Understanding Clipping in Zeroth-order Optimization

Saket Gollapudi SAKET312@CS.WASHINGTON.EDU
University of Washington, Seattle, Washington, USA

Elisa Bertino BERTINO@PURDUE.EDU
Purdue University, West Lafayette, Indiana, USA

Sewoong Oh SEWOONG@CS.WASHINGTON.EDU
University of Washington, Seattle, Washington, USA

Abstract

Zeroth-order optimization has emerged as a resource-efficient alternative to SGD when fine-tuning LLMs, thanks to the inherent low dimensionality of the loss landscape. It relies on two-point estimate of the gradient that only accesses the function value and not the gradient, bypassing resource heavy backpropagation. Recently, it was accidentally discovered that per-sample clipping can improve performance of the trained model, while running differentially private version of zeroth-order optimization [43] with very large privacy parameter ϵ (corresponding to very little privacy). In this paper, we systematically investigate this phenomenon and demonstrate that (i) the optimal choice of learning rate η is inversely proportional to the clipping threshold c, satisfying $\eta c =$ constant; (ii) clipping drives the zeroth-order optimization to a different solution with higher test accuracy; and (iii) this phenomenon is only observed with per-sample clipped zeroth-order optimization and not other zeroth-order alternatives that also provide robust estimates of the gradient.

1. Introduction

Zeroth-Order (ZO) optimization methods allow one to minimize an objective function without access to its gradients. This is necessary (i) when the learner has access to the function evaluation only, as in finding adversarial examples for a black-box model with API access [8, 24, 37], or (ii) when computing the gradient is prohibitively expensive, as in back-propagation on a model with tens of billions of parameters [27, 43]. Compared to First-Order (FO) methods that use gradient information to iteratively update the model, a major bottleneck in adopting ZO optimization in practice is the dimension-dependent convergence rate. ZO methods typically search over all dimensions, wasting a lot of iterations on unnecessary directions with small improvements, resulting in a factor of d slower convergence compared to FO counterparts for d-dimensional optimization problems [13].

The first practical breakthrough, therefore, of ZO methods came in a relatively lower dimensional problem of adversarial attacks, where the optimization is over the input pixels of, say, the Inception-v3 network [35]: $d = 299 \times 299 \times 3 = 268,203$ [8]. Empirically, ZO methods proved to be successful in finding adversarial examples for a black-box model where we only have API access to the function output [41]. The next breakthrough in applying ZO methods came in LLM fine-tuning [27, 43]. [27] theoretically showed that the convergence of ZO methods only scales as the effective rank of the Hessian matrix instead of the number of parameters to be optimized. Empirical evidence suggests that in LLM fine-tuning, the effective rank is significantly smaller than the number of parameters. Since, ZO methods do not require back-propagation, which is memory heavy, this makes ZO fine-tuning a memory-efficient alternative for LLM fine-tuning.

© S. Gollapudi, E. Bertino & S. Oh.

Consider fine-tuning model weights $w \in \mathbb{R}^d$ given n samples, $\min_{x\in\mathbb{R}^d}(1/n)\sum_{i=1}^n f_i(w)$, where f_i is the loss on the i-th sample. Given access to a FO oracle, Stochastic Gradient Descent (SGD) updates the model as per mini-batch gradient $g(w_t) = (1/|B_t|)\sum_{i\in B_t}\nabla f_i(w_t)$ with learning rate η: $w_{t+1} = w_t - \eta\, g(w_t)$. *Zeroth-Order SGD (ZO-SGD)* only accesses the ZO oracle that can compute the function value, and uses a two-point estimate of the mini-batch gradient:

$$g_\lambda(w_t) = \frac{1}{|B_t|}\sum_{i\in B_t}\frac{f_i(w_t+\lambda u_t) - f_i(w_t-\lambda u_t)}{2\lambda}u_t\ , \tag{1}$$

where $\lambda > 0$ is the smoothing parameter and $u_t \in \mathbb{R}^d$ is a vector with i.i.d. standard Gaussian entries. This approach can also be applied to SVRG and momentum methods [9, 23]. While empirically measuring the performance of ZO-SGD under differential privacy, [43] first reported that adding *per-sample* clipping to ZO-SGD, which we call *clipped ZO-SGD*, can improve performance of LLM fine-tuning:

$$g_{\lambda,c}(w_t) = \frac{1}{|B_t|}\sum_{i\in B_t}\mathrm{Clip}_c\Big(\frac{f_i(w_t+\lambda u_t) - f_i(w_t-\lambda u_t)}{2\lambda}\Big)u_t\ , \tag{2}$$

where $\mathrm{Clip}_c : \mathbb{R}\to\mathbb{R}$ is defined as $\mathrm{Clip}_c(a) = a$ if $|a|\le c$ and $\mathrm{Clip}_c(a) = \mathrm{sign}(a)\cdot c$ otherwise.

In this paper, we systematically investigate the gain of clipping in ZO-SGD in four fine-tuning tasks of SNLI [5], MNLI [39], SST-2 [32], and Trec [21] on two pretrained base models of RoBERTa [26] and OPT [45]. In all these settings, we identify a *clipping trade-off*: optimal choice of the learning rate inversely scales with the clipping threshold c. More importantly, we demonstrate that the pairs of (η, c) spanning this optimal trade-off are not functionally identical; the models fine-tuned with enough clipping achieve higher train loss but better test accuracy. This indicates that clipping drives ZO-SGD to a different basin that generalizes better. We further demonstrate this by comparing the prediction pattern between two models (Figure 3) and comparing the ensemble model accuracy (Figure 4). Adding other forms of robust mean estimator to ZO-SGD, we demonstrate in Figure 5 that the above phenomenon is only present in clipped ZO-SGD of Eq. (2). Further investigation reveals that clipping drives ZO-SGD to be significantly miscalibrated (Figure 6). This suggests that the higher accuracy of clipped ZO-SGD is related to how it focuses on a (large) subset of data that it is confident on and neglects a small subset of data that it is not confident on, adaptively.

2. Trade-off between clipping threshold and learning rate

A survey of related work is provided in App. A, a detailed experimental set-up in App. C.

The *clipping trade-off* at $\eta\, c =$ constant. To understand how the optimal choice of the learning rate η scales with the clipping threshold c, we sweep through a range of parameters and measure the test accuracy averaged over four fine-tuned models trained on four separate tasks (Figure 1, left). We make two important observations: (i) the optimal choice of the learning rate $\eta^*(c)$ for each clipping threshold c lies on the line satisfying $\eta c = 0.0005$ highlighted by the dark blue cells in the diagonal excluding $c = \infty$; and (ii) the best average test accuracy of 0.8636 is achieved with clipping, in this case with $c = 50$. This phenomenon, which we call the *clipping trade-off*, is quite robust. The same behavior is demonstrated on each individual fine-tuning task (Figures 8-14) and across two model families (OPT family in Figure 16) in LM fine-tuning.

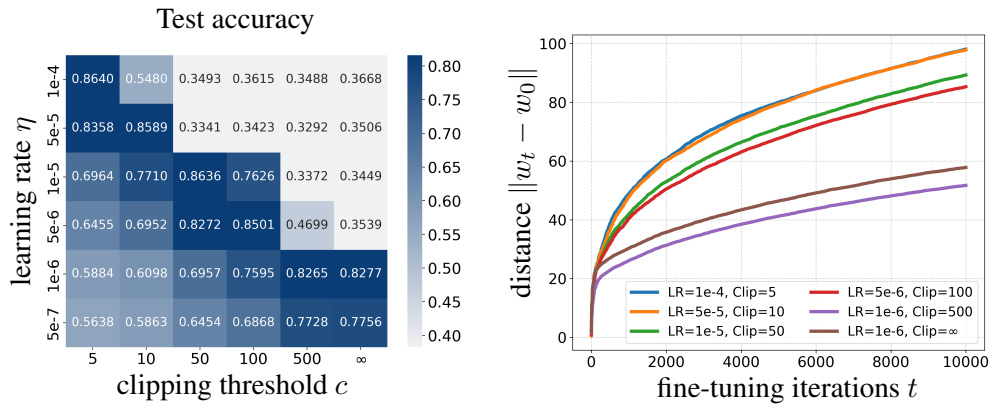

Figure 1: (left) Sweeping through the learning rate η and the clipping threshold c, we observe that the optimal learning rate for each clipping threshold lies on the curve $\eta \times c = 5 \times 10^{-4}$. Each cell shows the test accuracy averaged over four models fine-tuned from RoBERTa [26] on SNLI [5], MNLI [39], SST-2 [32] and Trec [21], respectively. $c = \infty$ is the no clipping baseline. (right) For those diagonal cells, tracking how fast SNLI fine-tuned weights w_t move away from the pre-trained checkpoint of RoBERTa w_0, we observe two distinct behaviors depending on the clipping threshold: $c \leq 100$ and $c \geq 500$.

One explanation of this inverse relation might be that the clipping in Eq. (2) is effectively scaling down the gradient estimate g_λ in Eq. (1) and the learning rates need to compensate for that in order to reach the similar optimal point. This would mean that models trained with the same effective speed of $\eta\, c = 5 \times 10^{-4}$ should be similar. In Figure 1 (right), we track the Euclidean distance from the initial pre-trained checkpoint: $\|w_t - w_0\|$ from Eq. (2). This reveals two distinct paths, possibly towards two distinct basins, indicating that clipping is not just rescaling the gradient estimates but rather driving the model towards potentially a better minima, one that can only be reached with clipping. This two basin hypothesis is further supported by the fact that clipping results in higher test loss (Figure 2 left) but better test accuracy (Figure 1 left). This trend also holds for train loss and train accuracy, as shown in Figure 7 in the appendix, suggesting that the gain of clipping comes from an implicit bias towards a better minima and not better optimization.

2.1. Two types of solutions along the clipping trade-off line

Previous experiments suggest that hyperparameter choices along the *clipping trade-off line* (i.e., $\eta c = 5 \times 10^{-4}$) converge to two different types of solutions depending on whether clipping is significant enough ($c \leq 100$). To visualize the geometry of these solutions similar to [15, 20], we project the high-dimensional test loss and test accuracy landscapes onto the 2D plane spanned by three model checkpoints along this line: w_1 ($\eta = 5e{-}5$, $c = 10$), w_2 ($\eta = 1e{-}5$, $c = 50$), and w_3 ($\eta = 1e{-}6$, $c = \infty$).

The train loss landscape (Figure 3, left) clearly shows how clipped ZO-SGD (w_1 and w_2) converges to points with higher loss compared to the unclipped baseline (w_3). Symmetrically, the test accuracy landscape (Figure 3, right) illustrates that w_1 and w_2 occupy a shared region of high accuracy, while w_3 is in a lower-accuracy region. Another way to measure how two models are different is the ensemble model [3, 29, 40]. Adding the logit output of two models and using the softmax to predict, known as an ensemble model, has been observed to give better performance than the

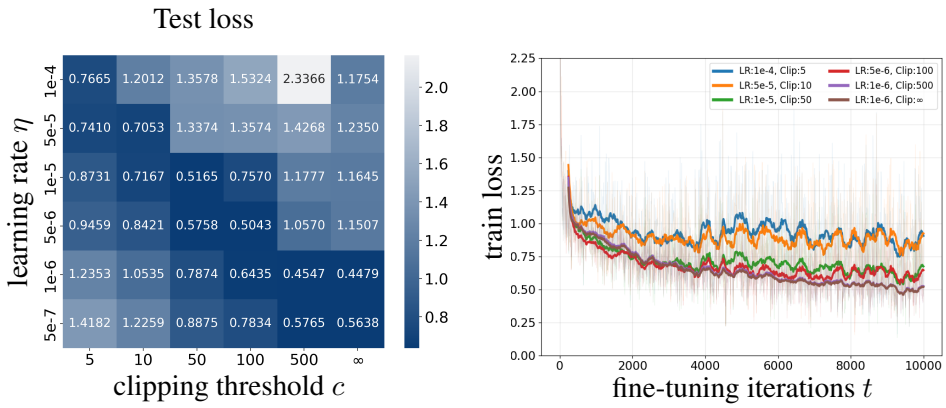

Figure 2: The same setting as Figure 1 but showing the average test loss over four models (SNLI, SST-2, MNLI, Trec) fine-tuned from RoBERTa (left) and the SNLI train loss of the configurations along the line $\eta c = 5 \times 10^{-4}$ (right). The higher loss achieved along the diagonal with clipping ($c \leq 100$) compared to the no-clipping best loss implies that the gain of clipping is from better generalization and not better optimization.

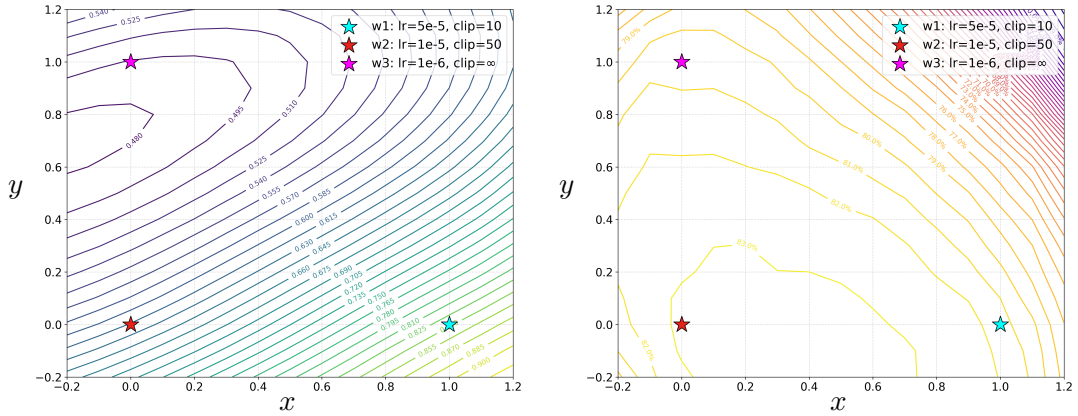

Figure 3: (Left) Train loss landscape projected down to (x, y)-plane evaluated at $w_2 + (w_1 - w_2)x + (w_3 - w_2)y$ for three model checkpoints w_1, w_2, w_3. (Right) Test Accuracy landscape projected down to (x, y)-plane evaluated at $w_2 + (w_1 - w_2)x + (w_3 - w_2)y$ for three model checkpoints w_1, w_2, w_3.

two original models. Figure 4 demonstrates how the ensemble of two clipped models w_1 and w_2 provides no gain (middle) while the ensemble of a clipped model w_1 and an unclipped model w_3 exploits the diversity between the two original models to get a performance gain (left). Figure 4 (right) shows that such a diversity can also be exploited by training two clipped models with different random seeds.

2.2. Connections to robust gradient estimation

One natural conjecture on why the clipped method converges to a better basin is that it provides a *robust estimate* of the gradient, in which case other robust gradient estimators might achieve a similar gain. We test this hypothesis on the following two robust estimators.

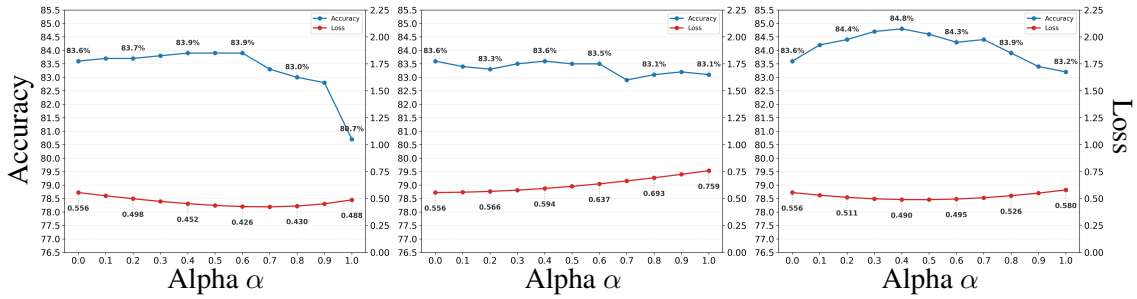

Figure 4: For models w_1, w_2, and w_3 from Figure 3, ensemble model of $\mathrm{SoftMax}((1-\alpha)\, h(x; w_2) + \alpha\, h(x; w_3))$, where h is the logit output, achieves better SNLI accuracy than the two original models (left) but two clipped models, w_2 and w_1, lack diversity and do not show ensemble gain (middle). Two clipped models can still get ensemble gain, if they use two different random seeds (right).

ZO optimization with *batch-clipping* applies clipping to the average gradient:

$$g_{\lambda,\bar{c}}(w_t) = \mathrm{Clip}_{\bar{c}}\Big(\frac{1}{|B_t|}\sum_{i\in B_t}\frac{f_i(w_t+\lambda u_t) - f_i(w_t-\lambda u_t)}{2\lambda}u_t\Big)\,. \tag{3}$$

When applied to the standard first-order SGD, batch-clipping has been empirically shown to achieve better generalization by converging to a wider and flatter basin for FashionMNIST, SVHN, and CIFAR10, when the noise in the gradient is heavy-tailed [38]. Their theoretical analysis shows that batch-clipping makes SGD with heavy-tailed noise exponentially slower to escape wider basins. Further, [42] showed that this batch-clipped SGD can achieve faster convergence under (L_0, L_1)-smoothness, where smoothness is proportional to the gradient norm: $\|\nabla^2\mathcal{L}(w)\| \le L_0 + L_1\|\nabla\mathcal{L}(w)\|$. It was further empirically demonstrated that the training loss of LSTM on Penn Treebank and ResNet20 on CIFAR 10 exhibit (L_0, L_1)-smoothness. In our case of ZO optimization, however, Figure 5 (left) shows that batch-clipping cannot drive the optimization to the better basin, and hence cannot achieve SNLI test accuracy above 82%.

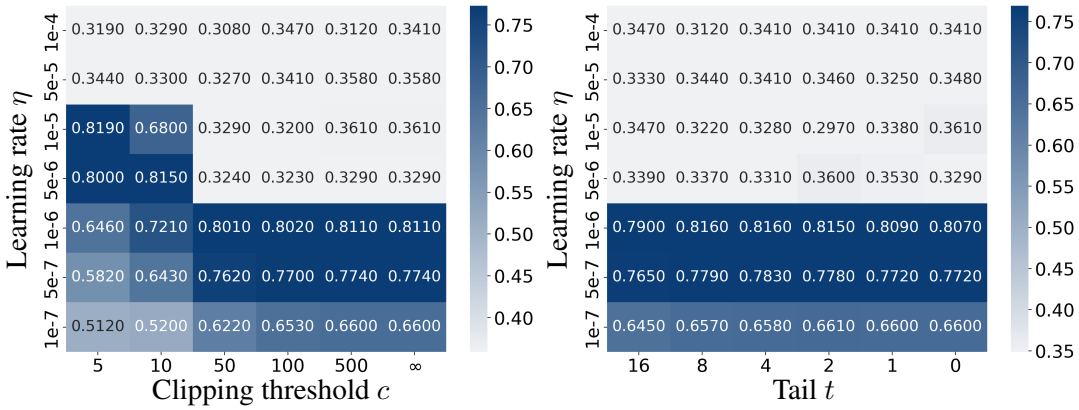

Figure 5: Repeating the same experiments from Figure 1 but with two other robust gradient estimators, $g_{\lambda,\bar{c}}$ in Eq. (3) (left) and $g_{\lambda,k}$ in Eq. (4) (right), demonstrates that clipping trade-off phenomenon only happens with the clipped ZO-SGD of Eq. (2).

Winsorized mean is a 1D robust mean estimator which adaptively clips the data according to k-th order statistics. We apply this to zeroth-order optimization:

$$g_{\lambda,k}(w_t) = \frac{1}{|B_t|} \sum_{i \in B_t} \mathrm{Clip}_{[L_t, U_t]}\Big(\frac{f_i(w_t + \lambda u_t) - f_i(w_t - \lambda u_t)}{2\lambda}\Big) u_t \ , \tag{4}$$

where L_t and U_t are the $(k+1)$-th and $(|B_t| - k)$-th order statistics of the mini-batch scalar finite differences, and $\mathrm{Clip}_{[L_t,U_t]}(a) = \max(L_t, \min(a, U_t))$. Similar to clipping outside the mean, Figure 5 (right) shows that Winsorized mean cannot achieve the test accuracy of clipped ZO-SGD.

2.3. Connections to generalization: calibration

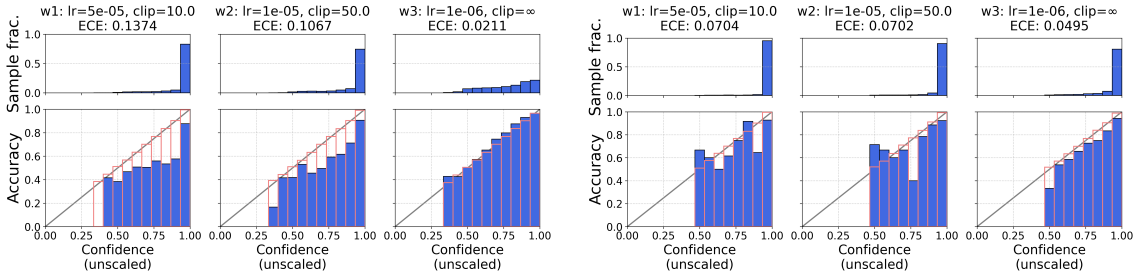

Figure 6: Model Calibration on SNLI (left) and SST-2 (right). Reliability diagrams comparing clipped models (w_1, w_2) to an unclipped baseline (w_3). The clipped models consistently display overconfidence (blue bars below the diagonal) and higher Expected Calibration Error (ECE).

To better understand the structural properties of these solutions, we analyze the models' reliability diagrams on SNLI and SST-2 in Figure 6. This suggests that clipping drives the model to a minima that is miscalibrated with higher loss but better accuracy. This is similar to other highly accurate or strongly regularized modern networks that tend toward systematic overconfidence [28]. When comparing configurations along the ηc boundary, the predictive behaviors diverge sharply. The unclipped baseline (w_3) remains well-calibrated (SNLI ECE: 0.0211), distributing its confidence mass more evenly across intermediate bin, accurately reflecting the model's true performance. Conversely, models trained with strict clipping (w_1, w_2) exhibit much higher ECE scores (0.1374 and 0.1067). Tight clipping concentrates the vast majority of predictions into the top confidence bin (> 0.9). While standard ZO-SGD finds minima with calibrated confidence, clipping drives the optimization into a fundamentally different basin. Ultimately, clipping trades ideal uncertainty calibration to discover a minima that rely on highly saturated feature activations.

Acknowledgment

This work is supported by NSF awards 2112471, 2229876, 2505865, and 2502281.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Kwangjun Ahn, Jingzhao Zhang, and Suvrit Sra. Understanding the unstable convergence of gradient descent. In *International Conference on Machine Learning*, 2022.

[3] Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2023.

[4] S Arora, Z Li, and A Panigrahi. Understanding gradient descent on the edge of stability in deep learning. In *International Conference on Machine Learning*, 2022.

[5] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, 2015.

[6] Aochuan Chen, Yimeng Zhang, Jinghan Jia, James Diffenderfer, Konstantinos Parasyris, Jiancheng Liu, Yihua Zhang, Zheng Zhang, Bhavya Kailkhura, and Sijia Liu. Deepzero: Scaling up zeroth-order optimization for deep model training. In *International Conference on Learning Representations*, volume 2024, pages 50185–50206, 2024.

[7] Jiahe Chen and Ziye Ma. Accelerating zeroth-order spectral optimization with partial orthogonalization from power iteration. *arXiv preprint arXiv:2605.09034*, 2026.

[8] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

[9] Xiangyi Chen, Sijia Liu, Kaidi Xu, Xingguo Li, Xue Lin, Mingyi Hong, and David Cox. Zo-adamm: Zeroth-order adaptive momentum method for black-box optimization. *Advances in neural information processing systems*, 32, 2019.

[10] Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private SGD: A geometric perspective. In *Advances in Neural Information Processing Systems*, volume 33, pages 13773–13782, 2020.

[11] Jeremy M Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021.

[12] A Damian, E Nichani, and JD Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *International Conference on Learning Representations*, 2023.

[13] John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

[14] Huang Fang, Xiaoyun Li, Chenglin Fan, and Ping Li. Improved convergence of differential private SGD with gradient clipping. In *International Conference on Learning Representations*, 2023.

[15] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in neural information processing systems*, volume 31, 2018.

[16] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[17] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International conference on machine learning*, pages 2137–2146. PMLR, 2018.

[18] Anastasia Koloskova, Hadrien Hendrikx, and Sebastian U Stich. Revisiting gradient clipping: Stochastic bias and tight convergence guarantees. In *International Conference on Machine Learning*, 2023.

[19] Yicheng Lang, Changsheng Wang, Yihua Zhang, Mingyi Hong, Zheng Zhang, Wotao Yin, and Sijia Liu. Powering up zeroth-order training via subspace gradient orthogonalization. *arXiv preprint arXiv:2602.17155*, 2026.

[20] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in neural information processing systems*, volume 31, 2018.

[21] Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

[22] Tianyi Lin, Zeyu Zheng, and Michael Jordan. Gradient-free methods for deterministic and stochastic nonsmooth nonconvex optimization. *Advances in Neural Information Processing Systems*, 35:26160–26175, 2022.

[23] Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. *Advances in neural information processing systems*, 31, 2018.

[24] Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. A primer on zeroth-order optimization in signal processing and machine learning. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020.

[25] Sijia Liu, Yicheng Lang, Soumyadeep Pal, Changsheng Wang, Yancheng Huang, Chongyu Fan, James Diffenderfer, Bhavya Kailkhura, and Yihua Zhang. Position: Zeroth-order optimization in deep learning is underexplored, not underpowered. *arXiv preprint arXiv:2605.15622*, 2026.

[26] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[27] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. In *Advances in Neural Information Processing Systems*, volume 36, pages 53038–53075, 2023.

[28] Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Ann Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lučić. Revisiting the calibration of modern neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 15682–15694. Curran Associates, Inc., 2021.

[29] Anshul Nasery, Jonathan Hayase, Pang Wei Koh, and Sewoong Oh. Pleas—merging models with permutations and least squares. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 30493–30502. IEEE, 2025.

[30] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

[31] Ohad Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research*, 18(1):1703–1713, 2017.

[32] Richard Socher, Alex Perelygina, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[33] Minhak Song, Liang Zhang, Bingcong Li, Niao He, Michael Muehlebach, and Sewoong Oh. Zeroth-order optimization at the edge of stability. *arXiv preprint arXiv:2604.14669*, 2026.

[34] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855. PMLR, 2022.

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[36] Liyan Tan, Yequan Zhao, Yifan Yang, Ruijie Zhang, Xinling Yu, and Zheng Zhang. Grzo: Group-relative zeroth-order optimization for large language model fine-tuning. *arXiv preprint arXiv:2606.02857*, 2026.

[37] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 742–749, 2019.

[38] Xingyu Wang, Sewoong Oh, and Chang-Han Rhee. Eliminating sharp minima from sgd with truncated heavy-tailed noise. In *International Conference on Learning Representations*, 2022.

[39] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, 2018.

[40] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022.

[41] Haishan Ye, Zhichao Huang, Cong Fang, Chris Junchi Li, and Tong Zhang. Hessian-aware zeroth-order optimization for black-box adversarial attack. *arXiv preprint arXiv:1812.11377*, 2018.

[42] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020.

[43] Liang Zhang, Bingcong Li, Kiran Koshy Thekumparampil, Sewoong Oh, and Niao He. DPZERO: Private fine-tuning of language models without backpropagation. In *International Conference on Machine Learning (ICML)*, 2024.

[44] Liang Zhang, Bingcong Li, Kiran Koshy Thekumparampil, Sewoong Oh, Michael Muehlebach, and Niao He. Zeroth-order optimization finds flat minima, 2025. URL https://arxiv.org/abs/2506.05454.

[45] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[46] Y Zhang, P Li, J Hong, J Li, Y Zhang, W Zheng, P-Y Chen, JD Lee, W Yin, M Hong, et al. Revisiting zeroth-order optimization for memory-efficient LLM fine-tuning: A benchmark. In *International Conference on Machine Learning*, 2024.

[47] Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jinfeng Yi, Mingyi Hong, Shiyu Chang, and Sijia Liu. How to robustify black-box ml models? a zeroth-order optimization perspective. *arXiv preprint arXiv:2203.14195*, 2022.

Appendix A. Related Work

A.1. Zeroth-Order Optimization

Zeroth-order (ZO) optimization has become increasingly relevant for scenarios where computing exact gradients is either computationally prohibitive or strictly impossible (see [25] for a survey of latest advances). Classical applications heavily focused on black-box optimization [16, 22, 30, 31], where access to the objective function is limited to forward evaluations. Significant algorithmic advances have resulted from this productive line of research, including [6, 7, 17, 19, 34, 36, 47]. Recently, however, there has been a significant push to adapt ZO techniques for memory-efficient training of large neural networks. Because backpropagation requires caching intermediate activations, scaling standard first-order methods to billions of parameters often exceeds available hardware memory. Recent approaches like MeZO [27, 43, 46] address this by utilizing a standard two-point ZO estimator, demonstrating that models can be effectively fine-tuned using only forward passes. This dramatically reduces the memory footprint, making ZO an attractive alternative for aligning and adapting modern language models under strict hardware constraints.

A.2. Gradient Clipping and Private Fine-Tuning

Beyond standard memory constraints, ZO methods are actively being adapted to solve challenges in differentially private (DP) machine learning. A fundamental mechanism in deep DP learning is gradient clipping, which tightly bounds the sensitivity of model updates to ensure no single training example disproportionately influences the weights [1, 10, 14, 18].

In traditional DP-SGD, clipping requires the explicit computation and materialization of per-example gradients, which exacerbates the memory bottleneck of backpropagation. To circumvent this, recent work integrates clipping directly into the zeroth-order estimation process. By clipping the scalar loss differences or the forward-pass directional derivatives, algorithms like DPZERO [43] can bound the sensitivity of the update step without ever performing backpropagation. This use of clipping seamlessly bridges the memory benefits of ZO optimization with the rigorous privacy guarantees of DP, allowing for the private fine-tuning of massive autoregressive architectures that would otherwise be impossible to train with standard first-order clipping techniques.

A.3. The Edge of Stability

The optimization dynamics of deep neural networks frequently exhibit a non-equilibrium behavior known as the Edge of Stability (EoS). Originally documented in first-order full-batch gradient descent [2, 4, 11, 12], the EoS phenomenon occurs when a network's training trajectory naturally gravitates toward and oscillates around regions where the maximum eigenvalue of the Hessian (often referred to as the "sharpness") hovers just above $2/\eta$, where η is the learning rate.

While first-order stability is dictated almost entirely by this maximum eigenvalue, the dynamics shift significantly under zeroth-order methods. Because ZO methods rely on stochastic gradient estimates, their mean-square linear stability is governed not just by the sharpness, but by the entire Hessian spectrum [33]. Specifically, theoretical bounds show that ZO methods operate at a distinct edge of stability characterized by a dependence on both the maximum eigenvalue and the Hessian trace. When trained with large step sizes, ZO optimization induces an implicit regularization effect that suppresses the growth of the Hessian trace, forcing the model to converge along a modified stability boundary uniquely dictated by the variance of the zeroth-order estimator.

Appendix B. Detailed Formulation of the Winsorized Mean Calculations

To isolate the specific dynamics induced by static gradient clipping, we benchmark our approach against alternative statistical robustification techniques, most notably the Winsorized mean. Unlike standard clipping, which applies a fixed, data-independent threshold c, Winsorization dynamically bounds the finite differences based on the empirical distribution of the current batch.

For a given batch B_t at iteration t, we first compute the scalar finite difference for each sample $i \in B_t$ along the random perturbation direction u_t:

$$v_i = \frac{f_i(x_t + \lambda u_t) - f_i(x_t - \lambda u_t)}{2\lambda} \ . \tag{5}$$

To apply Winsorization, we sort these scalars to obtain the order statistics:

$$v_{(1)} \leq v_{(2)} \leq \cdots \leq v_{(|B_t|)} \ . \tag{6}$$

For a chosen trimming parameter k (where $0 < k < |B_t|/2$), we define the dynamic lower and upper bounds as the $(k+1)$-th and the $(|B_t| - k)$-th order statistics, respectively:

$$L_t = v_{(k+1)} \quad \text{and} \quad U_t = v_{(|B_t|-k)} \ . \tag{7}$$

The k-Winsorized method replaces the k lowest values with L_t and the k highest values with U_t. The zeroth-order gradient estimator is then computed as the mean of these bounded values:

$$g_\lambda(x_t) = \left(\frac{1}{|B_t|} \sum_{i \in B_t} \text{Clip}_{[L_t, U_t]}(v_i) \right) u_t \ , \tag{8}$$

where the bounding function is defined as:

$$\text{Clip}_{[L_t, U_t]}(a) = \begin{cases} L_t & \text{if } a < L_t \\ a & \text{if } L_t \leq a \leq U_t \\ U_t & \text{if } a > U_t \end{cases} \ . \tag{9}$$

This formulation ensures that the gradient estimate remains robust to extreme outlier perturbations without completely discarding the mass of those samples, offering a data-dependent alternative to standard threshold clipping.

Appendix C. Experimental Setup

The primary goal of our evaluation is to understand how per-sample gradient clipping affects the convergence behavior of Zeroth-Order (ZO) optimization. Our study is motivated by an observation from the original DPZero framework by Zhang et al. [43]. Although their work focused on differential privacy, they noted that clipping also improved optimization in the non-private setting:

> "...the non-private baseline MeZO also appears to benefit from clipping. For instance, without clipping, the original MeZO encounters non-convergence issues at a stepsize of 5×10^{-6}. Conversely, incorporating clipping permits the use of larger stepsizes and yields better results. A thorough investigation of this phenomenon is reserved for future research."

This paper investigates that phenomenon directly. We adapt the DPZero framework for large language model fine-tuning while removing all privacy-related components by disabling Gaussian noise injection ($\sigma = 0$). This isolates the clipping threshold (c) and learning rate (η) as the only variables under study.

The original DPZero work evaluated both RoBERTa and OPT models using a few-shot learning setup with at most 512 training samples per class. We follow the same setup and evaluate performance across tasks of varying complexity: sentiment analysis on SST-2 [32], natural language inference on SNLI [5] and MNLI [39], and question classification on TREC [21].

For each model–dataset pair, we perform a grid search over learning rates ($\eta \in [5 \times 10^{-7}, 1 \times 10^{-4}]$) and clipping thresholds ($c \in [1.0, 500.0]$), including an unclipped baseline ($c = \infty$).

We then evaluate several aspects of clipped ZO optimization. First, we study the trade-off between clipping and learning rate by measuring final validation accuracy and convergence speed, allowing us to characterize the observed "diagonal trend" between η and c. Second, we identify failure modes by tracking training loss to determine when models either diverge due to high variance or fail to learn because clipping is too restrictive. Finally, we analyze how clipping shapes the loss landscape by measuring distance from initialization ($\|w_t - w_0\|$) and evaluating linear mode connectivity between checkpoints. Together, these experiments map how different (η, c) settings influence convergence behavior and solution geometry.

Appendix D. Additional experimental results for clipped zeroth-order optimization: RoBERTa

We provide the train accuracy and loss when sweeping through learning rate η and clipping threshold c in the figure below.

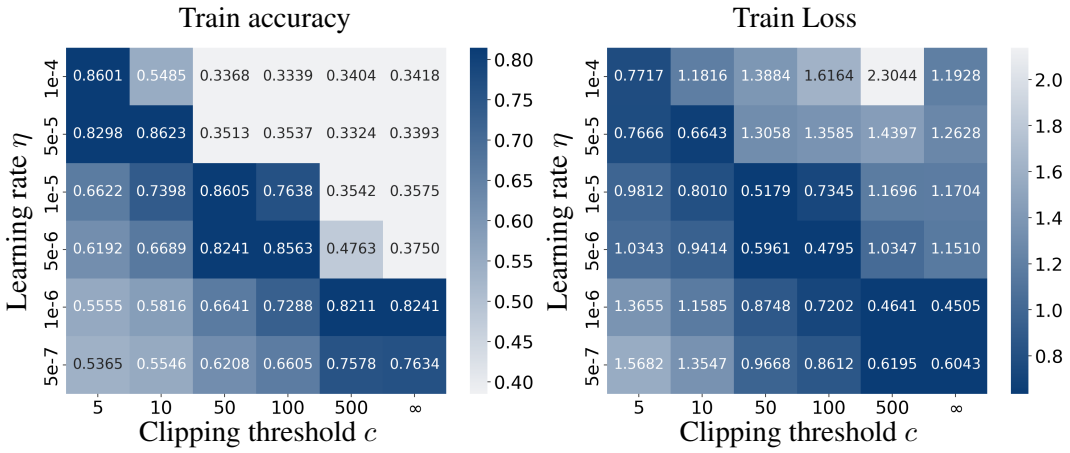

Figure 7: Each cell shows the train accuracy (left) and train loss (right) averaged over four models fine-tuned from RoBERTa [26] on SNLI [5], MNLI [39], SST-2 [32] and Trec [21], respectively. $c = \infty$ is the no clipping baseline.

We provide extended empirical results for the fine-tuning of RoBERTa (355M) using the clipped zeroth-order (ZO) estimator. We evaluate the performance across four standard downstream classification tasks: SST-2, MNLI, SNLI, and TREC. Across all datasets, we observe a consistent, structural interplay between the learning rate (η) and the clipping threshold (c).

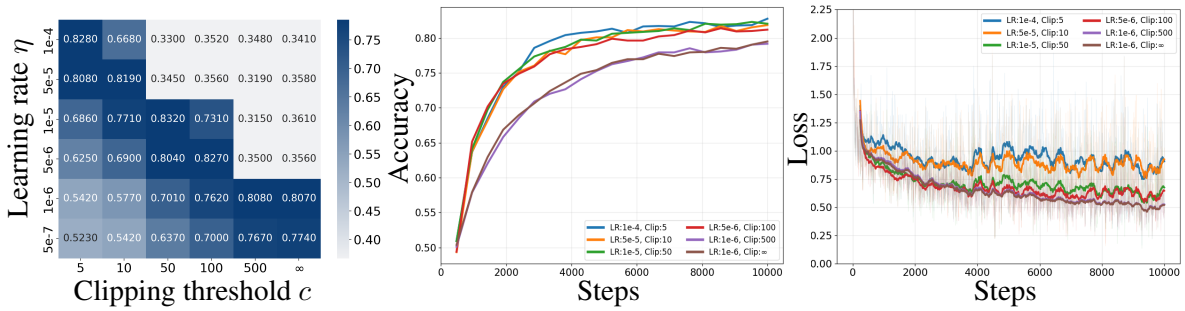

Figure 8: **Left:** Test accuracy grid for SNLI on RoBERTA after sweeping through the learning rate η and the clipping threshold c, we observe that the optimal learning rate for each clipping threshold lies on the curve $\eta \times c = 5 \times 10^{-4}$. **Middle:** Learning curves for η and c pairs along the ηc line depicting the various rates of convergence. **Right:** SNLI train loss of the configurations along the line $\eta\, c$ = constant line. The higher loss and higher accuracy achieved along the diagonal ($\eta c = 5 \times 10^{-4}$) with clipping ($c \leq 100$) compared to the no-clipping best loss implies that the gain of clipping is from better generalization and not better optimization.

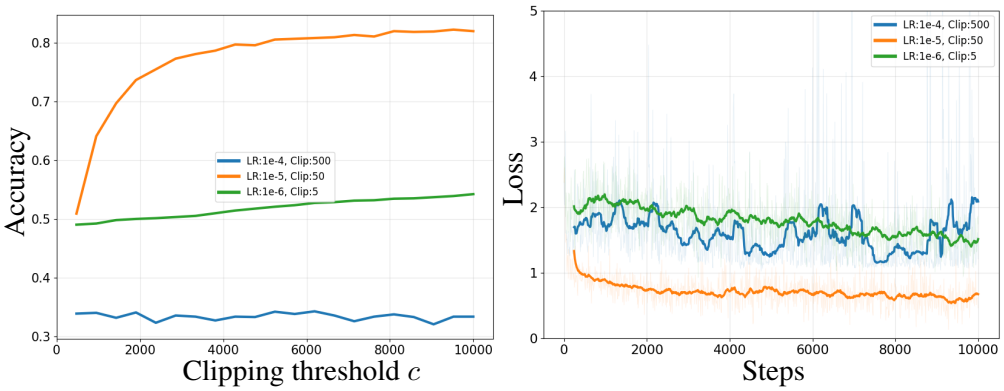

Figure 9: SNLI Learning curves (**left**) and loss curves (**right**) comparing configurations that deviate from the ηc diagonal ($\eta = 10^{-4}, c = 500$ and $\eta = 10^{-6}, c = 5$) against the configuration $\eta = 10^{-5}, c = 50$. Off-diagonal configurations show two failure modes: high η with large c leads to unstable training and random-guess accuracy (about 33%), while low η with small c yields stable but overly slow convergence, plateauing at 49–57% accuracy within 10,000 steps.

D.1. Observed Trends: SNLI

The Stanford Natural Language Inference (SNLI) dataset [5] is a widely adopted benchmark formatted as a 3-class problem to predict entailment, contradiction, or neutral relationships between sentence pairs. Training with the clipped zeroth-order estimator reveals a strict, inversely proportional trend between the learning rate (η) and the clipping threshold (c). The test accuracy heatmaps demonstrate that as the learning rate increases, the clipping threshold must be correspondingly decreased to maintain stability. The peak performance zone, yielding the highest test accuracies ($\sim$83%), lie along this optimal diagonal band (e.g., $\eta = 5 \times 10^{-5}$ paired with $c = 10$, and $\eta = 1 \times 10^{-5}$ paired with $c = 50$) satisfying the *clipping trade-off* at $\eta\, c =$ constant.

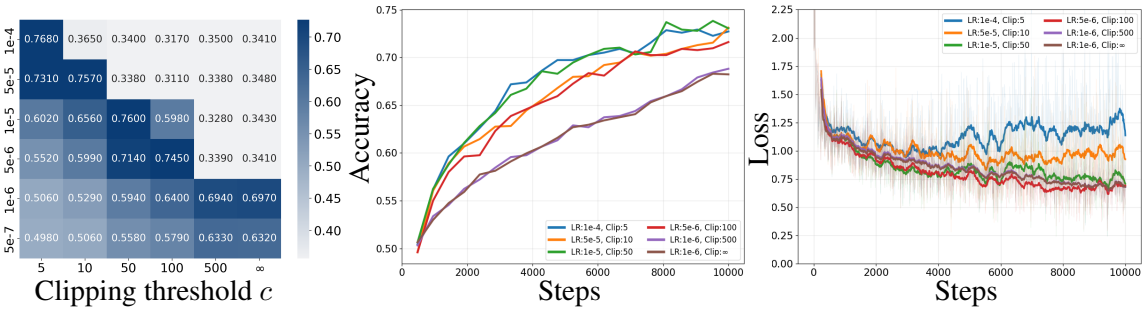

Figure 10: **Left:** MNLI test accuracy for RoBERTa across learning rates η and clipping thresholds c. The best learning rate at each threshold follows $\eta c = 5 \times 10^{-4}$. **Middle:** Learning curves for (η, c) pairs along this line, showing different convergence rates. **Right:** MNLI training loss for configurations satisfying $\eta c =$ constant. As with SNLI (Appendix D.1), clipped runs ($c \leq 100$) achieve higher loss but better accuracy than the best unclipped run, suggesting clipping improves generalization rather than optimization.

Configurations deviating from this diagonal exhibit two distinct failure modes. In the upper-right region of the hyperparameter grid (high η, large c), the models fail to learn entirely. The test loss trajectories exhibit extreme, volatile spikes, and accuracy flatlines at the random guessing baseline of approximately 33%. Conversely, configurations in the lower-left region (low η, small c) produce stable and smooth loss curves. However, their convergence is excessively slow; these models fail to reach peak accuracy within the allocated 10,000-step limit, stalling between 49% and 57% accuracy (Figure 9).

Further examination of the continuous learning trajectories reveals that hyperparameter configurations split into distinct behavioral groups. Aggressive pairings—characterized by larger learning rates and smaller clipping thresholds (e.g., $\eta = 10^{-4}, c = 5$ and $\eta = 5 \times 10^{-5}, c = 10$)—exhibit rapid initial convergence, quickly reaching near-peak accuracy within the first 2,000 to 4,000 iterations. However, this speed is accompanied by a highly variant and elevated training loss profile. Instead, conservative configurations with lower learning rates and larger clipping (e.g., $\eta = 10^{-6}, c = 500$ or $c = \infty$) display a much shallower ascent in accuracy, but descend cleanly to lower, smoother training loss plateaus. The magnitude of the clipping threshold dictates the optimization trajectory, steering the models to converge into distinctly different solutions.

D.2. Observed Trends: MNLI

Like SNLI, the Multi-Genre Natural Language Inference (MNLI) corpus [39] is a widespread 3-class sentence understanding benchmark requiring models to predict entailment, contradiction, or neutral relationships across diverse text genres.

Empirical results on the MNLI dataset perfectly mirror the structural dynamics observed in SNLI, confirming that the clipping trade-off is a robust and consistent phenomenon across different datasets. The test accuracy heatmap reveals that the highest performance (approximately 76%) also lies strictly along the established diagonal cells, maintaining an inversely proportional relationship between the learning rate (η) and the clipping threshold (c).

Deviating from this optimal band yields two distinct failure modes. In the high η and large c regime, failing to sufficiently clip the large learning rates results in the accuracy remaining trapped

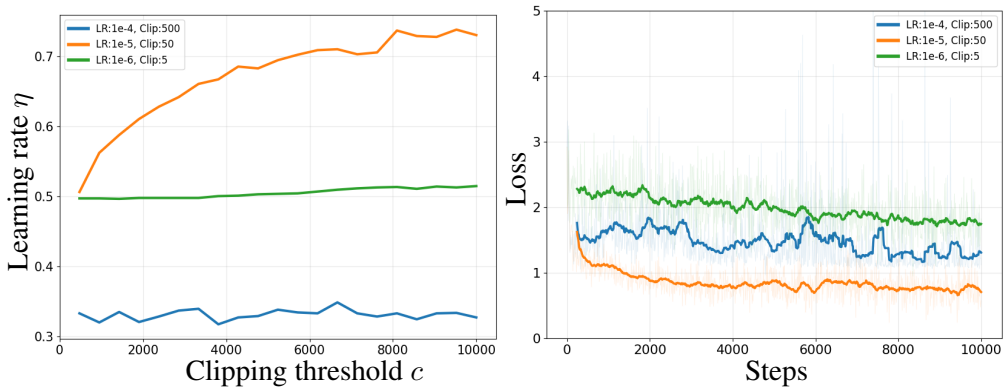

Figure 11: MNLI learning curves (**left**) and loss curves (**right**) comparing configurations that deviate from the ηc diagonal ($\eta = 10^{-4}, c = 500$ and $\eta = 10^{-6}, c = 5$) against the configuration $\eta = 10^{-5}, c = 50$. Similar to SNLI, the off-diagonal settings perform worse: high η with loose c causes unstable training and random-guess accuracy, while low η with strict c converges slowly and plateaus below the baseline.

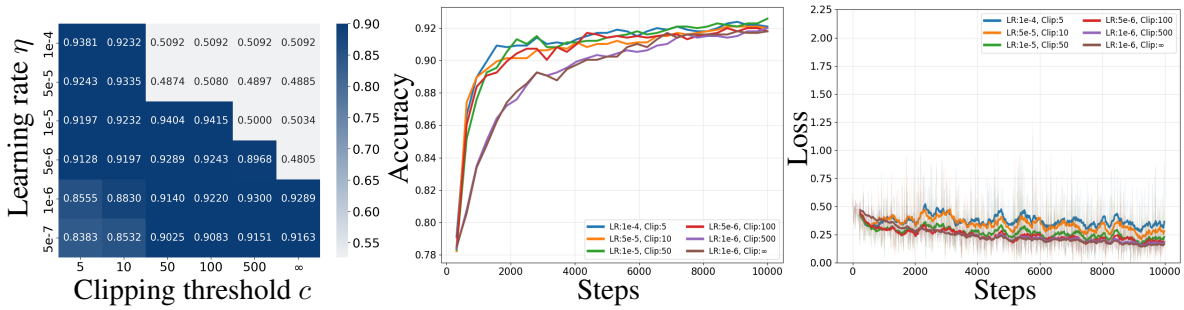

Figure 12: The clipping trade-off trend is less prominent for Roberta fine-tuned on SST-2.

at the 33% random-guessing baseline. On the opposite end of the spectrum, employing a low η with a small c produces smooth, linear accuracy climbs. However, the step sizes are too small, preventing the model from traversing the loss landscape quickly enough to reach a similar performance of the configurations along the diagonal band within the 10,000-step computational limit.

D.3. Observed Trends: SST-2

The Stanford Sentiment Treebank (SST-2) [32] is a standard binary classification task designed to predict the sentiment of movie reviews. Training with the clipped zeroth-order estimator also shows the trade-off. The optimal performance is concentrated along the diagonal band, maintaining the inversely proportional relationship between the learning rate (η) and the clipping threshold (c). Configurations parameterized along this optimal band (e.g., $\eta = 1 \times 10^{-5}$ paired with $c = 50$) achieve the highest convergence, consistently exceeding 93% test accuracy. Outside this optimal diagonal exposes severe optimization bottlenecks similar to SNLI and MNLI.

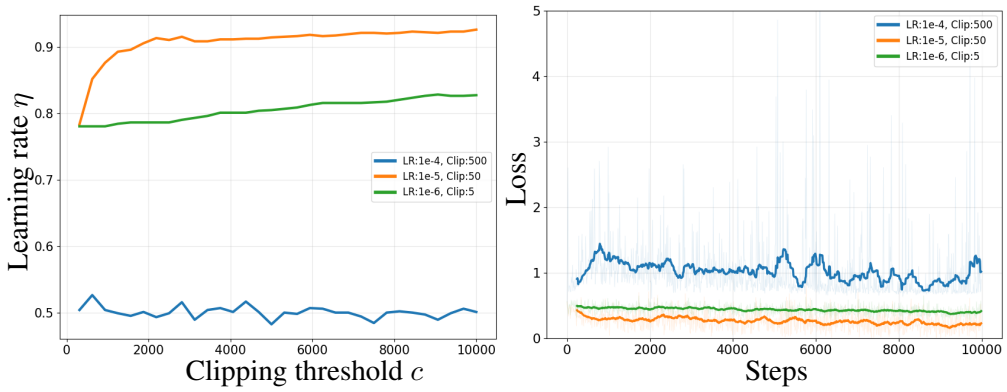

Figure 13: SST-2 learning (**left**) and loss (**right**) curves comparing off-diagonal configurations ($\eta = 10^{-4}, c = 500$ and $\eta = 10^{-6}, c = 5$) with $\eta = 10^{-5}, c = 50$. Similar to SNLI and MNLI, the off-diagonal settings perform worse: high η with loose c leads to unstable training and chance-level accuracy, while low η with strict c converges slowly and plateaus below the baseline.

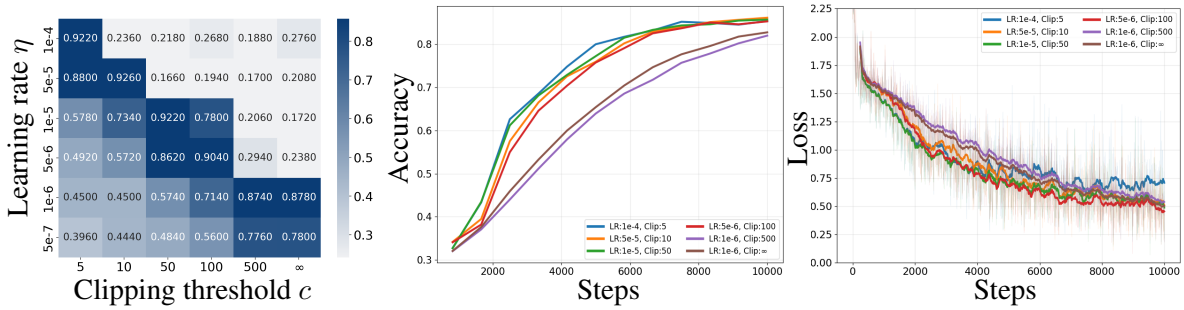

Figure 14: Clipping trade-off trends for RoBERTa fine-tuned on TREC, showing the same *clipping trade-off* behavior observed in D.1, D.2, and D.3.

D.4. Observed Trends: TREC

The TREC dataset [21] is a question classification benchmark requiring models to categorize open-domain, fact-based questions into one of six distinct semantic categories. Training with the clipped zeroth-order estimator on this 6-class problem confirms the clipping trade-off. Configurations parameterized away from this stability boundary exhibit severe performance degradation 15.

Appendix E. Additional experiments: OPT

E.1. Observed Trends: SST-2

To verify that the clipping trade-off generalizes to beyond Roberta for more advanced architectures, we extend our evaluation to the OPT [45] model fine-tuned on the SST-2 binary classification benchmark [32]. The test accuracy heatmap demonstrates that the OPT architecture strongly preserves the diagonal scaling trend previously observed in encoder-only models. Peak performance forms a distinct structural band where higher learning rates (η) strictly require tighter clipping thresholds (c). Accuracies exceeding 90%, and reaching up to 92.8%, are clustered entirely along this optimal

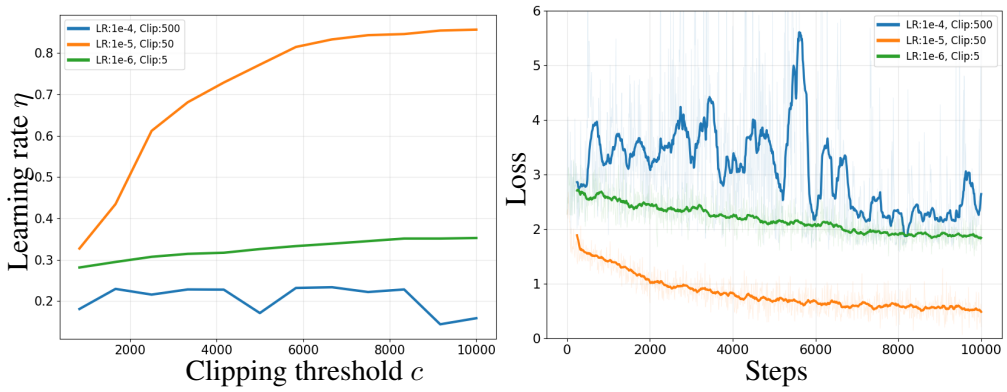

Figure 15: TREC learning curves (**left**) and loss curves (**right**) comparing configurations that deviate from the ηc diagonal ($\eta = 10^{-4}, c = 500$ and $\eta = 10^{-6}, c = 5$) against the configuration $\eta = 10^{-5}, c = 50$. Similar to SNLI, SST-2, and MNLI, the off-diagonal settings perform worse: high η with loose c causes unstable training and random-guess accuracy, while low η with strict c converges slowly and plateaus below the baseline.

diagonal (e.g., $\eta = 2 \times 10^{-5}$ paired with $c = 5.0$, and $\eta = 1 \times 10^{-6}$ paired with $c = 100.0$). Deviating from this boundary yields familiar failure modes. In the upper-right regime (e.g., $\eta = 1 \times 10^{-4}$ with $c \geq 10.0$), the unconstrained variance causes complete model failure, with accuracy catastrophically collapsing to the binary random-guessing baseline of approximately 49% to 54%. In the overly constrained lower-left region (e.g., $\eta \leq 1 \times 10^{-6}$ with $c \leq 1.0$), the model avoids this collapse but stagnates, plateauing at a sub-optimal 57% to 58% accuracy within the given training budget.

An analysis of the continuous learning trajectories reveals unified convergence for configurations situated on the optimal diagonal. Despite utilizing drastically different hyperparameter pairings, all sampled configurations successfully navigate to a similar peak accuracy of 91% to 93%. However, the speed of this traversal varies significantly. Configurations parameterized with a higher

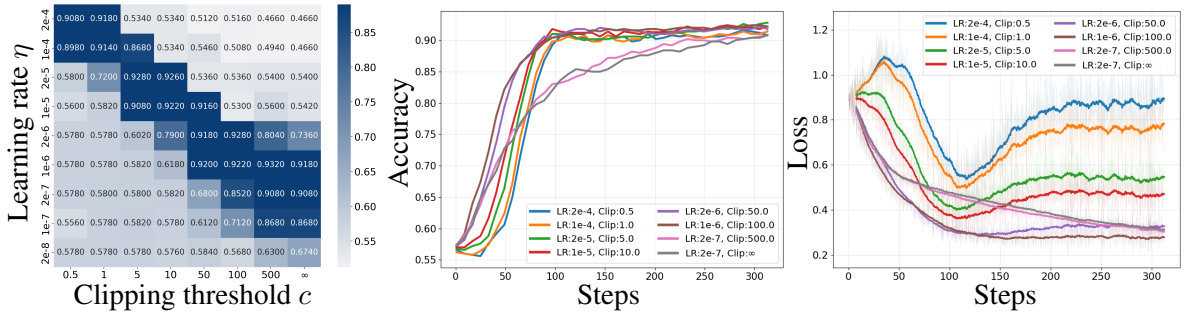

Figure 16: Experiments on private fine-tuning OPT (1.3B) for SST-2 with 2. **Left:** Test accuracy % when varying the stepsize and clipping threshold together. **Middle:** Accuracy plots depicting the convergence of the diagonal band configurations. **Right:** (Smoothed) training loss curves of the diagonal band configurations. Consistent with previous experiments, we observe a similar behavior with clipping and zeroth order methods converging with different behaviors when changing models from Roberta to OPT.

base learning rate (e.g., $\eta = 2 \times 10^{-5}, c = 5.0$) rapidly optimize the objective, achieving 90% accuracy in fewer than 50 epochs. Conversely, conservative setups utilizing lower learning rates and looser clipping thresholds (e.g., $\eta = 1 \times 10^{-6}, c = 100.0$) exhibit a much shallower ascent, requiring 150 to 200 epochs to cross the same performance threshold.

Examining the corresponding training loss profiles exposes a direct correlation between the learning rate and the zeroth-order estimator's variance. High-η configurations inherently display highly volatile, thick loss bands throughout the training run, reflecting the larger stochastic perturbations. In contrast, configurations with low learning rates (e.g., $\eta = 2 \times 10^{-7}, c = 500.0$) produce tightly clustered, smooth loss curves.

Appendix F. Gradient Estimator Spread

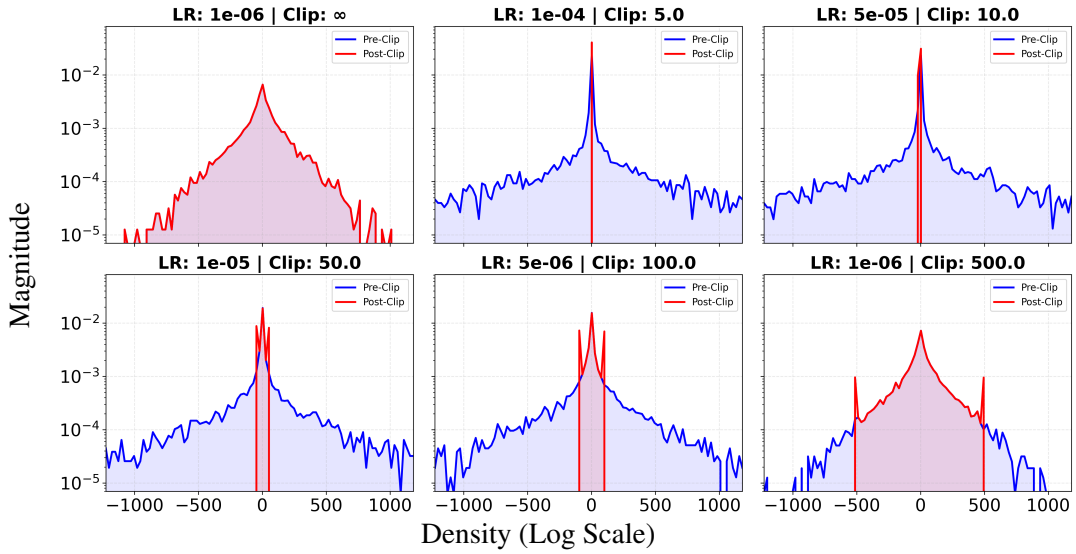

Figure 17: Histogram depicting the spread of the gradient estimator before and after clipping

To physically observe how the clipping threshold controls the optimization step, we visualize the empirical distribution of the per-sample zeroth-order gradient estimators before and after clipping. Figure 17 and Figure 18 display these distributions across six parameterizations sampled from the optimal diagonal, utilizing both raw counts to show the central mass and log-scaled density to highlight the tail behavior.

Examining the linear-scale count distributions highlights a distinct shift in the central mass of the estimator. While unclipped ZO estimation ($c = \infty$) naturally produces a smoothly decaying, wide-shouldered distribution around zero, the introduction of clipping severely distorts this geometry. Specifically, strict clipping constraints visually stretch the limits of the effective distribution while simultaneously inducing significantly thinner, sharper central peaks. This geometric transformation occurs because the clipping function forcibly redistributes the probability mass from the continuous heavy tails, pushing the values inward to aggregate exactly at the structural limits ($\pm c$). Consequently, the gradient estimator transitions from a broad, continuous spread into a highly con-

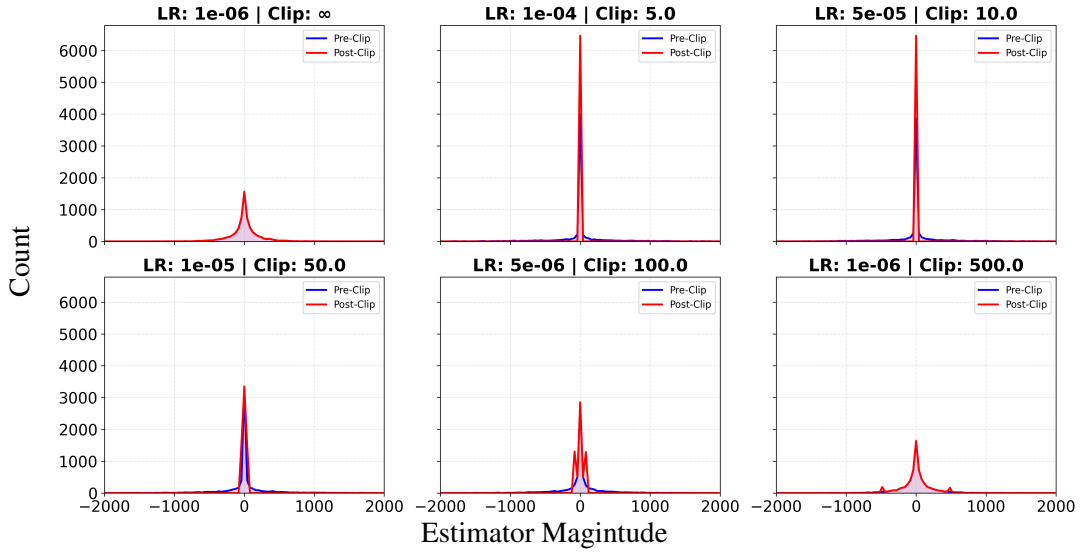

Figure 18: Histogram depicting raw counts of gradient estimator before and after clipping

centrated, truncated structure where the update signal is strictly confined to the immediate center and the absolute clipping boundaries.

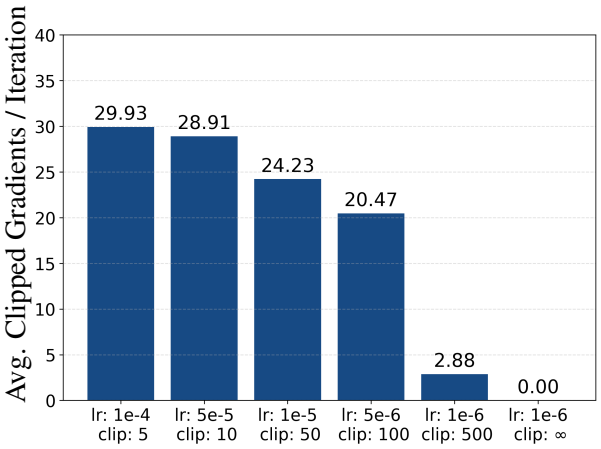

Figure 19: Average number of gradient estimates clipped per iteration (batch size = 64) for learning rate and clipping threshold configurations on the *clipping trade-off* line.

Furthermore, comparing the distributions along the $\eta c \approx$ constant boundary illustrates exactly how clipping compensates for step size. To quantify this effect, Figure 19 tracks the average number of clipped gradient estimates per iteration (using a batch size of 64) for configurations traversing the optimal diagonal. In aggressive setups (e.g., $\eta = 10^{-4}, c = 5.0$), the strict threshold aggressively squashes a massive fraction of the gradient estimates—nearly half the batch (averaging 29.93 out of 64 samples) is forcibly pushed to the boundary. This extreme suppression fundamentally alters the

aggregated update vector, which is precisely what allows the optimizer to survive the massive 10^{-4} step size without shattering.

Conversely, as we move down the diagonal to more conservative setups (e.g., $\eta = 10^{-6}, c = 500.0$), the clipping mechanism engages far less frequently, averaging fewer than 3 clipped samples per batch. In these regimes, the wider threshold allows the natural shape of the estimator's central mass to remain largely intact, acting merely as a sparse safeguard to truncate the most extreme, landscape-shattering outliers.

Appendix G. Future Work

While this work empirically maps the structural boundaries of clipped zeroth-order optimization, the discovery of the $\eta c \approx$ constant trade-off and its corresponding capabilities and behaviors opens several critical avenues for future research. Specifically, the interplay between per-sample variance bounding, first-order dynamics, and the edge of stability requires formal theoretical characterization.

Our findings present an exciting opportunity to broaden the current theoretical understanding of the Edge of Stability (EoS). Recent literature establishes that standard zeroth-order optimization naturally gravitates toward an equilibration where the maximum eigenvalue of the Hessian matches the theoretical $2/\eta$ boundary, inherently favoring flat minima characterized by a smaller trace of the Hessian [33, 44]. However, our preliminary curvature tracking indicates that tightly clipped configurations possess a unique capacity to deviate from this threshold, exhibiting continuous curvature growth without destabilizing. Future research should deeply examine this behavior to formalize whether clipped methods eventually match the $2/\eta$ equilibration, or if they are governed by an entirely distinct "Clipped EoS." Understanding the exact mathematical mechanisms that allow a bounded optimizer to survive in regions of extreme local sharpness will significantly advance ZO optimization theory.

Finally, the potential universality of these dynamics presents a compelling direction for future research beyond zeroth-order methods. It remains an open question whether the behaviors observed are exclusive to zeroth-order methods, or if they represent a fundamental geometric property of bounded updates. Future work must rigorously investigate whether applying strict per-sample clipping to standard first-order optimization (such as SGD) induces similar trends. If these dynamics do generalize, a promising avenue is to test whether artificially augmenting first-order updates with heavy-tailed stochasticity, while tightly enforcing per-sample bounds, can more effectively explore the optimization landscape. By mimicking the variance profile of ZO estimators, this strategy could equip first-order methods to systematically escape sharp local minima and settle into flatter, generalized representations.