LEAF: An Adaptation Framework against Noisy Data on Edge through Ultra Low-Cost Training

Zihan Xia
University of California San Diego
La Jolla, USA
z5xia@ucsd.edu

Jinwook Kim
Sk Hynix inc.
Icheon, Korea
jinwook4.kim@sk.com

Mingu Kang
University of California San Diego
La Jolla, USA
m7kang@ucsd.edu

ABSTRACT

The deployment of machine learning in real-life computer vision applications is often subject to the degraded input data by various noise sources and time-varying conditions. Continual re-training is one potential solution to adapt to the real-time noise sources and achieve the target performance. However, this approach requires massive computing complexity for the training, particularly demanding on resource-constrained edge devices. To address this challenge, for the first time, we present a hardware-efficient framework named LEAF, tailored for the adaptation to degraded images (ADI) scenario. Leveraging both qualitative and quantitative profiling of CNN dynamics when operating on degraded images, we propose two techniques: 1) selective experience replay (SER)-based unimportant image skipping to reduce both the forward pass (FwP) and backward pass (BwP) costs, and 2) pseudo noise dithering (PND)-assisted extremely low precision quantization (3/4-bit) for error gradients to enable nearly full integer computing. Extensive experiments on CIFAR10 and Tiny ImageNet datasets employing VGGNet and ResNet, along with five different types of image degradations with varying strengths demonstrate virtually no accuracy degradation ($< 0.5\%$) at 3/4-bits with $3.64 \sim 8.40\times$ reduced forward and backward processing.

KEYWORDS

data degradation, on-device learning, low-cost training

ACM Reference Format:
Zihan Xia, Jinwook Kim, and Mingu Kang. 2024. LEAF: An Adaptation Framework against Noisy Data on Edge through Ultra Low-Cost Training. In *61st ACM/IEEE Design Automation Conference (DAC '24), June 23–27, 2024, San Francisco, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3649329.3656256

1 INTRODUCTION

Real-world computer vision applications such as sensors, surveillance systems and autonomous driving, are often affected by various types of noise sources in input data. For example, *Motion Blur* can result from camera shake or movement of the target object, while adversarial environments and conditions, such as poor illumination or hazy weather, introduce additional noise. *White Gaussian* noise is also frequently generated due to imperfections in image sensor circuitry. For this reason, machine learning (ML) models, which are trained on well-crafted "clean" datasets, often experience significant performance degradation in real-life deployments.

One possible solution is to leverage existing degradation removal techniques, which have been a long-standing research topic in the field of computer vision, to eliminate or reduce the distortion and restore original images. There has been extensive research to remove common distortions including white Gaussian noise [11], salt-and-pepper noise [4], motion blur [21], non-uniform illumination [12] and so on. However, this approach is computationally expensive and furthermore aims to enhance the image for human vision rather than optimizing it for machine vision to achieve higher classification performance. Pei's [15] work highlights this aspect based on the observation that widely used dehazing and motion-blur removal algorithms have minimal impact on improving CNN-based classification performance.

An alternative and more effective solution to boost the performance is re-training the ML models, which have been pretrained on clean datasets, directly on degraded data. For instance, Zhou *et al.*'s experiments show that re-training can recover most of the accuracy loss from various degradation types and levels effectively [30]. However, it is not feasible to predict the type and strength of degradations of real-time input in a realistic scenario. Furthermore, transmitting the acquired data to a centralized cloud incurs long latency and privacy concerns. For this reason, enabling *in situ* adaptation to unpredictable distortions is strongly mandated in such a scenario, often operated in computing-/memory-constrained edge devices. However, current on-device training frameworks [3, 13, 17, 18, 24] have not yet observed and analyzed the inherent features hidden behind the scenario of ADI, which has a significantly different nature compared to conventional training from scratch. As a result, they cannot fully leverage its great potential for computing and memory savings.

This challenge has motivated us to design Light Edge Adaptation Framework (LEAF) to unlock its great potential toward highly efficient *in situ* neural network adaptation to degraded input data. The key contributions of this paper are as follows.

- We profile and analyze the characteristics of neural networks when operating on data with various types of degradation. Based on this analysis, for the first time, we propose a dedicated framework, dubbed LEAF, for the hardware-efficient adaptation.
- We propose two practical techniques: 1) SER-based redundant image skipping technique to save both FwP and BwP training costs; 2) PND-assisted extremely low-precision (3/4 bits) quantization for error gradients to enable nearly full-integer training.
- Extensive experiments on two datasets and network models (CIFAR10/ResNet-20 [5] and Tiny ImageNet/VGG16 [19]), with

ACM ISBN 979-8-4007-0601-1/24/06
https://doi.org/10.1145/3649329.3656256

two levels and five types of degradations (Gaussian Blur, Motion Blur, Radial Light, Salt Pepper, White Gaussian) demonstrate that LEAF achieves $< 0.5\%$ accuracy loss on average while enabling 3/4-bit error gradient quantization (with weights and activations being quantized) and using only $< 18\%$ images for learning across all testbenches.

2 RELATED WORK

2.1 Quantized Training of Neural Networks

The quantization during a training process poses greater challenges than during an inference, primarily because gradients exhibit wide, sharp, long-tailed, and evolving characteristics [23, 31]. Several studies have attempted to minimize the quantization error of gradients either through clipping-value searching methods [23, 31] or assuming a prior distribution of gradients [28]. These approaches have successfully achieved 8-bit integer quantization for gradients, although this direction has resulted in significantly increased algorithmic complexity. Alternatively, multiple different quantization methods have been applied based on the types of data [26, 29]. Notably, Sun et al. [22] introduced a unique radix-4 floating-point (FP) format, enabling a 4-bit training.

However, aforementioned studies have not explored quantization in the context of ADI use-cases, which exhibits a unique potential for ultra low-bit integer quantization for gradients.

2.2 On-Device Learning

Training neural networks on edge devices incurs significant challenges due to their restricted computational power and memory capacities. Han et al. [13] addressed this by integrating quantization, sparse tensor updating, and computation graph scheduling, to showcase a training on microcontrollers. E2Train [24] implements energy-saving techniques by selectively skipping layers, minibatches, and bit-level gradient computations. Additionally, recent works suggest skipping the BwP for data deemed non-essential for training [3, 17, 18]. For instance, the prediction confidence (PC) metric (i.e., the dot product of the one-hot label and the `Softmax` output) can be used to determine whether the image with a high PC could skip the BwP.

Unlike the above approaches, we explore further dramatic efficiency improvements in training, leveraging the distinctive characteristics of ADI tasks.

3 LIGHT ADAPTION FRAMEWORK ON EDGE

3.1 Overview of the Proposed Framework

Figure 1 visualizes the various types and degrees of image degradations studied in this paper (L: light, H: heavy). To efficiently adapt neural networks to real-life noisy data, we propose the LEAF, as summarized in Figure 2.

Network level: Intuitively, in the scenario of ADI, not all images contribute equally to learning. Our profiling results suggest that PC is an effective metric for measuring image importance. Thus, only images with a low PC are required for the intensive BwP. We note that images deemed important in the current training epoch are likely to play an important role in the learning in subsequent epochs. As such, we only need to select from images marked as

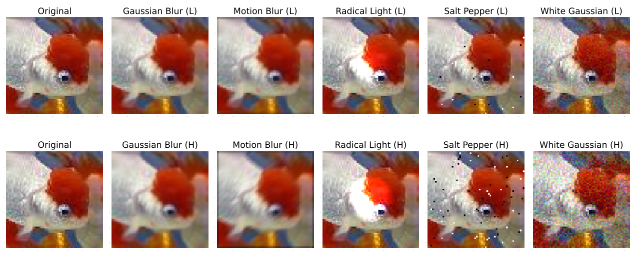

Figure 1: Visualization of different types and levels of degraded images.

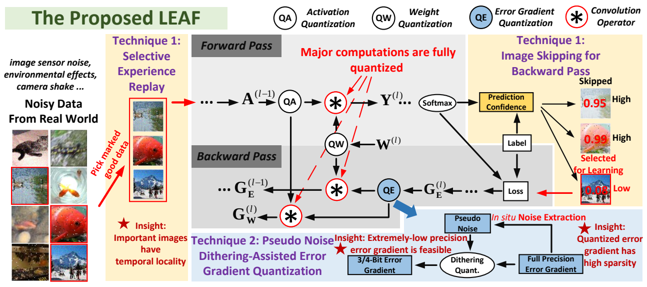
The Proposed LEAF
QA Activation Quantization
QW Weight Quantization
QE Error Gradient Quantization
Convolution Operator
image sensor noise, environmental effects, camera shake ...
Noisy Data From Real World
Pick marked good data
Technique 1: Selective Experience Replay
Insight: Important images have temporal locality
Forward Pass
Major computations are fully quantized
$A^{(l-1)}$
$Y^{(l)}$
$W^{(l)}$
Softmax
Backward Pass
$G_E^{(l-1)}$
$G_W^{(l)}$
$G_E^{(l)}$
Label
Loss
Technique 1: Image Skipping for Backward Pass
Prediction Confidence
Skipped
0.95 High
0.99 High
Selected for Learning
0.08 Low
Technique 2: Pseudo Noise Dithering-Assisted Error Gradient Quantization
Insight: Extremely-low precision error gradient is feasible
Pseudo Noise
In situ Noise Extraction
3/4-Bit Error Gradient
Dithering Quant.
Full Precision Error Gradient
Insight: Quantized error gradient has high sparsity

Figure 2: The proposed LEAF framework.

important in the preceding epoch for the current epoch's training. In this way, both FwP and BwP can be saved for processing many redundant data.

Data level: In Figure 2, the error gradient $\mathbf{G_E}$ is involved in computing both the weight gradient $\mathbf{G_W}$ and the error gradient propagation. Thus, quantizing $\mathbf{G_E}$ facilitates full-integer computing for the majority of training computations. Our analysis suggests that, in the ADI scenario, 3/4-bit quantization is sufficient to maintain on-par accuracy by employing the dithering technique [16]. Unlike traditional dithering, which requires costly random number generators (RNGs), we propose a method to extract noise directly from $\mathbf{G_E}$ itself, which has an entropy approaching that of the ideal uniform distribution. We also observe that $\mathbf{G_E}$ is highly sparse ($> 80\%$) during the training, which will further reduce the computing cost.

3.2 Network-Level: Low-Cost Image Skipping with Selective Experience Replay

In this section, we introduce the proposed Cost-Reduced Image Skipping with Selective Prioritized Experience Replay (ISER). We first conduct an empirical investigation of the correlation between PC and the L2-norm of weight gradients. This is to verify the feasibility of PC-based BwP skipping in the context of ADI. Subsequently, we introduce a general cost metric for evaluating various image skipping frameworks. Finally, by leveraging the observed temporal locality of critical images, we propose the SER to enable the skipping of FwP for computing the PC, thereby minimizing the overall cost.

Figure 3 visualizes the relation between PC and $||\mathbf{G_W}||_2$. The high Pearson's correlation coefficient r_{xy} indicates that PC is an effective estimator of $||\mathbf{G_W}||_2$. A low $||\mathbf{G_W}||_2$ implies minimal changes in network weights using this gradient, supporting the feasibility of skipping images with high PC. Additionally, the training process reveals a decreasing proportion of images contributing to learning as epochs progress, highlighting significant potential for energy savings.

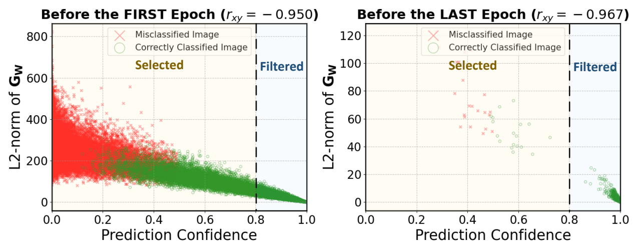

Figure 3: Distribution of the L2-norm of weight gradients with respect to all images in the Tiny ImageNet training set. Assuming a threshold of 0.8, all images in the "filtered zone" will be skipped for BwP.

Table 1: Average temporal locality in the training for ADI

	Gaussian Blur	Motion Blur	Radial Light	Salt Pepper	White Gaussian
L	0.92	0.91	0.91	0.91	0.91
H	0.91	0.92	0.93	0.92	0.92

However, computing PC to judge the images to skip is challenging since it requires FwP for every image in each training epoch to calculate the dot product between the `Softmax` output and one-hot label. To quantify this cost, we introduce a unitless model for evaluating the efficiency of various image skipping frameworks [3, 17, 18, 24].

$$C \triangleq \sum_{i=1}^{E} c_1 \times r_i \times |\mathcal{D}_i| + c_2 \times (1 - r_i) \times |\mathcal{D}_i|, \quad (1)$$

where E is the number of training epochs, r_i the skipping ratio in epoch i, $\mathcal{D}_i$ the candidate image set for epoch i, c_1 the cost of evaluating whether a candidate image should participate in the learning, and c_2 the cost of learning an image. $|\mathcal{D}_i|$ is the cardinality of set $\mathcal{D}_i$. Note that the FwP generally takes 1/3 operations and training time of the entire training loop [8, 22, 25]. For our framework and other PC-based methods [3, 17, 18], we define $c_1 = 1$ and $c_2 = 3$. Thus, cost C will be dominated and bounded by the first term as r_i approaches 1. In E2Train [24], we define $c_1 = 0$ and $c_2 = 3$ since FwP is not required for skipping decisions.

To minimize C while preserving the accuracy, we propose the SER based on the observation that critical images have temporal locality across training epochs. To analyze the temporal locality, we evaluate all images in the training set and collect the set of identified important images, $\mathcal{D}_i^* = \{image | PC(image) < Threshold, image \in training\, set\}$, at the beginning of the i-th epoch. The temporal locality at i-th epoch is then defined as:

$$TL(i) = \frac{|\mathcal{D}_i^* \cap \mathcal{D}_{i-1}^*|}{|\mathcal{D}_i^*|} \in [0, 1]. \quad (2)$$

The average temporal locality during the re-training for ADI is $TL = (\sum_{i=2}^{E} TL(i))/(E-1)$, as detailed in Table 1. We can observe that $> 90\%$ of critical images are from the preceding epoch in all scenarios. In light of this insight, the candidate image set $\mathcal{D}_i$ for a certain training epoch can be replaced by the images used for learning in the previous epoch (i.e., $\mathcal{D}_i \leftarrow \mathcal{D}_{i-1}^*$), instead of the entire training set as done in prior works [3, 17, 18]. Such epochs are termed "replay epochs". In the replay epoch, the PC is calculated for the set $\mathcal{D}_i$ to further narrow down the images with low PC, which will be used for the learning. By doing so, a majority of FwP is skipped by processing the limited set $\mathcal{D}_i$ and the computation of BwP is further minimized for the images with low PC.

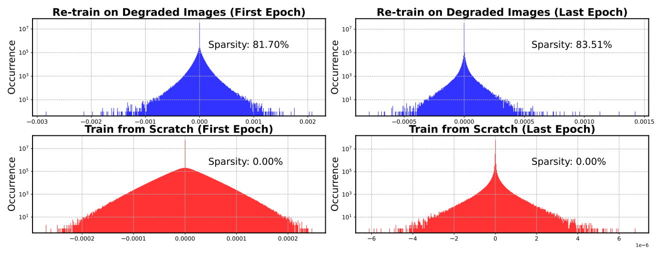

Figure 4: Distribution of error gradients. Data are extracted from the first mini-batch and the second conv. layer of VGG16.

Finally, we outline the procedure of the proposed ISER as follows: 1) Configure the frequency (K) of replay epochs. The training will follow the loop of "1 normal epoch + K replay epochs"; 2) Perform FwP to identify critical images based on PCs. Once the number of collected critical images reaches the minibatch size, this collected set of images will be sent for BwP computation.

3.3 Data-Level: Low-Precision Error Gradient Quantization with Pseudo Noise Dithering

In this section, we address two critical questions regarding ADI: $Q1$: Is low-precision quantization of $\mathbf{G_E}$ feasible?, $Q2$: If feasible, how can the cost of quantization be reduced? The profiling results presented in this section are derived from experiments on Tiny ImageNet/VGG16, subjected to White Gaussian degradation with a standard deviation of 0.09.

Unlike training-from-scratch, the dynamics of $\mathbf{G_E}$ in the ADI scenario remain less investigated. We visualize $\mathbf{G_E}$ in Figure 4 and summarize our observations as follows: 1) $\mathbf{G_E}$ shows a notable concentration near zero (notice the log-scale y-axis), implying that a limited number of neurons are important, and 2) a majority of $\mathbf{G_E}$ are zeros, even without quantization (explained in Section 4.3). As such, the ADI scenario gives a good opportunity for the ultra low-bit quantization.

To enable the extreme-low bit width quantization, we deliberately add noise to $\mathbf{G_E}$, which has been proven effective to make quantized gradients maintain an unbiased expectation compared to the original ones [23, 29, 31]. We propose to use the following dithering symmetric uniform quantization (SUQ) function:

$$Q_{dither}(\mathbf{x}_q) = \Delta \cdot round(\frac{clip(\mathbf{x}, \alpha)}{\Delta} + \eta), \quad (3)$$

where α is the clipping threshold, $clip(\mathbf{x}, \alpha) = \min(\max(\mathbf{x}, -\alpha), \alpha)$, $\Delta = \alpha/(2^{B-1} - 1)$ the quantization step size, B the bitwidth. And η is a noise vector, each element of which follows a uniform distribution $U(-0.5, 0.5)$.

To evaluate the impact of quantized error gradients on the angle change of weight gradients, we investigated the cosine similarity between the weight gradients $\mathbf{G_W}$ computed from the quantized $\mathbf{G_E}$ and full-precision $\mathbf{G_E}$ in Figure 5. The network was updated using full-precision $\mathbf{G_E}$ while we calculate the weight gradients using both the quantized and full-precision $\mathbf{G_E}$ to observe the difference. As depicted, 8-bit/4-bit $\mathbf{G_E}$ almost preserves the direction of $\mathbf{G_W}$. Interestingly, even a 3-bit quantization demonstrates a high

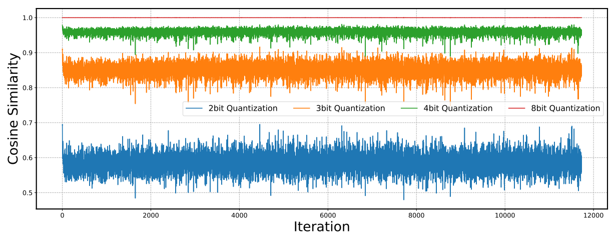

Figure 5: Average cosine similarity across all layers in the ADI task with White Gaussian noise.

Table 2: Entropy of noise obtained from error gradients.

Random Source	Exp.	Mant.	Mant. Last 16 bits	Mant. Last 8 bits
Ideal Entropy	8	23	16	8
Obtained Entropy[1]	5.0004	19.2733	15.8088	7.9978

1. Average entropy over all layers and training epochs.

Table 3: Average ratio of images used for learning

	Gaussian Blur	Motion Blur	Radial Light	Salt Pepper	White Gaussian
L	0.8548%	2.4560%	4.8135%	3.3821%	4.1781%
H	3.9615%	14.5855%	17.9928%	8.0857%	12.0385%

cosine similarity, thereby indicating the feasibility of employing low-bitwidth quantization of $\mathbf{G_E}$ in ADI tasks.

Equation (3) suggests that each error gradient needs a random number. However, on edge devices, the generation of a substantial volume of high-quality random numbers with adequate parallelism is not feasible, as outlined in [20, 27]. To address this, we propose a simple yet effective approach to acquire random numbers with minimal energy cost. Typically, the execution of the quantization operation is performed by high-precision FP computing units or embedded microprocessors, owing to its high sensitivity and minimal computational demands. Based on the insights that randomness is an inherent aspect of neural network training, we introduce the PND method. This method generates a random number for the error gradient by directly using the last K bits of its mantissa of the FP error gradient, thereby effectively bypassing the use of costly RNG. Note that the single-precision floating-point format (i.e., FP32) is extensively supported by existing AI processors as a special function processor for the purpose of format conversion and non-linear activation functions, as indicated in [10, 14], which can be readily exploited for the purpose of PND.

To explain the adequacy of the pseudo-noise in ADI tasks, we analyze the entropy of the obtained noise. Note that zero-valued $\mathbf{G_E}$ elements are excluded from this analysis because quantizing zero invariably yields zero. Ideally, we want to achieve the maximum entropy (i.e., uniform distribution). As depicted in Table 2, while the exponent exhibits limited randomness, the last 8 bits of the mantissa have randomness nearly equivalent to an ideal uniform distribution. For this reason, our proposed PND employs the last eight bits of the mantissa as pseudo noise sources.

Table 4: The configuration of degradation parameters

Degradation	Gaussian Blur (g)		Motion Blur (l)		Radial Light ((r, s))		Salt Pepper (p)		White Gaussian (v)	
Strength	L	H	L	H	L	H	L	H	L	H
Tiny ImageNet	0.5	0.7	2	4	(16, 2)	(16, 4)	0.005	0.02	0.05	0.09
CIFAR10	0.5	0.7	2	3	(8, 2)	(8, 4)	0.005	0.02	0.02	0.05

4 EVALUATION

4.1 Experiments Setup

Datasets and networks. Throughout the experiments, we utilize two testbenches: CIFAR10/ResNet-20 and Tiny ImageNet/VGG16 along with five noise models for the image degradation. The CIFAR-10 [6] dataset has $60,000$ 32×32 colour images in 10 classes. The Tiny ImageNet [7] dataset contains $100,000$ images of 200 classes downsized to 64×64 colored images.

Training strategies. Initially, we trained full-precision models. Then, quantization-aware-training with state-of-the-art quantization methods [1, 9] was performed based on the full-precision model. The computationally intensive convolutional layers and fully-connected layers were quantized. The first and last layers are not quantized as a common practice. The quantization levels were set to 4 bits for CIFAR10 and 8 bits for Tiny ImageNet, both for weights and activations. Then, we simulated the deployment of quantized models in the noisy environment. We fine-tuned the quantized model on degraded images for 15 epochs, using a batch size of 128 and the SGD optimizer with a momentum of 0.9. The initial learning rate was 0.001 and was divided by 10 after 10 epochs. The running mean and variance of batch normalization (BN) layers were frozen, which is a common setting for on-device fine tuning [2]. Note that freezing BN layers had a minimal accuracy impact, with a +0.02% difference for Tiny ImageNet/VGG16 and −0.13% for CIFAR10/ResNet-20. The full-precision accuracy of Tiny ImageNet and CIFAR10 was 61.70% and 91.29%, respectively. The quantized model accuracy was 61.79% and 91.33% for Tiny ImageNet and CIFAR10, respectively.

Baseline formulation. Alongside comparisons with various existing works, we built a baseline without using any proposed cost-saving techniques, thereby representing the maximum achievable accuracy. In detail, the weights and activations of the baseline were quantized. However, the error gradients were kept at full-precision and the entire dataset was used for training.

Synthesis of degraded images. The degraded images were synthesized from the original datasets. We followed the methods introduced in [15] to synthesize Gaussian Blur (kernel size 3, standard deviation g), Salt Pepper (density p), and White Gaussian (zero mean, standard deviation v) noises. Motion Blur was simulated using a linear kernel at a 45-degree angle with length l. Radial Light was created by multiplying images with a circular scaling map (radius r), where the scaling factor decreased linearly from the center (s) to the circumference (1). We generated different strengths of degradations by configuring the parameters as shown in Table 4.

4.2 Evaluating the Image Skipping

To evaluate the proposed ISER's potential in reducing training cost and its impact on accuracy, we benchmark ISER using the complex Tiny ImageNet/VGG16. Figure 6 compares the results with both the

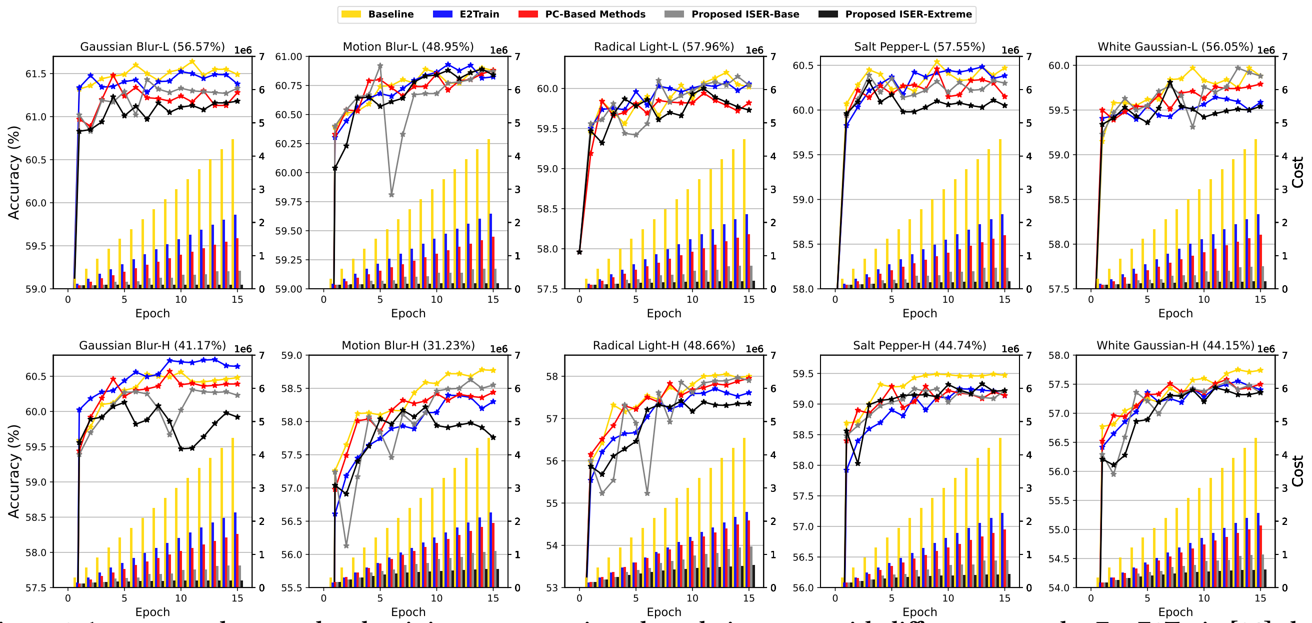

Figure 6: Accuracy and accumulated training cost on various degradation types with different strengths. For E2Train [24], data were reported from the average of three runs. For an apple-to-apple comparison, the threshold for our method and [3, 17, 18] is set to the same constant 0.995 (i.e., skipping when PC>0.995).

Table 5: Accuracy comparison with the baseline and existing works (each cell denotes results of light/heavy degradation)

Degradation	Gaussian Blur	Motion Blur	Radial Light	Salt Pepper	White Gaussian	Avg. Acc. Loss	Gaussian Blur	Motion Blur	Radial Light	Salt Pepper	White Gaussian	Avg. Acc. Loss
Test Case	Tiny ImageNet / VGG16						CIFAR10 / ResNet-20					
No Re-training	56.57/41.17	48.95/31.23	57.96/48.66	57.55/44.74	56.05/44.15	5.23/16.94	88.75/71.78	82.96/58.71	89.28/81.06	89.22/80.40	85.15/60.28	3.09/17.52
Baseline	61.64/60.56	60.90/58.78	60.20/58.05	60.54/59.49	59.97/57.75	0.00/0.00	90.91/90.14	90.13/86.59	90.02/87.02	90.23/89.03	89.54/87.06	0.00/0.00
SUQ (Dithering)	61.60/60.70	61.01/58.79	60.15/58.12	60.53/59.47	59.95/57.67	0.00/-0.02	90.95/89.84	90.07/86.32	90.05/86.67	90.35/88.93	89.65/86.71	-0.05/0.27
SUQ (PND)	61.68/60.69	61.01/58.72	60.20/57.96	60.40/59.59	60.02/57.76	**-0.01/-0.02**	90.97/89.83	89.93/86.15	89.96/86.55	90.27/89.01	89.78/86.77	**-0.02/0.29**
WAGEUBN	61.43/60.11	60.69/57.28	59.63/56.90	60.30/58.82	59.82/56.82	0.28/0.94	90.89/89.52	89.91/84.07	89.79/84.49	90.09/88.07	89.45/84.92	0.14/1.75
DOREFA	Does Not Converge						89.01/85.54	86.97/84.12	88.96/85.24	89.45/88.09	88.99/86.23	1.49/2.13

baseline and the other works. Specifically, [3,17,18] use common PC-based image skipping. E2Train [24] randomly drops mini-batches with a 50% probability. We established two ISER configurations: 1) ISER-Base: one normal epoch is followed by two replay epochs (i.e., K=2); 2) ISER-Extreme: only the first epoch is the normal epoch, followed by replay epochs for the remainder of the training.

In comparison to the baseline, which uses all images for learning, ISER-Base results in a mild accuracy loss of $-0.02 \sim 0.25\%$ while achieving $3.64 \sim 8.40\times$ reductions in the training cost, defined in (1). In contrast, ISER-Extreme offers even greater cost savings ($6.71 \sim 35.59\times$), at the expense of a $0.01 \sim 0.63\%$ accuracy drop. This is because the model learns on a subset of training set, thereby introducing training bias. Compared to E2Train [24], ISER-Base not only achieves $1.84 \sim 4.15\times$ more savings in costs but also maintains comparable accuracy with only 0.027% difference on average. Besides, ISER-Base incurs $1.64 \sim 2.85\times$ lower cost compared to PC-based methods [3, 17, 18] while achieving slightly better accuracy. In summary, while all of prior image skipping methods effectively recover most accuracy losses, our method stands out as the most cost-effective.

Table 3 details the percentage of images utilized from the entire training set in the ISER-Base configuration. On average, only 3.14% of images are crucial for learning in scenarios with light image degradations, whereas heavy distortion requires about 11.33% of images for effective accuracy recovery.

4.3 Evaluating the Error Gradient Quantization

We evaluate the efficacy of 1) the dithered quantization defined by (3) in ADI scenarios, and 2) the proposed PND to minimize the cost of generating random numbers for all benchmarks by measuring the accuracy. We set the clipping threshold α to $\max(|\mathbf{G_E}|)$ for easy hardware implementation, with the maximum taken over each error gradient map. Moreover, the results are compared to two well-known training quantization frameworks: 1) WAGEUBN [26], which introduces a shift-quantization function for the error gradient, and 2) DOREFA [29], which proposes another form of dithered quantization.

Interestingly, our findings suggest that extremely low bitwidth quantization of error gradients is viable. As shown in Table 5, except for the baseline, all evaluated methods employ 3-bit quantized error gradients. Compared to the baseline, our proposed quantization with PND fully compensates for the accuracy loss observed in Tiny ImageNet/VGG16. On CIFAR10/ResNet-20, the average accuracy loss is $\sim 0/0.29\%$ under conditions of light/heavy degradation. This performance significantly outperforms that of WAGEUBN [26] and DOREFA [29]. Notably, DOREFA [29] fails to converge in the Tiny ImageNet dataset evaluation. This issue potentially arises

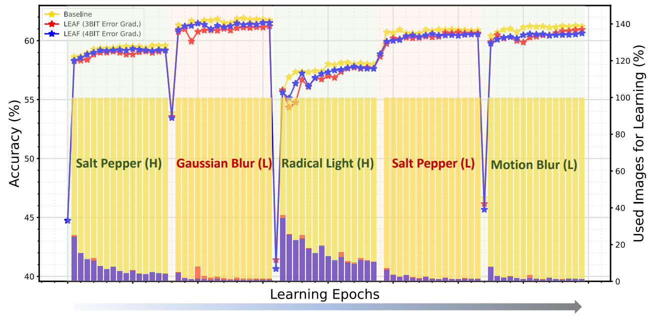

Figure 7: The accuracy trace and percent of used images for learning at every epoch.

from its quantization approach, which fails to preserve zero-valued error gradients as zeros. This inadequacy becomes unacceptable, particularly for highly sparse and low-precision data.

We also examined the sparsity of error gradients. For Tiny ImageNet/VGG16 and CIFAR10/ResNet-20, sparsity levels reach as high as 91.7% and 81.8%, respectively. This high degree of sparsity can be attributed to the combined effects of freezing BN, employing extremely-low bit width quantization, and the inherent nature of ADI. Note the freezing BN reduces the back propagation of error gradients to simple linear transforms, thereby preserving the zeros introduced by the Rectified Linear Unit.

4.4 Evaluating the LEAF Framework

We evaluate the proposed LEAF framework, integrating the ISER and extremely-low bitwidth quantization using PND. The challenging Tiny ImageNet/VGG16 was selected for this evaluation.

We focus on the adaptation capability of LEAF under time-varying degradations. We randomly selected five different degradation configurations and adapted the neural network to these sequentially incoming degradations. Figure 7 illustrates that LEAF effectively tracks the accuracy performance of the baseline, while utilizing significantly less data for learning (as indicated by the bars) and maintaining only 3/4-bit precision for error gradients. Notably, with 4-bit quantization, the accuracy loss is successfully contained to be less than 0.5%.

Furthermore, we observe that models trained on one type of degradation generally struggle to perform well when exposed to new distortions (except for Salt Pepper (L)). This observation highlights the significance of real-time, on-device adaptation in enabling true edge intelligence, emphasizing the necessity and relevance of the LEAF framework in such scenarios.

5 CONCLUSION

In this work, we presented the LEAF framework, which adapts to incoming noisy data through ultra-low-cost training. We proposed SER-based image skipping to identify and learn only critical images, achieving up to 8.40× cost savings compared to the baseline within 0.25% accuracy loss. We also introduced PND-assisted error gradient quantization, enabling quantization as low as 3 bits and observed sparsity greater than 80% without losing accuracy on Tiny ImageNet dataset. With the synergy of network-level and data-level optimizations, the proposed LEAF paves the way for future energy-efficient acceleration for realizing adaptation on the edge.

ACKNOWLEDGMENTS

This work was supported in part by PRISM, center in JUMP 2.0, an SRC program sponsored by DARPA.

REFERENCES

[1] Jungwook Choi et al. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018.
[2] Seungkyu Choi et al. An optimized design technique of low-bit neural network training for personalization on iot devices. In *DAC*, pages 1–6, 2019.
[3] Seungkyu Choi et al. Accelerating on-device dnn training workloads via runtime convergence monitor. *IEEE TCAD*, 42(5):1574–1587, 2023.
[4] Bo Fu et al. A salt and pepper noise image denoising method based on the generative classification. *Multimedia tools and applications*, pages 12043–12053, 2019.
[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
[6] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.
[7] Ya Le et al. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.
[8] Jinsu Lee et al. An overview of energy-efficient hardware accelerators for on-device deep-neural-network training. *IEEE OJ-SSCS*, pages 115–128, 2021.
[9] Junghyup Lee et al. Network quantization with element-wise gradient scaling. In *CVPR*, pages 6448–6457, 2021.
[10] Sae Kyu Lee et al. A 7-nm four-core mixed-precision ai chip with 26.2-tflops hybrid-fp8 training, 104.9-tops int4 inference, and workload-aware throttling. *IEEE JSSC*, pages 182–197, 2022.
[11] Stamatios Lefkimmiatis. Universal denoising networks : A novel cnn architecture for image denoising. In *CVPR*, pages 3204–3213, 2018.
[12] Mading Li et al. Restoration of unevenly illuminated images. In *ICIP*, pages 1118–1122, 2018.
[13] Ji Lin et al. On-device training under 256kb memory. In *Advances in Neural Information Processing Systems*, pages 22941–22954. Curran Associates, Inc., 2022.
[14] Jinwook Oh et al. A 3.0 tflops 0.62v scalable processor core for high compute utilization ai training and inference. In *2020 IEEE Symposium on VLSI Circuits*, pages 1–2, 2020.
[15] Yanting Pei et al. Effects of image degradation and degradation removal to cnn-based image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1239–1253, 2021.
[16] L. Schuchman. Dither signals and their effect on quantization noise. *IEEE Transactions on Communication Technology*, 12(4):162–165, 1964.
[17] Dongyeob Shin et al. Prediction confidence based low complexity gradient computation for accelerating dnn training. In *DAC*, pages 1–6, 2020.
[18] Dongyeob Shin et al. Low complexity gradient computation techniques to accelerate deep neural network training. *IEEE TNNLS*, 2023.
[19] Karen Simonyan et al. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
[20] Chunyou Su et al. Towards high performance low bitwidth training for deep neural networks. *Journal of Semiconductors*, 2020.
[21] Jian Sun et al. Learning a convolutional neural network for non-uniform motion blur removal. In *CVPR*, pages 769–777, 2015.
[22] Xiao Sun et al. Ultra-low precision 4-bit training of deep neural networks. In *NeurIPS*, pages 1796–1807, 2020.
[23] Shuai Wang et al. Gradient distribution-aware int8 training for neural networks. *Neurocomput.*, jul 2023.
[24] Yue Wang et al. E2-train: Training state-of-the-art cnns with over 80% energy savings. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *NeurIPS*, volume 32. Curran Associates, Inc., 2019.
[25] Simon Wiedemann et al. Dithered backprop: A sparse and quantized backpropagation algorithm for more efficient deep neural network training. In *CVPRW*, pages 3096–3104, 2020.
[26] Yukuan Yang et al. Training high-performance and large-scale deep neural networks with full 8-bit integers. *Neural Networks*, pages 70–82, 2020.
[27] Geng Yuan et al. You already have it: A generator-free low-precision dnn training framework using stochastic rounding. In *ECCV*, pages 34–51. Springer, 2022.
[28] Kang Zhao et al. Distribution adaptive int8 quantization for training cnns. In *AAAI*, pages 3483–3491, 2021.
[29] Shuchang Zhou et al. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.
[30] Yiren Zhou et al. On classification of distorted images with deep convolutional neural networks. In *ICASSP*, pages 1213–1217, 2017.
[31] Feng Zhu et al. Towards unified int8 training for convolutional neural network. In *CVPR*, pages 1966–1976, 2020.