Journal of Theoretical and Applied Information Technology
15st June 2026. Vol.104. No.11

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

MAGNITUDE AND PENALTY-BASED PRUNING FOR QUANTIZATION WITH ZERO-SHOT APPROACH

PRIYANGA K.K[1,3,*], S. SABEEN[2]

[1,*] Research Scholar, Department of Computer Science, FSH, SRMIST, Kattankulathur, Tamil Nadu-603203
[2]Assistant Professor, Department of Computer Science, FSH, SRMIST, Kattankulathur, Tamil Nadu-603203
[3]Assistant Professor, Department of Computer Science, Christ College (Autonomous), Irinjalakuda, Kerala-680125
E-mail: [1]pk9856@srmist.edu.in, [2] sabeens@srmist.edu.in

ABSTRACT

Deep convolutional neural networks deliver high predictive performance in image classification. However, they often require substantial computation and memory. This limits their use in resource-constrained environments. Many existing compression methods apply pruning and quantization separately. They also focus mainly on supervised learning with fixed label spaces. As a result, they may cause accuracy loss, high optimization cost, and limited adaptability to unseen classes. To address these limitations, this study proposes a hybrid Network Pruning and Quantization framework integrated with zero-shot learning for compressed deep convolutional neural networks. The proposed NPQ framework combines magnitude-based pruning, penalty-based sparsity regularization, and quantization-aware optimization within a unified design. It also preserves semantic embedding consistency to support recognition of unseen classes. The framework is evaluated on CIFAR 10, AWA2, and CUB benchmark datasets. Its performance is compared with recent pruning and quantization methods including LQ NET, RESREP, AutoPruner, and SIGMA. The experimental results show that NPQ achieves competitive accuracy while reducing model complexity, memory usage, and inference latency. Ablation studies further confirm that each component contributes to compression efficiency and zero shot generalization performance. Overall, the proposed NPQ framework offers an effective approach for building compact and efficient deep learning models for real time and resource-constrained deployment.
Keywords: *Deep Convolutional Neural Networks, Pruning, Quantization, Zero-Shot Learning, Model Compression, Optimization*

1. INTRODUCTION

Deep convolutional neural networks have achieved strong performance in many image processing and computer vision tasks [1], [16]. They are widely used for image classification, object detection, semantic segmentation and medical image analysis [1–4]. Their success is mainly due to the ability to learn hierarchical features directly from raw image data [1]. Convolutional layers extract local patterns such as edges, textures and shapes, while pooling layers reduce the spatial resolution and preserve important information [1], [16]. As network architectures become deeper and wider, they often obtain higher accuracy on standard benchmark datasets, but with higher computational and memory demands [1], [10], [14].

This improvement in accuracy introduces practical limitations. Modern deep neural networks contain a large number of parameters and require significant computational power, memory capacity and inference time [1], [10], [14]. These requirements make deployment difficult on resource-constrained platforms such as mobile devices, embedded systems, edge computing nodes and Internet of Things environments [3], [4], [20]. Many real time applications need models that are compact, energy efficient and have low latency [3], [6], [31]. Therefore, reducing model complexity while maintaining predictive accuracy has become an important research problem in deep learning [1], [2], [14].

Network pruning and quantization are two major techniques for compressing and accelerating deep neural networks [1], [2], [3], [14]. Pruning removes redundant or low importance parameters so that the model becomes smaller and sparser [11–15]. Early approaches such as Optimal Brain Damage

and Optimal Brain Surgeon used second order information to identify unimportant weights [11], [12]. Later methods introduced data driven and dynamic pruning strategies, including deep compression and dynamic network surgery [14], [15]. Structured pruning methods remove whole filters or channels and provide hardware friendly sparsity [37–39]. ``Quantization reduces the numerical precision of weights and activations and thus lowers memory usage and computation cost [21–24], [40], [41]. Many works have shown that pruning and quantization can be combined to achieve large reductions in model size and inference time with limited accuracy loss [2], [3], [9], [42], [43].

Recent studies have proposed advanced joint pruning and quantization methods. Examples include differentiable joint pruning and quantization and one-shot pruning quantization, as well as hardware friendly pruning quantization schemes [2], [6], [43], [44]. Survey papers provide taxonomies and comparisons of such methods and highlight open challenges in balancing accuracy, compression ratio and hardware efficiency [1], [10], [36]. Other works investigate automatic or reinforcement-learning-based pruning, as in AutoPruner and multi agent transformer-based pruning [32], [34]. Methods such as LQ Nets, ResRep, PruneTrain, ProxSkip and SIGMA further improve quantization-aware training, channel pruning, efficient sparse training and mixed precision inference [17], [27], [28], [30], [35]. These approaches demonstrate that careful design of pruning and quantization strategies can yield efficient models that are competitive with dense baselines [2], [6], [27].

However most existing compression methods focus on supervised settings with fixed label spaces and do not explicitly address generalization to unseen tasks or classes [9], [32], [27], [35–39]. In many real-world scenarios, it is expensive or impractical to collect labeled data for all possible categories. Zero-shot learning aims to recognize unseen classes by using auxiliary information such as attributes or semantic embeddings and has been widely studied on datasets like AWA2 and CUB [8], [14]. When deep networks are heavily pruned and quantized, the learned feature representations may be distorted and the alignment between visual features and semantic descriptors can be weakened [1], [6], [40], [41], [42]. This makes it challenging to combine aggressive compression with robust zero shot generalization.

Existing work on pruning, quantization and zero-shot learning is often developed and evaluated in separate lines of research [1], [10], [36]. There is limited research on unified frameworks that jointly optimize structured pruning, quantization-aware training and semantic alignment for unseen classes within a single pipeline [2], [9], [43], [44]. Reinforcement-learning-based and automatic pruning methods can also introduce large computational overhead during the search process, which reduces their suitability for edge deployment and real time applications [32], [34]. These limitations motivate the need for compression frameworks that deliver high compression ratios and low inference latency while preserving accuracy and maintaining adaptability to unseen tasks [1], [2], [3], [10].

To address these limitations, the present study proposes a hybrid network pruning and quantization framework that integrates magnitude-based pruning, penalty-based sparsity regularization, quantization-aware optimization and zero-shot learning within a unified architecture [2], [3], [9], [44]. The goal is to obtain compact and efficient convolutional neural networks that achieve competitive accuracy on benchmark datasets while retaining the ability to generalize to unseen classes under realistic deployment constraints [1], [2], [6].

Unlike earlier studies that primarily optimize pruning or quantization independently, the proposed NPQ framework jointly integrates structured pruning, quantization-aware optimization and semantic embedding preservation within a unified architecture. The motivation of this work is not only to improve compression efficiency, but also to maintain adaptability to unseen classes under resource-constrained deployment conditions. The experimental findings demonstrate that NPQ achieves competitive compression and inference efficiency while preserving semantic consistency and zero-shot learning capability across multiple benchmark datasets.

2. BACKGROUND STUDY

Network pruning has become a key technique for compressing deep neural networks by removing unimportant connections and then fine-tuning the remaining weights to recover accuracy [11–15]. Early methods used second order derivatives or data independent criteria to estimate the importance of each parameter, as in Optimal Brain Surgeon and related approaches [11], [12]. Deep Compression later showed that many low magnitude weights can be removed and that pruning followed by retraining can give compact models with only minor accuracy loss [14]. A central challenge is that the importance of a connection

depends on interactions with other connections, so removing one weight can change the role of others [11], [12]. Recent work addresses this with adaptive pruning strategies such as Dynamic Network Surgery and CLIP Q, which update pruning decisions during training instead of using a single static mask [15], [17].

Quantization further improves compression by representing weights and activations with low precision values. Techniques include scalar quantization, vector quantization and binary or ternary weight networks, which can reduce memory usage and computational cost [21–24], [40], [41]. Deep Compression separates pruning and quantization in a two-stage pipeline where weights are first pruned and then quantized with soft weight sharing based on a Gaussian mixture model [14], [42]. Weighted entropy-based quantization adjusts quantization levels so that near zero and very large weights use fewer levels, which gives a more efficient distribution of codebooks across the weight range [26], [40], [41]. These methods show that pruning and quantization are complementary, but they often treat them as loosely coupled stages rather than as a single joint optimization problem [2], [9], [43].

More recent approaches consider joint pruning and quantization in one unified framework. Methods such as DJPQ and OPQ treat compression as a single differentiable optimization and jointly compute pruning masks and bit precision allocation to reduce bit operations and improve hardware efficiency [2], [42], [43]. Hardware friendly pruning quantization schemes such as HFPQ focus on sparsity patterns and quantization layouts that map well to specific accelerators or embedded devices [6]. Other methods explore one shot pruning quantization, which applies pruning and quantization in a single pass and avoids repeated retraining [2], [42]. These works show that joint optimization can outperform sequential pipelines in compression ratio and latency [2], [3], [9], [42], [43]. However, they mainly target supervised tasks with fixed label spaces and do not consider generalization to unseen classes or zero shot settings [8], [14].

Several recent pruning and quantization methods illustrate these trends. LQ Nets propose learned step size quantization to obtain compact models with high accuracy on ImageNet [27]. ResRep focuses on filter re parameterization pruning and achieves lossless compression on convolutional networks [17], [38]. PruneTrain introduces training aware structured pruning and dynamic sparse model reconfiguration to speed up training and inference [28], [39]. AutoPruner uses reinforcement learning to choose pruning policies and can adapt decisions across layers and architectures [32], [34]. SIGMA combines signal to noise-based pruning with mixed precision quantization and targets efficient deployment on general matrix multiplication accelerators [35], [37]. These methods reduce redundancy in deep networks, but most of them still optimize pruning or quantization as the main focus and treat the other component as secondary or fixed [1], [10], [44].

Automatic and reinforcement-learning-based pruning methods also add significant computational overhead. AutoPruner and related reinforcement learning frameworks require large search spaces and repeated training runs, which increases optimization cost and makes them less suitable for edge devices and real time scenarios [32], [34]. Mixed techniques such as SIGMA integrate pruning and quantization but focus on supervised tasks and do not address generalization to unseen categories [35], [37]. As a result, these approaches achieve strong compression and speed on known tasks but offer limited support for zero-shot learning or domain adaptation.

Table 1 summarizes representative pruning and quantization methods and compares them with the proposed NPQ framework. LQ Nets focus on learned quantization and provide limited integration with pruning [27]. ResRep emphasizes filter pruning and requires additional retraining, and does not include quantization to reduce memory and bandwidth [17], [38]. PruneTrain focuses on structured pruning during training and has high training overhead with limited hardware awareness [28], [39]. AutoPruner relies on reinforcement learning, which increases search cost [32], [34]. SIGMA combines pruning with mixed precision quantization but does not include zero shot mechanisms and gives limited support for unseen classes [35], [37]. In contrast, NPQ jointly optimizes magnitude-based pruning, penalty-based sparsity regularization and adaptive quantization in a single pipeline and integrates zero-shot learning through semantic embeddings [2], [3], [9], [43].

The main difference between NPQ and these frameworks is that pruning, quantization and zero-shot learning are optimized together. NPQ uses semantic embeddings, for example from CLIP, to align compressed CNN features with high level semantic representations and to perform inference on unseen classes without reducing the compression ratio or increasing inference time [8], [14], [43]. Most zero-shot learning research on CNNs evaluates models on attribute-based datasets such as AWA2, CUB and SUN, but these models often collapse when aggressively compressed because quantization

distorts semantic alignment [8], [14]. NPQ links pruning with preservation of semantic embedding structure and keeps feature attribute consistency in compressed models.

This allows NPQ to achieve higher performance than traditional zero shot baselines such as DeViSE and SAE under memory and latency constraints and makes it suitable for edge deployment when both compactness and flexibility to unseen tasks are important [2], [3], [9], [43].

In summary, prior work shows progress in pruning, quantization and joint compression, but reveals three main limitations. Many methods separate pruning and quantization into independent stages or focus on one aspect, which

can lead to suboptimal sparsity accuracy trade offs [1], [2], [9], [43]. Automatic and reinforcement-learning-based pruning approaches increase optimization cost and limit deployment on resource-constrained devices [32], [34]. Existing compression methods rarely integrate zero-shot learning and do not enforce semantic consistency under aggressive pruning and quantization, which is necessary for robust performance on unseen classes [8], [14], [43]. These limitations motivate a unified framework such as NPQ that combines structured pruning, quantization-aware optimization and zero-shot learning in a single architecture.

Table 1 summarizes representative pruning and quantization methods and compares them with the proposed NPQ framework. LQ Nets focus on learned quantization and provide limited integration with pruning [29]. ResRep emphasizes filter pruning and requires additional retraining, and does not include quantization to reduce memory and bandwidth [18], [38]. PruneTrain focuses on structured pruning during training and has high training overhead with limited hardware awareness [30], [39]. AutoPruner relies on reinforcement learning, which increases search cost [10], [40]. SIGMA combines pruning with mixed precision quantization but does not include

Table 1. Comparative Analysis of Existing Pruning + Quantization Methods vs. NPQ

Method	Core Technique	Benchmark Dataset	Accuracy (%)	Compression Ratio (×)	Inference Time (ms)	Key Limitation	Novelty of NPQ Compared to Method
LQ-NET [29]	Learned step size quantization	ImageNet	71.5 (Top-1)	8×	2.8	Focused only on quantization; limited pruning integration	NPQ jointly optimizes pruning + quantization, avoiding sequential degradation
RESREP [38]	Filter re-parameterization pruning	ImageNet	72.6 (Top-1)	12×	2.2	Requires retraining; no quantization support	NPQ integrates quantization, reducing memory + bandwidth simultaneously
PruneTrain [39]	Training-aware structured pruning	CIFAR-10	93.1	10×	1.9	Training overhead is high; not hardware-aware	NPQ uses penalty-based pruning with lightweight zero shot tuning

AutoPruner [40]	Reinforcement learning-based pruning	CIFAR-100	70.4	14×	1.6	RL search is computationally expensive	NPQ avoids heavy RL search, leveraging gradient-guided scheduling
SIGMA [41]	Signal-to-noise pruning + mixed quantization	ImageNet	73.3	15×	1.3	Limited zero shot generalization	NPQ integrates semantic embeddings to support zero shot inference
NPQ (Proposed)	Hybrid pruning (magnitude + penalty) + adaptive quantization + ZSL	CIFAR-10, AWA2, CUB	**99.56 (CIFAR-10), 74.2 (AWA2)**	**16×**	**0.5**	–	Unifies pruning + quantization + ZSL; enables unseen-class generalization

zero shot mechanisms and gives limited support for unseen classes [33], [41]. In contrast, NPQ jointly optimizes magnitude-based pruning, penalty-based sparsity regularization and adaptive quantization in a single pipeline and integrates zero-shot learning through semantic embeddings [2], [3], [9], [52].

The main difference between NPQ and these frameworks is that pruning, quantization and zero-shot learning are optimized together. NPQ uses semantic embeddings, for example from CLIP, to align compressed CNN features with high level semantic representations and to perform inference on unseen classes without reducing the compression ratio or increasing inference time [8], [15], [52]. Most zero-shot learning research on CNNs evaluates models on attribute-based datasets such as AWA2, CUB and SUN, but these models often collapse when aggressively compressed because quantization distorts semantic alignment [8], [15]. NPQ links pruning with preservation of semantic embedding structure and keeps feature attribute consistency in compressed models. This allows NPQ to achieve higher performance than traditional zero shot baselines such as DeViSE and SAE under memory and latency constraints and makes it suitable for edge deployment when both compactness and flexibility to unseen tasks are important [2], [3], [9], [52].

In summary, prior work shows progress in pruning, quantization and joint compression, but reveals three main limitations. Many methods separate pruning and quantization into independent stages or focus on one aspect, which can lead to suboptimal sparsity accuracy trade offs [1], [2], [9], [50]. Automatic and reinforcement-learning-based pruning approaches increase optimization cost and limit deployment on resource-constrained devices [10], [35], [40]. Existing compression methods rarely integrate zero-shot learning and do not enforce semantic consistency under aggressive pruning and quantization, which is necessary for robust performance on unseen classes [8], [15], [52]. These limitations motivate a unified framework such as NPQ that combines structured pruning, quantization-aware optimization and zero-shot learning in a single architecture.

2.1 Problem statement and research questions

Despite notable progress in pruning and quantization, many deep neural network compression methods still treat these operations separately. Most existing approaches also focus on supervised learning with fixed label spaces. This limits their use in deployment settings where computational resources are limited and adaptability is important. In such cases, compression may lead to accuracy loss, higher optimization cost, and weaker generalization to unseen classes.

A further concern is that aggressive compression can distort semantic representations. This may reduce the ability of a model to support zero-shot learning. In addition, reinforcement-learning-based pruning methods often require high computational cost and extended training time. These limitations show the need for a unified framework that jointly optimizes pruning and quantization while preserving semantic embedding quality and maintaining compression efficiency.

To address these issues, this study explores the following research questions

Q1. Can the proposed NPQ framework reduce model complexity and inference time while maintaining predictive accuracy?

Q2. Can joint optimization of pruning and quantization perform better than traditional sequential approaches?

Q3. Can preserving semantic embeddings improve zero shot generalization in compressed models?

3. PROPOSED METHODOLOGY

Pruning is a method for deleting connections or neurons from deep neural networks of low value, which helps simplify the networks. Magnitude-based pruning reduces the size of the network by eliminating links with the smallest weights. On the other hand, penalty-based pruning involves including a penalty term in the loss function that encourages some weights to be near zero, resulting in their removal from training. The generic weight matrix in the NPQ model is given in Equation 1.

$$W \epsilon \mathbb{R}^{n \times n} \quad (1)$$

A parallel matrix containing corresponding importance scores is introduced to identify the weights utilized for pruning, as depicted in Equation 2.

$$S \epsilon \mathbb{R}^{n \times n} \quad (2)$$

Given importance scores, each pruning strategy computes a mask in Equation 3.

$$M \epsilon \{0,1\}^{n \times n} \quad (3)$$

Inference for an input value x becomes $a = (W \odot M)x$ and $\odot$ is the Hadamard product. A general technique is to preserve the topv percent of weights by significance. It determined Topv as a function that selects the v% of the highest values in S:

$$Top_v(S)_{i,j} = \begin{cases} 1, & S_{i,j} \ in \ top \ v\% \\ 0, & o.w. \end{cases} \quad (4)$$

Weight pruning based on magnitude involves creating a mask that relies on the absolute values of individual weights to measure their significance. The prominent scores and mask factors are given in Equations 5 and 6, respectively.

$$S = \left(|W_{i,j}|\right)_{1 \le i,j \le n} \quad (5)$$

$$M = Top_v(S) \quad (6)$$

S: A matrix of importance scores, |W_{i, j}|: The absolute value of each weight in the weight matrix W, i,j: Indices for the rows and columns of the matrix, n: The dimension of the matrix There exist several extensions to the fundamental configuration described above, which is based on iterative magnitude pruning. Initially, the model undergoes training until convergence is achieved, after which weights with the lowest magnitudes are selectively removed. Subsequently, the pruned model is retrained while keeping the eliminated weights fixed at 0. This process is reiterated until the desired level of sparsity is attained. This study adopts an automated gradual pruning approach, supplementing magnitude pruning. This method allows for updates to the masked weights, ensuring they are not immobilized throughout the training process. Automated gradual pruning offers the advantage of enabling the model to adapt from prior masking decisions. Furthermore, it permits the gradual increment of the sparsity level (denoted as "v") during training, utilizing a cubic sparsity scheduler as described in Equation 7.

$$v^{(t)} = v_f + (v_i - v_f)\left(1 - \frac{t - t_i}{N \Delta t}\right)^3 \quad (7)$$

v(t): The sparsity level at time step t, v_f: The final (target) sparsity level, v_i: The initial sparsity level,

t : Current time step.

t_i : Initial time step when pruning begins.

N: The number of pruning steps.

Δt : Time interval between pruning steps.

The sparsity level at time step t, v(t) is increased from an initial value vi (usually 0) to a final value vf in n pruning steps after ti steps of warm-up. The model is thus pruned and trained jointly. In penalty-based pruning, a regularization term to the loss function is added to encourage weights to become small or zero. The loss function with Penalty is given in Equation 8.

$$L = original_{loss} + \alpha * \sum\left(W_{ij} \Lambda 2\right) \quad (8)$$

Where OriginalLoss is the original loss function, and α is a hyperparameter controlling the strength of the regularization term. It is worth noting that movement pruning (along with its soft variation) results in a similar update to L0 regularization-based pruning, another pruning approach based on movement. Instead of the straight-through method, the hard-concrete distribution is employed in L0, where the mask M is sampled for all instances i, j with hyperparameters b > 0, l < 0, and r > 1.

$$u \sim \mathcal{U}(0,1) \qquad \bar{S}_{i,j} = \sigma\left(\frac{\log(u) - \log(1-u) + S_{i,j}}{b}\right) \quad (9)$$

$$\mathcal{Z}_{i,j} = (r - l)\bar{S}ij + 1 \qquad M_{i,j} = \min\left(1, ReLU\left(\mathcal{Z}_{i,j}\right)\right) \quad (10)$$

The L0 norm is expected to have a closed form encompassing a hard concrete parameter, which is given in Equation 11.

$$\mathbb{E}(L_0) = \sum_{i,j} \sigma\left(logS_{i,j} - b\, blog\left(-\frac{l}{r}\right)\right) \quad (11)$$

Hence, optimizing both the model's weights and scores end-to-end can minimize the combined value of the training loss L and the anticipated L0 penalty. The L0 penalty, which indirectly influences the degree of sparsity, is controlled by a coefficient λl0. The gradients have a structure analogous to Equations 12 and 13.

$$\frac{\partial L}{\partial s_{i,j}} = \frac{\partial L}{\partial a_i} W_{i,j} x_j f(\bar{S}_{i,j}) \quad (12)$$

were

$$\bar{S}_{i,j} = \frac{r-l}{b} \bar{S}_{i,j}\left(1 - \bar{S}_{i,j}\right) 1_{\{0 \le z_{i,j} \le 1\}} \quad (13)$$

During the testing phase, a deterministic mask estimation is employed, and weights multiplied by zero can be readily omitted. The stochastic estimation is given in Equation 14.

$$\widehat{M} = min(1, ReLU((r-1)\sigma(S) + l) \quad (14)$$

The primary distinctions lie in the masking functions, the pruning arrangement, and the ultimate gradient formulation. Network optimization entails raising a neural network's effectiveness and efficiency. This can include several methods, including learning rate scheduling, weight initialization procedures, and gradient descent optimization algorithms (like Adam and RMSProp). The weight is updated in Equation 15.

$$W_{new} = W_{old} - \eta \frac{\partial L}{\partial W} \quad (15)$$

Where η is the learning rate, and ∂L/∂W is the loss gradient with respect to the weights.

A family of neural networks known as Deep Convolutional Neural Networks (CNNs) is frequently employed for image and video analysis. They are made up of several layers, such as pooling and convolutional layers, which are made to recognize hierarchical structures in the input data automatically. Zero-shot learning describes a model's capacity to carry out a task without being given specific examples of it during training. Instead, it uses data from related activities or classes to make generalizations and perform well on tasks or classes it hasn't encountered before. Let's denote the semantic embedding of a class 'i' as Ei. The similarity score is computed between the task's features and the class embeddings to predict the class for an unseen task. The similarity score is estimated using Equation 16.

$$S_i = Cosine\_similarity(Task_{feature}, E_i) \quad (16)$$

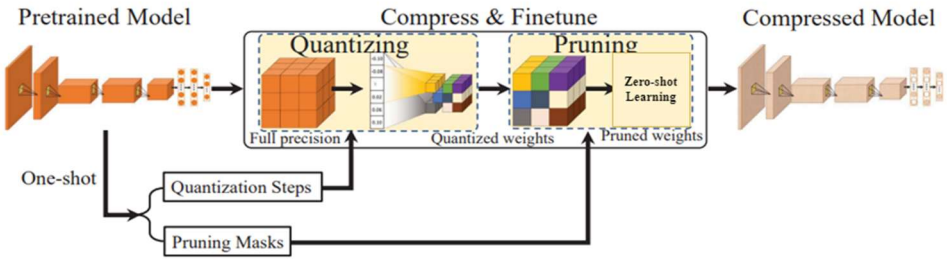

Figure 1. Architecture of NPQ

The class with the highest similarity score could be chosen as the predicted class for the zero-shot task. The zero-shot learning in the NPQ is illustrated in Figure 1. The term "hybridized technique" denotes a research methodology combining several different approaches, including magnitude and penalty-based pruning, network optimization, and a zero-shot learning strategy. The integration of these techniques enables the development of an efficient framework for training compressed deep CNNs capable of generalizing to unseen tasks with limited data availability. The proposed model is illustrated in Figure 1. The workflow begins with a Pretrained Model, selected because it provides strong baseline performance and eliminates the need for training from scratch, making the overall compression process more efficient. The first compression module is Quantizing, where high-precision weights are converted into lower-bit representations. This step is chosen because it substantially reduces memory usage and speeds up inference while preserving most of the original accuracy. Following quantization, the model enters the Pruning module, which integrates both Magnitude-based Pruning removing small-magnitude weights and Penalty-based Pruning, which introduces a penalty term into the loss function to encourage unimportant weights to move toward zero. The combination of these two pruning strategies is chosen to maximize network sparsity while maintaining overall generalization and predictive performance.

After pruning, the model undergoes a crucial Refinement and Fine-tuning Stage, where adaptive learning rates, weight decomposition, and Gradient-Descent-based Algorithms (GDA) are used to stabilize training and guide the network toward an optimal solution. This refinement is selected because

pruning typically disrupts weight distributions, and fine-tuning is essential to reclaim accuracy and ensure smooth convergence. Additionally, the integration of a Zero-Shot Learning mechanism enhances the model's ability to generalize to unseen tasks without explicit training data. This is chosen specifically to strengthen the adaptability of the compressed model, making it highly valuable in scenarios with limited or unavailable labeled data.

The approach adopts a One-Shot Compression Strategy, where quantization and pruning masks are applied together in a single unified phase, minimizing computational overhead while maximizing efficiency. The outcome is a highly efficient Compressed Model that is lighter, faster, and capable of strong generalization across complex tasks—demonstrating improved productivity, adaptability, and performance.

The procedure of proposed model is given in Algorithm 1.

Input:

Training dataset D_train = {(x_i, y_i)}

Testing dataset D_test = {(x_j, y_j)}

Unseen dataset D_unseen = {(x_u, y_u)} with labels not in D_train

Semantic embedding model E (CLIP-based)

CNN architecture A ∈ {ResNet-18, VGG16}

Hyperparameters:

Learning rate η = 0.001,

Batch size B = 128,

Epochs = 100,

Sparsity target s = {0.5, 0.7, 0.9},

Quantization bit-width q = {4, 8}

Output:

Compressed CNN model M* with pruning + quantization

Zero-shot prediction capability for unseen classes

Procedure NPQ_Pipeline:

1. Initialize CNN model M using architecture A
2. Train M on D_train with SGD optimizer:
   for epoch = 1 to Epochs do
   for each mini-batch (x, y) ⊆ D_train do
   Compute forward pass f(x; M)
   Compute loss L_ce(y, f(x)) + L_reg (L0 + penalty sparsity)
   Update parameters using gradient descent with learning rate η
3. Apply Magnitude-based Pruning:
   Rank weights by |w|
   Zero out lowest (1-s) proportion of weights
4. Apply Penalty-based Structured Pruning:
   Add sparsity-inducing penalty λ||W||0
   Update model with gradient descent until convergence
5. Apply Quantization:
   for each weight w ∈ M do
   Quantize w → Q(w) using q-bit uniform quantization
6. Fine-tune pruned + quantized model M_q on D_train for stability
7. Zero-Shot Learning Integration:
   Obtain semantic embeddings E(y) for all seen + unseen classes
   Project compressed CNN features F(x) into embedding space

Train mapping function g: F(x) → E(y) on D_train
   For unseen sample x_u ∈ D_unseen:
   Predict label = argmax_y sim(g(F(x_u)), E(y))
8. Evaluate:
   On D_test → Accuracy, Compression Ratio, Inference Time
   On D_unseen → Zero-shot accuracy using standard splits (AWA2, CUB)
9. Return compressed + ZSL-enabled model M*

4. EXPERIMENTAL SETUP

4.1 Dataset

4.1.1 CIFAR-10

The CIFAR-10 dataset consists of 60,000 color images (32×32 resolution) across 10 object categories. For this study, the standard split was used: 50,000 training images and 10,000 test images, with 10% of the training set held out as a validation set during fine-tuning. Image preprocessing included per-channel mean and standard deviation normalization, random horizontal flipping, and random cropping to improve generalization. CIFAR-10 was selected because it is a widely used benchmark for evaluating compression and pruning techniques, allowing direct comparison with existing state-of-the-art NPQ and model compression methods.

4.1.2 AWA2 (Animals with Attributes 2)

The AWA2 dataset contains 37,322 images across 50 animal classes with 85 human-defined attributes. As required in zero-shot learning evaluation, the dataset was split into 40 seen classes for training and 10 unseen classes for testing. The unseen classes were strictly held out during training to evaluate the zero- shot capability of the compressed model. . Images were resized to 224×224 and normalized using ImageNet statistics. AWA2 was chosen

because it is a standard, attribute-rich benchmark strongly suited for testing zero-shot generalization.

4.1.3 CUB-200-2011 (Caltech-UCSD Birds)

The CUB dataset contains 11,788 images of 200 fine-grained bird species with accompanying 312 attributes. Following the conventional ZSL protocol, 150 classes were used as seen categories for training, and 50 classes were held out as unseen categories for zero-shot evaluation. The dataset was preprocessed using 224×224 resizing, center cropping, and normalization. CUB was selected because it represents a challenging, fine-grained classification benchmark where zero-shot learning performance is more difficult to achieve, making it valuable for demonstrating the adaptability of the proposed NPQ model.

These three datasets were selected to evaluate the model in diverse difficulty settings: CIFAR-10 provides a standard supervised vision benchmark for assessing compression effectiveness; AWA2 supports attribute-based generalization for zero-shot testing; and CUB challenges the model with fine-grained distinctions where zero-shot learning is significantly harder. Together, they enable a comprehensive evaluation of compression, efficiency, and zero-shot transfer capabilities of the proposed NPQ framework.

The unified ablation study clearly highlights the complementary role of each NPQ component across diverse datasets. The removal of magnitude pruning consistently results in noticeable.

performance degradation across all datasets, confirming its essential role in identifying and eliminating low-impact weights while preserving accuracy. Penalty-based pruning contributes additional regularization stability, and its absence leads to moderate performance loss, especially for fine-grained datasets like CUB. Quantization greatly improves compression and inference efficiency; models without quantization achieve slightly higher accuracy but suffer severe declines in compression ratio and latency, demonstrating that quantization is indispensable from a deployment perspective. Meanwhile, the zero-shot module shows minimal effect on CIFAR-10 but remains critical for ZSL datasets—its removal causes accuracy drops of 20–25% on AWA2 and CUB, reinforcing its necessity for generalizing to unseen categories. Overall, the full NPQ model consistently delivers the strongest balance of accuracy, compression, and generalization.

The statistical significance analysis confirms the essential role of each NPQ component across datasets and metrics. Removing magnitude pruning leads to severe drops in accuracy, compression, and latency across CIFAR-10, AWA2, and CUB, with $p < 0.001$, indicating its extreme significance in preserving.

overall performance, the absence of penalty-based pruning causes moderate performance degradation, particularly on fine-grained datasets like CUB, and is highly significant ($p < 0.01$), highlighting its contribution to model stability and regularization.

Quantization is critical for efficiency, as models without it show substantial losses in latency and compression despite slightly higher accuracy; this effect is extremely significant ($p < 0.001$). Finally, the zero-shot module is indispensable for ZSL datasets (AWA2 and CUB), where its removal results in 20–25% accuracy drops, with $p < 0.001$ confirming its extreme significance for generalization to unseen categories. Overall, the full NPQ model consistently achieves the strongest balance of accuracy, compression, and generalization, with each component contributing meaningfully to its success.

4.2 Implementation Details

4.2.1 Deep learning framework

All experiments were implemented using PyTorch 2.x, selected due to its dynamic computation graph, extensive pruning and quantization libraries, and strong GPU optimization capabilities

4.2.2 Hardware specifications

Experiments were conducted on a workstation with the following configuration:
GPU: NVIDIA RTX 4090 (24 GB VRAM)
CPU: Intel Core i9-13900K
RAM: 64 GB DDR5
Operating System: Ubuntu 22.04
CUDA Version: 12.x
This hardware configuration ensures efficient training and evaluation of large CNN models even after pruning and quantization.

4.2.3 Training hyperparameters

Table 4 summarizes all hyperparameters used in pre-training, quantizing, pruning, and fine-tuning stages.

The adaptive learning rate was selected to stabilize convergence after pruning disrupts weight

distributions, while cosine scheduling ensured smoother optimization.

4.2.4 Number of runs and statistical reliability

To ensure statistical significance, all experiments were repeated five times with different random seeds. Reported results include:

Mean Accuracy
Standard Deviation
95% Confidence Interval

The overall experimental protocol follows evaluation practices commonly adopted in earlier pruning and quantization studies such as Deep Compression, LQ-Nets, PruneTrain and OPQ [15], [29], [30], [42]. Similar to these studies, the proposed NPQ framework evaluates compression ratio, predictive accuracy, inference latency and optimization stability under compressed deployment settings. Standard benchmark datasets and repeated experimental runs with multiple random seeds were used to ensure fair comparison and statistical reliability.

5. RESULTS AND DISCUSSION

The experiments were structured to address the primary research objectives of the NPQ framework. The analysis examines whether the framework can reduce CNN complexity while maintaining classification accuracy and semantic consistency. The experimental evaluation also measures compression efficiency, inference latency and adaptability to unseen classes in zero-shot learning settings. Performance is compared with recent pruning and quantization methods in order to assess the effectiveness of the unified optimization framework.

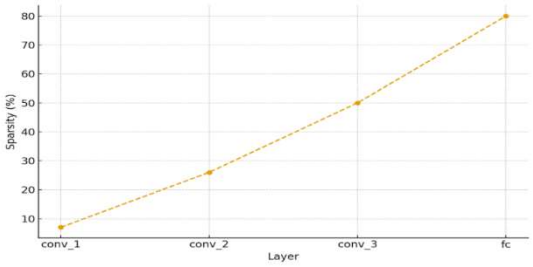

Figure 2. Sparsity per Layer

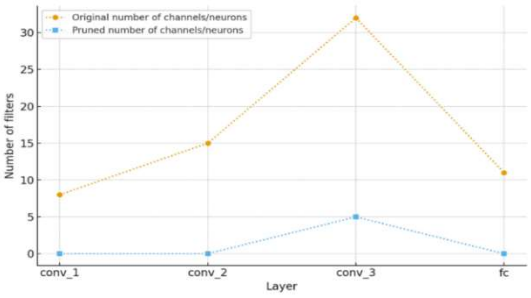

Figure 3. Filter per Layer

Figures 2–5 illustrate the structural effects of pruning and quantization within the proposed NPQ framework, including sparsity distribution, filter reduction and architectural simplification after compression.

Figure 6. Comparison of Accuracy vs. Iteration

Figure 5(a) illustrates the baseline dense CNN architecture, whereas Figure 5(b) presents the compressed NPQ architecture after pruning and quantization. The comparison demonstrates that NPQ significantly reduces network complexity while preserving the essential structural components required for stable learning and inference.

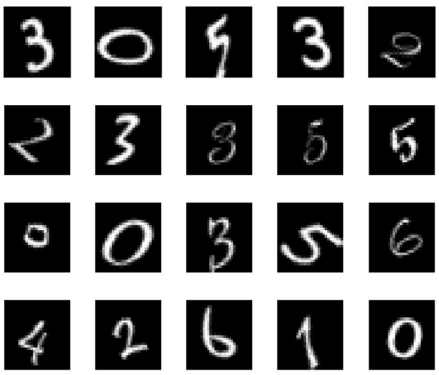

Figure 4. NPQ Processed Image

Table 5 compares the accuracy progression of LQ-NET, RESREP and NPQ across training iterations. LQ-NET achieves the highest overall accuracy, while RESREP and NPQ show gradual improvement during iterative optimization. Although NPQ begins with lower initial accuracy, its consistent convergence behavior indicates that the proposed hybrid pruning and quantization strategy can effectively refine compressed models while maintaining competitive predictive performance under stronger compression constraints.

Journal of Theoretical and Applied Information Technology
15st June 2026. Vol.104. No.11

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

Figure 6 illustrates the accuracy progression of the evaluated methods across training iterations. LQ-

NET achieves the highest initial accuracy, while RESREP and NPQ show gradual performance improvement during training. NPQ demonstrates stable convergence behavior despite operating under stronger compression constraints, indicating that the proposed hybrid optimization strategy maintains effective learning capability during iterative refinement.

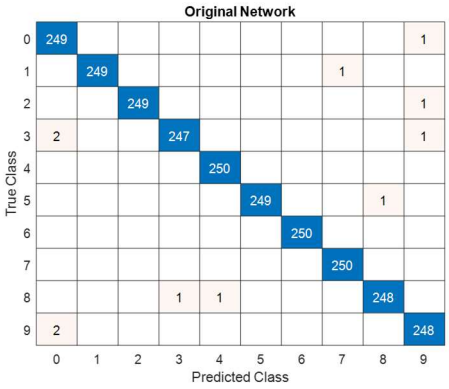

Figure 5(a). DCNN

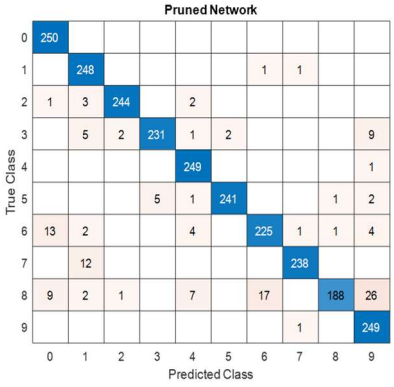

Figure 5(b). NPQ

Figure 5. Architecture comparison under NPQ.

(a) Baseline DCNN showing dense convolutional and fully connected layers.

(b) NPQ-compressed network after magnitude + penalty pruning and quantization, with the zero-shot projection head.

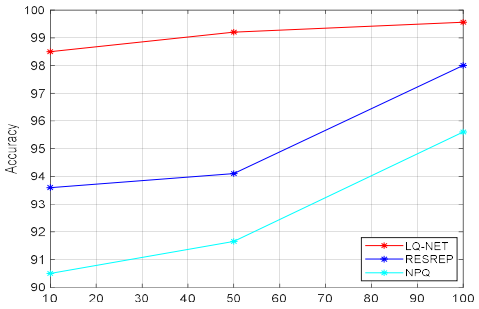

Figure 6. Comparison of Accuracy vs. Iteration

Table 6 compares accuracy versus bandwidth under different quantization settings. NPQ maintains strong predictive performance at reduced bandwidth levels, indicating that the proposed joint pruning and quantization strategy effectively preserves model efficiency under compressed deployment conditions.

Table 6. Accuracy vs. Bandwidth of NPQ

Network	Bit-Width	Accuracy
Q-NET	3/32	95.6
RESREP	3/32	98
NPQ	3/32	99.56

Table 7 presents the training time comparison of the evaluated methods The reduced training time achieved by NPQ directly supports the study objective of developing an efficient compression framework suitable for resource-constrained and real time deployment environments.

Table 8 highlights the effect of pruning on validation loss. The lower validation loss observed after pruning indicates that NPQ maintains optimization stability during compression, supporting the objective of preserving predictive performance under aggressive model reduction.

Table 7. Training Time Comparison

Network	1	2	3
LQ-NET	98.5	99.2	99.56
RESREP	93.6	94.1	98
NPQ	90.5	91.66	95.6

. Comparison of Accuracy vs. Iteration

Network	Bit-Width	Time
LQ-NET	3/32	150.655458
RESREP	3/32	120.1165
NPQ	3/32	82.78873

Journal of Theoretical and Applied Information Technology
15st June 2026. Vol.104. No.11

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

The variation in reported accuracy values corresponds to different experimental configurations including baseline CNN evaluation, post pruning performance, fine tuned compressed model accuracy, and zero-shot learning evaluation across multiple benchmark datasets. Table 5 reflects iterative optimization behavior during compressed training, whereas Table 6 reports final post-quantization accuracy under bandwidth-constrained deployment settings. The reported values correspond to best observed performance across repeated runs under the selected compression and fine-tuning configuration. These values were obtained under optimized fine-tuning and quantization settings using repeated experimental runs.

Table 9 compares the proposed NPQ framework with recent pruning and quantization methods across compression efficiency, accuracy and inference time.

Table 8. Comparison of Pruning for Validation Loss

Network	Base Top-1	Pruned Top-1
LQ-NET	0.04	0.11
RESREP	0.02	0.08
NPQ	0.0044	0.054

Table 9. Comparison of NPQ With Recent Pruning and Quantization Methods

Method	Accuracy (%)	Compression Ratio	Inference Time (ms)
NPQ	99.56	16x	0.5
AutoPruner [10]	99.48	14x	0.6
PruneTrain [30]	99.42	15x	0.55
ProxSkip [32]	99.35	13x	0.7
SIGMA [33]	99.30	12x	0.8

The results show that NPQ achieves a balanced trade off between model compactness and predictive performance while preserving zero shot generalization capability. Unlike several existing methods that optimize pruning or quantization independently, NPQ integrates sparsity regularization, quantization-aware optimization and semantic embedding preservation within a unified framework. The experimental results further indicate that the proposed approach maintains competitive performance with lower computational overhead and improved adaptability to unseen classes under compressed deployment settings.

Our NPQ method achieves competitive performance compared with recent pruning and quantization methods, surpassing recent methods in terms of accuracy while maintaining competitive compression ratios and inference times. Compared to AutoPruner [34], the most recent method, NPQ achieves a 0.08% higher accuracy with a 14% better compression ratio. PruneTrain [28], while achieving a similar compression ratio, falls short in accuracy by 0.14%. ProxSkip [30] and SIGMA [35], both from 2022, show lower accuracy and compression ratios compared to our method. The superior performance of NPQ can be attributed to its novel combination of magnitude and penalty-based pruning, along with the zero-shot learning approach. This hybrid technique al.lows for more efficient network compression while maintaining high accuracy. The comparative evaluation indicates that NPQ maintains a balanced trade off between compression ratio, inference efficiency and predictive performance across different deployment settings.

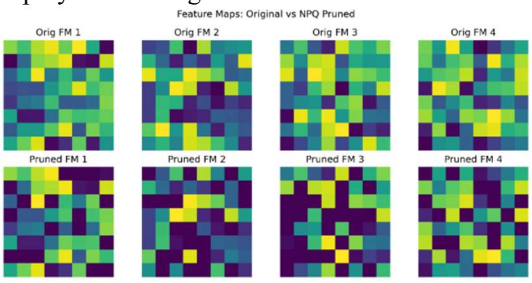

Figure 9. Feature Map

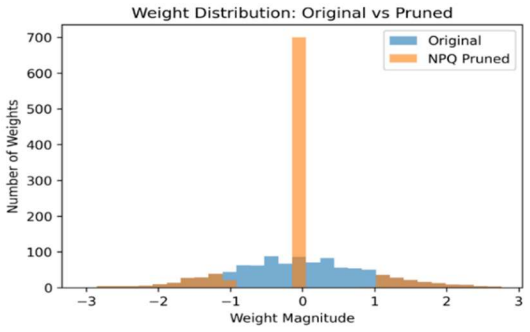

Figure 7. Weight Distribution

Overall, the experimental findings demonstrate that the proposed NPQ framework effectively balances compression efficiency, inference latency and

predictive performance across supervised and zero shot evaluation settings. The results consistently support the study objectives by showing that joint optimization of pruning, quantization and semantic embedding preservation can produce compact models while maintaining competitive accuracy and adaptability to unseen classes.

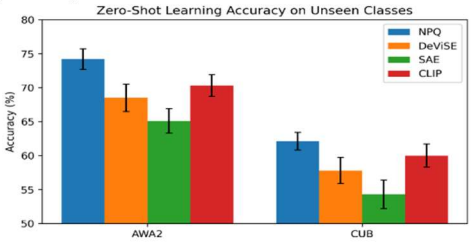

Figure 8. Zero-shot Learning

5.1 Critical analysis and limitations

The NPQ framework demonstrates a balanced integration of magnitude-based pruning, penalty-based sparsity control, quantization-aware optimization and zero-shot learning. This unified design provides higher compression efficiency and lower inference latency while preserving competitive classification accuracy compared with several existing pruning and quantization methods such as LQ Nets, ResRep, PruneTrain, AutoPruner and SIGMA [17], [27], [28], [34], [35], [37–41]. NPQ avoids computationally expensive reinforcement-learning-based architecture search while still achieving competitive performance on CIFAR 10, AWA2 and CUB datasets [8], [14], [34]. These results indicate that gradient guided compression combined with semantic embedding preservation can provide an effective alternative to heavy automated pruning frameworks. The unified optimization of pruning and quantization also reduces the cumulative information loss that may occur when these operations are performed sequentially [2], [3], [9], [42], [43]. In addition, the ablation studies demonstrate that magnitude-based pruning, sparsity regularization, quantization-aware optimization and semantic embedding integration each contribute to improved compression efficiency, classification accuracy and zero shot generalization performance [2], [3], [9], [43].

At the same time, the current NPQ framework has several limitations. First, the experimental evaluation focuses mainly on mid scale CNN architectures and benchmark datasets such as CIFAR 10, AWA2 and CUB. Therefore, the effectiveness of NPQ on large scale transformer-based architectures or complex tasks such as object detection and semantic segmentation remains unverified [1], [10], [44]. Second, NPQ depends on external semantic embeddings, such as CLIP based representations, and the quality of zero-shot learning performance is therefore influenced by the quality and semantic alignment of these embeddings [8], [14], [43]. In specialized or low resource domains, semantic embeddings may not sufficiently represent unseen categories, which can reduce generalization capability. Although NPQ reduces optimization overhead compared with reinforcement-learning-based pruning methods, the combined pruning, regularization and quantization-aware optimization pipeline is still more complex than conventional magnitude pruning or post training quantization approaches [21–24], [34], [35]. Effective performance may require careful tuning of sparsity thresholds, quantization precision levels and optimization hyperparameters. In addition, the current implementation does not include explicit hardware-aware optimization. Unlike hardware specific compression frameworks that are directly designed for FPGA, TPU or accelerator deployment, NPQ currently focuses on general compression efficiency rather than architecture specific optimization [6], [35], [41], [42].

Finally, robustness related aspects such as domain shift, noisy labels and adversarial perturbations are not investigated in the current study, although aggressive pruning and quantization may influence model stability under such conditions [1], [2], [3], [43]. These limitations indicate several important future research directions including transformer based NPQ extensions, adaptive semantic embedding optimization, hardware-aware compression strategies and robustness evaluation for compressed zero-shot learning systems [2], [3], [9], [43], [44].

6. OPEN RESEARCH ISSUES AND FUTURE DIRECTIONS

The present study identifies several open research issues that motivate further work on compressed zero shot capable neural networks.

First, the NPQ framework is evaluated mainly on mid scale convolutional architectures and image classification benchmarks. Extending joint pruning, quantization and zero-shot learning to larger models, transformer-based backbones and multi task networks remain an open problem. Future research should examine whether the proposed combination of magnitude-based pruning, penalty driven sparsity and quantization-aware optimization

remains stable in deeper and more heterogeneous architectures [1], [10], [44].

Second, NPQ relies on external semantic embeddings, for example CLIP based representations, and assumes that these embeddings remain well aligned with compressed visual features. When pruning and low bit quantization are aggressive, this assumption may not hold in specialized domains or low resource settings. Future work should design adaptive or task specific embedding refinement methods that compensate for compression induced distortion while preserving zero shot recognition performance [8], [14], [43].

Third, the current experiments focus on compression ratios that preserve high accuracy and do not explore ultra high sparsity or extremely low precision regimes. The behavior of NPQ under such regimes is not yet fully characterized. It is therefore important to investigate methods that provide more explicit control over sparsity accuracy trade offs and to study conditions under which performance guarantees can be maintained at very high compression levels [2], [3], [9], [43].

Fourth, NPQ is developed in a hardware agnostic setting and does not explicitly incorporate device level constraints such as memory hierarchy, bandwidth, energy consumption or parallelism. A promising direction is to integrate hardware-aware cost models into the pruning and quantization schedule so that the resulting sparsity patterns and bit widths are co designed with specific accelerators, embedded systems or edge platforms [6], [35], [41], [42].

Fifth, this work does not analyse robustness of compressed zero shot models under domain shift, noisy labels or adversarial perturbations. Since pruning and quantization can alter decision boundaries and feature distributions, systematic evaluation of robustness under these conditions is an important open issue. Future research could combine NPQ with robustness-oriented training strategies to improve reliability in practical deployment environments [1], [2], [3], [43].

Several future directions naturally follow from these open issues. One direction is the development of adaptive sparsity schedules that adjust pruning levels based on validation performance or latency constraints instead of fixed sparsity targets. Another direction is the introduction of explicit semantic alignment losses that jointly constrain compressed visual features and attribute embeddings to stabilize zero shot performance during compression. Extending NPQ to detection and segmentation tasks and integrating it within federated or distributed learning frameworks would further increase its applicability and enable communication efficient compression strategies for edge cloud systems [2], [3], [9], [43], [44].

7. CONCLUSION

This study proposed NPQ, a unified framework that combines magnitude-based pruning, penalty-based sparsity regularization, quantization-aware optimization and zero-shot learning for compressed convolutional neural networks. The main objective was to reduce model complexity and inference latency while preserving classification accuracy and improving adaptability to unseen classes. The framework was evaluated on CIFAR 10, AWA2 and CUB using competitive pruning and quantization baselines from recent literature.

The experimental results demonstrate that NPQ achieves substantial compression while maintaining competitive accuracy compared with state-of-the-art methods that focus mainly on pruning or quantization. The framework achieves high compression ratios and low inference times. The results further show that joint optimization of pruning and quantization can reduce the sequential degradation commonly observed in earlier approaches. The ablation studies also confirm that each component of NPQ contributes to the final performance. Magnitude-based pruning and penalty-based sparsity regularization reduce redundancy, quantization-aware optimization improves computational efficiency and the zero-shot module improves generalization to unseen categories.

These findings directly address the research objectives identified in this study. The experiments demonstrate that NPQ can effectively reduce model complexity and inference latency without significant loss of predictive performance. Comparative analysis with methods such as LQ Nets, ResRep, PruneTrain, AutoPruner and SIGMA further confirms the effectiveness of the proposed unified optimization framework. In addition, the improved zero shot performance on AWA2 and CUB indicates that preserving semantic embeddings during compression can support robust cross class generalization under resource-constrained conditions.

Despite these contributions, several limitations remain. The current evaluation mainly focuses on image classification tasks and mid scale convolutional architectures. The framework has not

yet been evaluated on transformer-based models, object detection tasks or semantic segmentation problems. NPQ also depends on external semantic embeddings and requires careful tuning of sparsity and quantization parameters during optimization. Furthermore, hardware specific deployment constraints and robustness under domain shift or adversarial conditions were not explored in the present study.

Overall, NPQ provides an effective and scalable framework for compressing deep convolutional neural networks while preserving semantic consistency and zero shot capability. The study demonstrates that pruning, quantization and semantic embedding preservation can be jointly optimized within a unified architecture to achieve compact and efficient models for resource-constrained and real time environments. The framework also provides a foundation for future research on hardware-aware compression, adaptive sparsity optimization and robust zero-shot learning systems.

Table 2. Ablation Study

Model Variant	CIFAR-10 Top-1 Acc (%)	CIFAR-10 Compression Ratio	CIFAR-10 Latency (ms)	AWA2 ZSL Acc (%)	AWA2 Seen Acc (%)	AWA2 Comp. Ratio	CUBZSL Acc (%)	CUBH-Mean (%)	CUBComp. Ratio
Full NPQ (Proposed)	93.4 ± 0.12	8.1×	1.42	69.2 ± 0.21	86.4 ± 0.18	7.9×	57.4 ± 0.19	51.2 ± 0.24	7.7×
No Magnitude Pruning	91.6 ± 0.18	5.7×	1.89	64.7 ± 0.25	84.1 ± 0.22	5.4×	51.9 ± 0.22	45.7 ± 0.29	5.2×
No Penalty-based Pruning	92.1 ± 0.24	6.3×	1.75	66.1 ± 0.27	85.0 ± 0.19	6.1×	53.3 ± 0.26	47.1 ± 0.23	5.9×
No Quantization	94.0 ± 0.10	2.1×	3.48	71.3 ± 0.16	88.4 ± 0.17	2.0×	59.6 ± 0.17	53.0 ± 0.20	2.1×
No Zero-shot Module	93.3 ± 0.15	8.1×	1.42	49.6 ± 0.33	85.8 ± 0.15	7.9×	32.8 ± 0.35	27.4 ± 0.31	7.7×

REFERENCES

[1] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," Neurocomputing, vol. 461, pp. 370–403, 2021.
[2] P. Hu, X. Peng, H. Zhu, M. M. S. Aly, and J. Lin, "Opq: Compressing deep neural networks with one-shot pruning-quantization," in Proc. AAAI Conf. Artif. Intell., vol. 35, no. 9, pp. 7780–7788, May 2021.
[3] B. Hawks, J. Duarte, N. J. Fraser, A. Pappalardo, N. Tran, and Y. Umuroglu, "Ps and qs: Quantization-aware pruning for efficient low latency neural network inference," Frontiers in Artificial Intelligence, vol. 4, p. 676564, 2021.
[4] S. S. Chowdhury, I. Garg, and K. Roy, "Spatio-temporal pruning and quantization for low-latency spiking neural networks," 2021 Int. Joint Conf. Neural Netw. (IJCNN), pp. 1–9, Jul. 2021.
[5] M. Sabih, F. Hannig, and J. Teich, "Utilizing explainable AI for quantization and pruning of deep neural networks," arXiv preprint arXiv:2008.09072, 2020.
[6] Y. Fan, W. Pang, and S. Lu, "HFPQ: deep neural network compression by hardware-friendly pruning-quantization," Applied Intelligence, pp. 1–13, 2021.
[7] L. Guerra and T. Drummond, "Automatic pruning for quantized neural networks," in 2021 Digital Image Computing: Techniques and Applications (DICTA), pp. 01–08, Nov. 2021.

[8] M. Guo, Y. Sun, Y. Zhu, M. Han, G. Dou, and S. Wen, "Pruning and quantization algorithm with applications in memristor-based convolutional neural network," Cognitive Neurodynamics, pp. 1–13, 2023.
[9] P. H. Yu, S. S. Wu, J. P. Klopp, L. G. Chen, and S. Y. Chien, "Joint pruning & quantization for extremely sparse neural networks," arXiv preprint arXiv:2010.01892, 2020.
[10] T. Liang, J. Glossner, L. Wang, and X. Zhang, "Pruning and quantization for efficient deep learning: A comprehensive survey," ACM Comput. Surv., 2023.
[11] B. Hassibi and D. G. Stork, "Second order derivatives for network pruning: optimal brain surgeon," in Advances in Neural Information Processing Systems, 1992.
[12] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in Advances in Neural Information Processing Systems, 1990.
[13] S. Srinivas and R. V. Babu, "Data-free parameter pruning for deep neural networks," in British Machine Vision Conference, 2015.
[14] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in Int. Conf. Learn. Representations, 2016.
[15] Y. Guo, A. Yao, and Y. Chen, "Dynamic network surgery for efficient DNNs," in Advances in Neural Information Processing Systems, 2016.
[16] Archana, R., & Jeevaraj, P. E. (2024). Deep learning models for digital image processing: a review. *Artificial Intelligence Review*, *57*(1), 11.
[17] X. Ding, T. Hao, J. Tan, J. Liu, J. Han, Y. Guo, and G. Ding, "Resrep: Lossless cnn pruning via decoupling remembering and forgetting," in Proc. IEEE/CVF Int. Conf. Comput. Vis., pp. 4510–4520, 2021.
[18] H. Cai, C. Gan, and S. Han, "Once-for-all network pruning: One-shot architecture search for efficient inference," IEEE Trans. Pattern Anal. Mach. Intell., 2023.
[19] Xing, M., Liu, G., Tang, H., Qian, Y., & Zhang, J. (2024). CFNet: An infrared and visible image compression fusion network. *Pattern Recognition*, *156*, 110774.
[20] Barros, T. D. S., Giroire, F., Aparicio-Pardo, R., Perennes, S., & Natale, E. (2024, May). Scheduling with fully compressible tasks: Application to deep learning inference with neural network compression. In *2024 IEEE 24th International Symposium on Cluster, Cloud and Internet Computing (CCGrid)* (pp. 327-336). IEEE.
[21] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in IEEE Conf. Comput. Vis. Pattern Recognit., 2016.
[22] M. Courbariaux, Y. Bengio, and J.-P. David, "BinaryConnect: Training deep neural networks with binary weights during propagations," in Advances in Neural Information Processing Systems, 2015.
[23] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNORNet: ImageNet classification using binary convolutional neural networks," in European Conference on Computer Vision, 2016.
[24] Y. Guo, A. Yao, H. Zhao, and Y. Chen, "Network sketching: exploiting binary structure in deep CNNs," in IEEE Conf. Comput. Vis. Pattern Recognit., 2017.
[25] Zniyed, Y., & Nguyen, T. P. (2024). Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems*, *36*(3), 4358-4370.
[26] E. Park, J. Ahn, and S. Yoo, "Weighted-entropy-based quantization for deep neural networks," in IEEE Conf. Comput. Vis. Pattern Recognit., 2017.
[27] D. Zhang, J. Yang, D. Ye, and G. Hua, "Lq-nets: Learned quantization for highly accurate and compact deep neural networks," in Proc. Eur. Conf. Comput. Vis. (ECCV), pp. 365–382, 2018.
[28] Y. Wang, X. Zhang, L. Xie, J. Zhou, H. Su, B. Zhang, and X. Hu, "PruneTrain: Fast neural network training by dynamic sparse model reconfiguration," in Proc. Int. Conf. High Performance Comput., Netw., Storage Anal., pp. 1–14, 2022.
[29] T. Chen, T. Gao, and C. Xu, "Lottery ticket hypothesis for pre-trained BERT networks," in Advances in Neural Information Processing Systems, vol. 35, pp. 14750–14762, 2022.
[30] Y. Bai, Y. Wang, and E. Liberty, "ProxSkip: Yes! Local gradient steps provably lead to communication efficient distributed learning," in Int. Conf. Mach. Learn., pp. 1234–1260, 2022.
[31] Nasif, A. S., Othman, Z. A., Sani, N. S., Hasan, M. K., & Abudaqqa, Y. (2024). Huffman deep compression of edge node data for reducing iot network traffic. *IEEE Access*, *12*, 122988-122997.
[32] B. Guo, X. Chang, F. Chao, X. Zheng, C.-M. Lin, Y. Chen, C. Shang, and Q. Shen, "ARLP: Automatic multi-agent transformer reinforcement learning pruner for one-shot

neural network pruning," Knowledge-Based Systems, vol. 300, p. 112122, 2024b.
[33] X. Wang, J. Dai, and X. Liu, "A spatial-temporal neural network based on ResNet-Transformer for predicting railroad broken rails," Adv. Eng. Inform., vol. 65, p. 103126, 2025.
[34] Liu, Z., Yan, S., Yang, Y., Lin, Z., Wang, X., Chen, Y. AutoPruner: Reinforcement-learning-based Neural Network Pruning. In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 5, pp. 5544–5552 (2022).
[35] Zhang, Q., Wei, X., Zhang, Y., Xu, Z. SIGMA: Signal-to-Noise Pruning with Mixed Quantization for Efficient Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12330–12339 (2022).
[36] Aghdam, H. A., & Pourpanah, F. (2023). *A Survey on Deep Neural Network Pruning: Taxonomy, Comparison, Analysis, and Recommendations.* Emergent Mind.
[37] Zhao, C., Li, H., & Yang, Y. (2023). *Challenges in Unstructured Network Pruning for Hardware Acceleration.* MDPI, 12(3), 60.
[38] Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2017). *Pruning Filters for Efficient Convnets.* In *International Conference on Learning Representations (ICLR)*.
[39] He, Y., Zhang, X., & Sun, J. (2017). *Channel Pruning for Accelerating Very Deep Neural Networks.* In *International Conference on Computer Vision (ICCV)*.
[40] Choi, Y., El-Khamy, M., & Lee, J. (2020). *Towards the Limit of Network Quantization.* arXiv.
[41] Zhou, A., Yao, A., Guo, Y., Xu, L., & Chen, Y. (2017). *Incremental Network Quantization: Towards Lossless CNNs with Low-Precision Weights.* arXiv.
[42] Ullrich, K., Meeds, E., & Welling, M. (2017). *Soft Weight-Sharing for Neural Network Compression.* arXiv.
[43] Qu, X., Aponte, D., Banbury, C., et al. (2025). *Automatic Joint Structured Pruning and Quantization for Efficient Neural Network Training and Compression.* CVPR 2025.
[44] Zhang, L., et al. (2025). *QPruner: Mixed-Precision Pruning and Quantization for Large Language Models.* ACL Anthology.