We thank the reviewers for their positive and constructive feedbacks. We tried our best to respond to all the raised issues and will reflect them in the final version.

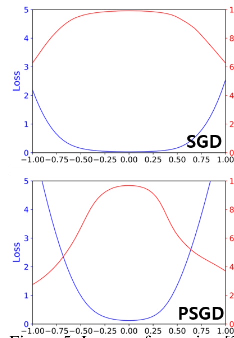

Figure 5: Loss surface using [35]; SGD (top) and PSGD (bottom)

[S1] PSGD is a Regularization method: Post-training quantization (PTQ) methods need a pre-trained model. However, a model pre-trained with SGD suffers from the problem shown in Fig. 1 (line 44-49). To tackle this issue, we train a compression-friendly model at full-precision (FP) with cross-entropy loss using PSGD. Our method can be considered as a regularization method equivalent to [1] (line 50-60, 80-89). After training, we can use simple layer-wise quantization to obtain a low-precision (LP) model without any data nor post-training (line 257), while PTQ methods need calibration data or additional computing phases. Note that our PSGD has a similar accuracy with the SGD-trained model at FP.

[S2] Comparison to other Post-training methods: We employ layer-wise quantization. ACIQ [2] uses channel-wise quantization (e.g. scale factor and zero point *per channel*) which attains higher performance at the expense of hardware-friendliness as noted in many prior works [7, 26, 34]. We had already cited the work and included the differences between the two methods in Sec. 2 (line 72-79). A similar rationale is given in Sec. 5.1 of a concurrent work [34] for not comparing with channel-wise methods. In Table 1 of ACIQ [2], the naive (channel-wise) baseline of ResNet-18 W4A4 (ImageNet) is 51.6% as opposed to 0.3% for ours (layer-wise). Hence, improving layer-wise quantization is a much more challenging problem that deserves attention because of its hardware efficiency. We have already compared with SOTA layer-wise methods in Table 2&3. Additionally, our PSGD can be combined with PTQ methods because we do not use any post-training. We performed additional experiments using a model trained with PSGD then post-processing with a concurrent PTQ work, LAPQ [34], using the official code. This attains 66.5% accuracy for W4A4, which is more than 3.1% and 6.2% points higher than that of PSGD-only and LAPQ-only respectively. Note that at lower bits such as W2A8, we attain 62.7% accuracy, while LAPQ has 1.3% accuracy.

[S3] Convergence analysis: Our algorithm is a variant of GD; the equivalent convergence analysis can be applied with the condition that the step-size, $0 < t \leq \frac{1}{L}$ where L is a Lipschitz constant. The detailed definition and proof are in [38].

Theorem: Given the scaling vector, $s(\cdot) \in \mathbb{R}^n$ and a convex, L-smooth function, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfies: $f(x_{i+1}) - f(x_i) \leq (\frac{s(x_i)^{\circ 2} - 2s(x_i)}{2L})^\intercal \nabla f(x_i)^{\circ 2}$, which is monotonically nonincreasing because r.h.s is always negative. As $i \rightarrow \infty$, $f(x_{i+1})$ converges to the optimum. ($\circ$ denotes the Hadamard operation)
Proof: Substituting x_{i+1} for $x_i - \frac{s(x_i)}{L} \circ \nabla f(x_i)$ into r.h.s of $f(x_{i+1}) - f(x_i) \leq \nabla f(x_i)^\intercal (x_{i+1} - x_i) + \frac{L}{2} \|x_{i+1} - x_i\|_2^2$, (which follows from the property of L-smoothness) yields the inequality. Given $s(x_i) = \frac{abs(x_i - \bar{x}_i) + \epsilon}{\|x_i - \bar{x}_i\|_\infty + \epsilon}$ (Appendix B), which satisfies $0 < s(x_i)_j \leq 1, \forall j \in [1, n]$ the r.h.s is always negative. ($abs(\cdot)$ is defined as the element-wise absolute value function).

Reviewer 1 (R1): Thank you for the positive feedback! We will include the suggested recent studies, revise the figures, and check with the English in the final version. Regarding Fig. 2, we will do our best to intuitively present our idea.

Reviewer 2 (R2): We are pleased the reviewer pinpointed the keypoints of the paper. ***I. Suggestion for comparison:*** These papers [36,37] propose to use LP arithmetic at the training phase or use different representation format to encode the parameters. While direct comparison may be difficult, we believe that the idea proposed by [37] can be incorporated into our pre-trained model for future work. An additional experiment applying a PTQ method is presented in [S2]. ***II. Evaluation with pruning:*** As pointed out, PSGD also achieves high sparsity as zero is included in the target set. The sparsity of ResNet-18@W4 (ImageNet) at LP is 72.4%! We will reflect ***I*** and ***II*** in the final version.

Reviewer 3 (R3): Thank you for the meaningful feedback. ***I. Convergence analysis:*** Please refer to [S3] for the convergence analysis. ***II. Position of our method:*** We apologize if we caused any confusion. Our method is not a post-training method and further details are in [S1]. ***III. Visualization of loss space:*** We respectfully disagree to the claim that empirical demonstration was not shown in a realistic case, as Sec. 5 and Fig. 4(b) compare the curvature of the solutions of a neural network. As suggested, we have also used official code of [35] to qualitatively assess the curvature in Fig. 5, using the same experimental setting of Sec. 5, which shows a similar tendency. ***IV. Stronger Baseline:*** Suggested baseline, ACIQ [2] is a channel wise quantization method. Detailed explanation regarding why channel-wise method was not compared is in [S2]. We will reflect all issues to avoid any confusion.

Reviewer 4 (R4): We are glad that the reviewer apprehended our novelty. ***I. Taylor Expansion:*** The equation is derived using Taylor expansion around $\boldsymbol{y}_t$ for the first equality. $\mathcal{F}^{-1}(\boldsymbol{y}_t - \eta \nabla_{\boldsymbol{y}}^{\mathcal{L}'}(\boldsymbol{y}_t)) = \mathcal{F}^{-1}(\boldsymbol{y}_t) + \mathcal{J}_{\boldsymbol{y}}^{\boldsymbol{x}}(\boldsymbol{y}_t)(\boldsymbol{y}_t - \eta \nabla_{\boldsymbol{y}}^{\mathcal{L}'}(\boldsymbol{y}_t) - \boldsymbol{y}_t) = \mathcal{F}^{-1}(\boldsymbol{y}_t) - \eta \mathcal{J}_{\boldsymbol{y}}^{\boldsymbol{x}}(\boldsymbol{y}_t) \nabla_{\boldsymbol{y}}^{\mathcal{L}'}(\boldsymbol{y}_t)$. ***II. Eq.(5):*** Eq.(5) is derived by selecting the scaling factor (Eq.(6)), $s(x) = \frac{1}{[f'(x)]^2}$, which is the scale multiplied to the gradient of the original space. Then, Eq.(5) can be interpreted as the warping function. The motive for Eq.(6) is explained in (line 137-142). ***III. Pruning baseline:*** We only considered single-shot pruning [22,25] because the intention of the experiment was to see the effectiveness of PSGD on making weights converge to zero (line 215-220). Comparing recent pruning methods and applying iterative pruning schedules to PSGD is our future work which is not the scope of this work. ***IV. Fig. 4:*** The intention of this section was to point out that PSGD solution cannot be found by standard SGD as it lies in a much sharper local minimum. The validity of the PSGD solution is explained in Sec. 3.4. and [S3]. Moreover, the solution is more quantization-friendly than that of SGD because it reduces the quantization error (refer to Fig.1 and Sec. 3.2). We will reflect raised issues for clearer understanding.

[34] Nahshan, Yury, et al. "Loss Aware Post-training Quantization." arXiv preprint arXiv:1911.07190 (2019).
[35] Li, Hao, et al. "Visualizing the loss landscape of neural nets." Advances in Neural Information Processing Systems. 2018.
[36] De Sa, Christopher, et al. "High-accuracy low-precision training." arXiv preprint arXiv:1803.03383 (2018).
[37] Tambe, Thierry, et al. "AdaptivFloat: A Floating-Point Based Data Type for Resilient Deep Learning Inference." arXiv preprint arXiv:1909.13271 (2019).
[38] EE236C, L. Vandenberghe. "1. Gradient method."