δ-Regularized Gradient Clipping for Stable Optimization: Analysis and Empirical Evaluation

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2025

Abstract

This work provides an extended empirical and theoretical analysis of δ-GClip, a variant of gradient clipping with a formal convergence guarantee which was proposed recently by Tucat et al. [21]. Our experiments analyze activation patterns, gradient dynamics, and dependence of δ on architectural scale across supervised benchmarks, diffusion models, and a lightweight protein-generation task. In particular, we show that combining Adam with a brief δ-clipping warm-up improves the stability and early-phase optimization in diffusion model training. Using the Kurdyka–Łojasiewicz framework by Scaman et al. [17] we further extend the convergence guarantees of δ-GClip beyond the squared-loss setting to more general smooth non-convex objectives.

1. Introduction

Gradient clipping is a widely used long-standing mechanism for controlling optimization dynamics in neural networks training, particularly in high-dimensional parameter regimes where gradients exhibit heavy-tailed variability or rapid growth [11, 16, 19, 24]. Despite its widespread use, most implementations rely on simple heuristics such as global-norm clipping with a fixed threshold, and their effect on the stability landscape of modern architectures remains poorly understood.

A recent contribution by Tucat et al. [21] introduced the δ-GClip method as a more principled variant of clipping that regulates the effective step size by enforcing a lower bound on the update magnitude. While not explicitly designed as a curvature-aware method, this mechanism implicitly reduces sensitivity to regions of high curvature by preventing excessively small steps when gradients become large. Importantly, Tucat et al. provided a convergence guarantee for quadratic objectives, i.e. based on the squared loss, but the empirical evaluation in the original work is limited in scope, leaving open questions about how δ-clipping interacts with depth, curvature, and gradient-norm dynamics in modern architectures with high-dimensional parameter spaces.

To address these questions, we reproduce and extend the experimental setups from the original work, adding explicit logging of activation frequencies, effective step sizes, and gradient-norm trajectories. Motivated by prior work showing that scaling analyses reveal universal patterns in optimization dynamics [2, 12], we perform a controlled depth-scaling study on ResNets of increasing depth. Scaling experiments provide a clean environment for probing stability regimes, curvature sensitivity, and early-phase optimization behavior as model size increases, allowing us to isolate the effect of δ-clipping before moving to more complex architectures.

Diffusion models represent an extreme high-gradient regime where optimization dynamics are notoriously unstable [7, 10, 13, 20]. Adaptive optimizers such as Adam are standard in this setting,

and clipping is often applied heuristically. Although several works analyze clipping in adaptive methods [6, 9, 18, 22], δ-GClip has not yet been evaluated in this context. We therefore investigate whether combining Adam with δ-clipping improves stability and early-phase gradient behavior in diffusion model training.

To test generality and probe δ-GClip's behavior in a data-constrained regime, we include a lightweight experiment on torsion-based protein diffusion models [23]. Protein structure datasets are orders of magnitude smaller than those used in large-scale image or language training, yet the models must learn complex distributions over a high-dimensional angular space. This setting naturally induces elevated gradient noise and high variability across individual sample gradients, making protein diffusion a sensitive testbed for assessing whether the stability benefits of δ-clipping are amplified when data, rather than model capacity or compute, is the primary bottleneck. The constrained angular geometry further sharpens optimization instabilities that may be masked in data-rich regimes, while remaining computationally manageable for controlled studies.

Finally, we discuss the possibility of extending the convergence argument for δ-GClip beyond the square loss considered in the original work. For this, we draw on the framework of Scaman et al. [17], which provides tools for analyzing optimization under more general smoothness and curvature assumptions comprising more loss functions than the squared loss, suggesting that the theoretical foundations of δ-clipping may be adaptable to a broader class of objectives.

2. Background on δ-GClip

The δ-GClip algorithm [21] is a regularized variant of standard gradient clipping in which the effective step size is lower-bounded by a parameter $\delta > 0$. Given η, $\gamma > 0$, instead of the usual clipping factor

$$h_{\text{GClip}}(x_t) = \eta \, \min\left\{1, \gamma/\|\nabla f(x_t)\|\right\},$$

δ-GClip uses

$$h(x_t) = h_{\delta\text{-GClip}}(x_t) = \eta \, \min\left\{1, \max\left\{\delta, \gamma/\|\nabla f(x_t)\|\right\}\right\},$$

which prevents the effective step size from collapsing when gradients become large. This modification preserves the stabilizing effect of clipping while introducing a lower-bounded update magnitude that interacts nontrivially with curvature. Following standard practice for stochastic gradient descent, all our experiments replace the full gradient $\nabla f(x_t)$ with the stochastic gradient g_t.

Tucat et al. prove that δ-GClip enjoys global convergence guarantees for sufficiently wide deep networks trained with the squared loss, based on the recently established PL* condition [14]. Although these assumptions are restrictive (full gradients, squared loss, and widths large enough for PL* to hold), the authors also provide empirical evidence that δ-GClip performs competitively on practical architectures far outside the theoretical regime, including ResNet-18, a VAE, a Vision Transformer, and a BERT-based classifier. These results highlight that the δ-regularization mechanism can remain effective even when gradients are noisy or heavy-tailed, and when curvature varies significantly across layers.

While the original analysis of δ-GClip primarily focuses on the squared loss regime, our empirical results, along with those of Tucat et al., suggest that its stability benefits extend to more complex optimization landscapes. To formally address this, we employ the framework of Scaman et al. [17] — which generalizes the Polyak-Łojasiewicz condition through Kurdyka-Łojasiewicz (ϕ-KL*) inequalities — to provide an extended convergence analysis for broader non-convex objectives, such as cross-entropy and denoising score matching (see Appendix A).

3. Evaluation on Established Benchmarks

We evaluate the proposed δ-GClip method on ResNet-18 and Variational Autoencoders (VAE), following the experimental protocol of Tucat et al. [21]. Our goal here is to explore a broader and practically relevant range of thresholds in order to better characterize the optimization behavior of δ-GClip across different scales. We sweep $\delta \in \{0.01, 0.03, 0.08, 0.2, 0.45, 0.9\}$ while keeping architectures and data protocols identical to Tucat et al. [21]. Final performance is measured via test accuracy for ResNet-18 and ELBO for VAE. Appendix B contains all hyperparameter settings, Appendix C defines the diagnostic metrics, Appendix D reports full ResNet-18 results, and Appendix E reports full VAE results.

Across the full sweep on ResNet-18, $\delta = 0.2$ yields the best overall performance. With this configuration, δ-GClip attains a final test accuracy of 95.22%, closely matching the SGD baseline (95.33%) while achieving a lower final training loss. This setting avoids the overly conservative behavior of Adam and the late-epoch instability characteristic of SGD, resulting in a stable and efficient optimization trajectory. In terms of update dynamics, $\delta = 0.2$ produces substantially smaller RUM than SGD while maintaining sufficiently large and consistent steps to ensure fast convergence. This balance reflects the role of δ as a lower bound on the effective step size.

For VAEs, gradient norms are substantially smaller than in the convolutional setting. As a result, for commonly used clipping thresholds $\gamma \in \{30, 50\}$ the ratio $\gamma/\|g_t\|$ exceeds all tested δ values, making δ inactive and reducing δ-GClip to standard GClip. Only for $\gamma = 1$ does δ become active (for $\delta \geq 0.08$), but this regime leads to overly conservative updates and inferior ELBO.

4. Linking Optimal δ to Model Scale and the Edge of Stability

Influence of Architecture Scale. Here, we investigate the impact of network depth and width on the selection of the optimal threshold δ^*. By construction, architectural scale influences the activation of the δ-controlled lower bound through its effect on gradient magnitudes: the mechanism in δ-GClip intervenes only when $\|\nabla f(x_t)\| > \gamma/\delta$, so the frequency of activation is directly determined by the gradient norm distribution induced by a given architecture. Full hyperparameter settings are provided in Appendix B.3 and complete results in Appendix G.

Although theoretical analyses show that very deep ResNets can exhibit either vanishing or exploding gradients depending on residual-branch scaling [15], empirical studies consistently find that gradient norms tend to decrease with depth in standard ResNet architectures [3, 5]. Consequently, for a fixed δ, the δ-controlled lower bound is triggered less frequently in deeper networks. This is precisely what we observe experimentally: deeper models produce fewer large-gradient events, so the activation threshold γ/δ is exceeded less often. As a result, the optimal threshold δ^* increases monotonically with depth. Increasing δ lowers the activation threshold and restores an intervention frequency comparable to that observed in shallower networks.

Across all tested depths, δ^*-GClip consistently outperforms standard GClip in both test accuracy and test loss, indicating that the additional lower-bound mechanism provides a measurable optimization advantage.

For width, we also observe a shift in the optimal δ^* as the model becomes wider. Narrow networks (*width* $\leq$ 2) perform best with moderate values in the range $\delta \approx 0.20$–0.45, whereas wider architectures (*width* = 8–10) achieve their strongest results at $\delta = 0.45$. These experiments were conducted using an ultra-wide CIFAR-ResNet, where increasing width primarily affects the magnitude and frequency of large-gradient events.

δ-Driven Proximity to the EoS. In our experiments, we also monitor the relationship between the top eigenvalue of the Hessian, $\lambda_{\max}(\nabla^2 f(x_t))$, and the stability threshold $2/h_{\delta\text{-GClip}}(x_t)$ imposed by the effective step size, where $\lambda_{\max}$ denotes the largest eigenvalue of the Hessian, estimated via standard power iteration on a single minibatch. Cohen et al. [8] demonstrated that gradient descent with a fixed learning rate η tends to operate at the Edge of Stability (EoS) — a regime where the leading Hessian eigenvalue hovers near $2/\eta$, with the optimizer progressing through a form of controlled instability. While standard gradient clipping disrupts this regime by aggressively reducing the effective step size during gradient spikes, the δ constraint introduces a principled floor on this adjustment. By preventing the effective learning rate from collapsing, δ-GClip bounds how far the optimizer can retreat from the high-curvature areas, effectively maintaining the trajectory within the EoS regime.

Across all tested depths, we observe a clear trend: increasing δ allows the model to converge to regions of higher sharpness and moves the optimization dynamics closer to EoS. Larger δ values make the effective step size less conservative by enforcing a higher lower bound on $h_{\delta\text{-GClip}}(x_t)$, which reduces the frequency of aggressive clipping and yields updates more similar to standard SGD. Consequently, this shifts the trajectory toward higher-curvature regions where the stability threshold $2/h_{\delta\text{-GClip}}(x_t)$ is more frequently challenged. Full sharpness measurements and effective learning-rate traces are provided in Appendix G.3.

5. Optimization Analysis in Diffusion Models

We next evaluate δ-GClip in a diffusion-model setting using a standard U-Net architecture for denoising score matching. Since our focus is on the behavior of δ-clipping rather than on competing with adaptive methods, we only briefly report a baseline experiment with δ-clipping alone (App. F.1) and concentrate instead on its interaction with Adam.

In this experiment, we apply δ-clipping only during an initial warm-up phase before switching to standard Adam, evaluating ten thresholds

$$\delta \in \{0.08, 0.10, 0.12, 0.15, 0.20, 0.25, 0.35, 0.50, 0.70, 0.95\}$$

with warm-up durations increasing from 2 to 12 epochs as δ grows. Full experimental details are provided in Appendix B.4, a full analysis is provided in Appendix F.2. Across the tested configurations, warm-up δ-clipping yields several improvements over pure Adam (FID 75.9), although the effect is not uniform across all values of δ. The strongest improvement is observed for $\delta = 0.15$ with 5 warm-up epochs (FID 67.5), which also outperforms standard GClip (FID 69.4). Larger thresholds such as $\delta = 0.95$ remain beneficial when paired with a longer warm-up window (FID 70.9), while intermediate values ($\delta = 0.20$–0.35) provide moderate gains. Although diffusion models exhibit complex and highly nonlinear optimization dynamics that make it difficult to identify a simple monotonic relationship between performance and δ, the overall trend suggests that a brief δ-clipping warm-up can improve stability and early-stage training behavior.

6. Optimization Analysis in Data-Constrained Protein Diffusion Models

To assess the behavior of δ-GClip in a regime where data scarcity, rather than model capacity, is the primary bottleneck, we conduct controlled experiments on a lightweight torsion-based protein diffusion model trained on a deliberately small dataset and with a small batch size (8). This setting

induces elevated gradient variance and pronounced optimization instability, making it a natural testbed for studying the stabilizing effect of gradient clipping. The model operates on angular coordinates (ϕ, ψ, ω) and is evaluated using training loss together with four distributional quality metrics (KL, JS, cluster L1, diversity ratio). A complete description of the model and dataset is provided in Appendix B.5, and the full experimental results are reported in Appendix H.

In the full-step configuration, δ-GClip is applied at every optimization update, modulating the effective learning rate according to the instantaneous gradient norm. Among all configurations, the combination of Adam with δ-GClip at $\delta = 0.90$ achieves the strongest overall performance, improving both optimization stability and sample quality relative to the Adam and Adam+GClip baselines. Specifically, it reduces the final training loss from 0.369 (Adam) and 0.383 (Adam+GClip) to 0.362, lowers JS divergence from 0.1735 and 0.1355 to 0.0706, and brings the diversity ratio from an over-dispersed 1.44 (Adam) toward an almost ideal 0.98. Intermediate thresholds such as $\delta \approx 0.45$ also outperform both baselines across most metrics, whereas smaller values ($\delta \leq 0.03$) behave similarly to standard clipping and provide only limited stabilization. Overall, the results indicate that larger δ values yield the most consistent improvements, while mid-range and small thresholds exhibit weaker or mixed effects in the data-limited protein diffusion setting.

7. Conclusions

In this work, we have provided an extensive empirical and theoretical characterization of δ-GClip, a regularized gradient clipping variant designed to stabilize high-dimensional learning dynamics. Our contributions and key findings are summarized as follows:

- **Theoretical Generalization**: We extended the original convergence analysis beyond the squared loss regime. By leveraging the Kurdyka-Łojasiewicz (ϕ-KL*) framework, we established global and local convergence guarantees for a broader class of non-convex objectives, including cross-entropy and denoising score matching.
- **Architectural Scaling Laws**: Through a systematic study of ResNet architectures, we identified a clear scaling relationship between model dimensions and optimization hyperparameters. We demonstrated that as network depth and width increase – leading to a natural decrease in gradient norms – the optimal threshold δ^* must shift upward to maintain consistent intervention frequencies and stability.
- **Stability in Generative Modeling**: We successfully integrated δ-GClip with adaptive optimizers in complex generative tasks. In diffusion models, a brief δ-clipping warm-up phase improved early-stage training stability and final FID scores compared to pure Adam.
- **Robustness under Data Constraints**: In torsion-based protein diffusion – a regime characterized by high gradient variance and limited data – δ-GClip significantly improved both optimization stability and distributional quality metrics, such as JS divergence and diversity ratios.
- **δ-Driven Proximity to the EoS.**: Our analysis shows that δ-GClip provides a principled mechanism for navigating the Edge of Stability by enforcing a functional floor on update magnitudes, allowing the model to converge to regions of higher sharpness without the catastrophic divergence or aggressive step-size collapse associated with standard clipping methods.

Overall, our results demonstrate that δ-GClip is a versatile and principled tool for modern deep learning, offering measurable advantages in stability and performance across a diverse range of high-dimensional architectures and data regimes.

References

[1] Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward and Newton-like methods. *Mathematical Programming*, 137(1):91–129, 2013.

[2] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences (PNAS)*, 121(27): e2311878121, 2024. URL https://arxiv.org/abs/2102.06701.

[3] David Balduzzi, Marcus Frean, Lennox Leary, JP Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? In *Proceedings of the 34th International Conference on Machine Learning*, pages 342–350, 2017.

[4] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, 2014.

[5] Jinghui Chen, Qin Li, and Jian Li. A comprehensive analysis of gradient norm equality in deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42 (12):3046–3060, 2020.

[6] Savelii Chezhegov, Yaroslav Klyukin, Andrei Semenov, Aleksandr Beznosikov, Alexander Gasnikov, Samuel Horváth, Martin Takáč, and Eduard Gorbunov. Clipping improves adam–norm and adagrad–norm when the noise is heavy–tailed. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2406.04443.

[7] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. doi: 10.48550/arXiv.2204.00227. URL https://arxiv.org/abs/2204.00227.

[8] Jeremy M. Cohen, Behrooz Ghorbani, Shankar Krishnan, Naman Agarwal, Sourabh Medapati, Michal Badura, Daniel Suo, David Cardoze, Zachary Nado, George E. Dahl, and Justin Gilmer. Adaptive gradient methods at the edge of stability. *arXiv preprint arXiv:2207.14484*, 2022. doi: 10.48550/arXiv.2207.14484. Revised version v2, April 2024.

[9] Eduard Gorbunov, Marina Danilova, and Alexander Gasnikov. Stochastic optimization with heavy-tailed noise via accelerated gradient clipping. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. doi: 10.48550/arXiv.2005.10785. URL https://arxiv.org/abs/2005.10785.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. doi: 10.48550/arXiv.2006.11239. URL https://arxiv.org/abs/2006.11239.

[11] Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. In *International Conference on Learning Representations (ICLR) Workshop*, 2018. URL https://arxiv.org/abs/1711.04623.

[12] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. URL https://arxiv.org/abs/2001.08361.

[13] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. *arXiv preprint arXiv:2312.02696*, 2024. doi: 10.48550/arXiv.2312.02696. URL https://arxiv.org/abs/2312.02696.

[14] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022. doi: 10.1016/j.acha.2021.12.003.

[15] Pierre Marion, Cosme Louart, and Romain Couillet. Scaling resnets in the large-depth regime. *Journal of Machine Learning Research*, 26(1):1–55, 2025.

[16] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/pascanu13.html.

[17] Kevin Scaman, Cédric Malherbe, and Ludovic Dos Santos. Convergence rates of non-convex stochastic gradient descent under a generic Lojasiewicz condition and local smoothness. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 19310–19327, 2022. URL https://proceedings.mlr.press/v162/scaman22a.html.

[18] Prem Seetharaman, Gordon Wichern, Bryan Pardo, and Jonathan Le Roux. Autoclip: Adaptive gradient clipping for source separation networks. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2020. doi: 10.1109/MLSP49062.2020.9231926. URL https://www.merl.com/publications/TR2020-132.

[19] Umut Şimşekli, Levent Sagun, and Mert Gürbüzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning (ICML)*, 2019. URL https://arxiv.org/abs/1901.06053.

[20] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021. doi: 10.48550/arXiv.2011.13456. URL https://arxiv.org/abs/2011.13456. Oral presentation.

[21] Matteo Tucat, Anirbit Mukherjee, Mingfei Sun, Procheta Sen, and Omar Rivasplata. Regularized gradient clipping provably trains wide and deep neural networks. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=ABT1XQLbOx.

[22] Guoxia Wang, Shuai Li, Congliang Chen, Jinle Zeng, Jiabin Yang, Dianhai Yu, Yanjun Ma, and Li Shen. Adagc: Improving training stability for large language model pretraining. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=ZQcDUhEOg9.

[23] Kevin E. Wu, Kevin K. Yang, Rianne van den Berg, Sarah Alamdari, James Y. Zou, Alex X. Lu, and Ava P. Amini. Protein structure generation via folding diffusion. *Nature Communications*, 15(1059), 2024. doi: 10.1038/s41467-024-45051-2. URL https://www.nature.com/articles/s41467-024-45051-2.

[24] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. *arXiv preprint arXiv:1905.11881*, 2020. doi: 10.48550/arXiv.1905.11881. URL https://arxiv.org/abs/1905.11881.

Appendix A. Theoretical Extension: Convergence Beyond the Squared Loss

The original convergence analysis for δ-GClip in Tucat et al. [21] is primarily situated within the regime of squared loss objectives, leveraging the PL* condition common in over-parameterized neural networks [14]. However, our empirical findings in Section 3 and Section 4 suggest that the stability benefits of the δ lower bound extend to more complex landscapes, such as those induced by cross-entropy and denoising score matching objectives. To bridge this gap, we draw upon the framework of Scaman et al. [17], which analyzes stochastic gradient descent under rather generic smoothness and curvature assumptions. Specifically, Scaman et al. [17] proposed the use of Kurdyka-Łojasiewicz (ϕ-KL*) inequalities as a generalization of the Polyak-Łojasiewicz condition, which allows for the analysis of convergence in much broader and more complex non-convex landscapes. We incorporate these inequalities into the convergence scheme for δ-GClip established by Tucat et al. [21], which allows us to extend the stability and convergence guarantees of δ-GClip to objectives beyond the squared loss. The rest of the reasoning is quite standard in optimization theory, we present it for the reader's convenience.

Recall that we consider the optimization of a differentiable objective function $f : \mathbb{R}^d \to \mathbb{R}$ using the δ-GClip update rule:

$$x_{t+1} = x_t - h(x_t)\nabla f(x_t), \quad h(x_t) := h_{\delta\text{-GClip}}(x_t) = \eta \min\left(1, \max\left(\delta, \gamma/\|\nabla f(x_t)\|\right)\right).$$

Here, $\eta > 0$ is the base learning rate, $\gamma > 0$ is the clipping threshold, and $\delta \in (0, 1]$ is the regularization parameter. Note that $\eta\delta \leq h(x_t) \leq \eta$.

Assumption 1 (β-smoothness) *The objective f is β-smooth, meaning its gradient ∇f is β-Lipschitz continuous. This implies for all $x, y \in \mathbb{R}^d$:*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\beta}{2}\|y - x\|^2$$

Under Assumption 1, the δ-GClip update satisfies the following inequality

$$f(x_{t+1}) \leq f(x_t) - \eta\left(1 - \frac{\beta\eta}{2}\right)\|\nabla f(x_t)\|^2, \quad \forall x_t \in \mathbb{R}^d.$$

It allows us to show that if $\eta < 2/\beta$ then $f(x_{t+1}) \leq f(x_t)$ for any $x_t \in \mathbb{R}^d$, and so given $S(x_0) := \{x \in \mathbb{R}^d \mid f(x) \leq f(x_0)\}$, a sub-level set defined by the initial point x_0, the sequence $\{x_t\}$ generated by δ-GClip remains in $S(x_0)$ for all $t \geq 0$.

Also, if η satisfies $C := (1 - \beta\eta/2) > 0$ (typically ensured by assuming $\eta < 1/\beta$), then using the lower bound for the effective step size, $h(x_t) \geq \eta\delta$, we get a stronger inequality

$$f(x_{t+1}) \leq f(x_t) - C\eta\delta\|\nabla f(x_t)\|^2, \tag{1}$$

which quantifies the guaranteed function decrease for a single iteration and ensures a minimum rate of progress proportional to δ.

Assumption 2 (Kurdyka-Łojasiewicz* Condition) *Extending the framework of Scaman et al. [17], we assume the existence of a non-decreasing function $\phi : \mathbb{R}^+ \to \mathbb{R}^+$ vanishing only at zero, such that:*

1. ***Global Case:*** *Let $f^* = \inf_{x\in\mathbb{R}^d} f(x)$ denote the global minimum value of f. For all $x \in \mathbb{R}^d$,*

$$\|\nabla f(x)\| \geq \phi(f(x) - f^*) \tag{2}$$

2. ***Local Case:*** *On a set $S \subset \mathbb{R}^d$, the objective function f satisfies*

$$\|\nabla f(x)\| \geq \phi(f(x)). \tag{3}$$

The KL* condition generalizes several standard landscape properties. The Global Case (2) is closely related to the Polyak-Łojasiewicz inequality, which has been shown to hold for over-parameterized neural networks in the Neural Tangent Kernel regime [14]. While the Local Case (3) represents a more standard assumption satisfied by a vast range of "tame" functions, including Logistic, MSE, and Softmax/Cross-Entropy objectives, which possess the KL* property on compact subsets of their domain [1, 4, 17]. In sufficiently over-parameterized models, the objective function f inherits these properties, allowing for the selection of a specific form of ϕ based on the underlying loss landscape.

We first establish the global convergence of δ-GClip iterations assuming existence of the global minimum f^*:

Theorem 1 (Global Convergence of δ-GClip) *Under Assumptions 1 and 2(2), if η is such that $C = (1 - \beta\eta/2) > 0$, the sequence $\{f(x_t)\}_{t\geq 0}$ converges to the global minimum f^*.*

Proof Summing (1) from $t = 0$ to T yields

$$\sum_{t=0}^{T} \|\nabla f(x_t)\|^2 \leq \frac{f(x_0) - f^*}{C\eta\delta}.$$

As $T \to \infty$, the series converges, implying $\lim_{t\to\infty} \|\nabla f(x_t)\| = 0$. By (2), $\phi(f(x_t) - f^*) \leq \|\nabla f(x_t)\|$. Since ϕ is non-decreasing and vanishes only at zero, it follows that $f(x_t) \to f^*$ as $t \to \infty$. ■

The rate of convergence is determined by the form of ϕ. If $\phi(s) = \sqrt{2\mu s}$ (the PL condition), the error $e_t := f(x_t) - f^*$ decays at a linear rate $O((1 - \rho)^t)$ with $\rho = 2\mu C\eta\delta$ [21]. For general growth conditions $\phi(s) = cs^\alpha$ where $\alpha \in (1/2, 1]$, the rate is sub-linear $O(t^{-1/(2\alpha-1)})$ [17]. In particular, for objectives like cross-entropy where $\alpha = 1$ as $e_t \to 0$, this yields a rate of $O(t^{-1})$.

While the Global Case of Assumption 1 implies convergence across the entire landscape, many practical loss surfaces only satisfy curvature constraints within restricted regions. By using the descent property established in (1) to ensure the sequence $\{x_t\}$ remains trapped within the sub-level set $S(x_0)$, we can establish convergence to the minimum value within that region.

Theorem 2 (Local Convergence of δ-GClip) *Suppose Assumptions 1 and 2 (3) hold on $S(x_0)$. If η is such that $C = (1 - \beta\eta/2) > 0$, the sequence $\{f(x_t)\}_{t\geq 0}$ converges to 0.*

Proof As we have shown above, $f(x_t)$ is monotonically non-increasing and bounded below in $S(x_0)$, thus there exist $f_\infty \in S(x_0)$ such that $f(x_t) \to f_\infty$ as $t \to \infty$. From descent inequality (1) and the local ϕ-KL* condition (3):

$$f(x_t) - f(x_{t+1}) \geq C\eta\delta\|\nabla f(x_t)\|^2 \geq C\eta\delta\phi(f(x_t))^2 \tag{4}$$

As $t \to \infty$, the difference $f(x_t) - f(x_{t+1})$ approaches zero, implying $\phi(f(x_t)) \to 0$. By the continuity of ϕ, we conclude $\phi(f_\infty) = 0$, which implies $f_\infty = 0$. ■

This result confirms that the δ lower bound acts as a critical stabilizer, preventing the vanishing gradients from stalling the optimization while maintaining the standard convergence guarantees of the KL framework.

Appendix B. Experimental Details

B.1. Image Classification (ResNet-18)

The classification experiments are conducted on the CIFAR-10 dataset using a standard ResNet-18 architecture trained with cross-entropy loss. The model is trained for 200 epochs with a batch size of 128. We employ a step-decay learning rate schedule, reducing the initial learning rate η by a factor of 10 at epochs 100 and 150.

All optimization runs use a weight decay of 5×10^{-4}. For the δ-GClip and GClip variants, the upper clipping bound is fixed at $\gamma = 1.0$. The specific configurations for all tested optimizers are summarized in Table 1.

Table 1: Key Hyperparameters for ResNet-18 experiments.

Hyperparameter	**Value**
Base Learning Rate (η)	0.1 (SGD) / 0.001 (Adam)
Batch Size	128
Total Epochs	200
LR Decay Milestones	[100, 150]
Upper Clipping Bound (γ)	0.1
δ-GClip Thresholds (δ)	$\{0.01, 0.03, 0.08, 0.2, 0.45, 0.9\}$

B.2. Variational Autoencoder (VAE)

The generative modeling experiments are conducted on the FashionMNIST dataset using a standard Variational Autoencoder (VAE) trained with the reconstruction + KL divergence objective. All models are trained for 100 epochs with a batch size of 128. Adam, SGD, GClip, and δ-GClip optimizers are evaluated under identical training conditions. For the clipping-based methods, we sweep the upper clipping bound $\gamma \in \{1, 30, 50\}$ and the δ thresholds $\{0.01, 0.03, 0.08, 0.2, 0.45, 0.9\}$. Gradient norms and RUM statistics are logged once per epoch to analyze the effect of clipping on update magnitudes.

B.3. Depth-Scaling Experiment (CIFAR-10, ResNet-8/20/32/44/50)

To study how the δ-GClip mechanism behaves as network depth increases, we conduct a depth-scaling experiment using the CIFAR-10 dataset and the standard CIFAR-ResNet family with depths $\{8, 20, 32, 44, 50\}$. All models follow the canonical architecture definitions for CIFAR-ResNets, using basic residual blocks without bottlenecks.

Each model is trained for 100 epochs with a batch size of 128. We use a step-decay learning rate schedule with an initial learning rate of $\eta = 0.1$ and a single decay by a factor of 10 at epoch 80. Weight decay is fixed at 5×10^{-4} for all runs.

For the clipping-based optimizers, we use an upper clipping bound of $\gamma = 0.1$ and sweep the same logarithmic-scale thresholds as in the main ResNet-18 experiment:

$$\delta \in \{0.01, 0.03, 0.08, 0.2, 0.45, 0.9\}.$$

All other training settings, including data preprocessing and augmentation, match those used in the ResNet-18 configuration described in Section B.1. Gradient norms, effective learning rates, and δ-activation statistics are logged once per epoch to analyze how the influence of δ evolves with increasing depth.

B.4. Diffusion Warm-Up Experiment

All diffusion experiments use a standard U-Net denoiser operating on 32×32 CIFAR-10 images under the DDPM framework with a linear β-schedule (1000 steps). The network follows a conventional six-level encoder–decoder structure with channel widths $(128, 128, 256, 256, 512, 512)$ and a single self-attention block at the 16×16 resolution. All models are trained for 50 epochs.

Training uses Adam with learning rate $\eta = 10^{-4}$, $(\beta_1, \beta_2) = (0.9, 0.999)$, batch size 128, and gradient clipping parameter $\gamma = 0.25$ for the GClip component. In warm-up configurations, δ-clipping is applied only during the first k epochs, after which training proceeds with standard Adam. The DDIM sampler with 100 steps is used for FID evaluation. All remaining implementation details follow standard diffusion-model practice.

The δ values and corresponding warm-up durations used in the study are listed in Table 2.

Table 2: Warm-up schedule: δ values and number of clipping epochs k.

δ	Warm-up epochs k
0.08	2
0.10	2
0.12	3
0.15	5
0.20	5
0.25	7
0.35	7
0.50	10
0.70	10
0.95	12

B.5. Protein Diffusion Experiment

All protein diffusion experiments use a lightweight Transformer-based denoiser operating on torsion-angle sequences (ϕ, ψ, ω) of length 64. The model follows a standard DDPM formulation with a linear β-schedule (300 steps) and predicts the noise $\varepsilon_\theta(x_t, t)$ under wrapped angular geometry. The dataset consists of a fixed, small collection of synthetic torsion sequences designed to emulate the multimodal structure of Ramachandran-like distributions while remaining explicitly data-constrained. This setting induces high gradient variance and optimization instability, making it a suitable testbed for evaluating the stabilizing effects of δ-clipping. All models are trained for 60 epochs.

Training uses Adam with learning rate $\eta = 10^{-3}$, $(\beta_1, \beta_2) = (0.9, 0.999)$, batch size 8, and gradient clipping parameter $\gamma = 0.25$ for the GClip component. Performance is evaluated using training loss and four distributional quality metrics (KL divergence, JS divergence, cluster L1, diversity ratio), which quantify fidelity and mode coverage of the generated torsion distributions.

B.5.1. FULL-STEP δ-CLIPPING: COMBINED CLIPPING AND ADAM

In the full-step configuration, δ-GClip is applied at every optimization step. At each update, the effective learning rate is rescaled according to the clipped ratio $h = \eta \cdot \min(1, \max(\delta, \gamma/\|g\|))$, after which Adam performs the parameter update. This setup exposes the direct interaction between the clipping threshold and the high-curvature, noise-amplifying dynamics characteristic of data-constrained protein diffusion training. The δ values used in this configuration are:

$$\delta \in \{0.01,\ 0.03,\ 0.08,\ 0.20,\ 0.45,\ 0.90\}.$$

All remaining implementation details follow standard diffusion-model practice.

B.5.2. WARM-UP δ-CLIPPING FOLLOWED BY ADAM

In the warm-up configuration, δ-GClip is applied only during the first k epochs, after which training proceeds with standard Adam at fixed learning rate $\eta = 10^{-3}$. The δ values and corresponding warm-up durations used in the study are listed in Table 3.

Table 3: Warm-up schedule for protein diffusion: δ values and number of clipping epochs k.

δ	Warm-up epochs k
0.01	2
0.03	4
0.08	6
0.20	8
0.45	10
0.90	13

Appendix C. Formal Definitions of Metrics

Here, we provide the mathematical formulations for the diagnostic metrics used to monitor the optimization dynamics of the δ-GClip method.

Effective Learning Rate In our algorithms, the effective learning rate η_{eff} is a stochastic counterpart of the step size $h(x_t)$ of δ-GClip, defined as:

$$\eta_{\text{eff},t} = \min\left(\eta, \eta \cdot \max\left(\delta, \frac{\gamma}{\|\mathbf{g}_t\|_2 + \epsilon}\right)\right) \tag{5}$$

where η is the base learning rate, γ is the upper clipping threshold, and δ is the lower-bound regularization parameter, and $\epsilon = 10^{-8}$ is a small constant added for numerical stability.

Relative Update Magnitude (RUM) The Relative Update Magnitude (RUM) quantifies the scale of the parameter update relative to the current weight norm. This metric allows for a normalized assessment of the step size across different stages of training. It is defined at step t as:

$$\text{RUM}_t = \frac{\|\eta_{\text{eff},t} \cdot \mathbf{g}_t\|_2}{\|\mathbf{x}_t\|_2 + \epsilon} \tag{6}$$

where $\mathbf{x}_t$ represents the model parameters, $\mathbf{g}_t$ is the stochastic gradient, $\eta_{\text{eff},t}$ is the effective learning rate after clipping.

Appendix D. Additional Results for ResNet-18

This appendix provides the full quantitative results for the ResNet-18 experiments, complementing the summary presented in Section 3. In addition to accuracy and loss, we report update-magnitude statistics and the δ-activation dynamics, which together offer a detailed view of how δ-GClip behaves across the full δ-sweep.

D.1. Full δ-Sweep Comparison

Table 4 summarizes the complete results across all optimizers and all tested δ values. The metrics include maximum and final test accuracy, final training loss, and both early-stage and overall RUM values. RUM quantifies the magnitude of parameter updates and therefore reflects the aggressiveness or conservativeness of the optimization process. As δ increases, RUM grows almost linearly, indicating that larger thresholds allow proportionally larger updates. Very small δ values enforce highly conservative steps, while very large values behave similarly to unclipped SGD. This monotonic relationship clarifies how δ governs the stability of the optimization trajectory.

The results show that $\delta = 0.2$ achieves the best balance between stability and learning speed: it produces substantially smaller RUM values than SGD while maintaining sufficiently large updates to avoid the stagnation observed in Adam and small-δ configurations. Larger thresholds such as $\delta = 0.45$ and $\delta = 0.9$ increase RUM to levels comparable to SGD, recovering its aggressive behavior and correspondingly higher final loss. Conversely, very small thresholds ($\delta \leq 0.03$) overly restrict the update magnitude, leading to slower convergence and reduced accuracy.

D.2. δ-Activation Dynamics

To better understand how the lower-bound threshold influences optimization, we compute the δ-activation signal for each experiment. For every epoch, we determine whether the update was governed by the δ constraint (i.e., whether $\delta > \gamma/\|g\|$). Table 5 reports (i) the number of epochs in which δ was active and (ii) the last epoch where activation occurred.

Table 4: Full comparison across all optimizers and δ values for ResNet-18. RUM Early is averaged over epochs 0–20, RUM Overall over all 200 epochs.

METHOD	MAX ACC	FINAL ACC	RUM EARLY	RUM OVERALL	FINAL LOSS
SGD_BASELINE	95.43	95.33	0.002879	0.001695	0.002107
DGCLIP_DELTA_0.2	95.22	95.22	0.000751	0.000595	0.001643
DGCLIP_DELTA_0.9	95.13	95.06	0.002692	0.001586	0.002132
DGCLIP_DELTA_0.45	94.98	94.86	0.001563	0.001014	0.001906
DGCLIP_DELTA_0.08	94.42	94.25	0.000381	0.000256	0.001154
DGCLIP_DELTA_0.03	93.86	93.66	0.000199	0.000108	0.000786
DGCLIP_DELTA_0.01	93.75	93.73	0.000128	0.000080	0.000740
STANDARD_GCLIP	93.46	93.46	0.000128	0.000080	0.000607
ADAM_BASELINE	93.19	92.97	0.000047	0.000033	0.008057

Table 5: δ-activation statistics across all dGClip configurations. "Active Epochs" counts how many epochs were governed by the δ lower bound; "Last Active" is the final epoch where δ was active.

METHOD	δ	ACTIVE EPOCHS	LAST ACTIVE
DGCLIP_DELTA_0.01	0.01	0	–
DGCLIP_DELTA_0.03	0.03	61	64
DGCLIP_DELTA_0.08	0.08	101	100
DGCLIP_DELTA_0.2	0.20	126	149
DGCLIP_DELTA_0.45	0.45	170	180
DGCLIP_DELTA_0.9	0.90	200	199

The activation statistics reveal a clear progression: as δ increases, the lower-bound constraint governs a larger fraction of training. For $\delta = 0.01$, the threshold is never active, and the method reduces to standard GClip. For $\delta = 0.03$, activation occurs intermittently during early training. For $\delta \geq 0.08$, the constraint becomes active for most epochs, and for $\delta \geq 0.45$ it dominates nearly the entire optimization process. This progression aligns with the RUM behavior in Table 4: more frequent activation corresponds to larger and more consistent update magnitudes.

The effective learning rate curves in Figure 1 further illustrate how δ shapes the optimization trajectory. Small thresholds produce rapidly decaying effective learning rates, mirroring the conservative behavior of Adam. Large thresholds maintain high effective learning rates throughout training, resembling SGD. The intermediate value $\delta = 0.2$ achieves a balanced profile: it prevents collapse into overly small steps while avoiding the instability associated with large, unclipped updates. This explains why $\delta = 0.2$ achieves the best overall performance in the main experiments.

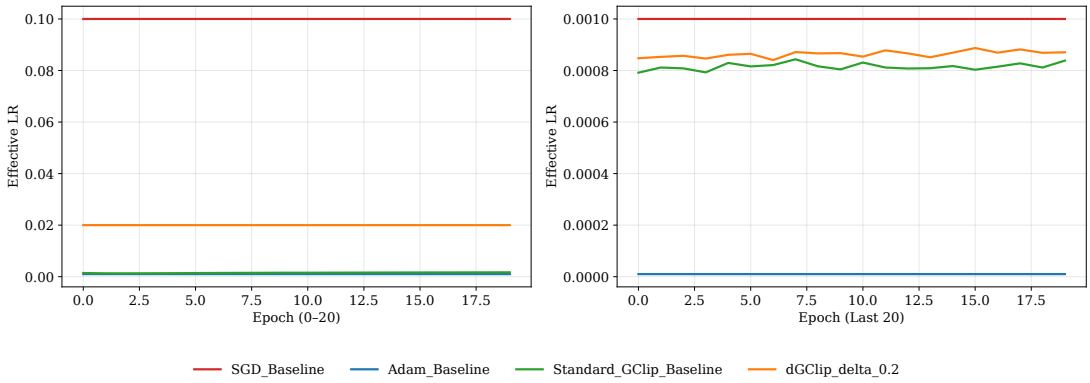


Figure 1: Effective learning rate dynamics during early training (left) and late training (right).

Appendix E. Additional Results for VAE on FashionMNIST

This appendix reports the full VAE results, including training loss, KLD, RUM statistics, and δ-activation metrics (Table 6). As discussed in Section 3, the gradient norms in this experiment are relatively small ($\|g\| \approx 12$–19), which makes δ-clipping mathematically impossible for $\gamma \in \{30, 50\}$ and all tested $\delta \leq 0.9$. For $\gamma = 1$, the ratio $\gamma/\|g\| \approx 0.07$ is sufficiently small for δ-clipping to occur, which happens only for $\delta \geq 0.08$.

The results show that for $\gamma \in \{30, 50\}$, δ–clipping is never activated, and the resulting behavior matches that of standard GClip. The training loss, KLD, and RUM statistics are nearly identical across different δ values, confirming that δ has no effect in these regimes. For $\gamma = 1$, δ–clipping becomes active only for $\delta \geq 0.08$, leading to consistently clipped updates and noticeably different loss values.

Appendix F. Additional Results for Diffusion Models

F.1. Baseline Diffusion Experiments

Table 7 reports the complete set of baseline diffusion-model experiments, including Adam, standard GClip, and the δ-GClip sweep. Adam and GClip were trained for 50 epochs, whereas all δ-GClip configurations were trained for 40 epochs. Despite this mismatch, several δ values achieve better generative behavior than standard GClip, indicating that δ-clipping provides a more effective early-stage optimization mechanism.

Moderate values such as $\delta = 0.15$ and $\delta = 0.6$ show consistently better convergence than GClip and exhibit higher early-stage update magnitudes (RUM), reflecting less aggressive gradient suppression and more effective exploration during the initial optimization phase.

F.2. Diffusion Warm-Up Experiment: Full Results

Table 8 reports the complete set of warm-up configurations evaluated in our diffusion experiments, including the optimizer variant, δ value, number of warm-up epochs, and the final training loss and FID after 50 epochs. These results correspond to the configurations described in Appendix B.4. For completeness, we also include the mean update magnitude (RUM) over training as a representative optimization statistic.

Table 6: Full VAE results across all dGClip configurations. Values shown as 0.0000 correspond to very small non-zero RUM magnitudes ($< 10^{-4}$) that round to zero at four decimal places.

γ	δ	Train	Test	KLD	RUM$_e$	RUM$_a$	δ–Act	Epochs
Reference Optimizers								
–	–	22.88	–	8.08	0.0004	0.0003	–	–
–	–	23.57	–	7.63	0.0004	0.0003	–	–
–	–	22.82	–	8.14	0.0004	0.0003	–	–
–	–	23.48	–	7.50	0.0004	0.0003	–	–
$\gamma = 1$								
1	0.01	27.26	–	6.35	0.0000	0.0000	0.00	0
1	0.03	27.26	–	6.46	0.0000	0.0000	0.00	0
1	0.08	26.75	–	6.67	0.0000	0.0000	0.99	99
1	0.20	25.04	–	7.14	0.0001	0.0001	1.00	100
1	0.45	24.06	–	7.56	0.0002	0.0002	1.00	100
1	0.90	23.55	–	7.49	0.0004	0.0003	1.00	100
$\gamma = 30$								
30	0	23.53	–	7.37	0.0004	0.0003	0.00	0
30	0.01	23.52	–	7.38	0.0004	0.0003	0.00	0
30	0.03	23.39	–	7.57	0.0004	0.0003	0.00	0
30	0.08	23.51	–	7.35	0.0004	0.0003	0.00	0
30	0.20	23.53	–	7.33	0.0004	0.0003	0.00	0
30	0.45	23.47	–	7.57	0.0004	0.0003	0.00	0
30	0.90	23.53	–	7.52	0.0005	0.0003	0.00	0
$\gamma = 50$								
50	0.01	23.39	–	7.66	0.0004	0.0004	0.00	0
50	0.03	23.38	–	7.72	0.0004	0.0003	0.00	0
50	0.08	23.39	–	7.72	0.0004	0.0003	0.00	0
50	0.20	23.68	–	7.49	0.0004	0.0003	0.00	0
50	0.45	23.52	–	7.40	0.0005	0.0004	0.00	0
50	0.90	23.52	–	7.42	0.0004	0.0004	0.00	0

RUM values are strictly positive; entries displayed as 0.0000 indicate non-zero magnitudes below the reporting precision.

Appendix G. Extended Results for Linking Optimal δ to Model Scale and Loss Landscape Sharpness

This appendix provides the extended experimental results supporting the analysis in Section 4. We organize the results into three parts: (i) depth scaling, (ii) width scaling, and (iii) sharpness measurements. Together, these results give a complete picture of how model scale influences the optimal threshold δ^*, the resulting optimization dynamics, and the curvature properties of the final solutions.

Table 7: Full baseline diffusion-model results for Adam, standard GClip, and the δ-GClip sweep. Adam and GClip were trained for 50 epochs; all δ-GClip models for 40 epochs.

Method	**FID**	**Final Loss**	**Grad Norm**	**RUM**$_{avg}$	**RUM**$_{10}$	**δ-Epochs**	**Last Active**
Adam	79.56	0.0309	0.1203	1.2×10^{-5}	1.6×10^{-5}	–	–
GClip	428.52	0.1385	0.8081	2.1×10^{-5}	2.5×10^{-5}	–	–
$\delta = 0.09$	415.88	0.1529	0.9704	2.3×10^{-5}	2.5×10^{-5}	0	–
$\delta = 0.15$	418.94	0.1499	0.9328	2.4×10^{-5}	2.9×10^{-5}	7	6
$\delta = 0.30$	420.15	0.1394	0.6636	2.7×10^{-5}	4.7×10^{-5}	13	12
$\delta = 0.60$	415.48	0.1297	0.3951	2.8×10^{-5}	6.5×10^{-5}	11	10
$\delta = 0.90$	437.64	0.1252	0.3091	2.9×10^{-5}	7.6×10^{-5}	10	9

Table 8: Warm-up diffusion experiment results sorted by FID. Adam baseline is highlighted.

Experiment	Method	δ	k (epochs)	Final Loss	FID	Mean RUM
d-gclip_delta0.15_clip5	δ-GClip	0.15	5	0.03070	**67.53**	4.72e-4
gclip_delta0_clip2	GClip	–	2	0.03068	69.44	4.72e-4
d-gclip_delta0.95_clip12	δ-GClip	0.95	12	0.03063	70.91	4.85e-4
d-gclip_delta0.50_clip10	δ-GClip	0.50	10	0.03064	71.79	4.85e-4
d-gclip_delta0.10_clip2	δ-GClip	0.10	2	0.03067	71.96	4.73e-4
d-gclip_delta0.08_clip2	δ-GClip	0.08	2	0.03068	73.27	4.73e-4
adam_delta0_clip0	**Adam**	–	0	**0.03093**	**75.90**	**4.91e-4**
d-gclip_delta0.25_clip7	δ-GClip	0.25	7	0.03099	76.19	4.79e-4
d-gclip_delta0.70_clip10	δ-GClip	0.70	10	0.03094	76.34	4.87e-4
d-gclip_delta0.20_clip5	δ-GClip	0.20	5	0.03097	77.89	4.80e-4
d-gclip_delta0.12_clip3	δ-GClip	0.12	3	0.03099	78.79	4.75e-4
d-gclip_delta0.35_clip7	δ-GClip	0.35	7	0.03097	79.06	4.84e-4

G.1. Depth-Scaling Results

Table 9 summarizes the optimal δ^* for each depth, along with the corresponding test accuracy, test loss, and a brief qualitative observation. These results highlight a clear trend: deeper networks consistently prefer larger values of δ, reflecting the decrease in gradient norms with depth and the resulting shift in the activation frequency of the δ-controlled lower bound.

Table 9: Summary of optimal δ^* across depths.

Depth	Optimal δ^*	Test Acc	Test Loss	Key Observation
8	**0.20**	0.8601	0.4184	Moderate lower bound is ideal.
20	**0.20**	0.9148	0.2817	Matches 0.45 accuracy but with lower loss.
32	**0.20**	0.9215	0.2938	Accuracy peaks at 0.20; loss slightly lower at 0.45.
44	**0.45**	0.9271	0.2735	Optimal δ shifts upward with depth.
50	**0.45**	0.9315	0.2598	Clear preference for higher δ.

The full quantitative results for all depths, including update-magnitude statistics and δ-activation behavior, are presented in Table 10. For clarity, the rows corresponding to the optimal δ^* at each depth are highlighted in bold.

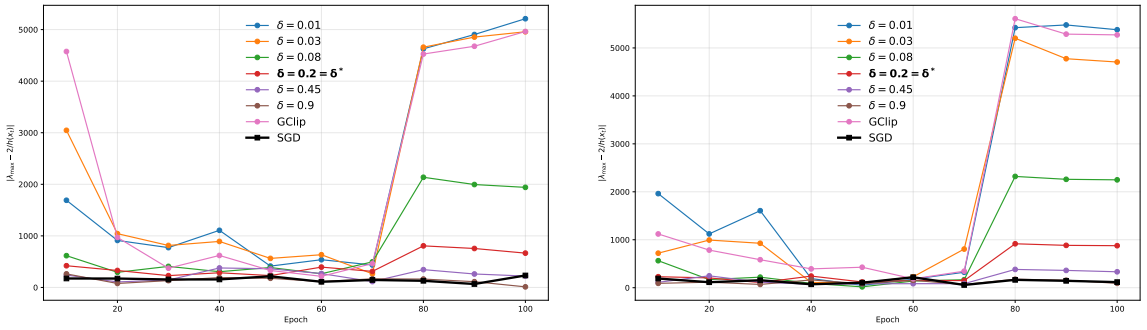
G.2. Width-Scaling Results

Table 11 reports the optimal threshold δ^* for each model width, together with a brief qualitative observation.

Table 12 reports the complete width-scaling results for all tested values of δ, grouped by model width. For each configuration, we include the final training and test metrics, the last-iteration gradient norm, the RUM statistic, and the average activation rate of the δ-controlled lower bound. The optimal δ for each width, as identified in Table 11, is highlighted in bold.

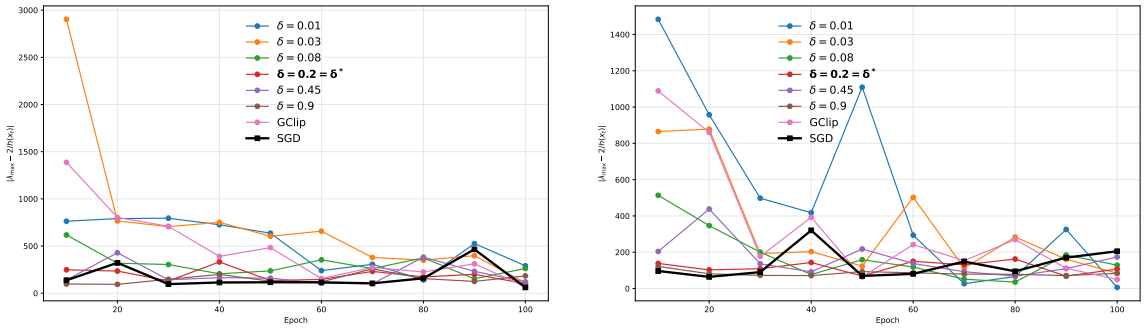
G.3. Sharpness Measurements

δ-Driven Proximity to the EoS. Table 13 reports the post-training sharpness statistics for all configurations, including the measured $\lambda_{\max}$, the effective learning rate $h(x_t)$, the corresponding EoS threshold $2/h(x_t)$, and the deviation $|\lambda_{\max} - 2/h(x_t)|$. These values summarize how closely each setting approaches the EoS boundary at convergence. To complement these endpoint measurements, Figure 2 and Figure 3 show the evolution of this deviation throughout training for depths 8 and 20, with and without learning rate scheduling respectively. We additionally evaluated a broader range of depths, and the same qualitative behavior consistently appears: larger δ values keep the effective step size from collapsing, reduce the extent of clipping, and maintain the optimization trajectory closer to the high-curvature EoS regime.



Depth 8 **Depth 20**

Figure 2: Sharpness deviation $|\lambda_{\max} - 2/h(x_t)|$ throughout training for two representative depths. Larger δ values consistently track the behavior of SGD, remaining close to the EoS boundary for the entire optimization trajectory, while smaller δ values behave similarly to standard GClip and stay farther from the EoS threshold. Note that a learning rate decay by a factor of 10 is applied at epoch 80, which causes a sharp increase in the EoS threshold $2/h$ and explains the visible spike in the deviation curves near the end of training. We observe the same qualitative pattern across all additional depths evaluated.



Depth 8 **Depth 20**

Figure 3: Sharpness deviation $|\lambda_{\max} - 2/h(x_t)|$ throughout training for two representative depths, trained with a constant learning rate (no scheduling). Without learning rate decay, the EoS threshold $2/h$ remains fixed throughout training, providing a stable reference for comparison. Larger δ values and SGD consistently stay closer to the EoS boundary across the entire optimization trajectory, while smaller δ values and standard GClip remain farther from the threshold. By the end of training, all methods converge to a similar deviation magnitude, yet the gap between large-δ methods and small-δ methods persists throughout the trajectory. We observe the same qualitative pattern across all additional depths evaluated.

Appendix H. Additional Results for Protein Diffusion Models

This appendix provides supplementary quantitative results for the protein diffusion experiments. Table 14 reports final training loss and average gradient norms for all optimization configurations, sorted by loss. Table 15 summarizes the distributional quality metrics (KL, JS, cluster L1, diversity ratio), sorted by JS divergence. These tables complement the main-text analysis by illustrating the relative performance of Adam, standard GClip, and all δ-GClip variants under the data-constrained regime.

Table 10: Full depth-scaling results for δ-GClip and GClip across all depths. RUM denotes the overall average Relative Update Magnitude across all epochs.

Depth	Method	δ	Test Acc	Test Loss	Train Loss	Grad Norm	RUM	δ Active Epochs
Depth 8								
8	dGClip	0.01	0.8413	0.4753	0.3742	2.91	0.00010	0
8	dGClip	0.03	0.8378	0.4779	0.3807	3.01	0.00010	10
8	dGClip	0.08	0.8573	0.4265	0.3343	2.31	0.00018	100
8	**dGClip**	**0.20**	**0.8601**	**0.4184**	**0.3274**	**1.85**	**0.00032**	**100**
8	dGClip	0.45	0.8535	0.4323	0.3337	1.60	0.00060	100
8	dGClip	0.90	0.8481	0.4509	0.3599	1.51	0.00111	100
8	GClip	–	0.8371	0.4737	0.3844	3.04	0.00010	0
Depth 20								
20	dGClip	0.01	0.8876	0.3724	0.0925	3.20	0.00009	0
20	dGClip	0.03	0.8919	0.3716	0.0824	2.93	0.00009	79
20	dGClip	0.08	0.9054	0.3160	0.0667	2.05	0.00015	100
20	**dGClip**	**0.20**	**0.9148**	**0.2817**	**0.0695**	**1.66**	**0.00028**	**100**
20	dGClip	0.45	0.9148	0.2818	0.0923	1.67	0.00060	100
20	dGClip	0.90	0.9078	0.2989	0.1277	1.75	0.00127	100
20	GClip	–	0.8901	0.3707	0.0999	3.41	0.00009	0
Depth 32								
32	dGClip	0.01	0.8985	0.3850	0.0477	3.07	0.00008	0
32	dGClip	0.03	0.8989	0.3916	0.0410	2.62	0.00009	79
32	dGClip	0.08	0.9124	0.3308	0.0319	1.76	0.00014	100
32	**dGClip**	**0.20**	**0.9215**	**0.2938**	**0.0420**	**1.67**	**0.00028**	**100**
32	dGClip	0.45	0.9190	0.2806	0.0529	1.61	0.00057	100
32	dGClip	0.90	0.9159	0.2909	0.0928	1.84	0.00130	100
32	GClip	–	0.8942	0.4046	0.0500	3.13	0.00008	0
Depth 44								
44	dGClip	0.01	0.8951	0.4403	0.0379	3.17	0.00007	0
44	dGClip	0.03	0.9029	0.4058	0.0259	2.31	0.00009	79
44	dGClip	0.08	0.9125	0.3488	0.0232	1.64	0.00014	100
44	dGClip	0.20	0.9240	0.3048	0.0292	1.56	0.00027	100
44	**dGClip**	**0.45**	**0.9271**	**0.2735**	**0.0401**	**1.56**	**0.00055**	**100**
44	dGClip	0.90	0.9160	0.2946	0.0776	1.86	0.00129	100
44	GClip	–	0.8932	0.4342	0.0373	3.11	0.00007	0
Depth 50								
50	dGClip	0.01	0.8927	0.4558	0.0316	3.07	0.00007	0
50	dGClip	0.03	0.8994	0.4203	0.0254	2.36	0.00009	79
50	dGClip	0.08	0.9197	0.3308	0.0179	1.42	0.00014	100
50	dGClip	0.20	0.9234	0.3036	0.0238	1.44	0.00026	100
50	**dGClip**	**0.45**	**0.9315**	**0.2598**	**0.0338**	**1.50**	**0.00053**	**100**
50	dGClip	0.90	0.9197	0.2887	0.0736	1.89	0.00131	100
50	GClip	–	0.8960	0.4501	0.0302	2.97	0.00007	0

Table 11: Optimal δ^* for each model width, together with the corresponding test loss and test accuracy.

Width	δ^*	**Test loss**	**Test acc**	**Observation**
1	0.20	0.27830	0.9199	Moderate threshold outperforms both small and very large values.
2	0.45	0.21411	0.9438	Clear improvement over smaller δ; large δ slightly worse.
4	0.45	0.19068	0.9517	Stable and consistently strongest among tested thresholds.
8	0.45	0.17104	0.9576	Best-performing configuration across all δ.
10	0.45	0.17039	0.9560	Matches width 8; large δ remains competitive but weaker.

Table 12: Full width-scaling results for all tested δ values. Optimal δ for each width is shown in bold.

Width	Method	δ	Test Acc	Test Loss	Train Loss	Grad Norm	RUM	δ Active Epochs
Width 1								
1	dgclip	0.01	0.8985	0.36394	0.07360	4.07573	0.00008336	0.0
1	dgclip	0.03	0.9005	0.35648	0.05931	3.65724	0.00009195	0.7136
1	dgclip	0.08	0.9141	0.33174	0.04513	2.57008	0.00014724	0.9790
1	**dgclip**	**0.20**	**0.9199**	**0.27830**	**0.05420**	**1.91045**	**0.00029314**	**0.9997**
1	dgclip	0.45	0.9160	0.29753	0.07109	1.50015	0.00061401	1.0
1	dgclip	0.90	0.9104	0.28577	0.11324	1.24908	0.00130746	1.0
1	gclip	NaN	0.8913	0.37581	0.07007	4.06753	0.00008396	0.0
Width 2								
2	dgclip	0.01	0.9120	0.35298	0.01376	4.16091	0.00011399	0.0
2	dgclip	0.03	0.9206	0.32790	0.01008	3.49569	0.00014420	0.6532
2	dgclip	0.08	0.9321	0.28997	0.00845	2.42790	0.00020873	0.8522
2	dgclip	0.20	0.9379	0.24846	0.01004	1.85928	0.00018987	0.9508
2	**dgclip**	**0.45**	**0.9438**	**0.21411**	**0.01599**	**1.49176**	**0.00034115**	**0.9979**
2	dgclip	0.90	0.9285	0.26634	0.04331	1.27661	0.00108188	1.0
2	gclip	NaN	0.9134	0.35492	0.01394	4.10394	0.00010643	0.0
Width 4								
4	dgclip	0.01	0.9193	0.36234	0.00779	4.20892	0.00014318	0.000026
4	dgclip	0.03	0.9296	0.30925	0.00564	3.38675	0.00016814	0.5977
4	dgclip	0.08	0.9342	0.26829	0.00502	2.31946	0.00031604	0.8086
4	dgclip	0.20	0.9473	0.22122	0.00426	1.81910	0.00024494	0.8751
4	**dgclip**	**0.45**	**0.9517**	**0.19068**	**0.00671**	**1.47672**	**0.00025905**	**0.9704**
4	dgclip	0.90	0.9442	0.22672	0.02013	1.26338	0.00084049	0.9998
4	gclip	NaN	0.9198	0.35170	0.00737	4.21326	0.00013060	0.0
Width 8								
8	dgclip	0.01	0.9265	0.31946	0.00662	4.26869	0.00013223	0.00023
8	dgclip	0.03	0.9318	0.29279	0.00487	3.29159	0.00015990	0.5499
8	dgclip	0.08	0.9415	0.24619	0.00516	2.23684	0.00032469	0.7811
8	dgclip	0.20	0.9486	0.20677	0.00335	1.78003	0.00030560	0.8474
8	**dgclip**	**0.45**	**0.9576**	**0.17104**	**0.00377**	**1.46702**	**0.00022662**	**0.9255**
8	dgclip	0.90	0.9502	0.18513	0.01054	1.24850	0.00062084	0.9980
Width 10								
10	gclip	NaN	0.9272	0.33388	0.00618	4.30822	0.00012068	0.0
10	dgclip	0.01	0.9292	0.32118	0.00630	4.30213	0.00012362	0.00043
10	dgclip	0.03	0.9305	0.31426	0.00396	3.27107	0.00014280	0.5388
10	dgclip	0.08	0.9408	0.25107	0.00474	2.21650	0.00034404	0.7742
10	dgclip	0.20	0.9519	0.18718	0.00315	1.76415	0.00030938	0.8372
10	**dgclip**	**0.45**	**0.9560**	**0.17039**	**0.00360**	**1.47360**	**0.00022757**	**0.9161**
10	dgclip	0.90	0.9546	0.17280	0.00890	1.24667	0.00055366	0.9960

Table 13: Table 13: Sharpness and EoS comparison across depths, grouped by δ. All reported quantities are measured after training has completed. Columns list the final sharpness $\lambda_{\max}$, the effective learning rate $h(x_t)$, the corresponding EoS threshold $2/h(x_t)$, and the absolute deviation $|\lambda_{\max} - 2/h(x_t)|$.

Depth	δ	$\lambda_{\max}$	$h(x_t)$	$2/h(x_t)$	$\lvert\lambda_{\max} - 2/h(x_t)\rvert$]
			$\delta = 0.01$		
8	0.01	948.72	3.47e-4	5755.96	4807.24
20	0.01	2160.31	3.28e-4	6098.65	3938.33
32	0.01	2509.18	3.64e-4	5500.41	2991.22
44	0.01	996.14	3.76e-4	5317.47	4321.33
50	0.01	2005.20	4.01e-4	4981.40	2976.19
			$\delta = 0.03$		
8	0.03	1071.75	3.39e-4	5898.25	4826.51
20	0.03	1409.80	3.65e-4	5477.70	4067.90
32	0.03	1855.42	4.45e-4	4491.97	2636.55
44	0.03	1352.95	5.34e-4	3743.15	2390.20
50	0.03	443.25	5.78e-4	3462.83	3019.58
			$\delta = 0.08$		
8	0.08	553.06	8.00e-4	2500.00	1946.94
20	0.08	553.78	8.09e-4	2471.80	1918.03
32	0.08	548.16	9.14e-4	2189.11	1640.95
44	0.08	545.54	9.89e-4	2021.32	1475.79
50	0.08	498.52	1.16e-3	1727.75	1229.23
			$\delta = 0.20$		
8	0.20	390.38	2.00e-3	1000.00	609.62
20	0.20	309.58	2.00e-3	999.97	690.39
32	0.20	456.89	2.00e-3	999.01	542.12
44	0.20	523.20	2.03e-3	986.06	462.86
50	0.20	242.24	2.08e-3	959.52	717.28
			$\delta = 0.45$		
8	0.45	258.93	4.50e-3	444.44	185.51
20	0.45	177.22	4.50e-3	444.44	267.22
32	0.45	222.80	4.50e-3	444.44	221.64
44	0.45	571.27	4.50e-3	444.44	126.82
50	0.45	175.98	4.50e-3	444.44	268.46
			$\delta = 0.90$		
8	0.90	215.48	9.00e-3	222.22	6.75
20	0.90	197.81	9.00e-3	222.22	24.41
32	0.90	117.87	9.00e-3	222.22	104.35
44	0.90	202.46	9.00e-3	222.22	19.76
50	0.90	155.03	9.00e-3	222.22	67.19

Table 14: Final loss and average gradient norm for all methods (sorted by loss).

Method	δ	Final loss	Avg. grad. norm
Adam+δ-GClip	0.90	0.3616	0.6331
Adam+δ-GClip	0.45	0.3674	0.6705
Adam	–	**0.3689**	**0.6462**
Adam+δ-GClip	0.08	0.3710	0.7863
Adam+δ-GClip	0.20	0.3717	0.7617
Adam+δ-GClip	0.03	0.3752	0.8115
Adam+GClip	–	**0.3828**	**0.8257**
Adam+δ-GClip	0.01	0.3828	0.8257

Table 15: Quality metrics for generated torsion distributions (sorted by JS divergence).

Method	δ	KL	JS	cluster L1	Diversity ratio
Adam+δ-GClip	0.90	0.2822	0.0706	0.1019	0.9783
Adam+δ-GClip	0.45	0.4955	0.1202	0.2742	1.2032
Adam+δ-GClip	0.01	0.5172	0.1332	0.2284	1.2017
Adam+GClip	–	**0.5244**	**0.1355**	**0.2308**	**1.2043**
Adam	–	**0.6591**	**0.1735**	**0.3576**	**1.4423**
Adam+δ-GClip	0.20	2.9229	0.2516	0.2474	0.5739
Adam+δ-GClip	0.03	3.4238	0.2810	0.1071	0.4686
Adam+δ-GClip	0.08	3.8079	0.2822	0.2303	0.4681