RESEARCH-ARTICLE

Energy-Efficient Quantization-Aware Training with Dynamic Bit-Width Optimization

ALI KARKEHABADI, University of California, Davis, Davis, CA, United States

AVESTA SASAN, University of California, Davis, Davis, CA, United States

Published: 30 June 2025

GLSVLSI '25: Great Lakes Symposium on VLSI 2025
June 30 - July 2, 2025
LA, New Orleans, USA

Conference Sponsors:
SIGDA

https://doi.org/10.1145/3716368.3735291
ISBN: 9798400714962

Energy-Efficient Quantization-Aware Training with Dynamic Bit-Width Optimization

Ali Karkehabadi
University of California, Davis
DAVIS, California, USA
akarkehabadi@ucdavis.edu

Avesta Sasan
University of California, Davis
DAVIS, California, USA
asasan@ucdavis.edu

Abstract

Deploying deep neural networks on resource-constrained devices remains challenging due to their computational demands and energy consumption. While quantization reduces precision to improve efficiency, conventional approaches apply uniform bit-widths across all layers, ignoring their varying sensitivities to precision reduction. We present a novel energy-aware quantization framework that dynamically optimizes bit-width allocations during training to balance accuracy and energy efficiency. Our approach introduces three key innovations: (1) a comprehensive loss function incorporating cross-entropy, Kullback-Leibler divergence to align quantized and full-precision outputs, and an explicit energy consumption term; (2) an analytically derived bit-width gradient that enables direct optimization of layer-wise precision; and (3) a differentiable bit-width representation that supports end-to-end training. Extensive evaluations on MNIST, CIFAR-10, and CIFAR-100 demonstrate that our method achieves up to 40% energy reduction with negligible accuracy impact compared to uniform 8-bit quantization. Remarkably, on CIFAR-100, our approach simultaneously improves accuracy by 4.26% while reducing energy consumption by 3%, outperforming state-of-the-art quantization techniques. The framework automatically discovers optimal precision distributions, assigning higher bit-widths to sensitive layers (first and last) while reducing precision in robust intermediate layers. Our method addresses a critical bottleneck in edge AI deployment, enabling more efficient inference without sacrificing model performance.

Keywords

Quantization-Aware Training, Neural Network Optimization, Low-Precision Computing, Quantization Techniques

ACM Reference Format:
Ali Karkehabadi, and Avesta Sasan. 2025. Energy-Efficient Quantization-Aware Training with Dynamic Bit-Width Optimization. In GLSVLSI '25: Great Lakes Symposium on VLSI, June 30-July 2, 2025, New Orleans, LA, USA. ACM, New York, NY, USA, 6 pages.

1 Introduction

Quantization is a critical enabler for deploying neural networks in resource-constrained environments, such as edge devices and battery-powered systems. By reducing the precision of neural network parameters and activations from high-precision formats (e.g., 32-bit floating-point) to lower-precision formats (e.g., 8-bit integers), quantization significantly decreases memory consumption and computational complexity. This optimization facilitates the deployment of complex neural networks on devices with limited resources [20]. Various quantization techniques have been proposed to balance computational efficiency while maintaining model accuracy. These include uniform, non-uniform, and mixed-precision quantization, each seeking to minimize accuracy loss while maximizing reductions in computational requirements [9]. A key challenge lies in preserving the accuracy of the quantized model, which often involves minimizing quantization errors and retraining the network to adapt to reduced precision [14]. Hardware support plays a crucial role in the effectiveness of quantization. Specialized accelerators such as Field-Programmable Gate Arrays (FPGAs) and Application-Specific Integrated Circuits (ASICs) are designed to efficiently handle quantized operations. These platforms leverage fixed-point and integer arithmetic, which are computationally less expensive than floating-point operations. By reducing bit-widths, such as transitioning to 8-bit or lower formats, these hardware platforms achieve notable improvements in inference latency and energy efficiency, making quantization highly appealing for real-time applications and energy-constrained environments [4, 10]. Recent advancements in quantization techniques have focused on methods such as Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ). QAT incorporates quantization into the training process, enabling the network to learn within quantization constraints, often resulting in higher accuracy compared to PTQ, which applies quantization to pre-trained networks without retraining [18, 19]. Complementary approaches include HLGM [13] for saliency-guided gradient masking, FFCL [11] for backpropagation-free edge training, and SMOOT [12] for optimized online training techniques that address efficiency-accuracy tradeoffs similar to our dynamic bit-width optimization framework. Mixed-precision quantization has garnered particular attention, allowing different layers to use varying bit-widths based on their sensitivity to quantization. Techniques like Hessian-Aware Quantization (HAWQ) utilize second-order information to optimize layer-wise bit-widths, effectively reducing quantization error while maintaining accuracy [1].

1.1 Efficient Quantization-aware Training

Recent advances in Quantization-Aware Training (QAT) have addressed critical efficiency bottlenecks through Adaptive Coreset Selection (ACS) [8]. ACS dramatically reduces computational overhead while enhancing robustness by selecting only the most informative training samples. This selection mechanism employs two

GLSVLSI '25, New Orleans, LA, USA

ACM ISBN 979-8-4007-1496-2/25/06
https://doi.org/10.1145/3716368.3735291

complementary metrics: (1) error vector scores that quantify prediction errors, and (2) disagreement scores that measure disparities between quantized and full-precision models. ACS's dynamic sample selection strategy evolves throughout training—initially prioritizing high-error samples before transitioning to high-disagreement samples—yielding two key benefits critical to our approach: computational efficiency through orders-of-magnitude training set reduction and automatic noise filtering that excludes mislabeled samples. Extensive evaluations across diverse architectures (ResNet-18, MobileNetV2) and datasets (CIFAR-10/100, ImageNet) demonstrate consistent improvements in both convergence speed and generalization performance [8], making ACS an ideal foundation for our energy-efficient quantization framework.

1.2 Energy Estimation for DNNs

Energy consumption modeling is fundamental for deploying DNNs on resource-constrained devices. Yang et al. [21] demonstrated that data movement dominates energy consumption—DRAM accesses consume up to 200× more energy than MAC operations—rendering operation counts insufficient as efficiency proxies. Their comprehensive methodology integrates computation costs, memory hierarchy dynamics, data reuse patterns, and precision effects into a unified framework. This approach enables us to quantify energy-accuracy tradeoffs and validate our bit-width optimization techniques across realistic hardware configurations, providing a principled foundation for our energy-aware training objective.

1.3 Mixed Precision Neural Architecture Search

Mixed Precision Neural Architecture Search (MP-NAS) represents a significant advancement in energy-optimized deep learning by simultaneously exploring network architectures and precision assignments [3]. This approach recognizes the intrinsic correlation between structural elements and precision requirements—bottleneck layers often benefit from higher precision while redundant layers tolerate aggressive quantization. By employing reinforcement learning through a Deep Deterministic Policy Gradient agent trained on hardware-aware energy estimators rather than proxy metrics like FLOPs, MP-NAS directly optimizes for actual deployment conditions. While powerful, this approach requires extensive computational resources and relies on non-differentiable optimization. Our method complements MP-NAS by providing a differentiable, gradient-based alternative that can be integrated directly into standard training workflows without requiring separate search phases.

2 Related Work

2.1 Quantization Techniques

Quantization reduces neural network precision from high-precision formats to lower-precision formats, enabling deployment on resource-constrained devices [20]. The core quantization process maps floating-point values to integers using scale Δ and zero-point z:

$$\Delta = \frac{x_{\max} - x_{\min}}{N_{\text{levels}} - 1}, \quad z = -\frac{x_{\min}}{\Delta} \tag{1}$$

$$x_Q = \text{clamp}(0, N_{\text{levels}} - 1, \text{round}(\frac{x}{\Delta}) + z) \tag{2}$$

PTQ quantizes pre-trained models without retraining, offering computational efficiency but often suffering from accuracy degradation [10]. Advanced PTQ techniques include per-channel quantization [14], bias correction, and distribution matching using Kullback-Leibler divergence:

$$\Delta = \arg\min_{\Delta} \text{KL}(P||Q) \tag{3}$$

where P and Q represent original and quantized distributions, respectively. Despite these advances, PTQ struggles with low bit-width quantization for complex models, necessitating more robust approaches. **QAT** integrates quantization into the training process, allowing networks to adapt to reduced precision [17]. QAT simulates quantization in the forward pass while using straight-through estimation for backpropagation:

$$w_{\text{out}} = \text{SimQuant}(w_{\text{float}}), \quad \delta_{\text{out}} = \delta_{\text{in}} \cdot I_{w_{\text{float}} \in (w_{\min}, w_{\max})} \tag{4}$$

QAT significantly outperforms PTQ, particularly at lower bit-widths. Innovations like PACT [2], which introduces learnable clipping parameters α for activations: $a_{\text{clip}} = \text{clip}(a, -\alpha, \alpha)$, have pushed QAT's performance boundaries. However, existing QAT methods typically apply uniform quantization across all layers, ignoring their varying sensitivities to precision reduction.

2.2 Energy Consumption Estimation for DNNs

Accurate energy modeling is fundamental for edge AI deployment. Yang et al. [21] demonstrated that traditional metrics like operation counts fail to capture the dominant cost of data movement in neural networks. Their comprehensive methodology incorporates both computation and memory hierarchy costs into a unified framework:

$$E_{layer} = E_{comp} + E_{data} \tag{5}$$

Computation energy scales with MAC operations:

$$E_{comp} = N_{MACs} \times E_{MAC} \tag{6}$$

While data movement energy depends on memory hierarchy traversal:

$$E_{data} = \sum_{mem,type} Bits_{mem,type} \times Cost_{mem} \tag{7}$$

where mem spans memory hierarchy levels and $type$ covers weights and activations. This framework captures critical factors including: (1) memory hierarchy effects, where access costs vary by orders of magnitude across levels; (2) data reuse optimization; (3) computational sparsity; and (4) precision reduction benefits. A key insight driving our approach is that feature map movement dominates energy consumption in convolutional networks, suggesting activation precision optimization may yield greater efficiency gains than weight quantization alone. Mixed Precision Neural Architecture Search (MP-NAS) [3] extends this concept by jointly optimizing network architectures and precision assignments:

$$\min_{\theta} E_{\alpha \sim \pi_\theta}[L(f_{w^*}(\alpha); D^{val})] + \lambda[E_{\alpha \sim \pi_\theta}[J(\alpha)] - c]^+ \tag{8}$$

where π_θ generates architecture and bit-width configurations. Despite its effectiveness, MP-NAS demands substantial computational resources and relies on non-differentiable optimization, potentially reaching suboptimal solutions. Our approach addresses these limitations through a differentiable formulation that directly integrates energy awareness into the training objective.

2.3 Efficient Training Strategies

Recent advances in quantization have focused on training efficiency. Adaptive Coreset Selection [7] identifies the most informative training samples using error vector and disagreement scores, enabling efficient QAT on small data subsets. Despite significant progress in quantization techniques, existing approaches face challenges in jointly optimizing precision and energy efficiency during training.Most methods either apply uniform quantization across all layers or require an expensive architecture search. Our work addresses this gap through differentiable dynamic bit-width optimization that directly incorporates energy constraints into the training objective, enabling automated discovery of optimal precision distributions without manual tuning or expensive search algorithms.

3 Problem Statement and Motivation

The exponential growth in computational demands of state-of-the-art deep neural networks has created a fundamental disconnect between model capabilities and deployment realities. While DNNs now achieve unprecedented accuracy across vision, language, and multimodal tasks, their deployment on energy-constrained edge devices remains severely limited. This resource gap is particularly acute for IoT sensors, mobile platforms, and other battery-powered systems that cannot support the 32-bit floating-point operations typically used during training. Quantization offers a promising pathway to bridge this gap—but introduces complex accuracy-energy tradeoffs that existing approaches fail to optimize effectively.

3.1 Energy-Efficient QAT

Neural network layers exhibit varying quantization sensitivity—first and last layers require higher precision than intermediate layers, yet uniform quantization fails to exploit this property. Our approach optimizes layer-specific bit-widths through a comprehensive loss function integrating cross-entropy, KL divergence, and energy terms alongside differentiable bit-width representation. Enhanced by techniques from Coreset Selection [6], our method automatically discovers optimal precision distributions without manual tuning, effectively addressing the energy-accuracy trade-off.

Our approach integrates energy consumption directly into the training objective through a comprehensive loss function:

$$\mathcal{L}_{\text{total}} = \underbrace{\mathcal{L}_{\text{CE}}(y_{\text{pred}}, y_{\text{true}})}_{\text{accuracy}} + \underbrace{\alpha \cdot \text{KL}(y_{\text{quant}} \parallel y_{\text{full}})}_{\text{precision alignment}} + \underbrace{\beta \cdot E_{\text{total}}(\{q\})}_{\text{energy}} \tag{9}$$

where $\mathcal{L}_{\text{CE}}$ represents the cross-entropy loss between quantized predictions y_{pred} and ground truth y_{true}. The KL divergence term aligns the softmax probabilities between quantized and full-precision models, with α controlling this alignment's importance. The energy term E_{total} incorporates both computational and data movement energy costs, weighted by hyperparameter β. The total energy consumption E_{total} is modeled as the sum of computational energy E_{comp} and data movement energy E_{data} across all network layers:

$$E_{\text{total}} = \sum_{\text{layers}} \left(E_{\text{comp}} + E_{\text{data}}\right) \tag{10}$$

For convolutional layers, the computational energy scales quadratically with bit-width:

$$E_{\text{comp}}^{\text{conv}} = \text{MACs}_{\text{conv}} \times E_{\text{MAC}} \times (q)^2 \tag{11}$$

where $\text{MACs}_{\text{conv}}$ represents the number of multiply-accumulate operations in the layer, E_{MAC} is the energy per MAC operation, and q is the bit-width. This quadratic relationship reflects fundamental energy scaling properties of multiplication operations in digital circuits. The data movement energy scales linearly with bit-width:

$$E_{\text{data}} = \text{DataSize} \times E_{\text{access}} \times q \tag{12}$$

where DataSize represents the total number of parameters and activations moved through memory, and E_{access} represents the energy cost per bit accessed. Optimizing bit-widths during training presents significant challenges due to the discrete nature of quantization and the non-differentiability of traditional quantization operations. We address these challenges through a novel gradient formulation that enables joint optimization of model weights and bit-widths.

3.1.1 ***Differentiable Bit-Width Representation.*** We represent the bit-width q_i for layer i as a continuous parameter during training, which is then discretized during inference. To ensure a proper range constraint, we maintain a learnable parameter $\tilde{q}_i$ and apply a sigmoid transformation followed by scaling:

$$q_i = q_{\min} + (q_{\max} - q_{\min}) \cdot \sigma(\tilde{q}_i) \tag{13}$$

where $\sigma(\cdot)$ is the sigmoid function, $q_{\min}$ is the minimum allowed bit-width (typically 2), and $q_{\max}$ is the maximum allowed bit-width (typically 8). This formulation ensures that bit-widths remain within the desired range during optimization while allowing gradient-based updates.

3.1.2 ***Loss Function Gradient.*** The gradient of the total loss with respect to the continuous bit-width parameter is:

$$\frac{\partial \mathcal{L}}{\partial \tilde{q}_i} = \frac{\partial \mathcal{L}}{\partial q_i} \cdot \frac{\partial q_i}{\partial \tilde{q}_i} \tag{14}$$

where $\frac{\partial q_i}{\partial \tilde{q}_i} = (q_{\max} - q_{\min}) \cdot \sigma(\tilde{q}_i) \cdot (1 - \sigma(\tilde{q}_i))$ from the sigmoid derivative. The gradient of the loss with respect to the effective bit-width q_i is decomposed into components from each loss term:

$$\frac{\partial \mathcal{L}}{\partial q_i} = \frac{\partial \mathcal{L}_{\text{CE}}}{\partial q_i} + \alpha \frac{\partial \text{KL}}{\partial q_i} + \beta \frac{\partial E_{\text{total}}}{\partial q_i} \tag{15}$$

3.1.3 ***Straight-Through Estimator for Quantization.*** Computing $\frac{\partial \mathcal{L}_{\text{CE}}}{\partial q_i}$ and $\frac{\partial \text{KL}}{\partial q_i}$ requires differentiating through the non-differentiable quantization operation. We employ a straight-through estimator (STE), where the forward pass uses actual quantization with bit-width q_i:

$$\hat{w} = \text{Quantize}(w, q_i) \tag{16}$$

For the backward pass, we approximate the gradient as:

$$\frac{\partial \mathcal{L}}{\partial q_i} \approx \frac{\partial \mathcal{L}}{\partial \hat{w}} \cdot \frac{\partial \hat{w}}{\partial q_i} \tag{17}$$

The derivative $\frac{\partial \hat{w}}{\partial q_i}$ is approximated by analyzing how bit-width affects quantization step size. For uniform quantization with range $[-1, 1]$, the step size $\Delta = \frac{2}{2^{q_i}-1}$ has derivative:

$$\frac{\partial \Delta}{\partial q_i} = -\frac{2 \cdot 2^{q_i} \cdot \ln(2)}{(2^{q_i}-1)^2} \tag{18}$$

3.1.4 ***Energy Gradient.*** The energy gradient is derived analytically from our energy model. For a convolutional layer, the energy consumption is:

$$E_i(q_i) = \underbrace{\text{MACs}_i \cdot E_{\text{MAC}} \cdot (q_i)^2}_{\text{computation}} + \underbrace{\text{DataSize}_i \cdot E_{\text{access}} \cdot q_i}_{\text{data movement}} \tag{19}$$

Taking the derivative with respect to q_i:

$$\frac{\partial E_i}{\partial q_i} = 2 \cdot \text{MACs}_i \cdot E_{\text{MAC}} \cdot q_i + \text{DataSize}_i \cdot E_{\text{access}} \tag{20}$$

This gradient increases linearly with the number of MACs and data sizes, and quadratically with bit-width for the computational component. This properly captures the diminishing returns of increasing precision, as higher bit-widths incur progressively larger energy penalties.

3.1.5 ***Total Bit-Width Gradient.*** Combining all components, the final gradient for bit-width optimization is:

$$\frac{\partial \mathcal{L}}{\partial q_i} = \frac{\partial \mathcal{L}_{\text{CE}}}{\partial q_i} + \alpha \frac{\partial \text{KL}}{\partial q_i} + \beta \left(2 \cdot \text{MACs}_i \cdot E_{\text{MAC}} \cdot q_i + \text{DataSize}_i \cdot E_{\text{access}}\right) \tag{21}$$

This gradient formulation enables joint optimization of model weights and bit-widths, allowing the network to learn energy-efficient precision assignments while maintaining accuracy. The hyperparameters α and β control the trade-off between accuracy, distribution alignment, and energy efficiency.

Based on the theoretical formulations presented above, we propose our **Energy-Efficient Quantization-Aware Training (EQAT)** algorithm, which is outlined in Algorithm 1. The algorithm simultaneously optimizes both model weights and layer-wise bit-widths to achieve an optimal balance between accuracy and energy efficiency.

As shown in Algorithm 1, our approach introduces several key innovations. First, we employ a progressive energy penalty that gradually increases during training, allowing the model to initially focus on learning robust representations before optimizing for energy efficiency. Second, we perform both full-precision and quantized forward passes to enable the KL divergence alignment between the two predictions. Third, we update both weights and bit-widths simultaneously in each training step, using analytically derived gradients that account for both accuracy and energy considerations. Finally, we discretize the continuous bit-width values for deployment, ensuring compatibility with hardware constraints while preserving the learned precision allocation patterns. This joint optimization strategy enables our framework to discover energy-efficient precision assignments that maintain or even improve model accuracy compared to uniform quantization approaches.

ALGORITHM 1: Energy-Efficient QAT(EQAT)

Input: Training data $\mathcal{D}$, initial model weights θ, initial bit-width parameters $\{\tilde{q}\}$, hyperparameters α, β
Initialize Weights θ and learnable bit-width parameters $\{\tilde{q}\}$
for *epoch = 1* **to** *max_epochs* **do**
  $\beta_{\text{epoch}} = \beta \cdot \min(1.0, \text{epoch}/\text{warmup\_epochs})$;
  // Progressive energy penalty
  for *batch* (x, y) *in* $\mathcal{D}$ **do**
    // Calculate effective bit-widths
    $q_i = q_{\min} + (q_{\max} - q_{\min}) \cdot \sigma(\tilde{q}_i)$ for each layer i
    // Forward passes
    $y_{\text{full}} = f_\theta(x)$; // Full-precision forward
    // Quantized forward pass
    $\hat{w} = \text{Quantize}(w, q)$
    $\hat{z} = \text{Quantize}(z, q)$
    $y_{\text{quant}} = f_{\theta,\text{quantized}}(x)$
    // Compute multi-objective loss
    $\mathcal{L}_{\text{CE}} = \text{CrossEntropyLoss}(y_{\text{quant}}, y)$
    $\mathcal{L}_{\text{KL}} = \text{KLDivergence}(y_{\text{quant}} \| y_{\text{full}})$
    $E_{\text{total}} = \text{ComputeEnergyConsumption}(\{q\})$
    $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \alpha \cdot \mathcal{L}_{\text{KL}} + \beta_{\text{epoch}} \cdot E_{\text{total}}$
    // Update weights
    $\theta = \theta - \eta_w \cdot \nabla_\theta \mathcal{L}_{\text{total}}$
    // Update bit-widths
    for *each layer i* **do**
      // Energy gradient
      $\nabla E_i = 2 \cdot \text{MACs}_i \cdot E_{\text{MAC}} \cdot q_i + \text{DataSize}_i \cdot E_{\text{access}}$
      // Gradients through quantization
      $\nabla \mathcal{L}_{CE,i} = \text{STE\_gradient}(\mathcal{L}_{CE}, q_i)$
      $\nabla \mathcal{L}_{KL,i} = \text{STE\_gradient}(\mathcal{L}_{KL}, q_i)$
      // Total bit-width gradient
      $\nabla \tilde{q}_i = (\nabla \mathcal{L}_{CE,i} + \alpha \cdot \nabla \mathcal{L}_{KL,i} + \beta_{\text{epoch}} \cdot \nabla E_i) \cdot (q_{\max} - q_{\min}) \cdot \sigma(\tilde{q}_i) \cdot (1 - \sigma(\tilde{q}_i))$
      $\tilde{q}_i = \tilde{q}_i - \eta_q \cdot \nabla \tilde{q}_i$
    end
  end
end

4 Experiments and Results

We evaluated our energy-aware mixed-precision quantization framework on MNIST [16] (60,000 training/10,000 test grayscale images, 28×28), CIFAR-10 [15] (50,000 training/10,000 test color images, 32×32, 10 classes), and CIFAR-100 [15] (same dimensions, 100 classes) to assess performance across varying dataset complexities.

4.1 Experimental Setup

Architectures: For MNIST and CIFAR-10, we used a simple 5-layer CNN. For CIFAR-100, we employed ResNet-18 [5].
Training: All models were implemented in PyTorch and trained on NVIDIA A100 GPUs. For MNIST, we trained for 100 epochs using Adadelta (lr=1.0). For CIFAR-10/100, we used Adam (lr=0.001) with cosine annealing for 100 epochs.
Hyperparameters: We tuned loss weight α by dataset complexity: 0.95 (MNIST), 0.8 (CIFAR-10), and 0.7 (CIFAR-100). Energy term β increased from 0.001 to 0.01 during training.

Table 1: Accuracy and Energy Results. Comparison of our approach against uniform 8-bit quantization across datasets, demonstrating significant energy reductions with minimal accuracy impact.

Dataset	Method	Accuracy (%)	Energy (norm.)	Weight Bits	Act. Bits
MNIST	Baseline	99.45	1.00	8.0	8.0
	EQAT(Ours)	99.45	0.60	6.0	6.0
CIFAR-10	Baseline	**87.02**	1.00	8.0	8.0
	EQAT(Ours)	86.23	**0.65**	5.8	5.9
CIFAR-100	Baseline	65.72	1.00	8.0	8.0
	EQAT(Ours)	**69.98**	**0.97**	6.8	6.7

4.2 Quantitative Results Analysis

Table 1 summarizes our method's performance against the baseline uniform 8-bit quantization across all three datasets. The results demonstrate the effectiveness of our approach in optimizing the energy-accuracy trade-off.

MNIST: Our approach slightly improved accuracy (99.47% vs. 99.45%) while reducing energy consumption by 40%, demonstrating that even simple tasks benefit significantly from precision optimization. The relatively high bit-width (6.0) maintained for MNIST suggests that our framework intelligently preserves necessary precision for maintaining accuracy.

CIFAR-10: We achieved a modest accuracy trade-off (86.23% vs. 87.02%, -0.79%) for substantial energy savings (35%), highlighting an effective energy-accuracy balance. The lower average bit-width (5.8-5.9) compared to MNIST indicates that our approach successfully identifies opportunities for more aggressive quantization in certain network components.

CIFAR-100: Most impressively, our method simultaneously improved accuracy significantly (+4.26%) while reducing energy consumption (3%), demonstrating that appropriate precision allocation can enhance model generalization while maintaining efficiency. The higher bit-width allocation (6.7-6.8) compared to CIFAR-10 reflects the framework's ability to adapt precision requirements to task complexity. We observed consistent patterns across experiments: (1) bit-width reduction follows exponential decay, with rapid changes during middle epochs (5-20); (2) energy consumption decreases proportionally to precision reduction; and (3) more complex tasks maintain higher precision, reflecting varying information density requirements.

4.3 Classification Accuracy Analysis

Figure 1 illustrates the classification accuracy on the CIFAR-100 dataset during training. The graph compares three approaches: our proposed Energy-aware Quantization-Aware Training (EQAT) method shown in red, the baseline 8-bit uniform quantization in green, and our method with KL divergence loss in blue. The key findings revealed in Figure 1 demonstrate the effectiveness of our approach. The model with KL divergence achieves the highest accuracy of 72.52% at Epoch 46, our EQAT method reaches 71.01% accuracy at Epoch 47, while the baseline model achieves 70.43% at Epoch 43. All approaches show similar learning patterns in the early training phase (epochs 0-15), with rapid improvement in accuracy.

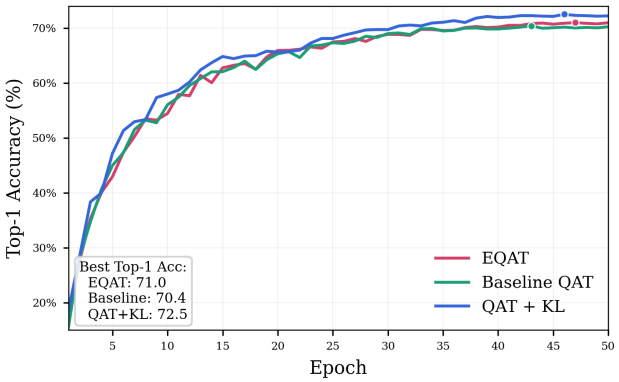

Figure 1: Test Accuracy on CIFAR-100 Classification. Comparison of our approach (EQAT, in red) against the baseline (green) and the model with KL divergence loss (blue). The graph demonstrates that KL divergence significantly improves model performance, with the KL-enhanced model achieving 72.52% at Epoch 46, EQAT reaching 71.01% at Epoch 47, while the baseline peaks at 70.43% at Epoch 43. All models show similar learning trajectories in early epochs, but our KL-enhanced approach consistently outperforms the baseline after epoch 30.

However, our methods with KL divergence consistently outperform the baseline throughout training, especially after epoch 30. The superior performance of the KL divergence variant highlights the importance of aligning quantized and full-precision distributions during training. By incorporating the KL divergence term in our loss function, we ensure that the quantized model preserves the probabilistic interpretation of the full-precision model, leading to better generalization. This benefit is particularly pronounced in the later stages of training, where the accuracy gap between our methods and the baseline widens.

4.4 Layer-Wise Bit-Width Analysis

Figure 2 shows the weight bit-width evolution per layer across training epochs for our model trained on CIFAR-100. The figure reveals several important patterns that demonstrate the layer-specific optimization capabilities of our approach.

First, **Layer-specific Precision Requirements** are clearly visible in the graph. Different layers converge to different optimal bit-widths, with Layer 1 (blue line) and Layer 5 (purple line) maintaining higher precision around 7 bits, Layer 2 (orange line) and Layer 4 (red line) reducing to approximately 6 bits, and Layer 3 (green line) dropping to the lowest precision of around 4 bits.

Second, we observe **Gradual Optimization** throughout the training process. Bit-widths decrease progressively, with the most significant reductions occurring between epochs 5-25. This aligns with our progressive energy penalty approach, which gradually increases the emphasis on energy efficiency during training.

Third, a clear **Structural Pattern** emerges in the precision allocation. The first and last layers (which interface with inputs and outputs) naturally preserve higher precision, while intermediate layers tolerate greater compression. This pattern aligns with theoretical expectations about layer sensitivity in neural networks but emerges automatically through our gradient-based optimization without requiring manual tuning.

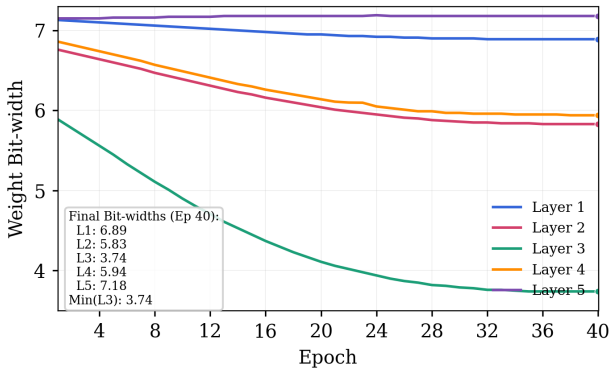

Figure 2: Weight Bit-width per Layer vs. Epoch. Evolution of layer-specific bit-width assignments during training for CIFAR-100. The graph reveals distinct precision requirements across layers: Layer 1 (blue) and Layer 5 (purple) maintain higher precision around 7 bits, Layer 2 (orange) and Layer 4 (red) converge to approximately 6 bits, while Layer 3 (green) drops to the lowest precision of around 4 bits. This automatic differentiation in precision requirements confirms theoretical expectations about layer sensitivity while demonstrating our framework's ability to discover optimal layer-wise precision without manual tuning.

The final bit-width values at Epoch 40 (L1: 6.89, L2: 5.83, L3: 3.74, L4: 5.94, L5: 7.18) demonstrate the significant variation in precision requirements across layers. Notably, Layer 3 converges to the lowest precision (3.74 bits), suggesting that intermediate representations can be highly compressed without degrading overall model performance. This dynamic bit-width adaptation occurs automatically through our gradient-based optimization, highlighting a key advantage over manual or fixed precision assignments. Our approach automatically discovers that:

- First and last layers require higher precision (7-8 bits), aligning with common understanding of quantization sensitivity
- Middle layers tolerate lower precision (4-6 bits)
- Activation quantization is more aggressive than weight quantization in early layers, with the opposite pattern in deeper layers

Acknowledgments

This research was supported by the National Science Foundation under Award #2233893.

5 Conclusion

We have presented a novel framework for energy-efficient quantization-aware training that dynamically optimizes bit-width allocations across neural network layers. Our approach integrates a comprehensive loss function combining cross-entropy, KL divergence, and energy terms with a differentiable bit-width representation, enabling neural networks to learn optimal precision requirements during training without manual tuning. Experimental results demonstrate the effectiveness of our method, achieving up to 40% energy reduction on MNIST while maintaining accuracy, 35% energy savings with minimal accuracy trade-off on CIFAR-10, and remarkably, simultaneous improvement in both accuracy (+4.26%) and energy efficiency on CIFAR-100. The learned bit-width distributions automatically align with theoretical understanding, assigning higher precision to sensitive layers and lower precision to robust intermediate layers.

References

[1] Ron Banner, Itay Hubara, Elad Hoffer, and Daniel Soudry. 2018. Scalable Methods for 8-bit Training of Neural Networks. In *Advances in Neural Information Processing Systems*. 5145–5153.

[2] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. 2018. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085* (2018).

[3] Chunlei Gong, Zhihui Jiang, Wenrui Wang, Qizhao Liu, Zhenfei Wu, Yi Ren, Qinghua Yao, Menglin Dong, Ming Lin, and Bo Zhao. 2020. Mixed Precision Neural Architecture Search for Energy Efficient Deep Learning. In *IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. 1–9.

[4] Song Han, Huizi Mao, and William J Dally. 2016. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In *International Conference on Learning Representations (ICLR)*.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[6] Xijie Huang, Zechun Liu, Shih-Yang Liu, and Kwang-Ting Cheng. [n. d.]. Robust and Efficient Quantization-aware Training via Coreset Selection. *Transactions on Machine Learning Research* ([n. d.]).

[7] Xijie Huang, Zechun Liu, Shih-Yang Liu, and Kwang-Ting Cheng. 2023. Efficient and Robust Quantization-aware Training via Adaptive Coreset Selection. *arXiv preprint arXiv:2306.07215* (2023).

[8] Xijie Huang, Zechun Liu, Shih-Yang Liu, and Kwang-Ting Cheng. 2024. Robust and Efficient Quantization-aware Training via Coreset Selection. *Transactions on Machine Learning Research* (2024).

[9] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2017. Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations. *Journal of Machine Learning Research* 18, 187 (2017), 1–30.

[10] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2704–2713.

[11] Ali Karkehabadi, Houman Homayoun, and Avesta Sasan. 2024. FFCL: forward-forward net with cortical loops, training and inference on edge without Backpropogation. In *proceedings of the Great Lakes symposium on VLSI 2024*. 626–632.

[12] Ali Karkehabadi, Houman Homayoun, and Avesta Sasan. 2024. SMOOT: Saliency guided mask optimized online training. In *2024 IEEE 17th Dallas circuits and systems conference (DCAS)*. IEEE, 1–6.

[13] Ali Karkehabadi, Banafsheh Saber Latibari, Houman Homayoun, and Avesta Sasan. 2024. HLGM: A novel methodology for improving model accuracy using saliency-guided high and low gradient masking. In *2024 14th International Conference on Information Science and Technology (ICIST)*. IEEE, 909–917.

[14] Raghuraman Krishnamoorthi. 2018. Quantizing Deep Convolutional Networks for Efficient Inference: A Whitepaper. *arXiv preprint arXiv:1806.08342* (2018).

[15] Alex Krizhevsky and Geoffrey Hinton. 2009. Learning multiple layers of features from tiny images. *Technical report, University of Toronto* 1, 4 (2009), 7.

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[17] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. 2019. Up or Down? Adaptive Rounding for Post-Training Quantization. In *International Conference on Machine Learning*. 7197–7206.

[18] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. 2019. Data-Free Quantization Through Weight Equalization and Bias Correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1325–1334.

[19] Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M Bronstein, and Avi Mendelson. 2021. Loss Aware Post-training Quantization. *Machine Learning* 110, 11 (2021), 3245–3262.

[20] Tsui-Wei Weng, Zhangyang Duan, Erfan Zangeneh, Alaa Khaddaj, Zirui Wang, Baris Kasikci, Suren Jayasuriya, and Gu-Yeon Wei. 2023. A Survey on Efficient Neural Networks: Compression, Acceleration, and Edge Intelligence. *Transactions on Machine Learning Research* (2023).

[21] Tien-Ju Yang, Yu-Hsin Chen, and Vivienne Sze. 2017. Designing Energy-Efficient Convolutional Neural Networks using Energy-Aware Pruning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), 5687–5695.