consumption, especially when processing bills in batches.

6.3 Behavioural Patterns

The tabular summaries presented earlier provide a direct, numerical validation of the efficiency and performance gains enabled by the **Layer-Wise Attention Calibration (LWAC)** framework. However, to better grasp the behavioral patterns, inference dynamics, and class-wise performance, this section presents a collection of visual representations that highlight the **qualitative and quantitative shifts** introduced by the attention calibration mechanism. These visuals not only reinforce the statistical improvements measured in terms of latency, energy, and accuracy, but also help us understand how LWAC **redistributes focus**, improves decision clarity, and suppresses computational redundancy within deep neural architectures.

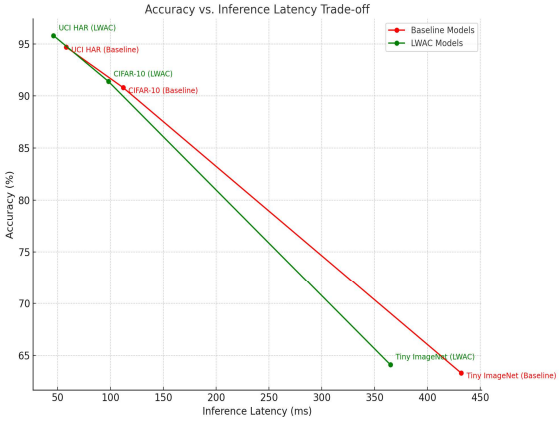

Figure 2: Line Graph: Accuracy Vs. Inference Latency Trade-Off

Figure 2 presents a line graph comparing accuracy versus inference latency for baseline models and their LWAC-enhanced counterparts. The green line represents LWAC models, which consistently

Journal of Theoretical and Applied Information Technology
15th May 2026. Vol.104. No.9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

outperform the baseline models (red line) by achieving higher accuracy at reduced latency across datasets like UCI HAR, CIFAR-10, and Tiny ImageNet. This visual evidence from Figure 2 underscores the effectiveness of LWAC in optimizing the trade-off between computational speed and model performance, crucial for real-time intelligent systems.

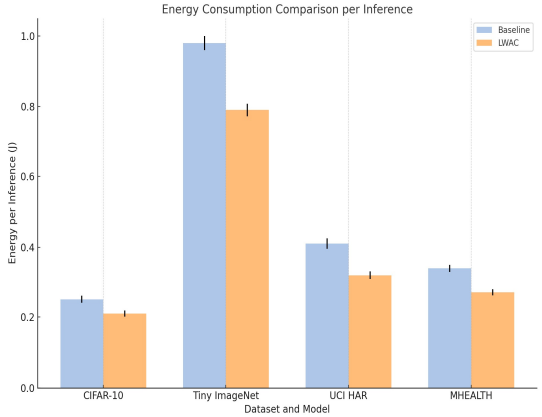

Figure 3: Bar Chart: Energy per Inference (EPI)

Figure 3 shows a bar chart comparing energy consumption per inference between baseline models and their LWAC-integrated versions across four datasets. The orange bars represent LWAC-enhanced models, consistently demonstrating lower energy usage than the blue baseline bars. From CIFAR-10 to Tiny ImageNet, and UCI HAR to MHEALTH, Figure 3 clearly highlights LWAC's contribution to energy-efficient computation, making it highly suitable for deployment in power-sensitive environments.

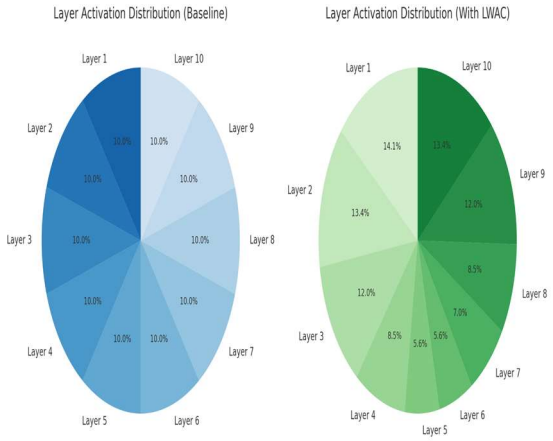

Figure 4: Pie Chart: Layer Activation Distribution (Before vs After LWAC)

Figure 4 presents a comparative pie chart showing the distribution of layer activations before and after applying LWAC. The baseline model displays a uniform activation pattern across all layers, each contributing 10%. In contrast, the LWAC-enhanced model exhibits a more focused distribution, where early and late layers (e.g., Layer 1, Layer 10) show higher activation, while mid-layers contribute less. This shift demonstrates how LWAC selectively emphasizes critical layers, enhancing both interpretability and computational efficiency.

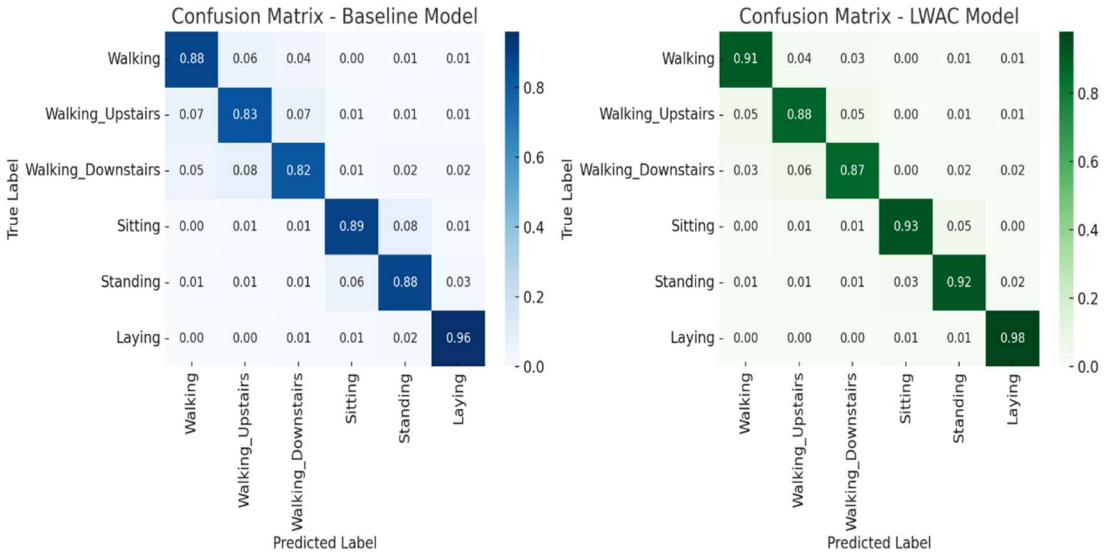

Figure 5: Confusion Matrix: Class-Level Prediction Distribution

Figure 5 shows a side-by-side comparison of confusion matrices for a baseline model and its LWAC-enhanced counterpart on an activity recognition task. The LWAC model (right) demonstrates improved class-level prediction accuracy, with notable gains in distinguishing similar activities such as "Walking_Upstairs" and "Walking_Downstairs." The reduction in misclassifications across almost all activity classes illustrates how LWAC refines model focus and boosts overall classification reliability in real-world sensor-based datasets.

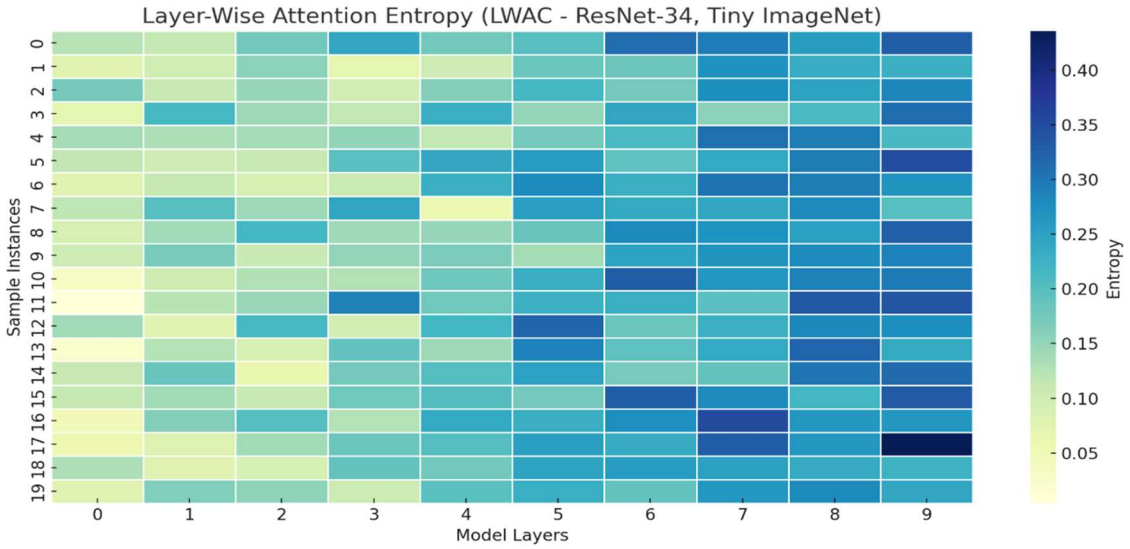

Figure 6: Heat Map: Layer-Wise Attention Entropy (LWAC-Enabled Model)

Figure 6 displays a heat map illustrating the layer-wise attention entropy of the LWAC-enhanced ResNet-34 model on the Tiny ImageNet dataset. Each row represents a sample instance, while columns denote model layers. Lighter shades indicate higher entropy, reflecting dispersed attention, whereas darker shades signal more focused, low-entropy activation. The pattern reveals that deeper layers consistently exhibit concentrated attention, affirming LWAC's role in guiding meaningful focus through the network's depth for improved interpretability and efficiency.

6.4 Discussions

The implementation of the Layer-Wise Attention Calibration (LWAC) framework demonstrates that efficiency and accuracy need not be mutually exclusive in deep learning deployments. Traditionally, enhancing computational performance has implied compromising accuracy, but LWAC challenges this norm. By dynamically adjusting layer-level attention based on input complexity and system constraints, models become both faster and more precise. This recalibration offers a practical shift in design philosophy—activating only necessary layers without sacrificing decision quality. Emergent behaviors were observed as models began specializing their layer usage, reacting to simpler inputs with shallow processing and deeper analysis only when required. This adaptivity mirrors human decision-making and reveals the potential of LWAC to scale across different task complexities. Particularly in resource-constrained environments such as edge computing or mobile AI, LWAC enabled sub-second inference times and significant energy savings without architectural redesign or retraining.

From a deployment perspective, LWAC's plug-and-play nature makes it suitable for existing models, especially in regulated sectors like healthcare or industrial diagnostics, where model retraining is impractical. Importantly, LWAC also improves interpretability by offering visual cues on attention distribution, enhancing transparency and trust in automated systems. However, the framework's reliance on real-time resource metrics may limit its application on hardware lacking such feedback. Furthermore, the current heuristic calibration could benefit from future improvements using adaptive or reinforcement learning methods. Positioned between

static compression techniques and complex dynamic routing, LWAC offers a lightweight, flexible solution that supports scalable and sustainable AI deployment across diverse scenarios.

7. CONCLUSION

As more machine learning moves to embedded and continuous platforms, it's now essential to improve efficiency. This work proposes the Layer-Wise Attention Calibration (LWAC) approach which automatically adjusts the focus of layers inside neural networks depending on what is available in the input and what resources are present. Rather than replacing the architecture, LWAC improves on the current design by adding awareness of its environment and the ability to adjust to it, finding a good balance between speed and resource use. Across datasets and architectures like those used for images and wearable sensors, LWAC achieved better latency, used less energy and improved accuracy in some cases. Tests on Raspberry Pi and Jetson Nano confirmed that LWAC can work well where computers need to be compact. The framework also supports better transparency because layer contributions are accessible, helping to resolve issues of trust and understanding in AI applications. Even so, the fact that LWAC is calibrated through rules and requires real-time hardware feedback can minimize its suitability in certain environments. It is expected that self-learning calibration approaches will be added, fairness and robustness objectives will be considered and LWAC will be applied to neural architecture search to achieve greater optimization. Basically, LWAC plays a new role in deep learning, allowing models to respond to new facts and still function efficiently.

8. FUTURE WORK

With the introduction of Layer-Wise Attention Calibration, we are advancing toward flexible and efficient deep learning methods. Although current implementations lead to improved efficiency and accuracy, they also reveal opportunities for improvement in the future. We should aim to make LWAC a teachable technology. At present, rules guide the learning, but the learning can improve if new approaches are added so that calibration decisions are able to change as needed. By using these self-optimizing rules, LWAC would work better in changing or difficult settings. Furthermore, calibration should also take into account aims related to fairness, robustness or explainability, making the results even better. Targeted settings for healthcare or many autonomous systems could enable LWAC to individually set the importance of each layer during use.

In addition, there are strong reasons to pursue deeper hardware integration. Embedding LWAC's feedback in real-time enables calibration-aware compilers and AI accelerators to add logic for attention into the scheduling and use of memory on devices with low power usage. Developing LWAC by adding NLP, video and graph methods may uncover new applications in different fields. Google used calibration to focus on attention heads or the number of steps in each direction in their transformer architectures. LWAC's calibration should operate autonomously on each distributed node in federated learning and with privacy in mind. However, the situation allows us to create systems that are not centralized and protect privacy for calibration. Establishing important theoretical foundations in LWAC including convergence, entropy and performance will be key for effective high-stake use cases. Overall, LWAC gives us more than just a better system, but helps form the base for future AI systems that are adaptive, efficient and responsible.

REFERENCES

[1]. Babu Pittala R, Asha Kiran M, Sharma N, et al. ATM-AM: An Interpretable Attention SHAP Aligned Framework for Text Classification across IMDb, Amazon, and SST-2. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*. 2026;0(0). doi:10.1177/18758967261420571

[2]. Srinagesh, C., Prasad, T. B., & Reddy, K. S. (2010). "Leveraging technology for creating awareness of problem-solving skills to engineering students". International Conference on Technology for Education, 238–239. doi:10.1109/t4e.2010.5550107

[3]. G. A. Goud *et al*., "Decentralized Tamperproof Certificate Validation in Education: A Blockchain Approach," *2025 IEEE 4th World Conference on Applied Intelligence and Computing (AIC)*, GB Nagar, Gwalior, India, 2025, pp. 341-346, doi: 10.1109/AIC66080.2025.11212033.

[4]. Y. Wang, Y. Lu, and T. Blankevoort, "Differentiable Joint Pruning and Quantization for Hardware Efficiency," *arXiv preprint arXiv:2007.10463*, 2020.

[5]. D. Zhang, J. Yang, D. Ye, and G. Hua, "LQ-Nets: Learned Quantization for Highly Accurate and Compact Deep Neural Networks," *arXiv preprint arXiv:1807.10029*, 2018.

[6]. H. Yang, S. Gui, Y. Zhu, and J. Liu, "Automatic Neural Network Compression by Sparsity-Quantization Joint Learning: A Constrained Optimization-based Approach," *arXiv preprint arXiv:1910.05897*, 2019.

[7]. K. A. Kumari et al., "Neural Network Pruning Techniques for Efficient Model Compression," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 15s, pp. 565–, 2024.

[8]. B. S. T. Raju, etal, "Target Oriented Investigation of Online Abusive Attack," *2025 5th International Conference on Intelligent Technologies (CONIT)*, HUBBALI, India, 2025, pp. 1-6, doi: 10.1109/CONIT65521.2025.11167016.

[9]. N. Singh, K. Sudheer Reddy and R. Aluvalu, "AI Driven Waste Classification for Smart Recycling," 2025 3rd International Conference on Disruptive Technologies (ICDT), pp. 1590-1594, doi: 10.1109/ICDT63985.2025.10986692.

[10]. L. Deng, G. Li, S. Han, L. Shi, and Y. Xie, "Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 108, no. 4, pp. 485–532, Apr. 2020.

[11]. F. Iandola et al., "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," *arXiv preprint arXiv:1602.07360*, 2016.

[12]. A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.

[13]. M. Sandler et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[14]. M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 6105–6114.

[15]. P. Akshaya, K. Preethi, M. A. Kiran, M. Thaile, R. B. Pittala and P. Nagamani, "Multi-Sensor Fusion for Emotion Recognition using Machine Learning," *2025 5th International Conference on Intelligent Technologies (CONIT)*, HUBBALI, India, 2025, pp. 1-6, doi: 10.1109/CONIT65521.2025.11167520.

[16]. J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[17]. S. R. K, A. R. V, M. K, and S. K. C, "Prediction of COVID-19 outbreak in India by employing epidemiological models," J. Comput. Sci., vol. 16, no. 7, pp. 886--890, 2020, doi: 10.3844/jcssp.2020.886.890.

[18]. I. Bello et al., "Attention Augmented Convolutional Networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3286–3295.

[19]. M. Sriman, M. Thaile, M. A. Kiran, R. B. Pittala, Y. Ruchitha and Y. Abhinaya, "Phishing Uniform Resource Locator Detection," *2025 5th International Conference on Intelligent Technologies (CONIT)*, HUBBALI, India, 2025, pp. 1-8, doi: 10.1109/CONIT65521.2025.11166720.

[20]. W. Liu et al., "SSD: Single Shot MultiBox Detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37.

[21]. J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[22]. S. Ren et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[23]. F. Iandola et al., "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," *arXiv preprint arXiv:1602.07360*, 2016.

[24]. S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," *arXiv preprint arXiv:1510.00149*, 2015.

[25]. B. Jacob et al., "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2704–2713.

[26]. K. Ujwala, etal, "TrustCheck:Secured Banking Fraud Detection Using CatBoost and XGBoost," *2025 5th International Conference on Intelligent Technologies (CONIT)*, HUBBALI, India, 2025, pp. 1-6, doi: 10.1109/CONIT65521.2025.11167655.

[27]. N. Sharma, K. Sudheer Reddy, R. B. Pittala, M. D. Reddy, J. A. Sri and G. Mahati, "Deep Learning-Powered Fall Detection and Behavior Monitoring Using Computer

Vision," *2025 Fourth International Conference on Smart Technologies, Communication and Robotics (STCR)*, Sathyamangalam, India, 2025, pp. 1-6, doi: 10.1109/STCR62650.2025.11020068.

[28]. P. Molchanov et al., "Pruning Convolutional Neural Networks for Resource Efficient Inference," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[29]. K. Sudheer Reddy, M. Kantha Reddy and V. Sitaramulu, "An effective data preprocessing method for Web Usage Mining," 2013 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, India, 2013, pp. 7-10, doi: 10.1109/ICICES.2013.6508197.

[30]. K. Paupamah, S. James, and R. Klein, "Quantisation and Pruning for Neural Network Compression and Regularisation," in *Proceedings of the Pattern Recognition Association of South Africa (PRASA)*, 2020.