HERO: Hessian-Enhanced Robust Optimization for Unifying and Improving Generalization and Quantization Performance

Huanrui Yang*, Xiaoxuan Yang*, Neil Zhenqiang Gong and Yiran Chen
Duke University
Durham, NC, USA
{huanrui.yang,xy92,neil.gong,yiran.chen}@duke.edu

ABSTRACT

With the recent demand of deploying neural network models on mobile and edge devices, it is desired to improve the model's generalizability on unseen testing data, as well as enhance the model's robustness under fixed-point quantization for efficient deployment. Minimizing the training loss, however, provides few guarantees on the generalization and quantization performance. In this work, we fulfill the need of improving generalization and quantization performance simultaneously by theoretically unifying them under the framework of improving the model's robustness against bounded weight perturbation and minimizing the eigenvalues of the Hessian matrix with respect to model weights. We therefore propose HERO, a Hessian-enhanced robust optimization method, to minimize the Hessian eigenvalues through a gradient-based training process, simultaneously improving the generalization and quantization performance. HERO enables up to a 3.8% gain on test accuracy, up to 30% higher accuracy under 80% training label perturbation, and the best post-training quantization accuracy across a wide range of precision, including a > 10% accuracy improvement over SGD-trained models for common model architectures on various datasets.

ACM Reference Format:
Huanrui Yang*, Xiaoxuan Yang*, Neil Zhenqiang Gong and Yiran Chen. 2022. HERO: Hessian-Enhanced Robust Optimization for Unifying and Improving Generalization and Quantization Performance. In *Proceedings of the 59th ACM/IEEE Design Automation Conference (DAC) (DAC '22), July 10–14, 2022, San Francisco, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3489517.3530678

1 INTRODUCTION

The rapid development of deep learning algorithms has seen the emergence of high-performance deep neural network (DNN) models. Models like VGG [21], ResNet [9], MobileNet [20], etc., have been deployed on mobile and edge applications to process data gathered in the wild. Extensive model deployment requires the model to generalize well to unseen data, and to maintain high performance under fixed-precision quantization for memory and computational efficiency on mobile and edge devices [10].

* Equal contribution. This work is supported in part by NSF 1937786, NSF 2112562, NSF 1955246, and ARO W911NF-19-2-0107.

DAC '22, July 10–14, 2022, San Francisco, CA, USA

ACM ISBN 978-1-4503-9142-9/22/07…$15.00
https://doi.org/10.1145/3489517.3530678

In most cases, DNN models are trained following the *empirical risk minimization* (ERM) setting, whose objective is to minimize the model loss $L_{\mathbb{S}}(\boldsymbol{W})$ induced by weight $\boldsymbol{W}$ on the training set $\mathbb{S}$. However, only minimizing the ERM objective may not lead to an ideal model for practical applications: the model may be overfitted to the training set and have low testing accuracy [7, 24], or be severely degraded by the post-training quantization process when deploying to the real world [1, 27]. Previous work has been contributing empirical methods to improve DNN generalizability or quantization performance individually. Methods like weight decay [14], batch normalization [12], stochastic model architecture [11, 22], and intensive data augmentation [4, 25] improve model generalizability, yet they are not contributing to quantization performance [1]. Quantization-aware training [19, 23, 27] regains the quantization performance via retraining on a specific quantization precision, yet they fail to perform well when the precision is changed on the fly [1], also hurting the generalization performance of the full-precision model. A DNN training method achieving both high generalization accuracy and high quantization robustness is still lacking.

Interestingly, we notice that previous theoretical analysis has shed light on unifying the pursuit of generalization and quantization performance. Foret et al. [7] show DNN's generalization gap is related to the model's robustness against ℓ_2 norm bounded weight perturbation, whereas the robustness against quantization is shown to be equivalent to the robustness against ℓ_∞ norm bounded weight perturbation [1]. However, the first-order approximation used to improve weight perturbation robustness in both [7] and [1] leaves a weak robustness guarantee and makes them only work against one of the ℓ_2 or ℓ_∞ perturbation bound, failing to work on both generalization and quantization performance as we show later.

In this work, we aim to improve DNN generalization and quantization performance simultaneously with a novel optimization method. As discussed in Section 3.1, we form our objective as improving the model's robustness against a general ℓ_p norm bounded weight perturbation. Further analysis with second-order Taylor expansion in Section 3.2 unveils that the minimal perturbation strength under both ℓ_2 and ℓ_∞ bound leading to a loss increase can be bounded by the maximum eigenvalue of the Hessian matrix with respect to the weight. Therefore, in Section 4.1, we propose an effective way to regularize Hessian eigenvalue. We further derive Hessian-Enhanced Robust Optimization, *HERO*, which efficiently performs the Hessian eigenvalue regularization through a gradient-based optimization process. HERO leads to a better generalization performance and a better robustness to quantization on all precision, as in Section 4.2. To the best of our knowledge, HERO is the first to make the following theoretical contributions:

- Unifying generalization and quantization performance under the framework of improving the model's robustness against a general ℓ_p norm bounded weight perturbation;
- Showing the weight perturbation robustness can be improved via regularizing Hessian eigenvalues with respect to the model weights during DNN training;
- Deriving gradient update rule to optimize the Hessian eigenvalue regularization, leading to simultaneous improvement on generalization and quantization performance.

Our theoretical analysis is well-supported by the empirical results. For generalization, HERO consistently achieves higher test accuracy, including a significant 2.58% and 3.78% accuracy gain with MobileNetV2 over SGD on CIFAR-10 and CIFAR-100 datasets, respectively. We further show the generalizability achieved by HERO is robust under the presence of label noise, where HERO outperforms SGD by $5 \sim 30\%$ on ResNet20 and $2 \sim 10\%$ on MobileNetV2 when training on CIFAR-10 with $20 \sim 80\%$ label perturbation. For quantization, HERO provides the best post-training accuracy under a wide range of precision, including a $> 10\%$ accuracy improvement over SGD-trained MobileNet and VGG model at ultra-low precision of 4-5 bits. HERO also beats state-of-the-art Gradient ℓ_1 [1] by a large margin under all quantization schemes. Additional ablation studies are also provided to verify our theoretical insights.

2 RELATED WORK

2.1 Improving Model Generalization

As recent research utilizes heavily over-parameterized DNN models, it's essential to prevent the model from overfitting to the training set so that it can generalize well to unseen data. Overfitting can be largely resolved via regularization and data augmentation. For regularization, previous work has developed weight decay [14], dropout [22], stochastic depth [11], etc. As for data augmentation, recent methods explore mixup [25], auto-augmentation [4], etc. However, theoretical understanding of why these methods help model generalization is still lacking. Sharpness aware minimization (SAM) [7] theoretically links the generalization ability of the model with the model performance under ℓ_2 norm bounded weight perturbation, and therefore to the smoothness of the loss surface. SAM provides an efficient optimization algorithm to improve generalization, yet the first-order approximations involved weakens its ability to guarantee performance. HERO builds upon the observation of SAM and proposes an efficient way to regularize the loss smoothness via Hessian eigenvalues, leading to a stronger theoretical guarantee on the generalization performance. Moreover, the effectiveness of previous methods on quantized models is not well understood, while HERO unifies the pursuit of generalization and quantization, solving both problems simultaneously.

2.2 Improving Quantization Robustness

Quantization is essential for deploying a DNN model onto mobile and edge devices, as it saves on-device memory and achieves both run-time speedup and less energy cost [10]. Moreover, the dynamic change of power and memory availability on the device would require changing the precision of a pretrained model on the fly [1]. However, directly quantizing a DNN model to a low precision (less than 8-bit) will lead to a severe accuracy drop. Straight-through estimator [3] enables the finetuning of quantized models to regain the lost accuracy [19, 23, 27]. However, the resulting model only works on the exact quantization precision it is trained on; modifying the precision requires a lengthy retraining process. Others aim to design quantization schemes or rounding functions that can minimize the post-training quantization loss [2, 26], yet these methods require extensive analysis of the model architecture and parameter distribution, making it hard to apply on the fly. The only previous work successfully achieving general robustness against all quantization precision is Gradient ℓ_1 [1], which applies ℓ_1 regularization on the gradient of the model. As this method is based on a first-order approximation to the quantization loss, our later analysis shows it is insufficient to guarantee robustness. HERO further introduces Hessian regularization, which can lead to a stronger guarantee on much higher quantization robustness.

2.3 Curvature Regularization

As we link the problem of generalization and quantization with the model performance under weight perturbation, we take inspiration from the related field of adversarial robustness, where extensive studies have been done towards DNN's robustness against adversarial perturbation on the input [8, 17]. One noticeable work is the curvature regularization (CURE), which shows that the robustness against input perturbation can be improved by regularizing the Hessian eigenvalues of the loss function with respect to the input [18]. HERO also applies the Hessian eigenvalue regularization, but is different from CURE as we are working with respect to the model weight, rather than the input. The regularization of HERO needs to be computed on the weight tensors from multiple layers, each having distinct value and gradient ranges. We tackle the challenge of adapting perturbation strength across different layers based on their weight distribution, as introduced in Section 4.1. Furthermore, we apply additional first-order regularization to the optimization process as introduced in Section 4.2, effectively leading to better generalization and quantization performance.

3 THEORETICAL ANALYSIS

3.1 Unifying Generalization and Quantization

Here we start with investigating the properties needed for a deep neural network model to have both good generalizability and high quantization performance.

Bounding Generalization Gap. Recently, a theoretical analysis was made by Foret et al. [7] on bounding the generalization gap of a deep neural network, which can be stated as:

Theorem 1. *For any $\rho > 0$, with high probability over training set $\mathbb{S}$ generated from distribution $\mathbb{D}$,*

$$L_{\mathbb{D}}(\boldsymbol{W}) - L_{\mathbb{S}}(\boldsymbol{W}) \leq \left[\max_{||\delta||_2 \leq \rho} L_{\mathbb{S}}(\boldsymbol{W}+\delta) - L_{\mathbb{S}}(\boldsymbol{W}) \right] + h(||\boldsymbol{W}||_2^2/\rho^2) \tag{1}$$

where L is the loss function, $\boldsymbol{W}$ denotes the weight of the model and $h : \mathbb{R}_+ \to \mathbb{R}_+$ is a strictly increasing function [7].

Note that the second term relating to $||\boldsymbol{W}||_2^2$ can be effectively minimized during training with weight decay [14], so the generalization gap is largely bounded by the model's performance under a weight perturbation δ bounded by its ℓ_2 norm.

Bounding Quantization Loss. The post-training quantization process can also be considered as a process of perturbing the model weights. Here we focus on the typical setting of a linear uniform weight quantization [19], where the weight distribution is separated into 2^n uniform-sized bins, and each bin is rounded into a n-bit quantized value. Suppose the quantization bin has a width of Δ, the rounding function will change each element of the weight by at most $\Delta/2$. So the weight perturbation induced by quantization is bounded by the ℓ_∞ norm, as $||\delta||_\infty := ||\boldsymbol{W}_q - \boldsymbol{W}||_\infty \leq \Delta/2$, where $\boldsymbol{W}$ and $\boldsymbol{W}_q$ denote the original and quantized weight, respectively. Therefore we can bound the loss increase introduced by quantization as:

Theorem 2. *For a linear uniform quantization with a bin width $\Delta = 2\rho$, we have*

$$L_{\mathbb{S}}(\boldsymbol{W}_q) - L_{\mathbb{S}}(\boldsymbol{W}) \leq \left[\max_{||\delta||_\infty \leq \rho} L_{\mathbb{S}}(\boldsymbol{W}+\delta) - L_{\mathbb{S}}(\boldsymbol{W})\right], \tag{2}$$

which is bounded by the model's performance under a weight perturbation δ bounded by its ℓ_∞ norm.

Unifying the Bounds. With the analysis on Theorem 1 and 2, we can unify the pursuit of generalization and quantization performance as understanding how the model loss changes under a general ℓ_p norm bounded weight perturbation. Specifically, we can derive lower bounds for the minimal strength needed for perturbation δ to induce an increase c in the model loss as:

$$\delta^* := \arg\min_{\delta} ||\delta||_p \quad s.t.\ L_{\mathbb{S}}(\boldsymbol{W}+\delta) - L_{\mathbb{S}}(\boldsymbol{W}) \geq c. \tag{3}$$

A larger lower bound on $||\delta^*||_p$ indicates larger perturbations can be allowed given a tolerance of loss increase $< c$, which is desired.

3.2 Finding Perturbation Lower Bound

With a sufficiently small perturbation δ, we can use Taylor expansion to well approximate the loss increase under weight perturbation with a quadratic function:

$$L_{\mathbb{S}}(\boldsymbol{W}+\delta) - L_{\mathbb{S}}(\boldsymbol{W}) \approx \nabla_{\boldsymbol{W}} L_{\mathbb{S}}(\boldsymbol{W})^T \delta + \frac{1}{2}\delta^T \boldsymbol{H}\delta, \tag{4}$$

where $\nabla_{\boldsymbol{W}} L_{\mathbb{S}}(\boldsymbol{W})$ and $\boldsymbol{H}$ denote the gradient and Hessian of the loss with respect to the weight $\boldsymbol{W}$, respectively. For simplicity, in the rest of the section, we denote $g := \nabla_{\boldsymbol{W}} L_{\mathbb{S}}(\boldsymbol{W})$. We can thus rewrite the objective in Equation (3) as:

$$\delta^* := \arg\min_{\delta} ||\delta||_p \quad s.t.\ g^T\delta + \frac{1}{2}\delta^T\boldsymbol{H}\delta \geq c, \tag{5}$$

In the following discussion, we provide the lower bound on the minimal $||\delta^*||_2$ and $||\delta^*||_\infty$ needed to induce a loss increase of c with respect to the properties of the loss function at weight $\boldsymbol{W}$. The bounds on the magnitude of other ℓ_p norm bounded weight perturbations can be similarly derived from our result using the equivalence of norms in finite-dimensional spaces.

Theorem 3. *Assume that $v := \lambda_{max}(H) \geq 0$ as the largest eigenvalue of the Hessian, and $n := ||\boldsymbol{W}||_0$ as the number of nonzero elements in $\boldsymbol{W}$, we have*

$$\frac{||g||_2}{v}\left(\sqrt{1+\frac{2vc}{||g||_2^2}} - 1\right) \leq ||\delta^*||_2, \tag{6}$$

$$\frac{|g|}{nv}\left(\sqrt{1+\frac{2nvc}{|g|^2}} - 1\right) \leq ||\delta^*||_\infty. \tag{7}$$

Our theorem can be proved similarly to Theorem 1 in [18].

Note that the lower bounds in Equations (6) and (7) both monotonically increase with the decrease of v, i.e., a smaller Hessian eigenvalue. This implies that under second-order approximation, having small Hessian eigenvalues is beneficial in limiting the loss increase under ℓ_p bounded weight perturbation, therefore inducing better generalization and quantization performance.

Interestingly, note that the bound in Equation (7) is also monotonically increasing with decreasing $|g|$, showing the effectiveness of the previously proposed gradient ℓ_1 regularization [1]. Meanwhile, even if we consider the case where gradient ℓ_1 is fully optimized, i.e., $|g| \to 0$, we have the lower bound

$$\lim_{|g|\to 0}\left[\frac{|g|}{nv}\left(\sqrt{1+\frac{2nvc}{|g|^2}} - 1\right)\right] = \sqrt{\frac{2c}{nv}}, \tag{8}$$

which may still be small if the Hessian eigenvalue v is large. This analysis unveils that optimizing gradient ℓ_1 is inadequate for the model's robustness against quantization, while further minimizing Hessian eigenvalues provides a stronger guarantee.

4 HESSIAN-ENHANCED TRAINING

4.1 Regularizing Hessian Eigenvalues

Following the conclusion of Theorem 3, here we aim to propose a regularization term that can minimize the squared sum of the Hessian matrix $\boldsymbol{H}$'s eigenvalues λ_i to encourage all eigenvalues to be small, thus minimizing the maximum eigenvalue v. This leads to our regularizer formulation:

$$L_r = \sum_i \lambda_i^2 = \mathbb{E}_{\boldsymbol{z}}||\boldsymbol{H}\boldsymbol{z}||^2,\ \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}). \tag{9}$$

With a finite difference approximation of the Hessian, we have $\boldsymbol{H}\boldsymbol{z} \approx \frac{\nabla L(\boldsymbol{W}+h\boldsymbol{z}) - \nabla L(\boldsymbol{W})}{h}$, where h is a small positive number. Note that sampling multiple $\boldsymbol{z}$ from the Gaussian distribution to compute the expectation may be costly; thus, we follow the observation made in CURE [18], where the regularization loss can be estimated by only focusing on selected directions leading to high curvature, which often occurs along the gradient direction, i.e., $\boldsymbol{z} = \nabla L(\boldsymbol{W})$ [6, 18]. Thus we can convert the regularization term in Equation (9) into

$$L_r(\boldsymbol{W}) = ||\nabla L(\boldsymbol{W}+h\boldsymbol{z}) - \nabla L(\boldsymbol{W})||^2,\ \boldsymbol{z} = \nabla L(\boldsymbol{W}) \tag{10}$$

where $h > 0$ is a small parameter determining the step size of the perturbation, and the $\frac{1}{h^2}$ term can be omitted by absorbing into the regularization strength parameter.

For a DNN model, L_r needs to be computed on the weight tensors from all the layers, each having distinct dimensions and gradient value ranges. To accommodate the diversity among layers, we propose to compute L_r in a layer-wise fashion, and scale the ℓ_2 norm of the perturbation $\boldsymbol{z}$ to match the weight value range in each layer. Specifically, for layer i we have

$$L_r^i(\boldsymbol{W}^i) = ||\nabla L(\boldsymbol{W}^i + h\boldsymbol{z}^i) - \nabla L(\boldsymbol{W}^i)||^2,$$
$$\boldsymbol{z}^i = \frac{\boldsymbol{W}^{i^2}}{||\boldsymbol{W}^i||_2}\frac{\nabla L(\boldsymbol{W}^i)}{||\nabla L(\boldsymbol{W}^i)||_2}. \tag{11}$$

The overall Hessian regularization is therefore computed as $L_r(\boldsymbol{W}) = \sum_{i=1}^{N} L_r^i(\boldsymbol{W}^i)$, summing over all the N layers in the model.

4.2 Hessian-enhanced Robust Optimization

In order to minimize $L_r(\boldsymbol{W})$ during DNN training, we provide an efficient and effective method to compute the gradient of $L_r^i(\boldsymbol{W}^i)$ with respect to $\boldsymbol{W}^i$. We start with defining $G(\boldsymbol{U}) := ||\nabla L(\boldsymbol{U}) - \nabla L(\boldsymbol{W}^i)||^2$, which allow us to convert $\nabla L_r^i(\boldsymbol{W}^i)$ to

$$\begin{aligned}\nabla L_r^i(\boldsymbol{W}^i) &= \nabla_{(\boldsymbol{W}^i+h\boldsymbol{z}^i)} G(\boldsymbol{W}^i + h\boldsymbol{z}^i) \cdot \nabla_{\boldsymbol{W}^i}(\boldsymbol{W}^i + h\boldsymbol{z}^i) \\ &\approx \nabla_{(\boldsymbol{W}^i+h\boldsymbol{z}^i)} G(\boldsymbol{W}^i + h\boldsymbol{z}^i).\end{aligned} \tag{12}$$

With this conversion, our regularization can be optimized with only one additional back propagation on the gradient difference G with respect to the perturbed weight $\boldsymbol{W}^i + h\boldsymbol{z}^i$, which is well supported by common deep learning libraries such as TensorFlow and PyTorch. Note that we discard the second-order term $\nabla_{\boldsymbol{W}^i}(\boldsymbol{z}^i)$ in the final derivation step, which has been proven to be an effective approximation by [7].

In the meantime, note that regularizing the Hessian eigenvalue is necessary yet insufficient for the robustness against generalization and quantization. Since the Hessian regularization only regularizes the second-order derivative but not the first-order one, the final "optimum" may end up on a flat but steep slope in the loss surface. Adding a first-order regularization on the gradient norm is needed to mitigate the problem and complete the robust optimization. However, directly adding the ℓ_p norm of the gradient to the overall loss function requires additional computation and an additional regularization strength parameter. So instead, we take inspiration from the previous sharpness-aware minimization (SAM) method [7], which shows replacing the gradient of the original weight $\nabla_{\boldsymbol{W}^i} L(\boldsymbol{W}^i)$ with the gradient of the perturbed weight $\nabla_{(\boldsymbol{W}^i+h\boldsymbol{z}^i)} L(\boldsymbol{W}^i + h\boldsymbol{z}^i)$ in the SGD process effectively serves as a first-order regularization on the gradient norm and loss sharpness. This replacement can be made without additional cost as we already have $\nabla_{(\boldsymbol{W}^i+h\boldsymbol{z}^i)} L(\boldsymbol{W}^i + h\boldsymbol{z}^i)$ computed in the computation of $L_r^i(\boldsymbol{W}^i)$.

With the approximation in Equation (12) and the addition of the first-order regularization in the SGD process, we can derive the gradient of our Hessian-enhanced robust optimization as

$$\nabla_{\boldsymbol{W}^i} = \nabla_{(\boldsymbol{W}^i+h\boldsymbol{z}^i)} L(\boldsymbol{W}^i + h\boldsymbol{z}^i) + \alpha \boldsymbol{W} + \gamma \sum_{i=1}^{N} \nabla_{(\boldsymbol{W}^i+h\boldsymbol{z}^i)} G(\boldsymbol{W}^i + h\boldsymbol{z}^i), \tag{13}$$

where $\alpha > 0$ denotes the weight decay and $\gamma > 0$ denotes the regularization strength of the Hessian regularization. Performing SGD optimization with the derived gradient $\nabla_{\boldsymbol{W}}$ in Equation (13) leads to the HERO algorithm.

5 EVALUATION

5.1 Experiment Setup

We evaluate HERO with three representative DNNs: ResNet20 [9], MobileNetV2 [20], and VGG19 with batch normalization (VGG19BN) [21] on the CIFAR-10 and CIFAR-100 datasets [13]. The parameter numbers of these networks are $0.27M$ (ResNet20), $2.30M$ (MobileNetV2), and $20.04M$ (VGG19BN). We further evaluate HERO with ResNet18 [9] using the ImageNet dataset [5] to validate the scalability of our method. The parameter number of ResNet18 is $11.17M$. We compare our approach with the stochastic gradient descent (SGD) and Gradient $\ell 1$ (GRAD L1) [1] training methods. We include GRAD L1 as a baseline because it is by far the state-of-the-art regularization method towards quantization robustness, yet only uses the first-order information of the quantization loss, in contrast to the second-order information used by HERO.

Table 1: Test accuracy on various models and datasets.

Dataset	Model	HERO	GRAD L1	SGD
CIFAR-10	ResNet20	**93.44%**	92.82%	92.82%
	MobileNetV2	**95.03%**	92.52%	92.45%
	VGG19BN	**94.79%**	93.41%	93.89%
CIFAR-100	ResNet20	**70.72%**	69.30%	69.52%
	MobileNetV2	**76.90%**	74.13%	73.12%
	VGG19BN	**76.09%**	74.05%	74.61%
ImageNet	ResNet18	**71.05%**	70.82%	70.74%

All methods utilize a cosine learning rate scheduler with an initial learning rate η of 0.1. We set the momentum as 0.9 and the weight decay α as 10^{-4}. For the CIFAR-10 and CIFAR-100 experiments, we apply basic data augmentations, such as random crop, padding, and random horizontal flip on the training set, and train the model for 200 epochs with batch size 128. For the ImageNet experiments, random resized crop and normalization are applied to the training set. We train the model for 100 epochs with batch size 256. Note that we train the model from scratch in all the experiments. All experiments are conducted using NVIDIA TITAN RTX GPUs.

For HERO, to select the Hessian regularization strength γ, we conduct a grid search over $\{0.01, 0.05, 0.1, 0.5, 1.0, 5.0\}$. For the weight perturbation step size h, we follow the previous experiment settings in [7] to utilize 0.5 for CIFAR-10 experiments and 1.0 for other experiments. For the GRAD L1 regularization strength, we follow the steps in [1] to run a grid search to find the best hyperparameter with the minimal sacrifice of the test accuracy.

5.2 Improving Model Generalization

As discussed in Theorem 1 and Equation (6), HERO is beneficial on limiting the loss increase under ℓ_2 bounded weight perturbation, thus realizing better generalization performance. In this subsection, we showcase HERO's effectiveness in improving model generalizability with experiments on the test accuracy comparison and the noisy-label training performance.

Test Accuracy. We evaluate the test accuracy of HERO and baseline methods in Table 1. For ResNet20, MobileNetV2 and VGG19BN models, HERO achieves 0.62%, 2.58% and 0.90% accuracy gain compared with SGD on CIFAR-10 dataset respectively. The performance also increases on CIFAR-100 by 1.20%, 3.78%, and 1.48% with respect to SGD. Notice that HERO enables a better test accuracy on compact models without enlarging the network size. For instance, on the CIFAR-10 and CIFAR-100 dataset, the MobileNetV2 test accuracy achieved by HERO can outperform the VGG19 test accuracy achieved by SGD, with $\sim 8.7\times$ fewer parameters. This further benefits the deployment of efficient models in the real world.

On the contrary, we find GRAD L1 method, which is designed against ℓ_∞ bounded weight perturbation, doesn't guarantee a consistent improvement of the test accuracy against SGD. This implies that generalizing the robustness against ℓ_∞ bounded to ℓ_2 bounded

Table 2: Test accuracy under noisy-label training.

(a) ResNet20

Noise ratio	20%	40%	60%	80%
HERO	**90.63%**	**88.71%**	**84.61%**	**72.11%**
GRAD L1	85.91%	78.66%	65.86%	48.28%
SGD	85.64%	78.73%	66.42%	42.17%

(b) MobileNetV2

Noise ratio	20%	40%	60%	80%
HERO	**91.70%**	**88.57%**	**81.73%**	**72.03%**
GRAD L1	89.00%	85.56%	79.73%	30.34%
SGD	89.28%	85.84%	80.49%	62.91%

weight perturbation isn't trivial. On the other hand, HERO provides both consistent promising generalization performance and robustness against quantization, as further discussed in Section 5.3.

To further validate the scalability of HERO, we test with the ResNet18 model on ImageNet. The result confirms that HERO can improve the generalization compared to GRAD L1 and SGD.

Noisy-Label Training. For models trained on real-world data, inevitable label noise will exist in the training dataset. Robustness against noisy labels in the training process is essential for the model's generalizability to the test data. Here we show that HERO is still robust under the presence of noisy labels.

We utilize ResNet20 and MobileNetV2 networks on the CIFAR-10 dataset for this experiment. First, we follow the symmetric noisy label generation in [16], where we uniformly sample a certain proportion (from 20% to 80%, namely *noise ratio*) of the training data and replace their labels with a uniform random sample from all the possible classes. We then train the model with the same training procedure on the perturbed training set, and evaluate the accuracy on the original clean test set. As shown in Table 2, HERO has the best test accuracy across all noise ratios among all three methods. Besides, the test accuracy of GRAD L1 and SGD drops dramatically at the high noise ratio of 80%; while the HERO approach still provides acceptable results. Therefore, HERO shows its robustness against the training label perturbation and achieve the best performance under noisy training label among all methods.

5.3 Improving Quantization Robustness

In Theorem 2 and Equation (2), we show that the loss change of uniform weight quantization is bounded by the model performance under a weight perturbation δ bounded by its ℓ_∞ norm, where lower quantization precision indicates a higher weight perturbation. Here we demonstrate the quantization robustness achieved by HERO with the post-training quantization to various precision. No quantization-aware finetuning is performed in these experiments.

The experiments on the CIFAR-10 dataset are shown in Figure 1 (a)-(c). The test accuracy for HERO across different quantization precision is consistently higher than that of GRAD L1 and SGD. Our observation matches with [1] that GRAD L1 can achieve better test accuracy to some extent under low weight precision compared with SGD. Yet, the second-order regularization introduced by HERO provides a better guarantee of quantization robustness.

More significantly, the HERO performance under low quantization precision shows a large improvement compared with baselines across all the precision. For instance, for the MobilenetV2 network,

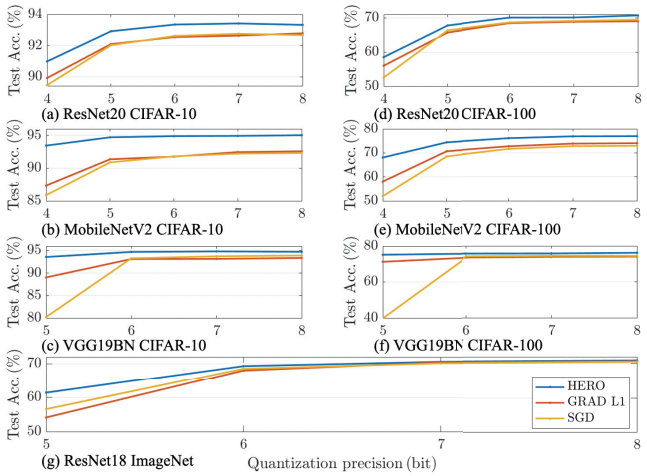


Figure 1: Post-training quantization accuracies with HERO, GRAD L1, and SGD: (a)-(c) ResNet20, MobileNetV2, VGG19BN experiments on CIFAR-10 dataset; (d)-(f) ResNet20, MobileNetV2, VGG19BN experiments on CIFAR-100 dataset; (g) ResNet18 experiments on ImageNet dataset.

test accuracy for HERO under 4-bit weight is 93.45%, significantly higher than the 87.34% and 85.88% achieved by GRAD L1 and SGD, respectively. DNN quantization with ultra-low precision is a challenging problem due to large perturbations on the weights, while HERO effectively provides robustness against such perturbation.

We also notice that a model with more parameters is more sensitive to quantization perturbation. In our case of the VGG19BN network, SGD with 5-bit quantization already leads to noticeable accuracy degradation compared to full precision results. In the meantime, HERO still retains a 93.57% test accuracy compared to the 89.03% and 80.22% accuracy of GRAD L1 and SGD, showing its effectiveness on larger models.

A similar trend can also be observed on other datasets. On the CIFAR-100 dataset, as shown in Figure 1 (d)-(f), the consistent trend that HERO outperforms GRAD L1 and SGD still holds across different quantization precision. Besides, in the low precision setting, HERO has an outstanding performance gain compared with baseline methods. For instance, on the MobileNetV2 network, HERO improves the test accuracy under 4-bit quantization by 10.05% and 16.10% compared with GRAD L1 and SGD, respectively. Our quantization result with ResNet18 on ImageNet dataset also shows that HERO can provide better quantization robustness across different quantization precision, as shown in Figure 1 (g).

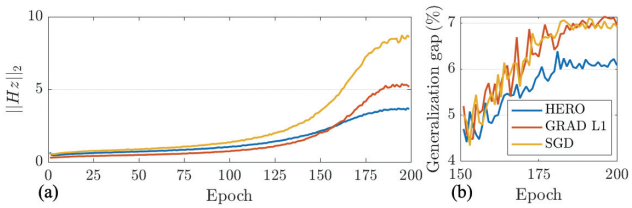
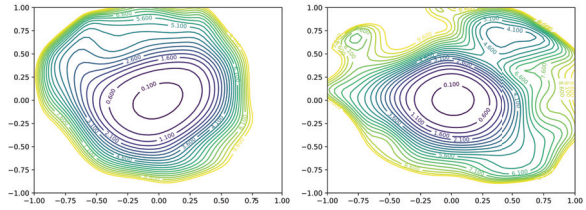


Figure 2: Hessian norm and generalization gap evolution through the training with HERO, GRAD L1, and SGD.



(a) HERO, test accuracy: 93.44% (b) SGD, test accuracy: 92.82%

Figure 3: Loss surface contour along 2 random directions around converged weights. Estimated on ResNet20 model on CIFAR-10 dataset trained with HERO and SGD.

5.4 Theoretical Insight Verification

Hessian Norm across Training Process. To show the effectiveness of HERO on regularizing Hessian eigenvalues, we visualize the evolution of the Hessian norm $||\boldsymbol{Hz}||_2$ throughout the training process in Figure 2 (a) following the setting in [18], with $\boldsymbol{z}$ being the perturbation defined in Equation (11). All curves are averaged over the entire CIFAR-10 training set. The generalization gap between training and test accuracy in the final 50 training epochs is shown in Figure 2 (b). Throughout the training process, the Hessian norm gets larger as the model overfits to the training set. Meanwhile, HERO helps keep the Hessian norm values at the lowest level towards the end of the training process, and thus leads to the lowest generalization gap as expected.

Loss Contour Visualization. We further showcase the weight perturbation robustness achieved by HERO with the loss contour in the neighborhood of the converged weights, with HERO in Figure 3 (a) and SGD in Figure 3 (b) plotted under the same scale. The contour is generated with the visualization tool provided by [15], which monitors the loss change while applying normalized adjustments to the weight along two random directions. Compared to that of SGD, the loss surface of HERO appears to be smoother, with a larger region within the inner contour circle indicating a 0.1 loss increase. This shows HERO is robust to larger perturbation within a tolerance of loss increase, which is well in line with Theorem 3.

The Necessity of Hessian-enhanced Method. In the derivation of HERO's gradient in Equation (13), we borrow the first-order regularization ($\nabla_{\boldsymbol{W}} = \nabla_{(\boldsymbol{W}^i+h\boldsymbol{z}^i)} L(\boldsymbol{W}^i + h\boldsymbol{z}^i)$) from SAM [7] alongside our Hessian regularization. SAM itself already leads to the state-of-the-art result on generalization performance [7], beating dropout [22] and Mixup [25]. Here we show the Hessian regularization introduced by HERO is still necessary, as it further increases the generalization and quantization performance over SAM. We compare HERO with the first-order only method (i.e., $\nabla_{\boldsymbol{W}^i} = \nabla_{(\boldsymbol{W}^i+h\boldsymbol{z}^i)} L(\boldsymbol{W}^i+h\boldsymbol{z}^i)+\alpha\boldsymbol{W}$) and SGD (i.e., $\nabla_{\boldsymbol{W}^i} = \nabla_{\boldsymbol{W}^i} L(\boldsymbol{W}^i)+\alpha\boldsymbol{W}$) in Table 3. For test accuracy on the full precision model, HERO provides an additional 1% gain over the first-order only method. Furthermore, HERO provides better robustness against quantization. For example, 4-bit weight quantization with the HERO model leads to a 1.6% accuracy drop, much smaller than the 2.5% drop achieved with the first-order regularization. The result confirms the necessity of including the Hessian regularization in the pursuit of both generalization and quantization performance.

Table 3: Ablation study on HERO, first-order only, and SGD gradient update rule. Results reported with MobileNetV2 network on CIFAR-10 dataset.

Quantization (bit)	4	6	8	Full
HERO	**93.45%**	**94.90%**	**95.03%**	**95.03%**
First-order only	91.61%	93.92%	94.00%	94.06%
SGD	85.88%	91.81%	92.33%	92.45%

6 CONCLUSION

This work proposes HERO, a Hessian-enhanced robust optimization method to improve the generalization and quantization performance of DNN models simultaneously. We provide novel insights on unifying generalization and quantization under improving weight perturbation robustness, theoretical analysis on enhancing the robustness with Hessian regularization, and empirical results showing the effectiveness of HERO. We hope this work helps on deploying DNN models onto real-world mobile and edge devices, and inspires further attention to the robustness against weight perturbation.

REFERENCES

[1] Milad Alizadeh et al. 2020. Gradient l1 regularization for quantization robustness. *arXiv preprint arXiv:2002.07520* (2020).
[2] Ron Banner et al. 2018. Post-training 4-bit quantization of convolution networks for rapid-deployment. *arXiv preprint arXiv:1810.05723* (2018).
[3] Yoshua Bengio et al. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432* (2013).
[4] Ekin D Cubuk et al. 2018. AutoAugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501* (2018).
[5] Jia Deng et al. 2009. ImageNet: A large-scale hierarchical image database. In *ICCV*.
[6] Alhussein Fawzi et al. 2018. Empirical study of the topology and geometry of deep networks. In *ICCV*.
[7] Pierre Foret et al. 2020. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412* (2020).
[8] Ian J Goodfellow et al. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).
[9] Kaiming He et al. 2016. Deep residual learning for image recognition. In *ICCV*.
[10] Mark Horowitz. 2014. 1.1 computing's energy problem (and what we can do about it). In *ISSCC*.
[11] Gao Huang et al. 2016. Deep networks with stochastic depth. In *ECCV*.
[12] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).
[13] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning multiple layers of features from tiny images*. Technical Report.
[14] Anders Krogh and John A Hertz. 1991. A simple weight decay can improve generalization. In *NeurIPS*.
[15] Hao Li et al. 2017. Visualizing the loss landscape of neural nets. *arXiv preprint arXiv:1712.09913* (2017).
[16] Junnan Li et al. 2020. DivideMix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394* (2020).
[17] Aleksander Madry et al. 2018. Towards deep learning models resistant to adversarial attacks. In *ICLR*.
[18] Seyed-Mohsen Moosavi-Dezfooli et al. 2019. Robustness via curvature regularization, and vice versa. In *ICCV*.
[19] Antonio Polino et al. 2018. Model compression via distillation and quantization. *arXiv preprint arXiv:1802.05668* (2018).
[20] Mark Sandler et al. 2018. MobileNetV2: Inverted residuals and linear bottlenecks. In *ICCV*.
[21] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[22] Nitish Srivastava et al. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* (2014), 1929–1958.
[23] Huanrui Yang et al. 2021. BSQ: Exploring bit-level sparsity for mixed-Precision neural network quantization. *arXiv preprint arXiv:2102.10462* (2021).
[24] Chiyuan Zhang et al. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530* (2016).
[25] Hongyi Zhang et al. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).
[26] Ritchie Zhao et al. 2019. Improving neural network quantization without retraining using outlier channel splitting. In *ICML*. 7543–7552.
[27] Shuchang Zhou et al. 2016. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160* (2016).