OPT2025: 17th Annual Workshop on Optimization for Machine Learning

# LOTION: Smoothing the Optimization Landscape for Quantized Training

**Mujin Kwun**[1]
**Depen Morwani**[1]
**Chloe Huangyuan Su**[1,2]
**Stephanie Gil**[2,1]
**Nikhil Anand**[1]
**Sham Kakade**[1,2]
[1] *Kempner Institute for the Study of Natural and Artificial Intelligence, Harvard University*
[2] *Department of Computer Science, Harvard University*

## Abstract

Optimizing neural networks for quantized objectives is fundamentally challenging because the quantizer is piece-wise constant, yielding zero gradients everywhere except at quantization thresholds where the derivative is undefined. Most existing methods deal with this issue by relaxing gradient computations with techniques like Straight Through Estimators (STE) and do not provide any guarantees of convergence. In this work, taking inspiration from Nesterov smoothing, we approximate the quantized loss surface with a continuous loss surface. In particular, we introduce LOTION, **L**ow-precision **O**ptimization via s**T**ochastic-no**I**se sm**O**othi**N**g, a principled smoothing framework that replaces the raw quantized loss with its expectation under unbiased randomized-rounding noise. In this framework, standard optimizers are guaranteed to converge to a local minimum of the loss surface. Moreover, when using noise derived from stochastic rounding, we show that the global minima of the original quantized loss are preserved. We empirically demonstrate that this method outperforms standard QAT on synthetic testbeds and on 150M- and 300M- parameter language models.

## 1. Introduction

While the performance of LLMs scales predictably with the size of the model and the amount of data it was trained on [12], these improved capabilities are accompanied by a corresponding cost when the model is deployed for inference. As a result, model compression and low-precision execution are becoming the default for training and serving LLMs on modern accelerators.

Although post-training quantization (PTQ) and quantization-aware training (QAT) directly alleviate this burden by compressing model weights and/or activations to low-precision formats, it turns the training objective into a highly *discontinuous* surface: every forward pass hard-assigns weights to a finite codebook, zeroing gradients almost everywhere. The usual workaround is the straight-through estimator (STE), which simply treats the non-differentiable quantizer as the identity in the backward pass. Despite some empirical successes, naive, identity-based STEs provide no guarantees and tend to become unstable in newer low-precision formats that quantize more aggressively, motivating a more principled alternative [10, 15, 18, 20, 22–25]. This work seeks a fully principled, parameter-free alternative which is applicable to a wide variety of rounding schemes.

We propose **LOTION—L**ow-precision **O**ptimization via s**T**ochastic-no**I**se sm**O**othi**N**g. Instead of modifying the gradient, LOTION *directly smooths the loss itself*: it trains on the expectation of the quantized loss under (possibly unbiased) stochastic-rounding noise. This expectation is differentiable almost everywhere, so any standard first- or second-order optimizer can be used with its usual convergence guarantees. A diagonal Gauss–Newton analysis reveals that LOTION is equivalent to adding a data-dependent regularization whose strength is tied to the error induced by per coordinate quantization and the hessian curvature in the coordinate, yielding the first explicit connection between randomized rounding and curvature-aware regularization.

**Our contributions.**

- We introduce **LOTION**, a loss-smoothing framework that replaces discontinuous quantized objectives with an almost everywhere differentiable surrogate obtained via stochastic rounding.
- Via a diagonal Gauss–Newton approximation, we derive a closed-form curvature-aware regularizer, providing a transparent interpretation of how stochastic rounding stabilizes training.
- We show that smoothing preserves *all* global minima of the original quantized problem and, when coupled with any convergent optimizer (e.g. Adam, Shampoo), inherits their convergence guarantees.
- As a proof of concept, we show that LOTION vastly exceeds the accuracy of widely used STE-based QAT and PTQ methods at INT4 precision on a synthetic linear regression task. Additionally, we pretrain 150M and 300M parameter language models and quantize them to INT4, INT8, and FP4, achieving lower post-quantization validation loss than PTQ and QAT baselines.

## 2. Problem Setting

We adopt the standard supervised learning setup. Let $(x, y) \sim \mathcal{D}$ denote input–label pairs and let $f(w; x)$ be the output of a neural network parameterised by weights $w \in \mathbb{R}^d$. With a per-example loss $\ell(\cdot, \cdot)$, the population loss is

$$\mathcal{L}(w) \;=\; \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\,\ell\big(f(w;x),\,y\big)\big].$$

Quantization constrains us to a finite codebook of representable weights. Let $\mathrm{cast} : \mathbb{R}^d \to Q$ denote the quantization operator mapping real-valued weights to $Q \subset \mathbb{R}^d$. We therefore seek

$$\min_{w\in\mathbb{R}^d} \; \mathcal{L}\big(\mathrm{cast}(w)\big).$$

**Remark 1** *The mapping $w \mapsto \mathcal{L}\big(\mathrm{cast}(w)\big)$ is piecewise constant; gradients vanish except on the measure-zero cell boundaries induced by the quantizer, where the gradients do not exist.*

## 3. LOTION: Smoothing the Loss

The core idea of our smoothing approach is to turn our non-differentiable (and discontinuous) optimization problem into a continuous one. In particular, the approach is to consider a stochastically perturbed optimization problem of the form:

$$\mathcal{L}_{\text{smooth},D}(w) = \mathbb{E}_{q\sim D_w}[\mathcal{L}(q)]$$

where $D_w$ represents a distribution over the points in $Q$, where the distribution is allowed to depend on $w$. For example, the Gaussian smoothing approach (analyzed by Nesterov [19]) would be to first sample $\epsilon \sim N(0, \sigma^2 I)$, and then take $q = \text{cast}(w + \epsilon)$. For this choice of $D(w)$, then $\mathcal{L}_{\text{smooth},D}$ will be a continuous and differentiable function for all orders.

In this work, we will consider a stochastic rounding approach, formally defined in B.2, which lets us make connections to prior work and helps us derive a more principled regularization approach.

### 3.1. General Case and a Gauss–Newton Regulariser

We define the smoothed objective

$$\mathcal{L}_{\text{smooth}}(w) \;:=\; \mathbb{E}_{\varepsilon\sim \text{RR}(w)}\big[\mathcal{L}(w+\varepsilon)\big],$$

where $\text{RR}(w)$ is the unbiased randomized-rounding distribution formally defined in Section B.1. Under random rounding, a parameter is rounded up or down with probability corresponding to the distance from the upper and lower quantization bin. For a twice-differentiable loss we have the second–order expansion

$$\mathcal{L}(w+\varepsilon) = \mathcal{L}(w) + g(w)^\top \varepsilon + \tfrac{1}{2}\,\varepsilon^\top H(w)\varepsilon + \mathcal{O}\big(\|\varepsilon\|^3\big),$$

with gradient $g(w) = \nabla\mathcal{L}(w)$ and Hessian $H(w) = \nabla^2\mathcal{L}(w)$. Taking expectations and using $\mathbb{E}[\varepsilon] = 0$ yields

$$\mathcal{L}_{\text{smooth}}(w) = \mathcal{L}(w) + \tfrac{1}{2}\,\text{tr}\big(H(w)\,\Sigma_\varepsilon(w)\big) + \mathcal{O}\big(\mathbb{E}[\|\varepsilon\|^3]\big),$$

where $\Sigma_\varepsilon(w) = \text{Cov}[\varepsilon]$.

**Gauss–Newton replacement.** The Hessian of the neural network $f$ can be decomposed as a sum of two terms:

$$\nabla_w^2 \ell = \underbrace{\nabla_w f^T \nabla_f^2 \ell \nabla_w f}_{G(w)} + \nabla_w \ell \nabla_w^2 f$$

where the first component is positive semi-definite for convex losses and is referred to as the Gauss-Newton component. As the full Hessian may introduce negative curvature, we therefore substitute it with the positive-semidefinite *Gauss–Newton* matrix. Dropping higher-order terms gives the working approximation

$$\boxed{\mathcal{L}_{\text{GN}}(w) \;=\; \mathcal{L}(w) + \tfrac{1}{2}\,\text{tr}\big(G(w)\,\Sigma_\varepsilon(w)\big)} \tag{1}$$

**Diagonal form under unbiased rounding.** The random rounding scheme is coordinate-wise, so $\Sigma_\varepsilon(w) = \text{diag}(\sigma_1^2, \ldots, \sigma_d^2)$ with

$$\sigma_i^2 \;=\; s_{B(i)}^2\,\Delta_i\,(1-\Delta_i),$$

where $s_{B(i)}$ is the shared scale of the block $B(i)$ being rounded and $\Delta_i \in [0,1]$ is the fractional part of $w_i/s_{B(i)}$, the distance to the lower quantization bin after scaling (see B.2 for more details). Writing $g_{ii}(w)$ for the $i$th diagonal element of $G(w)$,

$$\mathcal{L}_{\text{GN}}(w) \;=\; \mathcal{L}(w) + \frac{1}{2}\sum_{i=1}^{d} g_{ii}(w)\,\sigma_i^2 \quad=\quad \mathcal{L}(w) + \frac{1}{2}\sum_{i=1}^{d} g_{ii}(w)\,s_{B(i)}^2\,\Delta_i\,(1-\Delta_i). \tag{2}$$

**Interpretation and optimization.** Equation 2 shows that randomized rounding injects an $\ell_2$-style *curvature-aware* ridge regularizer. Because the bound preserves every global minimizer of the original quantized objective (lemma 4), adding this term merely smooths the landscape; all optimal quantized solutions remain attainable. In practice, the diagonal terms of the Gauss-Newton component can be obtained by either using another backpropagation with sampled labels as done in Sophia [14] or we can use the empirical Fisher approximation by accumulating the square of the gradients observed in practice as done by Adam [13].

## 4. Experiments

In this section, we provide both synthetic and large language model experiments to compare the performance of our smoothed loss with analogous QAT and PTQ baselines. The synthetic experiments are lightweight and were run on A100s and H100s in less than an hour per run. Both the synthetic and language model experiments can run on any modern hardware, as all computations are done in FP32 with simulated weight quantization or rounding. The PTQ runs are trained end-to-end in FP32, and model checkpoints are naively clamped (standard quantization, see B.1) or rounded (stochastic rounding, see B.2) for evaluations. The QAT baseline simulates weight quantization in the forward pass, and Rounding Aware Training (RAT) simulates stochastic rounding. Both perform backward pass operations in full precision using the straight-through estimator. We train LOTION in full precision and round for evaluations.

### 4.1. Quadratic Loss

We begin with a linear regression toy problem where each input $x \in \mathbb{R}^d$ (with $d = 12000$) is sampled from a Gaussian distribution whose covariance follows a power-law spectrum ($\lambda_i \propto 1/i^{1.1}$ for $i = 1, ..., d$) that mimics the spectrum for Hessians observed in modern neural networks. The target is given by $w^{*\top} x$ for a predetermined $w^*$.

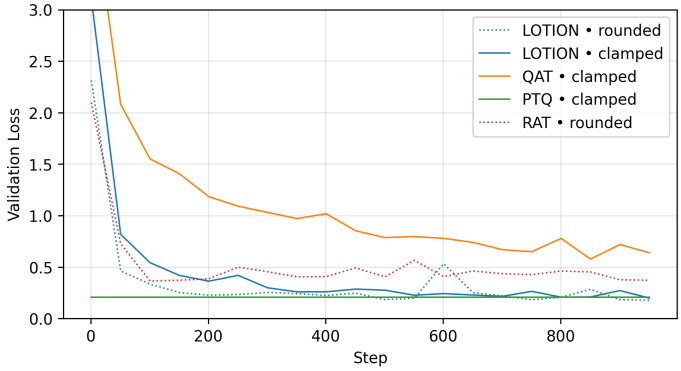
We compare quantized loss, obtained by directly casting or rounding weights, across all methods. As shown in Figure 1, LOTION outperforms both the PTQ and QAT methods in quantized validation loss. The behavior of QAT and RAT on the quantized validation loss is jagged and plateaus, while LOTION continues to decrease the quantized validation loss.

Additionally, we use this same setting to explore whether scaling model size can compensate for quantization noise, nullifying performance gaps between methods. We reproduce these experiments on a two-layer network, sweeping over the hidden dimension size. These results, provided in Figure C.1, show that LOTION maintains its performance gap.

### 4.2. Large Language Model

In addition to our toy-model experiments, we pretrain models at 150M (Figure 2) and 300M scales (Figure 5) to evaluate whether LOTION outperforms traditional QAT methods at realistic model sizes. We train and evaluate our models on C4 [21] using OLMo [11] following the hyperparameter settings of [26]. Both sets of models are trained with Chinchilla-optimal token budgets (20x as many tokens as model parameters) [12].

As in the linear regression setting, LOTION outperforms both QAT and PTQ baselines on quantized loss, especially at lower bit-width. We reproduce these results both at a larger model size (Figure 5) and with a larger training data budget (Figure 6), showing that LOTION continues




| Method | Val. loss |
|---|---|
| LOTION (round) | 0.1673 |
| LOTION (clamp) | 0.19834 |
| PTQ (clamp) | 0.20564 |
| RAT (round) | 0.37145 |
| QAT (clamp) | 0.52832 |

Figure 1: A comparison of INT4 quantized/rounded validation loss between LOTION, QAT, and PTQ, with summary table. The labels "clamp" and "round" denote whether weights are clamped or rounded for evaluation.

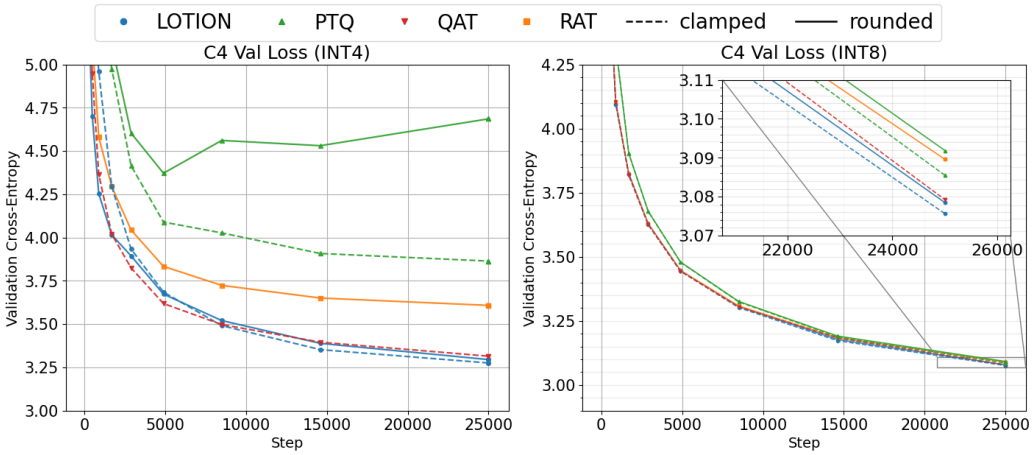



Figure 2: Quantized and rounded validation loss at INT4 (**Left**) and INT8 (**Right**) precision for LOTION, QAT, RAT and PTQ.

Figure 3: Final validation cross-entropy (150M)

| Method | Metric | INT4 | INT8 |
|---|---|---|---|
| PTQ | rounded | 4.686 | 3.092 |
| PTQ | clamped | 3.864 | 3.085 |
| RAT | rounded | 3.608 | 3.090 |
| QAT | clamped | 3.315 | 3.079 |
| LOTION | rounded | 3.295 | 3.078 |
| LOTION | clamped | **3.276** | **3.076** |

to decrease the quantized validation loss while QAT plateaus. Furthermore, we compare these methods when quantizing to FP4 precision in Figure 7. FP4 is generally favored over INT4 due to its

non-uniform quantization bin boundaries, allowing for superior inference accuracy [16]. We show that LOTION outperforms QAT even in this precision format.

## 5. Discussion and Limitations

With increasing scales of modern neural networks, low-precision execution is becoming a necessity to serve them on low-memory devices. Thus, devising better mechanisms for obtaining quantization-friendly networks is an important challenge. Previous methods such as QAT use straight-through estimators but do not provide guarantees of convergence.

In comparison, LOTION smooths the loss surface of quantized loss while preserving the global minima. This preserves the guarantees from traditional optimization literature about convergence to a stationary point on the smoothed loss surface. We believe that extending the current empirical results to real neural networks is an important research direction to verify the efficacy of LOTION in practical settings.

LOTION in particular uses stochastic rounding noise to help smooth the loss surface, resulting in an almost everywhere differentiable loss. However, the loss surface is still not completely smooth due to the undefined derivatives at the quantization bin boundaries. Using other noise distributions for obtaining a smooth loss surface while preserving the global minima property is an interesting research direction.

## References

[1] Chaim Baskin, Evgenii Zheltonozhkii, Tal Rozen, Natan Liss, Yoav Chai, Eli Schwartz, Raja Giryes, Alexander M. Bronstein, and Avi Mendelson. Nice: Noise injection and clamping estimation for neural network quantization. *Mathematics*, 9(17). ISSN 2227-7390. doi: 10.3390/math9172144. URL http://dx.doi.org/10.3390/math9172144.

[2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. URL https://arxiv.org/abs/1308.3432.

[3] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

[4] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale, 2022. URL https://arxiv.org/abs/2208.07339.

[5] Zhen Dong, Zhewei Yao, Amir Gholami, Michael Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision, 2019. URL https://arxiv.org/abs/1905.03696.

[6] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011. URL http://jmlr.org/papers/v12/duchi11a.html.

[7] John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. Randomized smoothing for stochastic optimization, 2012. URL https://arxiv.org/abs/1103.4296.

[8] Alexandre Défossez, Yossi Adi, and Gabriel Synnaeve. Differentiable model compression via pseudo quantization noise, 2022. URL https://arxiv.org/abs/2104.09987.

[9] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization, 2020. URL https://arxiv.org/abs/1902.08153.

[10] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks, 2019. URL https://arxiv.org/abs/1908.05033.

[11] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. Olmo: Accelerating the science of language models, 2024. URL https://arxiv.org/abs/2402.00838.

[12] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[14] Hong Liu, Zhiyuan Li, David Leo Wright Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3xHDeA8Noi.

[15] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm, 2018. URL https://arxiv.org/abs/1808.00278.

[16] Gunjan Mehta, Justin Xin, Riyad Islam, Yiheng Zhang, Asfiya Baig, Akhil Goel, and Sandro Cavallari. NVIDIA TensorRT unlocks FP4 image generation for NVIDIA Blackwell GeForce RTX 50 Series GPUs. *NVIDIA Technical Blog*, May 2025. URL https://developer.nvidia.com/blog/nvidia-tensorrt-unlocks-fp4-image-generation-for-nvidia-blackwell-geforce-rt

[17] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization, 2020. URL https://arxiv.org/abs/2004.10568.

[18] Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training, 2022. URL https://arxiv.org/abs/2203.11086.

[19] Yu Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, May 2005. ISSN 0025-5610. doi: 10.1007/s10107-004-0552-5. URL https://doi.org/10.1007/s10107-004-0552-5.

[20] Eunhyeok Park and Sungjoo Yoo. Profit: A novel training method for sub-4-bit mobilenet models, 2020. URL https://arxiv.org/abs/2008.04693.

[21] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[22] Charbel Sakr, Steve Dai, Rangharajan Venkatesan, Brian Zimmer, William J. Dally, and Brucek Khailany. Optimal clipping and magnitude-aware differentiation for improved quantization-aware training, 2022. URL https://arxiv.org/abs/2206.06501.

[23] Juncheol Shin, Junhyuk So, Sein Park, Seungyeop Kang, Sungjoo Yoo, and Eunhyeok Park. Nipq: Noise proxy-based integrated pseudo-quantization, 2023. URL https://arxiv.org/abs/2206.00820.

[24] Matteo Spallanzani, Gian Paolo Leonardi, and Luca Benini. Training quantised neural networks with ste variants: the additive noise annealing algorithm, 2022. URL https://arxiv.org/abs/2203.11323.

[25] Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley Osher, Yingyong Qi, and Jack Xin. Understanding straight-through estimator in training activation quantized neural nets, 2019. URL https://arxiv.org/abs/1903.05662.

[26] Rosie Zhao, Depen Morwani, David Brandfonbrener, Nikhil Vyas, and Sham Kakade. Deconstructing what makes a good optimizer for language models, 2025. URL https://arxiv.org/abs/2407.07972.

[27] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients, 2018. URL https://arxiv.org/abs/1606.06160.

## Appendix A. Related Work

**Quantization-Aware Training (QAT) vs Post-Training Quantization (PTQ).** Neural network quantization is typically performed either during training (QAT) or after training (PTQ). QAT methods often rely on the *straight-through estimator* (STE)—a heuristic gradient approximation introduced by Hinton and formalized by Bengio et al. [2]. STE enables gradient-based training by pretending the quantizer is the identity during backpropagation. Despite its popularity, STE lacks convergence guarantees and is known to cause gradient instability, especially at 4- or 2-bit precision. Numerous works attempt to mitigate this via gradient scaling, learned quantization scales [9], or progressive bit-reduction schedules [17]. PTQ methods, by contrast, quantize pretrained models post hoc and optimize auxiliary calibration objectives. Layer-wise curvature metrics, such as the Hessian trace [5], or sensitivity analyses are often used to allocate bits across layers. While PTQ avoids instability, it often underperforms QAT at very low precision.

**Loss Smoothing via Noise and Stochastic Rounding.** Another line of work introduces noise to smooth the quantized objective, sidestepping non-differentiability. This idea is rooted in classical techniques such as Nesterov smoothing [19] and randomized smoothing [6, 7]. In neural network quantization, additive noise has been used to simulate quantization effects during training [1, 8], with improved empirical stability. Recent methods extend this idea to fully differentiable quantization-aware training. For instance, NIPQ [23] replaces hard quantization with a noise proxy, allowing optimization over bit-widths and scales. However, NIPQ makes two limiting assumptions: (a) it uses only a scalar proxy (Hessian trace) for curvature, and (b) it relies on a fixed rounding scheme, limiting generality. Moreover, such proxies typically lack convergence guarantees and must be manually tuned.

**Quantization and Curvature-Awareness.** Several works relate quantization error to curvature of the loss landscape. For instance, HAWQ [5] and its successors use the Hessian spectrum to guide bit allocation. Other approaches estimate layer sensitivity using Hutchinson-based Hessian trace approximations [27]. Some training-time regularization methods penalize sharp minima [18], promoting flatness to better tolerate quantization. These methods highlight the importance of curvature in quantization-aware optimization, but often rely on global heuristics or require non-standard solvers.

**Our Approach: LOTION.** LOTION introduces a loss-level smoothing framework that avoids heuristic gradient modifications. Instead of backpropagating through a non-differentiable cast function, LOTION directly optimizes the *expected quantized loss* under unbiased stochastic rounding noise. This yields a smooth, differentiable objective that preserves all global minima of the original quantized loss and supports a broad range of rounding schemes. Unlike NIPQ, which uses a scalar proxy, LOTION derives a curvature-aware regularizer via a diagonal Gauss–Newton approximation, making explicit the connection between quantization noise and second-order structure. This principled formulation provides both empirical stability and theoretical guarantees absent from prior methods.

## Appendix B. Quantization formats

### B.1. Fine-Grained Shared-Scale Integer Quantization

We study the standard symmetric signed integer quantization method in which parameters are scaled into the desired dynamic range and partitioned into uniform blocks as in the absolute maximum quantization method described in LLM.INT8() [4]. We prefer absolute maximum quantization because it prevents overflow and avoids the computation and memory burden of methods like zero-point asymmetric schemes that are particularly costly for weight-only quantization. Absolute max quantization is often use in practice, notably in the FP8 "fine-grained shared-scale" format adopted by DEEPSEEK [3]. Parameters are partitioned into blocks $B$ (possibly as small as a single element). For each input block $w_i$ we store one floating-point (FP16) scale $s_B$ and an $n$-bit integer tensor $z$ defined by

$$s_B \;=\; \frac{\max_{i\in B}|w_i|}{2^{n-1}-1}, \qquad z_i \;=\; \left\lfloor \frac{w_i}{s_B} \right\rceil, \qquad [\mathrm{cast}(w)]_i \;=\; s_B\, z_i \quad (i\in B).$$

Because $|w_i| \le (2^{n-1}-1)\, s_B$ by construction, the rounded values satisfy $z_i \in [-(2^{n-1}-1), 2^{n-1}-1]$ and thus lie safely within the representable range; no explicit clipping step is required.

### B.2. Randomized Rounding

To define a general notion of randomized rounding, let us first denote the support of the cast function as $Q$, i.e, $Q = \{x \mid \exists w \text{ s.t. } \mathrm{cast}(w) = x\}$. Also, let us define a notion of distance between two finite sets $(S_1, S_2)$ of points in $\mathbb{R}^d$ as $d(S_1, S_2) = \max\{\|w_1 - w_2\|_2 | w_1 \in S_1, w_2 \in S_2\}$.

**Definition 2 (Randomized Rounding)** *Randomized Rounding (RR) is a function from $\mathbb{R}^d \to \mathbb{P}[Q]$ that satisfies the following three properties:*

1. $\forall w \in \mathbb{R}^d, \mathbb{E}_{q\sim RR(w)}[q] = w$
2. *$RR$ is continuous and locally bounded [1] (with $RR(w)$ having a finite support), where the continuity is defined with respect to $\mathcal{W}_2$ distance on $\mathbb{P}[Q]$ and $L_2$ distance on $\mathbb{R}^d$.*
3. *$\forall w \in Q$ which satisfy $cast(w) = w$, $RR(w)$ has probability $1$ on $w$.*

Smoothed loss defined with respect to randomized rounding satisfies some nice properties.

**Lemma 3** *For any loss function $L(w)$ which is continuous w.r.t $L_2$ norm and any $f : \mathbb{R}^d \to \mathbb{P}[Q]$ satisfying the 2nd axiom above, $\mathbb{E}_{q\sim f(w)}[L(q)]$ is also continuous w.r.t the $L_2$ norm.*

**Lemma 4** *For any $f : \mathbb{R}^d \to \mathbb{P}[Q]$ which satisfies the 3rd axiom above,*

$$min_{w\in\mathbb{R}^d}\mathbb{E}_{q\sim f(w)}[L(q)] = min_{w\in\mathbb{R}^d} L(cast(w))$$

The above two lemmas combined show that $\mathbb{E}_{q\sim RR(w)}[L(q)]$ is a continuous function whose global minima matches the global minima of the quantized loss function. Thus, we have a better optimizable function, which does not impact the global minimum of the loss surface.

1. Locally bounded means that for any compact set $D \subset \mathbb{R}^d$, $d(\mathrm{spt}(\mathrm{RR}(D)), \{0\})$ is finite

EXAMPLE: SHARED-SCALE INTEGER ROUNDING

We provide an example of a randomized rounding scheme corresponding to the casting function defined in Section B.1. Consider a scalar $z_i'$ as defined below:

$$z_i' \;=\; \tfrac{w_i}{s_B}.$$

The randomized rounding scheme is defined as below:

$$RR(w) = \begin{cases} s_B z_i' \text{ if } z_i = z_i' \\ s_B \lfloor z_i' \rfloor \text{ w.p. } \lceil z_i' \rceil - z_i' \\ s_B \lceil z_i' \rceil \text{ w.p. } z_i' - \lfloor z_i' \rfloor \end{cases}$$

We use this rounding scheme for LOTION with integer formats.

### B.3. Warm-Up: Quadratic Losses

To visualize how we incorporate randomized rounding into LOTION, we first consider a setting where the (population) loss is quadratic

$$\mathcal{L}(w) \;=\; \tfrac{1}{2}\,(w - w^\star)^\top H\,(w - w^\star) + C, \qquad H \succeq 0,$$

e.g. $\mathcal{L}(w) = \frac{1}{2}\|Aw - b\|_2^2$ with $H = A^\top A$ and $C = \frac{1}{2}\|b\|_2^2 - \frac{1}{2}\|Aw^\star\|_2^2$. Let $\varepsilon$ denote the *randomized-rounding noise*, so $\mathrm{RR}(w) = w + \varepsilon$ with $\mathbb{E}[\varepsilon] = 0$ and $\Sigma_\varepsilon := \mathrm{Cov}[\varepsilon]$. Because the stochastic rounding noise is zero-mean, expanding the quadratic and taking expectations gives a closed form:

$$\boxed{\mathcal{L}_{\text{smooth}}(w) \;=\; \mathbb{E}\big[\mathcal{L}(w+\varepsilon)\big] \;=\; \mathcal{L}(w) + \tfrac{1}{2}\,\mathrm{tr}\big(H\,\Sigma_\varepsilon\big)} \tag{3}$$

**Covariance of unbiased randomized rounding.** For the fine-grained shared-scale rule in Section B.1, each coordinate $i$ (belonging to block $B(i)$ with scale $s_B$) is rounded independently:

$$z_i' \;=\; \frac{w_i}{s_B}, \quad \varepsilon_i \;=\; s_B\big(R_i - z_i'\big), \quad R_i = \begin{cases} \lfloor z_i' \rfloor & \text{w.p. } \lceil z_i' \rceil - z_i', \\ \lceil z_i' \rceil & \text{w.p. } z_i' - \lfloor z_i' \rfloor. \end{cases}$$

Writing $\Delta_i := z_i' - \lfloor z_i' \rfloor \in [0,1]$,

$$\mathrm{Var}[\varepsilon_i] \;=\; s_B^2\,\Delta_i\,(1-\Delta_i) \;\;\le\;\; \tfrac{1}{4}\,s_B^2.$$

Since distinct coordinates round independently, $\Sigma_\varepsilon = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$ with $\sigma_i^2 = s_{B(i)}^2 \Delta_i(1-\Delta_i)$.

**Interpretation: An Implied Regularizer** Plugging this diagonal covariance into equation 3 yields an $\ell_2$-style *ridge* term:

$$\mathcal{L}_{\text{smooth}}(w) \;=\; \mathcal{L}(w) \;+\; \tfrac{1}{2}\sum_{i=1}^{d} H_{ii}\, s_{B(i)}^2\, \Delta_i(1-\Delta_i).$$

Thus, randomized rounding *exactly* adds a data-dependent diagonal regularizer whose strength is dependent on the curvature of the hessian and the expected rounding error in the given coordinate. This makes the smoothed objective strictly smoother than the original yet preserves all global minima (See lemma 3 and 4).

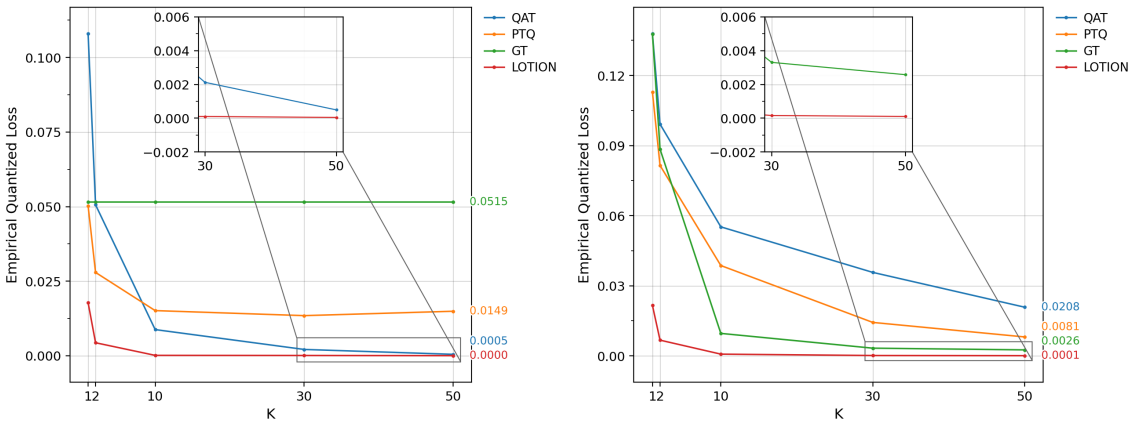



Figure 4: Quantized validation loss (**Left**) and rounded validation loss (**Right**) as a function of the hidden dimension, $K$, of a two layer linear network for LOTION, QAT, GT, and PTQ.

## Appendix C. Additional Synthetic Experiments

### C.1. Linear Network

In this section, we use our toy model to explore whether scaling the model can compensate for quantization noise, thereby eliminating meaningful performance differences between methods. We conduct experiments on a two-layer linear network given by

$$f(x) = \frac{1}{k} W_2 W_1 x$$

where $W_2 \in \mathbb{R}^{1\times k}, W_1 \in \mathbb{R}^{k\times d}$ and $x \in \mathbb{R}^d$. We again consider the input distribution to be Gaussian with $d = 12000$ with a power-law decaying spectrum given by $\lambda_i \propto \frac{1}{i^{1.1}}$. The targets are given by $y = {w^*}^\top x$, for $w^*$ sampled from a Gaussian distribution. The following lemma holds for the above network as $k \to \infty$.

**Lemma 5** *For the uniform INT fine-grained, shared-scale quantization scheme, the quantized loss for $f(x)$ goes to 0 as $k \to \infty$.*

The above lemma holds as all the elements of the outputs $W_2$ can be set to 1 and each row of the first layer $W_1$ can be equal to a stochastically rounded $w^*$. We provide a proof for this lemma in Appendix A.

We expect that as $K$ grows, the smoothed loss and the quantized loss for various methods will decrease. We plot these results in Figure 4. For each value of $K$, we plot the lowest quantized loss achieved at this model size. PTQ_ROUND and PTQ_CLAMP are models that are trained in full precision and rounded or clamped to measure quantized loss. GT_ROUND and GT_CLAMP are initialized from models where all elements of $W_2$ are set to 1 and where the rows of $W_1$ are w*. These models are then rounded for GT_ROUND or clamped for GT_CLAMP. Following lemma 5, GT_ROUND's quantized loss goes to 0 as $k \to \infty$. As shown in the figure, LOTION outperforms all methods in quantized loss even as we scale up model size.

## Appendix D. Additional Language Model Experiments

### D.1. 300M parameter model

In addition to the 150M parameter model sweep, we train 300M parameter models and similarly evaluate validation loss after quantizing with LOTION and our various baselines. As with the 150M models, LOTION outperforms QAT and PTQ at both INT4 and INT8 precision.

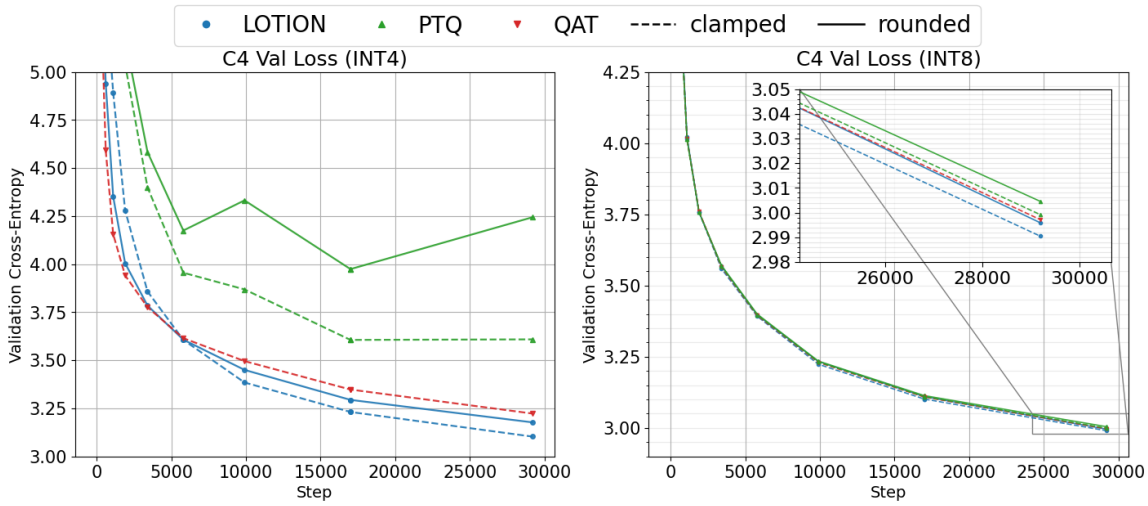



Figure 5: Quantized and rounded validation loss at INT4 (**Left**) and INT8 (**Right**) precision for LOTION, QAT, and PTQ.

Table 1: Final validation cross-entropy (300M)

| Method | Metric | INT4 | INT8 |
|---|---|---|---|
| PTQ | rounded | 3.9745 | 3.0045 |
| PTQ | clamped | 3.6062 | 2.9992 |
| QAT | clamped | 3.2230 | 2.9972 |
| LOTION | rounded | 3.1772 | 2.9959 |
| LOTION | clamped | **3.1031** | **2.9905** |

### D.2. Overtrained models

In figure 6, we also include training runs for our 150M parameter model trained to 5x chinchilla (a token budget that is 100x larger than the number of parameters) to validate that LOTION continues to decrease the quantized loss while QAT plateaus.

### D.3. FP4

Quantization to FP4 is generally favored over INT4 because of its non-uniform quantization scheme that can represent small values while accounting for rare large outliers. This increased precision

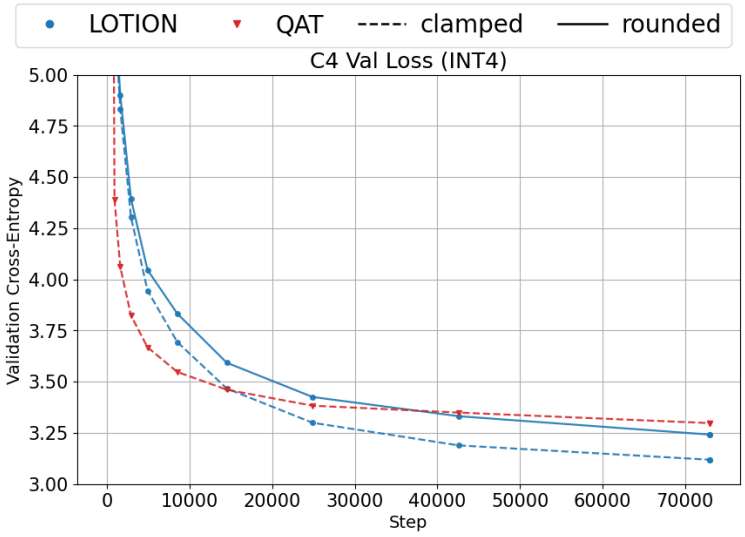



Figure 6: Quantized and rounded validation loss at INT4 (**Left**) precision for LOTION, QAT, and PTQ.

for smaller values tends to lead to lower overall quantization error and higher inference accuracy. In figure 7, we validate that LOTION grants performance gains over QAT even with this modern precision format.

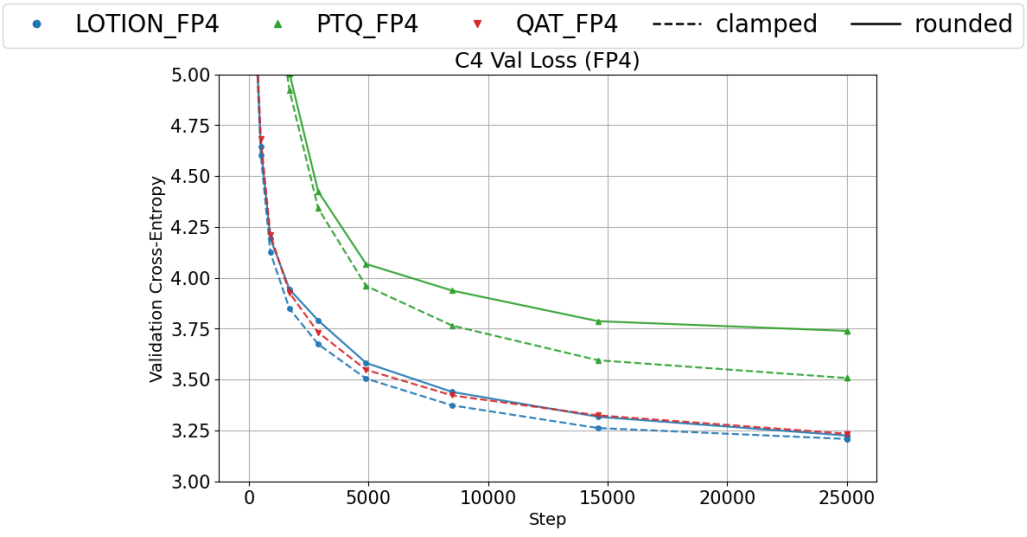



Figure 7: Quantized and rounded validation loss at FP4 (**Left**) precision for LOTION, QAT, and PTQ.