Edge of Stability Selectively Shapes Learning Across the Data Distribution

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2026

Abstract

Existing analyses of the edge of stability (EoS) treat it as a global property of optimization. We show that it is also selective: the stability constraint redistributes learning across subsets of the training distribution, amplifying progress on some groups while suppressing progress on others. Using a branching intervention that enters or exits the EoS regime from the same training state, we causally demonstrate this trade-off and identify two necessary conditions for a group to benefit. First, its aggregate gradient must align with the top Hessian eigenvector. We isolate this mechanism with a controlled perturbation that preserves distance but randomizes direction, destroying alignment and eliminating the advantage. Second, the group must sustain non-vanishing gradient magnitude over time. Under cross-entropy loss, gradient saturation decouples confidently classified groups, shifting the advantage to output-outliers, whose gradients persist. Together, these results show that EoS functions not only as a stability boundary, but as a mechanism governing the allocation of learning across the data distribution.

1. Introduction

Deep neural networks exhibit strong sensitivity to optimizer and hyperparameters. Training choices such as learning rate, batch size, and optimizer affect which solution is found [19, 22, 39], unlike in the classical convex setting where these choices do not affect which minimum is reached. Understanding the mechanisms underlying this implicit bias is a key objective in the theory of deep learning.

One structural explanation comes from the *edge of stability* (EoS) literature: under full-batch and large-batch gradient descent, the top Hessian eigenvalue self-stabilizes near the stability threshold that depends on the optimizer and hyperparameters [3, 5, 8, 9, 16, 18, 20, 38]. At this threshold, the optimizer operates at the boundary of discrete-time stability, constraining which regions of the loss landscape remain accessible during training.

While the EoS phenomenon is well established, far less is understood about its consequences. In particular, it remains unclear whether operating near the stability threshold provides any functional benefit, or how these stability constraints shape optimization across the data distribution. Prior work has primarily characterized EoS through curvature and optimization trajectories in parameter space, leaving open how these dynamics influence which examples are learned during training. We ask:

Which subsets of the training distribution benefit from EoS?
Which do not? What governs this allocation?

To study this allocation, we define four prototype groups from the input geometry that vary in input typicality, label consistency, and boundary proximity, independent of the model or loss. We find

that EoS induces a *selective* learning regime: the stability constraint distributes optimization effort unevenly, amplifying subsets whose gradients persistently align with the top Hessian eigendirection while suppressing others. These findings refine the role of curvature in optimization [11, 14, 15, 22]. It also counters classical intuition: although the Descent Lemma treats the sharpest direction as a stability boundary that constrains learning, alignment with it is precisely what determines how learning is allocated across the distribution.

Our paper makes the following contributions:

- **EoS is selective, not global** (§3). The stability constraint is not a uniform bottleneck.
- **Alignment $\times$ persistence governs selectivity** (§2.2, §4, §I). Curvature predicts EoS benefit.
- **Geometry shifts the beneficiary** (§D). Varying the dataset geometry shifts the data subset that benefits from EoS, with preliminary single-seed evidence that the test-time advantage shifts accordingly between adversarial robustness and generalization to displaced inputs.

2. Setup: Measuring and Intervening on EoS Selectivity

2.1. Prototype Groups

We construct four prototype groups directly from the input-space data distribution, based on geometric properties relative to the class centroid (Figure 1): *inliers* (closest to the centroid), *boundary points* (k-NN label-ambiguous), *output-outliers* (inliers with flipped labels), and *input-outliers* (inliers extrapolated away from the opposite class's centroid). Construction details are in Appendix B.3.

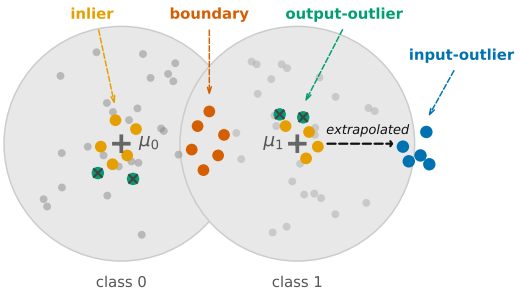

Figure 1: **Conceptual taxonomy of prototypes.** Data samples are categorized based on geometric proximity in input space relative to class-specific cluster centroids (μ_0, μ_1).

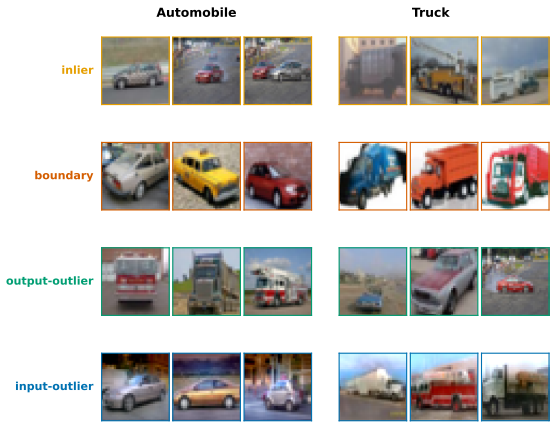

Figure 2: **Input-space visualization of prototype groups for CIFAR-10.** Three representative samples per class (automobile vs. truck) are shown.

2.2. Measurements

We study full-batch gradient descent, $\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t)$, with Hessian $H_t = \nabla^2 L(\theta_t)$ and top eigenpair (λ_1, v_1). The edge of stability (EoS) is the regime in which the sharpness approaches

the discrete-time stability threshold, $\eta\lambda_1 \approx 2$ [8]. At this boundary, updates develop a period-two oscillation along the top eigendirection v_1. Self-stabilization theory predicts that the cycle-averaged dynamics then include a sharpness-reducing correction, so a subset is affected by EoS in proportion to how strongly its gradient couples to the sharpness-controlling direction [10]. This motivates measuring, for each prototype group k with loss ℓ_k, its coupling to the unstable mode.

We track two quantities. First, the directional alignment

$$\cos^2 \theta_k = \frac{(\nabla \ell_k \cdot v_1)^2}{\|\nabla \ell_k\|^2} \tag{1}$$

measures whether group k points along the EoS-constrained direction. Second, the curvature influence

$$C_k = (\nabla \ell_k \cdot v_1)^2 = \|\nabla \ell_k\|^2 \cos^2 \theta_k \tag{2}$$

combines alignment with gradient magnitude. Thus C_k can fall either because the group rotates away from v_1 or because its gradient vanishes. Appendix I relates this single-mode statistic to the subset-level self-stabilization selector $\langle \nabla \ell_k, \nabla \lambda_1 \rangle$, and Appendix A gives full definitions on metric construction.

2.3. Intervention

To test whether EoS causally changes subset-level learning, we use a matched branching intervention. Each run is trained until the first time t^* at which λ_1 reaches $2/\eta$. From the same checkpoint, the *baseline* branch continues at learning rate η and remains at EoS, while the *exit* branch halves the learning rate, raising the stability threshold and leaving the EoS regime until its EoS at t^{**}. Since the two branches share data, initialization, architecture, and trajectory up to t^*, post-branch differences in the prototype losses $\ell_k = \frac{1}{|P_k|} \sum_{i \in P_k} \ell(f_\theta(x_i), y_i)$ isolate the effect of remaining at the stability boundary. We use controlled interventions to test the two factors in Equation (2): random-direction displacement removes alignment, while cross-entropy saturation removes gradient persistence.

3. Selective Learning at the Edge of Stability

3.1. The Selective Trade-off

Figure 3 shows the effect of the branching intervention under MSE training. After the exit branch leaves the EoS regime at t^*, prototype losses begin to diverge between the two runs. The divergence is group-specific: input-outlier and output-outlier loss decrease faster under the baseline ($\Delta \ell_k > 0$), while inlier and boundary loss decrease faster under the exit branch ($\Delta \ell_k < 0$). The stability constraint does not uniformly slow or accelerate learning across the data subsets; instead, it redistributes optimization, concentrating progress on some groups at the expense of others. The selective trade-off replicates across alternative architectures (CNN, ResNet), optimizers, and class pair (Appendix E). This trade-off raises a natural question: what determines which group benefits from EoS?

4. Mechanism: Alignment and Gradient Persistence

Two properties are jointly necessary to capture the EoS advantage: directional alignment and gradient persistence. We isolate each experimentally.

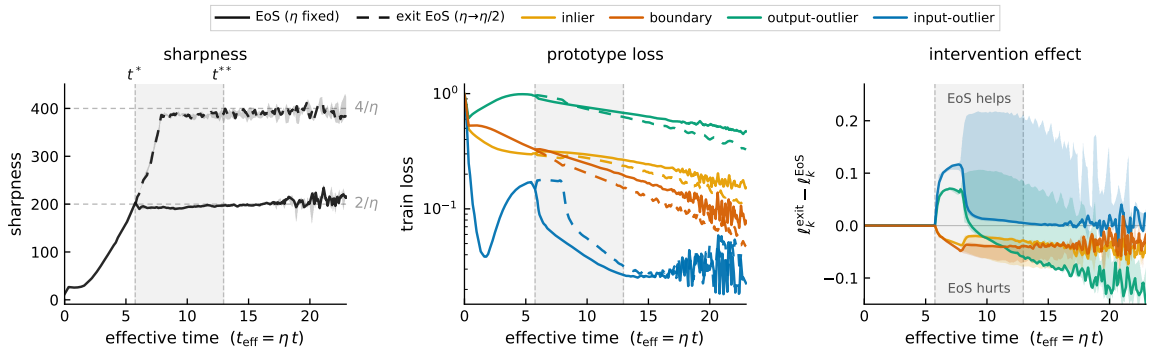

Figure 3: **EoS creates selective trade-offs across prototype groups.** Baseline run (solid) enters EoS at t^*, while the exit branch (dashed) subsequently leaves. **Left:** sharpness confirms the two branches occupy distinct stability regimes. **Middle:** prototype losses diverge post-branch, with input-outliers and output-outliers benefiting from EoS while boundary and inlier progress is suppressed. **Right:** the intervention effect $\Delta\ell_k = \ell_k^{\text{exit}} - \ell_k^{\text{EoS}}$ is shown where positive values indicate EoS achieves lower loss for that group.

4.1. Directional alignment.

In the baseline construction, input-outliers are displaced along a shared direction v_{diff}, yielding a directionally coherent group. To test whether this coherence drives their curvature dominance, we construct a counterfactual in which each input-outlier is displaced by the same distance but in a random direction orthogonal to v_{diff}. The two conditions match in centroid distance, group size, and labels, differing only in directional structure (Appendix F).

Under coherent displacement along v_{diff}, input-outlier's $\cos^2\theta_k$ and curvature influence $(\nabla\ell_k \cdot v_1)^2$ dominate (Figure 4). The input-outliers benefit from EoS while the progress of other groups is suppressed. Under incoherent displacement, input-outlier alignment collapses, curvature influence falls by an order of magnitude, and the selective intervention effect dissolves. Geometric atypicality alone—large distance from its own class centroid—is not sufficient. Natural CIFAR-10 points show the same emergence of distance–alignment coupling near EoS onset (Appendix G). A group benefits at EoS when its per-example gradients share a coherent direction that aligns with $\mathbf{v}_1$.

4.2. Gradient persistence.

Directional coherence determines *which direction* a group pushes the Hessian, but the coupling also requires sustained gradient *magnitude*. We test this by comparing MSE and CE training on identical data (Figure 5).

Under MSE as the specified loss function, per-example gradients scale with the residual and persist even for confidently classified points. Conversely, under CE, gradients vanish as points are learned and confidence grows. The experiment reveals that while the gradient for input-outliers points towards $\mathbf{v}_1$ across both loss functions, $(\nabla\ell_k \cdot v_1)^2$ collapses by orders of magnitude in CE as gradient norms shrink. The functional consequence is that output-outliers, with the most elevated curvature influence, becomes the sole beneficiary at the edge of stability.

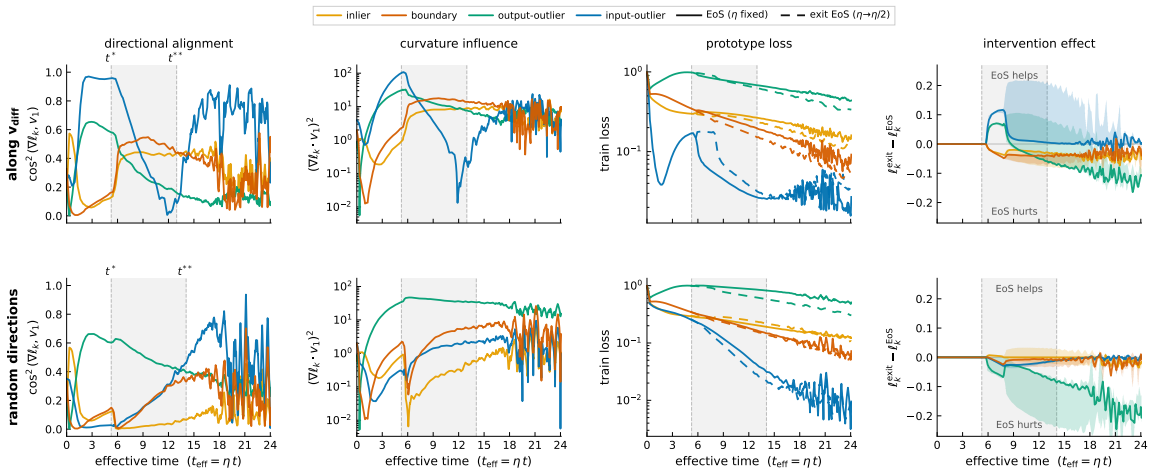

Figure 4: **Directional alignment is necessary for the selective EoS advantage.** Only the input-outlier displacement direction differs. Coherent displacement yields high alignment, high curvature influence, and an EoS input-outlier advantage. Random displacement collapses alignment and eliminates the advantage.

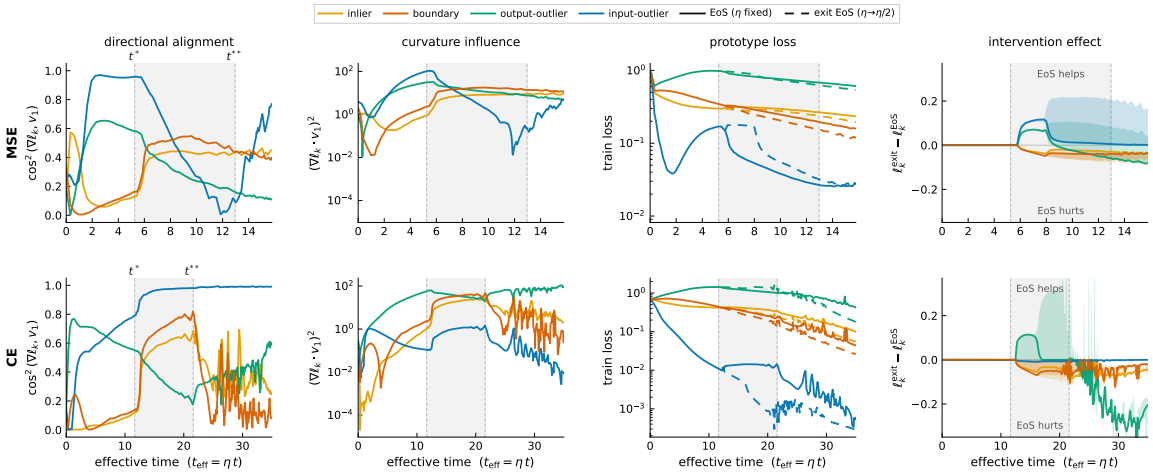

Figure 5: **Gradient persistence determines which group retains curvature influence.** Only loss differs. Under MSE, input-outliers retain curvature influence and benefit from EoS. Under CE, their gradients saturate, C_k collapses, and the EoS advantage shifts to output-outliers.

5. Discussion

We show that the edge of stability acts as a selective mechanism over the data distribution. A branching intervention from a shared training state reveals that remaining at EoS does not uniformly improve or suppress learning, but redistributes progress across prototype groups. The beneficiary is predicted by curvature influence, $(\nabla \ell_k \cdot v_1)^2$ (alignment with the top Hessian eigendirection and persistence of gradient magnitude). Controlled interventions confirm both factors are necessary. These results suggest that, in controlled full-batch settings, EoS is not merely a stability boundary but a mechanism that determines which subsets receive concentrated optimization.

References

[1] Kwangjun Ahn, Jingzhao Zhang, and Suvrit Sra. Understanding the unstable convergence of gradient descent. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 247–257. PMLR, 2022. URL https://proceedings.mlr.press/v162/ahn22a.html.

[2] Kwangjun Ahn, Sébastien Bubeck, Sinho Chewi, Yin Tat Lee, Felipe Suarez, and Yi Zhang. Learning threshold neurons via edge of stability. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/3e592c571de69a43d7a870ea89c7e33a-Abstract-Conference.html.

[3] Arseniy Andreyev and Pierfrancesco Beneventano. Edge of stochastic stability: Revisiting the edge of stability for SGD. *arXiv preprint arXiv:2412.20553*, 2024. URL https://arxiv.org/abs/2412.20553.

[4] Arseniy Andreyev and Pierfrancesco Beneventano. Edge of stochastic stability, 2025. URL https://github.com/arseniqum/edge-of-stochastic-stability. Software, Apache 2.0 license.

[5] Arseniy Andreyev, Advikar Ananthkumar, Marc Walden, Tomaso Poggio, and Pierfrancesco Beneventano. Momentum further constrains sharpness at the edge of stochastic stability. *arXiv preprint arXiv:2604.14108*, 2026. doi: 10.48550/arXiv.2604.14108.

[6] Sanjeev Arora, Zhiyuan Li, and Abhishek Panigrahi. Understanding gradient descent on the edge of stability in deep learning. In *International Conference on Machine Learning*, pages 948–1024. PMLR, 2022.

[7] Lei Chen and Joan Bruna. Beyond the edge of stability via two-step gradient updates. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 4330–4391. PMLR, 2023. URL https://proceedings.mlr.press/v202/chen23b.html.

[8] Jeremy 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.

[9] Jeremy Cohen, Behrooz Ghorbani, Shankar Krishnan, Naman Agarwal, Sourabh Medapati, Michal Badura, Daniel Suo, Zachary Nado, George E. Dahl, and Justin Gilmer. Adaptive gradient methods at the edge of stability. In *NeurIPS 2023 Workshop on Heavy Tails in Machine Learning: Structure, Stability, and Dynamics*, 2023. URL https://openreview.net/forum?id=dHGNgkUcGd.

[10] Alex Damian, Eshaan Nichani, and Jason D. Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *International Conference on Learning Representations*, 2023.

[11] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, volume 70, pages 1019–1028. PMLR, 2017.

[12] Vitaly Feldman. Does learning require memorization? A short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959. ACM, 2020. doi: 10.1145/3357713.3384290.

[13] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1e14bfe2714193e7af5abc64ecbd6b46-Abstract.html.

[14] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

[16] Rustem Islamov, Michael Crawshaw, Jeremy Cohen, and Robert Gower. Non-euclidean gradient descent operates at the edge of stability. *arXiv preprint arXiv:2603.05002*, 2026. doi: 10.48550/arXiv.2603.05002. URL https://arxiv.org/abs/2603.05002.

[17] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence*, 2018. URL https://arxiv.org/abs/1803.05407.

[18] Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof Geras. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1g87C4KwB. arXiv:2002.09572.

[19] Stanislaw Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, September 2018.

[20] Stanisław Jastrzębski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. On the relation between the sharpest directions of DNN loss and the SGD step length. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkgEaj05t7. arXiv:1807.05031.

[21] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgIPJBFvH.

[22] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

[25] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL http://www.cs.toronto.edu/~kriz/cifar.html.

[26] Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: The catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

[27] Kaifeng Lyu, Zhiyuan Li, and Sanjeev Arora. Understanding the generalization benefit of normalization layers: Sharpness reduction. In *Advances in Neural Information Processing Systems*, volume 35, pages 34689–34708, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/dffd1c523512e557f4e75e8309049213-Abstract-Conference.html.

[28] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[29] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/10ce03a1ed01077e3e289f3e53c72813-Abstract.html.

[30] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, volume 34, 2021. doi: 10.48550/arXiv.2107.07075. arXiv:2107.07075.

[31] Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/0987b8b338d6c90bbedd8631bc499221-Abstract.html.

[32] Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. doi: 10.1016/0041-5553(64)90137-5.

[33] Elan Rosenfeld and Andrej Risteski. Outliers with opposing signals have an outsized effect on neural network optimization. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=kIZ3S3tel6.

[34] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: Beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://openreview.net/forum?id=UmvSlP-PyV.

[35] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Empirical Methods in Natural Language Processing*, pages 9275–9293. Association for Computational Linguistics, 2020. URL https://aclanthology.org/2020.emnlp-main.746.

[36] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

[37] Lei Wu, Zhanxing Zhu, and Weinan E. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017. URL https://arxiv.org/abs/1706.10239.

[38] Chen Xing, Devansh Arpit, Christos Tsirigotis, and Yoshua Bengio. A walk with sgd. *arXiv preprint arXiv:1802.08770*, 2018. doi: 10.48550/arXiv.1802.08770. URL https://arxiv.org/abs/1802.08770.

[39] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. doi: 10.48550/arXiv.1611.03530. arXiv:1611.03530.

[40] Xingyu Zhu, Zixuan Wang, Xiang Wang, Mo Zhou, and Rong Ge. Understanding edge-of-stability training dynamics with a minimalist example. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=p7EagBsMAEO.

Appendix A. EoS setup

A.1. Edge of stability

Gradient descent in deep networks often operates at the *edge of stability* (EoS), a regime in which training remains near the boundary between stable and unstable updates without diverging [8]. Consider full-batch gradient descent $\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t)$ with Hessian $H_t = \nabla^2 L(\theta_t)$ and eigenpairs (λ_i, v_i). A perturbation along v_i is multiplied by $1 - \eta\lambda_i$ at each step, so discrete-time stability requires $\eta\lambda_i < 2$. EoS is the regime in which the top eigenvalue λ_1 (sharpness) saturates the bound:

$$\eta\lambda_1 \approx 2.$$

The multiplier along the corresponding eigendirection $\mathbf{v}_1$ is then near -1, producing sign-alternating oscillations that are linearly unstable yet remain bounded throughout training [8].

A.2. Metrics

Per group loss ℓ_k. For each prototype group $k \in \{\text{inlier, boundary, input-outlier, output-outlier}\}$ with index set P_k, we define $\ell_k = \frac{1}{|P_k|} \sum_{i \in P_k} \ell(f(x_i), y_i)$ as the average loss over the examples in group k. Tracking ℓ_k over training reveals the learning order across the groups and which groups are differentially affected by the stability constraint at EoS.

Directional coupling. Let $\nabla \ell_k$ denote the gradient of the loss restricted to prototype group k. We measure the directional coupling of group k to the EoS-constrained mode by

$$\cos^2\theta_k \;=\; \frac{(\nabla \ell_k \cdot v_1)^2}{\|\nabla \ell_k\|^2} \;\in\; [0, 1], \tag{3}$$

where v_1 is the top Hessian eigenvector (assumed unit norm). When $\cos^2\theta_k \approx 1$, the group gradient is nearly aligned with the top eigendirection $\mathbf{v}_1$, and when $\cos^2\theta_k \approx 0$, it is nearly orthogonal. This quantity measures how strongly a group's gradient aligns with the direction constrained by EoS dynamics.

The link to learning comes from self-stabilization. At EoS, the oscillation–stabilization cycle produces net parameter movement primarily along $\mathbf{v}_1$ [10]. Consequently, loss decreases predominantly for groups whose gradients are aligned with $\mathbf{v}_1$, while groups with orthogonal gradients make limited progress (Figure 6).

Curvature influence. While $\cos^2\theta_k$ measures direction, it does not capture gradient magnitude. We report the squared projection

$$(\nabla \ell_k \cdot v_1)^2 = \|\nabla \ell_k\|^2 \cdot \cos^2\theta_k, \tag{4}$$

which quantifies a group's effective curvature influence along $\mathbf{v}_1$. This follows from the quadratic form

$$\nabla \ell_k^\top H \nabla \ell_k = \sum_i \lambda_i (\nabla \ell_k \cdot v_i)^2,$$

where the contribution of the top eigendirection is $\lambda_1 (\nabla \ell_k \cdot v_1)^2$. Since λ_1 is shared across groups at a given step, $(\nabla \ell_k \cdot v_1)^2$ ranks groups by their curvature influence in the dominant direction. The metric can decrease either through rotation away from $\mathbf{v}_1$ or shrinking gradient magnitude. We isolate each effect in Section 4.

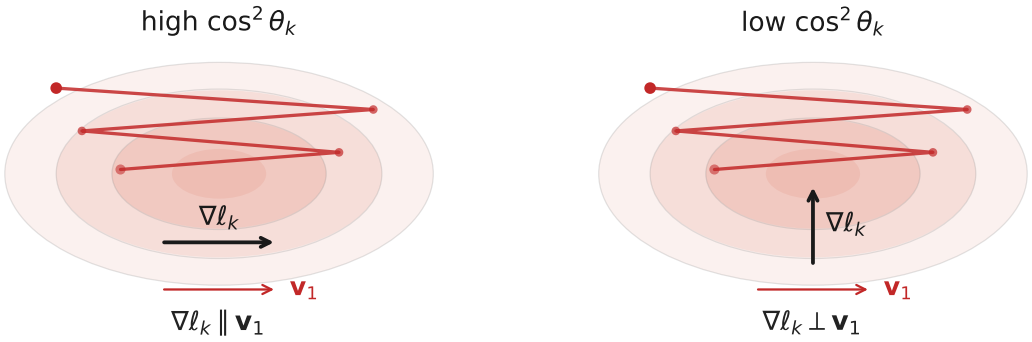

Figure 6: **Directional coupling at EoS.** The optimizer oscillates along $\mathbf{v}_1$ (red zigzag). When a group's gradient $\nabla\ell_k$ aligns with $\mathbf{v}_1$ (left), self-stabilization reduces loss for that group. When $\nabla\ell_k$ is orthogonal to $\mathbf{v}_1$ (right), the group is decoupled from the oscillation and its loss does not benefit.

Appendix B. Experimental setup

B.1. Architecture

We use a fully connected MLP that flattens each input image to a vector, then applies two hidden linear layers of width 512 with ReLU activations, followed by a final linear classifier to 2 output classes.

B.2. Training procedure and hyperparameters

Our primary experiments use full-batch GD, which is deterministic given a fixed initialization. We varied the initialization seed to obtain different optimization trajectories. Five pre-determined identical seeds were used in all plots shown in main text. An initial scaling of 0.2 was applied to all runs.

B.3. Prototype Taxonomy

Definition. We partition the training distribution into four groups defined by the joint geometry of $P_{X,Y}$, independent of any trained model (Figure 1). *Inliers* are high-density points near the class centroid μ_c with correct labels. *Boundary points* are in-distribution examples near the inter-class boundary, identified by high label ambiguity in their local neighborhood. *Input-outliers* are geometrically atypical inputs, far from μ_c in input space, that retain correct labels. *Output-outliers* are high-density inputs assigned an incorrect label. Representative examples from each group are shown in Figure 2. Inliers and boundary points are identified from the existing training data by ranking centroid distance and k-NN label ambiguity respectively, while input-outliers and output-outliers are synthetically constructed to isolate the effects of input-space atypicality and label inconsistency.

Construction. We instantiate the taxonomy on a binary CIFAR-10 task (automobile vs. truck, $n = 10{,}000$) [25]. Inlier candidates are the $M = 3m$ points per class with smallest centroid distance $\|x_i - \mu_c\|$; boundary candidates are the m points per class whose k-NN label composition ($k = 50$)

is closest to uniform. From the inlier candidate pool we sample three disjoint subsets of size $m = 25$ per class: the first retains original inputs and labels (inliers); the second is assigned flipped labels $1-c$ (output-outliers); the third is extrapolated by pushing each example away from the opposite class's centroid as $x_i \pm \alpha\, v_{\text{diff}}$ (sign $+$ if $y_i = 1$, $-$ if $y_i = 0$), where $v_{\text{diff}} = \mu_1 - \mu_0$ is the unnormalized centroid difference (input-outliers). We set $\alpha = 3$, chosen so input-outliers are by construction the most distant group from their class centroid; resulting pixel values may lie outside the valid input range, but we do not clip in order to preserve the displacement magnitude. Boundary points are drawn directly from the ambiguity pool. The final training set contains $n = 10{,}000$ examples, including 200 prototype-labeled points (50 per group, 25 per class) tracked throughout training.

B.4. Setup and Branching Intervention Design

We train a two-hidden-layer MLP (width 512, ReLU) with full-batch gradient descent on a binary CIFAR-10 task (automobile vs. truck, $n = 10{,}000$) for 10,000 steps. Prototype groups are constructed as described in Section B.3. The default learning rate is $\eta = 0.01$ under mean square error (MSE). All quantities are plotted against effective time $t_{\text{eff}} = \eta\, t$, which normalizes for step size.

To test whether the stability constraint causally affects subset-level learning, we branch each run from a shared trajectory at time t^*, defined as the onset of EoS (detected as the first time λ_1 reaches $2/\eta$). The *baseline* branch continues at the original learning rate, remaining at EoS. The *exit* branch reduces the learning rate by half ($\eta \to \eta/2$), increasing the stability threshold to $4/\eta$ and allowing for training to promptly exit the EoS regime. We denote t^{**} as the time at which the exit branch reaches its new stability threshold $4/\eta$. The interval $[t^*, t^{**}]$ is the window over which the two branches occupy different stability regimes. Since architecture, data, and initialization are shared up to t^*, the branching intervention identifies the causal effect of continuing at the original learning rate versus dropping it at EoS onset. The intervention separates the branches into EoS and non-EoS regimes, so post-branch divergence in prototype-level loss provides evidence about the consequences of remaining at EoS.

All figures show medians across 5 seeds. Shading indicates interquartile range where shown.

B.5. Branching Intervention Implementation

We isolate the effect of a learning-rate drop at the edge of stability by running each configuration as a matched pair:

- a *baseline* trained at η for the full step budget;
- a *fork* that follows the baseline trajectory up to a fork step t^* and then applies a single scheduled drop to $\eta' = c\eta$ (we use $c = 0.5$).

The fork step t^* is defined as the first logged step at which $\lambda_{\max}(\nabla^2 \mathcal{L})$ crosses $2/\eta$, the EoS threshold for full-batch GD. We utilize a learning rate of 0.01 for all plots.

Because GD is deterministic given a fixed initialization and has no optimizer state beyond the weights, the fork is implemented by resuming from a checkpoint of the baseline taken just before t^* and continuing at η up to t^*, at which point the drop is applied. All other settings are identical to the baseline—dataset and initialization seeds, architecture, loss, batch size, step budget, and the fixed prototype subset.

Both branches log all diagnostics on a uniform every-32-steps grid spanning the full training window, ensuring the baseline and fork are sampled at identical step indices so their per-prototype

curves can be compared index-for-index without interpolation. The only quantity that differs between a baseline and its fork is the scheduled $\eta \to c\eta$ at step t^*. This makes the fork a minimal counterfactual for the learning-rate drop and supports the causal language in Section 3. Both the fork and the baseline are compared via distance traveled $\eta *$ step to provide a valid comparison of speed across different learning rates.

B.6. EoS detection and timing

The effect size depends on when t^* falls relative to the onset of the second EoS regime. When t^* lies well after the sharpness plateau is established, the post-fork branches converge to similar outcomes and the trade-off is attenuated. This is consistent with the mechanism: the intervention acts by redirecting a constraint-shaped trajectory, so once that shaping has occurred it has less leverage.

The logged t^* has two sources of small offset relative to the true EoS onset. First, the every-32-steps logging grid limits detection resolution: t^* can only be flagged at a logged step, so the actual crossing may have occurred up to 32 steps earlier.

Second, the cubic term of the Taylor expansion around a point adds a jitter around $2/\eta$ [7]. Specifically, the period-2 orbit of GD exists for $\eta \in \big(2/f''(\bar{x}), 2/(f''(\bar{x}) - \epsilon \cdot f^{(3)}(\bar{x}))\big)$, so the bifurcation is governed by the curvature at the orbit, $f''(\bar{x}) - \epsilon \cdot f^{(3)}(\bar{x})$, rather than the curvature at the minimum, $f''(\bar{x})$, which our $2/\eta$ crossing criterion targets. Together, these two effects can cause the logged t^* to lead or lag the visible onset of oscillation by a small number of steps. This is consistent with the small offsets visible in some figures and does not affect the branching design: both branches share the trajectory up to t^*, so the post-fork comparison is unaffected by a few-step mistiming in the detection of EoS onset.

B.7. Compute and Codebase

Compute. All experiments ran on a single NVIDIA L40S GPU (48 GB) on an internal academic SLURM cluster, with 8 CPU cores and 64 GB of system RAM per job. A 10,000-step run takes approximately 10 minutes for the MLP and 30 minutes for CNN/ResNet models. The number of GPU-hours is on the order of 15 for all reported experiments.

Codebase. Our implementation builds on the open-source codebase (https://github.com/arseniqum/edge-of-stochastic-stability) [3] (Apache 2.0; 4), which provides the CIFAR-10 training pipeline and curvature/sharpness logging used in prior work on edge of stochastic stability. We extend this framework with (i) a prototype taxonomy and synthetic outlier construction, (ii) an EoS branching intervention, (iii) per-group alignment diagnostics (gradient norms and cosine alignment), and (iv) per-subset loss and sharpness metrics.

B.8. Limitations and scope

Our experiments are restricted to relatively small models and datasets, primarily because tracking EoS dynamics requires repeated estimates of the top Hessian eigenvalue via Hessian–vector products, whose cost scales with both model and dataset size. This constraint is common in the EoS literature, including the settings of [3, 8], from which we adapt our experimental setup. Consequently, whether the observed selectivity persists for more classes, larger datasets, or higher-resolution inputs remains open. For the same reason, we focus on full-batch training: a major open direction for the field is to understand in what way the self-stabilization argument [10] could be generalized to mini-batch

optimizers. Extending the corresponding subset-level analysis to mini-batch optimizers requires these additional theoretical foundations. Our prototype taxonomy is also defined in pixel space, which maps most transparently to gradient geometry in MLPs. In convolutional architectures, our experiments suggest that the same predictive quantity, $(\nabla \ell_k \cdot v_1)^2$, remains informative, but the identity of the dominant group can shift as learned features reshape the input-to-gradient mapping. Extending prototype construction to representation space is therefore a natural next step. Finally, our robustness and generalization results are preliminary; systematic evaluation across architectures, seeds, and broader distribution shifts remains future work.

Appendix C. Related Work

Our work connects three threads: training dynamics at EoS, sample-centric analyses of which examples drive learning, and curvature-based implicit regularization. We argue these are linked: the directional structure at EoS governs how optimization effort is distributed across the data, connecting where the optimizer moves in parameter space to what it learns.

C.1. Training Dynamics at the Edge of Stability

Onset of EoS. EoS is characterized by training in which the top Hessian eigenvalue grows along the optimization trajectory before reaching an instability boundary [8, 18, 20]. This phenomenon was denoted as progressive sharpening by [8]. In full-batch gradient descent, this gives rise to the edge of stability, where sharpness saturates near $2/\eta$ while the loss continues to decrease [8]. EoS describes a regime where sharpness saturates near $2/\eta$ while loss continues to decrease [8]. Recent work has shown that EoS-like phenomena extend beyond full-batch GD to adaptive optimizers, stochastic training, and momentum dynamics [3, 5, 9]. Our goal is complementary: we focus on the cleanest setting for EoS, full-batch gradient descent, and use this controlled regime to show that the global stability constraint is also a selective mechanism over the data distribution.

Self-stabilization at EoS. A mechanistic account is provided by Damian, where oscillations along the top Hessian eigenvector $\mathbf{v}_1$ bound curvature via self-stabilization [10]. Specifically, these oscillations can feed back through higher-order derivatives to reduce sharpness, yielding an implicit projected dynamics near the stability boundary [10]. Prior work characterizes EoS through global dynamics—fast oscillation along $\mathbf{v}_1$ and slow sharpness-reducing drift along flat directions [6, 27, 40]. We instead ask which examples contribute to these dynamics, revealing a selective effect over the data distribution that scalar sharpness obscures.

C.2. Sample-Centric Learning

A parallel literature studies which individual examples drive learning [12, 13, 24, 30, 31, 33–36]. Rare or atypical examples can disproportionately influence generalization, i.e. memorization of rare subpopulation members is necessary for near-optimal generalization on long-tailed distributions [12, 13]. Toneva [36] also tracks forgetting events and finds rare examples are repeatedly forgotten and relearned [36]. Other works score rare example on difficulty throughout training and reveal ambiguous examples (those which are flip throughout training) are key for out-of-distribution generalization [35]. Rare examples can also dominate gradient signal and sharpening dynamics [30, 31, 33]. In these approaches, example difficulty is defined relative to the model state. We instead define example groups directly from the data distribution and ask how optimization treats them, shifting the question from which examples are hard to which are structurally favored. Other studies [30] also introduce GraNd (per-example gradient norm) and EL2N (per-example error-vector norm) as scores that rank training examples by influence on optimization. Our curvature-influence score $(\nabla \ell_i \cdot \mathbf{v}_1)^2 = \|\nabla \ell_i\|^2 \cos^2 \theta_i$ adds the $\cos^2 \theta_i$ factor on top of GraNd's $\|\nabla \ell_i\|^2$. The $\cos^2$ factor is what makes the score predictive of the EoS beneficiary's identity, and it is invisible to GraNd and EL2N.

C.3. Implicit regularization through curvature.

Curvature has been widely linked to generalization: sharp minima correlate with worse performance [22], while optimization methods and hyperparameters implicitly bias toward flatter solutions [14, 15, 19, 21, 22, 26, 29, 37]. Other methods make the flatness objective explicit or directly bias the final iterate toward flatter regions, including Sharpness-Aware Minimization and stochastic weight averaging [14, 17]. Much of the flatness literature treats curvature as a property of the final solution or of the algorithm's implicit bias. Work on EoS studies curvature dynamically along the training trajectory, including its convergence and implicit-regularization effects [1, 2, 6, 27, 40]. We add a data-level perspective: at EoS, curvature acts selectively across examples, with functional consequences for which subsets generalize and which are robust.

C.4. Our positioning

This work draws on three threads: the dynamics of training at EoS, sample-centric analyses of which examples drive learning, and implicit regularization through curvature. The threads are usually treated separately, and this paper's contribution sits where they meet: the directional structure of EoS governs how optimization effort is distributed across the data distribution, connecting where the optimizer goes in parameter space to what it learns from the data. The closest prior work [33], observes that small groups of outliers with large-magnitude features have an outsized effect on sharpening and EoS dynamics; we extend this with a model-independent prototype taxonomy and analysis on initial data distribution.

Appendix D. Generalizing the Alignment Principle

We suggest preliminary results here on the implications of directional alignment and gradient persistence on generalization. Sections 3 and 4 imply a clear prediction: if alignment×persistence drives the effect, then changing only the data geometry, while holding the model, optimizer, and loss fixed, should change the dominant group and, in turn, the functional benefit conferred by EoS.

D.1. Performance on Harder Class Boundaries.

We verify this prediction on a more challenging class pair (cat vs. dog, $n = 10{,}000$), where the relative geometry of the prototype groups shifts. With $\alpha = 3$, boundary points are the most distant group from the centroid, not the input-outliers. Correspondingly, they dominate $\mathbf{v}_1$ and become the primary beneficiary of EoS. Increasing α to 10 restores input-outliers as the most distant group and transfers the advantage back to them (Appendix E.3).

D.2. Does Edge-of-Stability Improve Generalization or Robustness?

Sections 3–4 established when a subset benefits from EoS. We now ask whether this training-time selectivity transfers to test-time behavior and share preliminary results.

Adversarial robustness. Figure 7 reports PGD adversarial accuracy ($\varepsilon = 0.03$, 10 steps, step size $\varepsilon/4$) on 50 test-set boundary points selected by k-NN ambiguity [28]. With $\alpha = 3$, boundary points dominate $\mathbf{v}_1$, and the EoS branch maintains higher adversarial accuracy than the exit branch after t^{**}. The stability constraint implicitly sharpens the decision boundary in the region most relevant to robust classification. With $\alpha = 10$, where input-outliers instead dominate, the pattern reverses: the exit branch now achieves higher adversarial accuracy on boundary points, because the EoS branch in this configuration has been concentrating its optimization budget elsewhere.

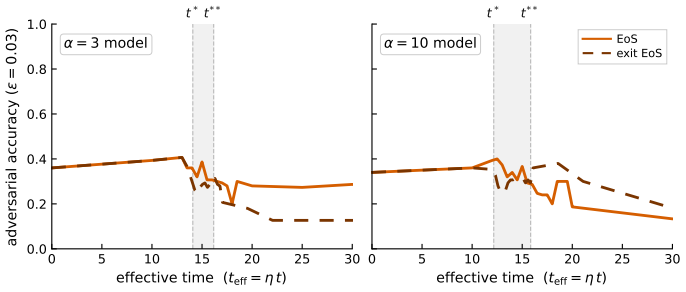

Figure 7: **EoS improves adversarial robustness only when boundary points dominate $\mathbf{v}_1$. Left** ($\alpha = 3$, boundary dominates $\mathbf{v}_1$): the EoS branch (solid) outperforms the exit branch (dashed) after t^{**}. **Right** ($\alpha = 10$, input-outlier dominates $\mathbf{v}_1$): the pattern reverses, and the exit branch performs better. Robustness gains appear only when EoS concentrates on the evaluated subset. Single seed.

Out-of-distribution generalization. Figure 8 reports MSE loss on input-outliers constructed at varying α_{test}, evaluated at the checkpoint immediately after t^{**} in training. With $\alpha = 3$, the exit branch achieves lower loss at large α_{test}, indicating no OOD advantage for input-outliers. With

$\alpha = 10$, where input-outliers dominate $\mathbf{v}_1$, the EoS branch achieves lower loss at large α_{test}: optimization on input-outliers at EoS during training appears to transfer to OOD generalization along v_{diff}.

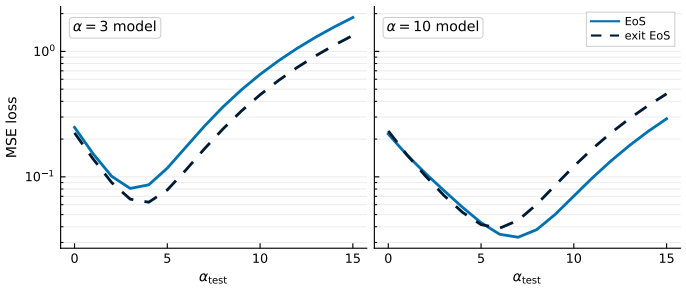

Figure 8: **EoS concentrates generalization on the dominant group.** Test MSE on input-outliers across α_{test} (after t^{**}). **Left** ($\alpha = 3$, boundary dominates $\mathbf{v}_1$): no OOD advantage. **Right** ($\alpha = 10$, input-outlier dominates $\mathbf{v}_1$): EoS improves OOD performance at large α_{test}. Single seed.

Appendix E. Architecture, Optimizer, Class Pair Robustness

Our primary results focus on an MLP trained with full-batch gradient descent under both mean-squared error and cross-entropy losses. In this section, we assess robustness across alternative architectures, optimizers, and class pair, across 3 seeds and with $\eta = 0.01$. We examine (i) the divergence in sharpness between baseline and exit-from-EoS runs, (ii) the curvature profile of the baseline, and (iii) how this curvature predicts the intervention effect, defined as $\Delta\ell_k = \ell_k^{\text{exit}} - \ell_k^{\text{EoS}}$.

E.1. Architecture robustness

The network's architecture determines how input-space geometry maps to gradient-space geometry. In an MLP, which processes raw pixel vectors, centroid distance translates directly into gradient atypicality: pixel-space outliers produce distinctly oriented gradients. Convolutional architectures transform this mapping through learned spatial features, pooling, and normalization, which can compress pixel-space differences that the MLP preserves. As a result, the group with the highest centroid distance need not be the group with the largest curvature influence.

The predictive quantity $(\nabla\ell_k \cdot v_1)^2$ remains consistent across architectures—what changes is which group achieves the highest value.

CNN.

The CNN consists of three convolutional layers with channel widths $64, 64, 128$, all using 3×3 kernels, stride 1, no padding, and ReLU activations, with 2×2 max-pooling after the second and third convolutional layers. The resulting feature map is flattened and passed through a fully connected hidden layer of width 512 with ReLU activation, followed by a linear classifier to C output classes.

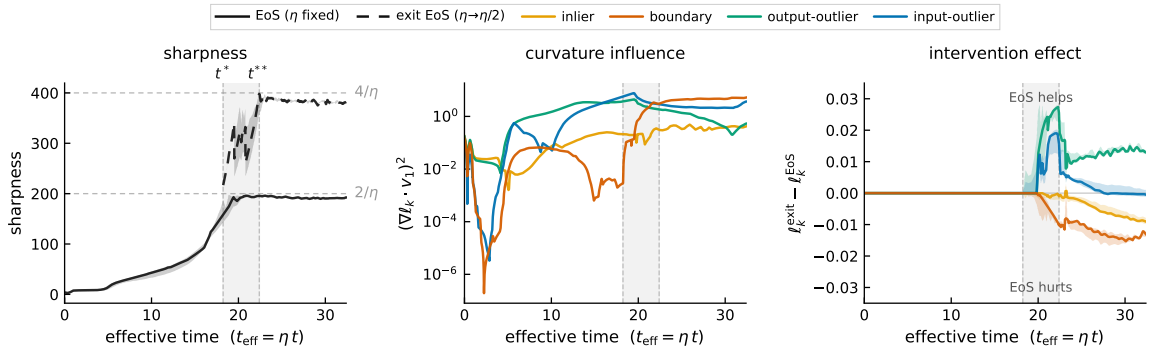

Figure 9: GD CNN MSE. Curvature influence is comparable for output-outliers and input-outliers, both benefit.

ResNet.

We use a batch-normalization-free ResNet-14 with an initial 3×3 convolutional stem of width 16, followed by three residual stages with channel widths $16, 32, 64$ and block counts $[2, 2, 2]$. Each residual block contains two 3×3 convolutions with ReLU activations and identity BatchNorm replacements; downsampling occurs in the first block of stages 2 and 3, followed by global average pooling and a final linear classifier to C output classes.

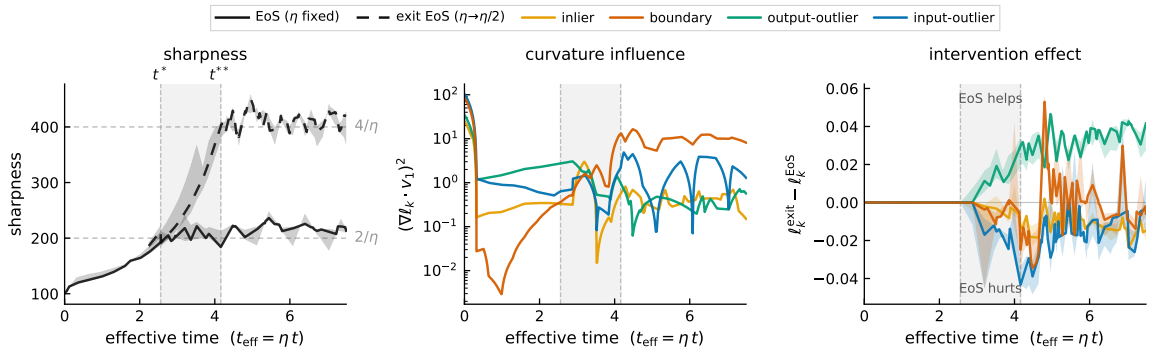

Figure 10: GD ResNet MSE. Curvature influence is highest for output-outliers, and it is the primary beneficiary of EoS.

E.2. Optimizer robustness

We primarily focused on optimizers which maintained a fixed curvature landscape. We did not extend this analysis to adaptive optimizers such as Adam [23] as it reshapes the curvature landscape at each step. Instead, stochastic gradient descent (SGD) and full-batch gradient descent with momentum add randomness and acceleration, respectively, to each step the optimizer takes down the loss landscape. For a large batch for Figure 11, we see that the highest curvature influence group (input-outliers) correspondingly is the primary beneficiary of EoS. Similarly for Figure 12, input-outliers have the highest curvature influence, and the intervention benefits them.

SGD (batch size=128).

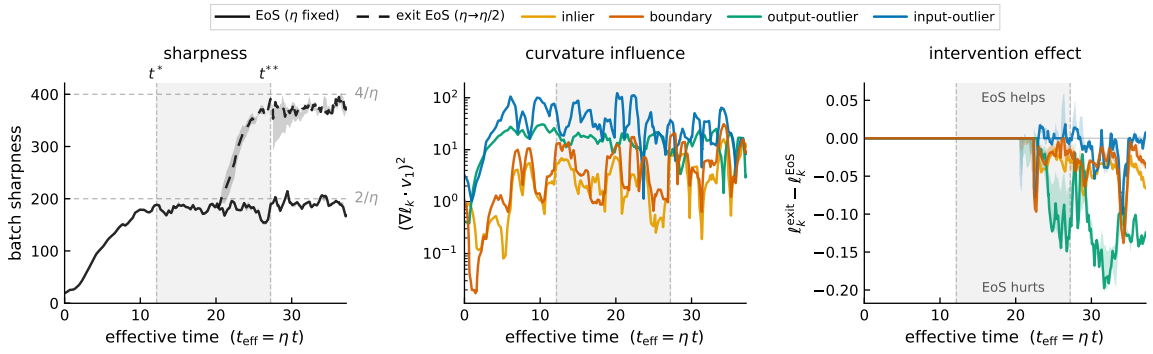

Figure 11: SGD MLP MSE. Curvature influence is highest for input-outliers, and it is the primary beneficiary of EoS.

GD with Momentum ($\beta = 0.9$).

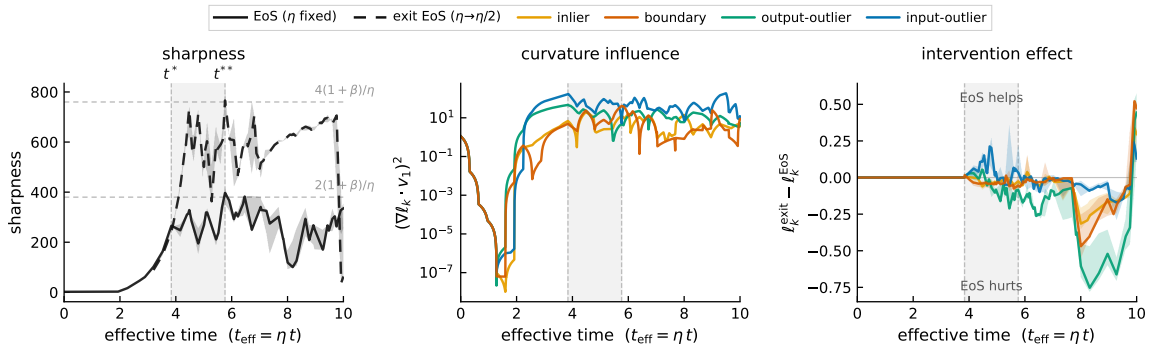

Figure 12: GD Momentum MLP MSE. Curvature influence is highest for input-outliers, and it is the primary beneficiary of EoS. Momentum implemented as in [32] and EoS threshold implemented described in large-batch momentum in [5]

E.3. Class pair robustness

A harder classification task. On the closer pair (3,5), $\alpha = 3$ is no longer sufficient to make input-outliers the most atypical subset; boundary points dominate atypicality instead (Figure 13). The alignment principle predicts that boundary points should therefore capture the EoS advantage on this pair, and they do (Figure 14).

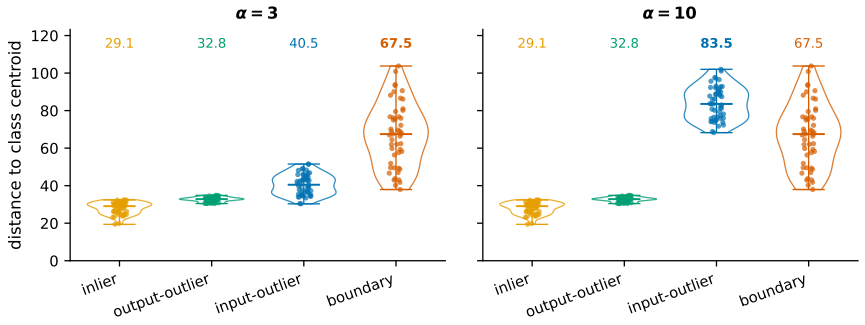

Figure 13: Distribution of centroid distance by prototype subgroup on the (3,5) class pair. Boundary points have the largest median distance under $\alpha = 3$ but input-outliers have the largest median distance under $\alpha = 10$.

Ablation on α. At $\alpha = 3$, boundary points are the primary beneficiaries at EoS. At $\alpha = 10$, input-outliers become the most distant from their class centroids and the EoS advantage transfers to them.

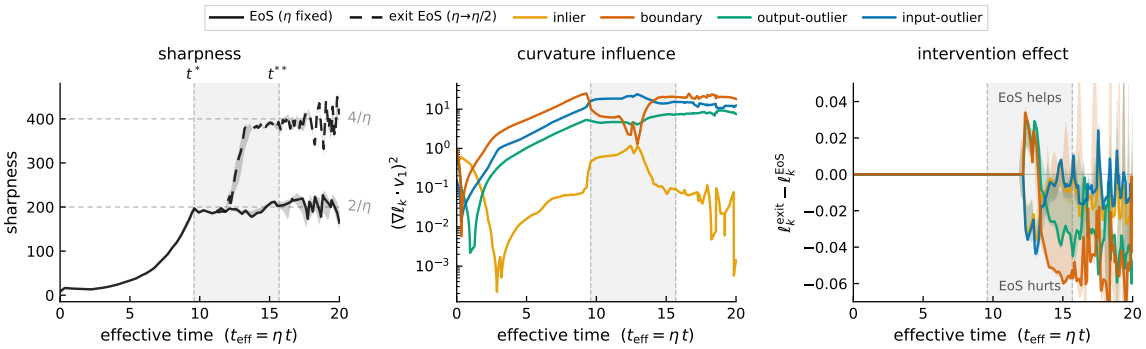

Figure 14: Curvature influence is highest for boundary, and it is the primary beneficiary of EoS.

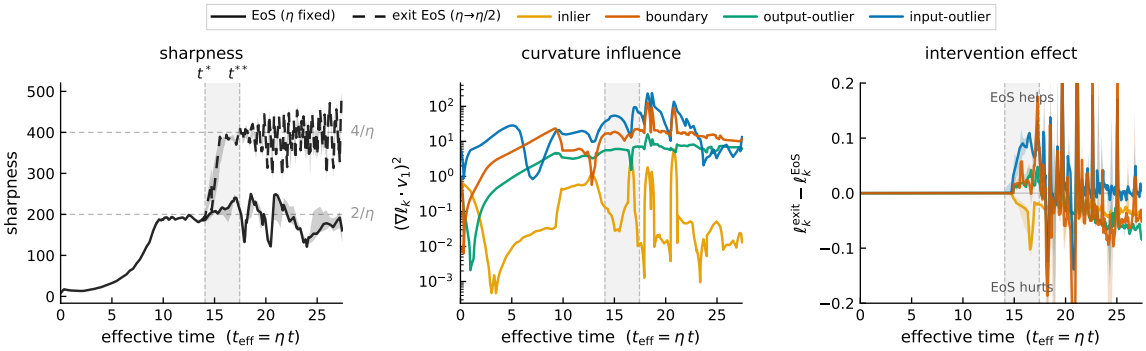

Figure 15: Curvature influence is highest for input-outliers, and it is the primary beneficiary of EoS.

Appendix F. Input-Outlier Construction Ablation

The conceptual schematic for the directional perturbation is shown in Figure 16. Figure 17 confirms that geometric atypicality of input-outlier is preserved under the random-direction control.

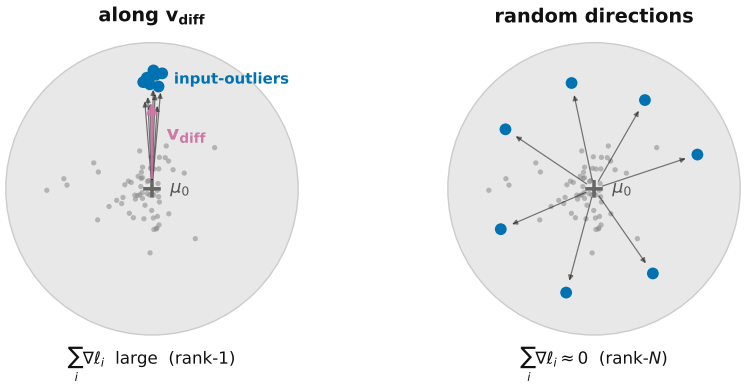

Figure 16: Schematic for coherent vs. incoherent input-outlier construction. **Left:** input-outliers displaced along a shared direction v_{diff}; per-example gradients reinforce, producing a concentrated curvature contribution along one direction. **Right:** same displacement distance but in random orthogonal directions; per-example gradients partially cancel, producing a diffuse contribution across many eigenvectors. Seed points, labels, and centroid distances are identical across conditions.

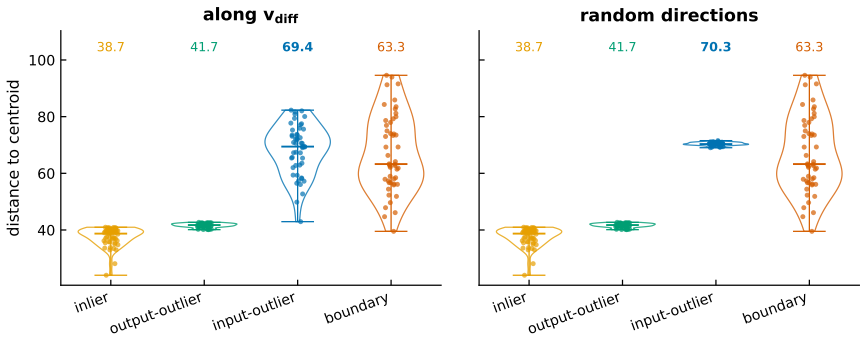

Figure 17: Centroid distance is preserved under the random-direction control (median 69.4 vs. 70.3 for along-v_{diff} and random-direction outliers, respectively). Geometric atypicality is conserved.

Appendix G. Natural points

We verify that the condition of gradient alignment holds on natural geometric outliers, with no synthetic displacement applied. Figure 18 plots the Spearman rank correlation coefficient between centroid distance and per-example alignment $\cos^2(\nabla \ell_i, v_1)$ across training: the correlation is near zero during progressive sharpening and rises at EoS onset.

Figure 19 shows the underlying per-example trajectories. Atypical points (red) exhibit high alignment that peaks just before EoS onset and then declines, while typical points (blue) remain weakly aligned over the same interval. This relationship is monotonic with respect to distance from the centroid near EoS onset. Later in training, the ordering reverses, with typical points eventually exceeding atypical ones in alignment. Centroid distance, a purely distributional quantity computed before training, correlates with which examples will dominate the unstable direction at EoS.

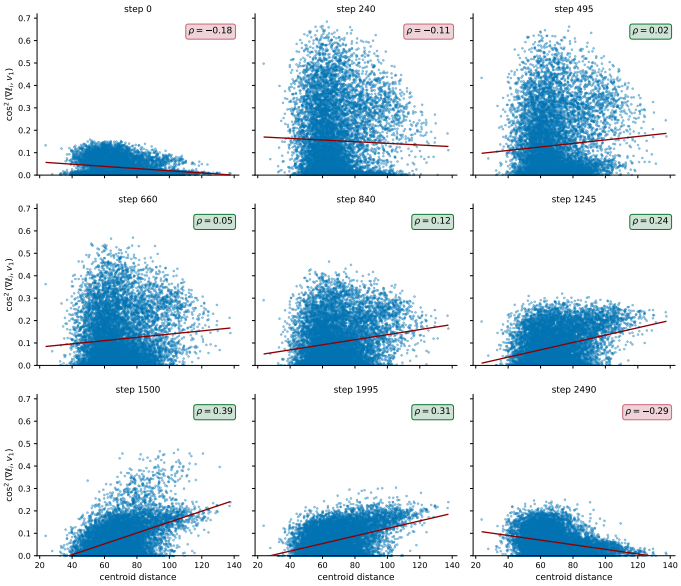

Figure 18: **Top:** beginning of training, initially negative correlation that increases; **Middle:** EoS onset, monotonically increasing correlation; **Bottom:** correlation peak when large-amplitude oscillations along $\mathbf{v}_1$ develop, then monotonically decrease to negative after peaking

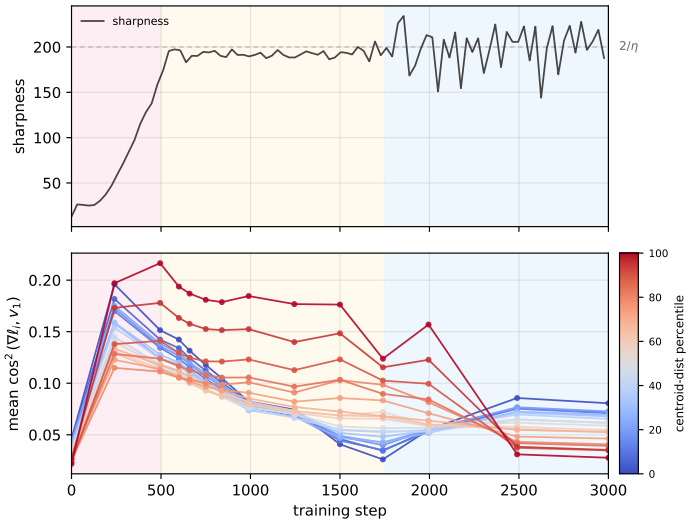

Figure 19: **Top:** correlation rises at EoS onset and peaks with sharpness oscillations. **Bottom:** alignment vs. centroid-distance percentile over training. Atypical points peak near EoS (step 660), then decline, and dominance shifts to typical points later in training.

Appendix H. Future Work

Our results suggest that EoS may have functional consequences beyond the subset-level training dynamics studied in the main text. In particular, if the subset that benefits from EoS is determined by curvature influence, then changing which subset dominates $\mathbf{v}_1$ should also change which downstream behavior is improved. This gives a concrete prediction: when boundary points dominate $\mathbf{v}_1$, EoS should preferentially concentrate optimization near the decision boundary and may improve adversarial robustness; when input-outliers dominate, EoS should instead concentrate optimization on distributional tails and may improve extrapolation along the outlier direction.

Preliminary evidence from our α-ablation is consistent with this prediction. Varying α changes the relative geometry of the prototype groups and shifts which group has the largest curvature influence. When boundary points dominate $\mathbf{v}_1$, the EoS branch shows improved robustness on boundary-like examples; when input-outliers dominate, the EoS branch instead shows improved performance on outlier-like test points. These results suggest that hyperparameters typically viewed as controlling convergence, such as learning rate and loss, may also influence which subset of the data distribution is emphasized during training.

A systematic study of these functional consequences remains future work. In particular, it would be valuable to test whether the dominant-curvature subgroup predicts robustness, out-of-distribution generalization, or memorization behavior across larger architectures, stochastic optimizers, and more realistic distribution shifts. Such experiments could clarify when controlled sharpness acts as a useful inductive bias, and when it merely reallocates optimization effort toward subsets that are not aligned with the desired test-time behavior.

Appendix I. Theory: EoS Self-stabilization as a Subset Selector

This appendix gives the local mathematical mechanism behind the subset-level effects measured in the main text. The starting point is the cubic self-stabilization model of [10]: at the edge of stability, gradient descent admits a slow, cycle-averaged description as projected gradient descent on the active sharpness constraint $S(\theta) = 2/\eta$. Under this assumption, we derive a first-order decomposition of the branch differential into two distinct contributions and isolate the two mechanisms tested experimentally—directional alignment and gradient persistence.

I.1. Notation and assumptions

Let $L : \mathbb{R}^d \to \mathbb{R}$ denote the full empirical training loss, and let

$$\ell_k(\theta) := \frac{1}{|P_k|} \sum_{i \in P_k} \ell(f_\theta(x_i), y_i)$$

be the loss restricted to prototype group k. Let $H(\theta) := \nabla^2 L(\theta)$, $S(\theta) := \lambda_{\max}(H(\theta))$, and $\mathbf{v}_1(\theta)$ denote the Hessian, sharpness, and unit top Hessian eigenvector respectively (assumed simple throughout). We compare two branches forked from $\theta^* = \theta_{t^*}$ at EoS onset, where $\eta S(\theta^*) \approx 2$. The baseline branch continues at learning rate η. The exit branch uses learning rate $c\eta$ with $c \in (0, 1)$, locally stable at the fork. We use the baseline learning-rate-scaled time

$$\tau := \eta(t - t^*),$$

where t denotes the gradient descent iteration index and t^* denotes the EoS onset step at which the branches fork. This scaling gives a continuous-time comparison variable aligned to the baseline branch. Define the cycle-averaged (two-step average) branch differential

$$\bar{\Delta}\ell_k(\tau) := \bar{\ell}_k(\theta_{\text{exit}}(\tau)) - \bar{\ell}_k(\theta_{\text{EoS}}(\tau)),$$

where the bar denotes the two-step average that suppresses the $O(\delta)$ phase term along $\mathbf{v}_1$ generated by the period-two EoS oscillation. Define the sharpness-gradient quantities

$$\alpha := -\langle \nabla L, \nabla S \rangle, \qquad \beta := \|\nabla S\|^2, \qquad \delta := \sqrt{2\alpha/\beta},$$

where δ measures the oscillation amplitude along $\mathbf{v}_1$ in the EoS regime [10], and progressive sharpening corresponds to $\alpha > 0$. We use the following standing assumptions.

(A1) Smoothness and simple top eigenvalue. $L \in C^3$ (L is thrice-differentiable) and each $\ell_k \in C^2$ (ℓ_k is twice-differentiable) in a neighborhood of θ^*, with simple largest Hessian eigenvalue.

(A2) Local EoS approximation. The baseline EoS trajectory admits a slow, cycle-averaged description as projected gradient descent on the active sharpness constraint $S(\theta) = 2/\eta$, in the sense of [10].

(A3) Stable exit. The exit branch is locally stable in the top direction at the fork, $c\eta S(\theta^*) < 2$.

(A4) Short-window Taylor regime. Post-fork comparisons are made in a window short enough that gradients and Hessians admit Taylor expansion around θ^* with $O(\tau^2)$ remainder.

(A5) Cycle averaging. Subset losses are compared via two-step averaging or short-window smoothing, suppressing the $O(\delta)$ instantaneous phase fluctuation along u at EoS.

I.2. Projected drift at the sharpness boundary

Lemma 1 (EoS slow drift)
Under (A1)–(A2), the cycle-averaged EoS drift at θ^* is

$$\dot{\theta}_{EoS} = -\nabla L - \frac{\alpha}{\beta}\nabla S.$$

Under (A3), the exit branch in baseline time τ has leading drift $\dot{\theta}_{exit} = -c\nabla L$.

Proof The tangent space of the active boundary $\mathcal{M} := \{\theta : S(\theta) = 2/\eta\}$ at θ^* is $\{z : \langle \nabla S, z\rangle = 0\}$. Removing the normal component of ∇L along ∇S gives

$$\nabla L|_{\mathcal{M}} = \nabla L - \frac{\langle \nabla L, \nabla S\rangle}{\|\nabla S\|^2}\nabla S = \nabla L + \frac{\alpha}{\beta}\nabla S,$$

so projected-gradient descent yields drift $-\nabla L - (\alpha/\beta)\nabla S$. The exit-branch drift in baseline time follows from (A3): a step of size $c\eta$ in baseline time $\tau = \eta(n - n^*)$ has slope $-c\nabla L$ to leading order. ■

The key observation is that the EoS branch is not ordinary gradient descent with oscillations: its slow drift carries an additional component along $-\nabla S$. Each subset feels this additional drift through the inner product $\langle \nabla \ell_k, \nabla S\rangle$.

I.3. Branch decomposition

Define the two local subset scores

$$R_k := \langle \nabla \ell_k, \nabla L\rangle, \qquad Q_k := \langle \nabla \ell_k, \nabla S\rangle.$$

R_k is the ordinary loss-gradient alignment of subset k. Q_k is the projection of the subset gradient onto the sharpness-control direction.

Proposition 2 (Per-subset branch decomposition) *Under (A1)–(A5), for short post-branch times,*

$$\bar{\Delta}\ell_k(\tau) \;=\; \underbrace{(1-c)\,R_k\,\tau}_{\textit{learning-rate confounder}} \;+\; \underbrace{\frac{\alpha}{\beta}\,Q_k\,\tau}_{\textit{EoS selector}} \;+ O_k(\tau^2) + O_k(\delta^2).$$

Proof All quantities below are evaluated locally near the branch point θ^* unless otherwise stated. Recall that

$$R_k := \langle \nabla \ell_k(\theta^*), \nabla L(\theta^*)\rangle, \qquad Q_k := \langle \nabla \ell_k(\theta^*), \nabla S(\theta^*)\rangle.$$

By Lemma 1, the cycle-averaged EoS branch has slow drift

$$\dot{\theta}_{\mathrm{EoS}} = -\nabla L - \frac{\alpha}{\beta}\nabla S,$$

while the exit branch, measured in baseline time τ, has leading drift

$$\dot{\theta}_{\text{exit}} = -c\nabla L.$$

We first compute the rate of change of the subset loss along the EoS branch. By the chain rule,

$$\frac{d}{d\tau}\bar{\ell}_k(\theta_{\text{EoS}}(\tau)) = \left\langle \nabla\ell_k(\theta_{\text{EoS}}(\tau)), \dot{\theta}_{\text{EoS}}(\tau) \right\rangle.$$

Using the local EoS drift from Lemma 1, this becomes

$$\frac{d}{d\tau}\bar{\ell}_k(\theta_{\text{EoS}}(\tau)) = \left\langle \nabla\ell_k(\theta_{\text{EoS}}(\tau)), -\nabla L(\theta_{\text{EoS}}(\tau)) - \frac{\alpha(\tau)}{\beta(\tau)}\nabla S(\theta_{\text{EoS}}(\tau)) \right\rangle.$$

Expanding the inner product gives

$$\frac{d}{d\tau}\bar{\ell}_k(\theta_{\text{EoS}}(\tau)) = -\left\langle \nabla\ell_k(\theta_{\text{EoS}}(\tau)), \nabla L(\theta_{\text{EoS}}(\tau)) \right\rangle - \frac{\alpha(\tau)}{\beta(\tau)}\left\langle \nabla\ell_k(\theta_{\text{EoS}}(\tau)), \nabla S(\theta_{\text{EoS}}(\tau)) \right\rangle.$$

For short post-branch times, Assumption (A4) allows us to replace the quantities along the branch by their values at θ^* up to first-order local errors:

$$\langle \nabla\ell_k(\theta_{\text{EoS}}(\tau)), \nabla L(\theta_{\text{EoS}}(\tau)) \rangle = R_k + O_k(\tau),$$

and

$$\langle \nabla\ell_k(\theta_{\text{EoS}}(\tau)), \nabla S(\theta_{\text{EoS}}(\tau)) \rangle = Q_k + O_k(\tau).$$

Likewise, $\alpha(\tau)/\beta(\tau) = \alpha/\beta + O(\tau)$ locally. Therefore

$$\frac{d}{d\tau}\bar{\ell}_k(\theta_{\text{EoS}}(\tau)) = -R_k - \frac{\alpha}{\beta}Q_k + O_k(\tau) + O_k(\delta^2).$$

The term $O_k(\delta^2)$ comes from replacing the instantaneous oscillatory EoS trajectory by its two-step cycle average. The leading $O(\delta)$ phase term cancels under the two-step average, leaving only second-order oscillation effects.

Now consider the exit branch. Again by the chain rule,

$$\frac{d}{d\tau}\bar{\ell}_k(\theta_{\text{exit}}(\tau)) = \left\langle \nabla\ell_k(\theta_{\text{exit}}(\tau)), \dot{\theta}_{\text{exit}}(\tau) \right\rangle.$$

Using $\dot{\theta}_{\text{exit}} = -c\nabla L$ gives

$$\frac{d}{d\tau}\bar{\ell}_k(\theta_{\text{exit}}(\tau)) = -c\left\langle \nabla\ell_k(\theta_{\text{exit}}(\tau)), \nabla L(\theta_{\text{exit}}(\tau)) \right\rangle.$$

Again Taylor-expanding around θ^* over the short window,

$$\langle \nabla\ell_k(\theta_{\text{exit}}(\tau)), \nabla L(\theta_{\text{exit}}(\tau)) \rangle = R_k + O_k(\tau),$$

so

$$\frac{d}{d\tau}\bar{\ell}_k(\theta_{\text{exit}}(\tau)) = -cR_k + O_k(\tau).$$

We now differentiate the branch differential

$$\bar{\Delta}\ell_k(\tau) := \bar{\ell}_k(\theta_{\text{exit}}(\tau)) - \bar{\ell}_k(\theta_{\text{EoS}}(\tau)).$$

Taking a derivative with respect to τ yields

$$\frac{d}{d\tau}\bar{\Delta}\ell_k(\tau) = \frac{d}{d\tau}\bar{\ell}_k(\theta_{\text{exit}}(\tau)) - \frac{d}{d\tau}\bar{\ell}_k(\theta_{\text{EoS}}(\tau)).$$

Substituting the two expressions above,

$$\frac{d}{d\tau}\bar{\Delta}\ell_k(\tau) = [-cR_k + O_k(\tau)] - \left[-R_k - \frac{\alpha}{\beta}Q_k + O_k(\tau) + O_k(\delta^2)\right].$$

Simplifying,

$$\frac{d}{d\tau}\bar{\Delta}\ell_k(\tau) = (1-c)R_k + \frac{\alpha}{\beta}Q_k + O_k(\tau) + O_k(\delta^2).$$

At the branching time, both branches start from the same parameter value, so

$$\bar{\Delta}\ell_k(0) = 0.$$

Integrating from 0 to τ gives

$$\bar{\Delta}\ell_k(\tau) = \int_0^\tau \frac{d}{ds}\bar{\Delta}\ell_k(s)\, ds.$$

Using the expression for the derivative,

$$\bar{\Delta}\ell_k(\tau) = \int_0^\tau \left[(1-c)R_k + \frac{\alpha}{\beta}Q_k + O_k(s) + O_k(\delta^2)\right] ds.$$

The leading terms are constant in this local expansion, so

$$\int_0^\tau \left[(1-c)R_k + \frac{\alpha}{\beta}Q_k\right] ds = \left[(1-c)R_k + \frac{\alpha}{\beta}Q_k\right]\tau.$$

The local Taylor error integrates as

$$\int_0^\tau O_k(s)\, ds = O_k(\tau^2),$$

and the cycle-averaging residual contributes

$$\int_0^\tau O_k(\delta^2)\, ds = O_k(\delta^2\tau).$$

For short fixed post-branch windows, we absorb this into the stated residual notation as $O_k(\delta^2)$. Therefore

$$\bar{\Delta}\ell_k(\tau) = (1-c)R_k\tau + \frac{\alpha}{\beta}Q_k\tau + O_k(\tau^2) + O_k(\delta^2).$$

This proves the decomposition. ■

Proposition 2 isolates two distinct mechanisms in the branching intervention. The first is an ordinary learning-rate confounder: lowering η by factor c slows progress on every subset by $(1-c)R_k\tau$. The second is the EoS-specific selector: the baseline branch carries an additional $-(\alpha/\beta)\nabla S$ drift, contributing $(\alpha/\beta)Q_k\tau$ to the differential. Whether subset k benefits from EoS depends on the sign of Q_k relative to the rate confounder.

Corollary 3 (Rate slowdown alone cannot produce mixed signs) *If $R_k \geq 0$ for all prototype groups k, then the learning-rate confounder $(1-c)R_k\tau$ is nonnegative for all k. A mixed-sign pattern across groups in $\bar{\Delta}\ell_k$ therefore cannot be explained by ordinary learning-rate slowdown alone.*

Proof $0 < c < 1$ implies $1 - c > 0$. With $\tau \geq 0$, the rate term has the sign of R_k. Nonnegativity of all R_k then forces nonnegative rate contributions for all subsets. ■

This is the theoretical basis for interpreting the selective trade-off in Figure 4. Mixed-sign $\bar{\Delta}\ell_k$ across groups requires either the EoS selector Q_k or another subset-dependent mechanism, not rate slowdown alone. In our empirical comparisons, we instead align branches by learning-rate-normalized time. This removes the leading-order speed difference along $-\nabla L$, so that the remaining first-order branch differential is governed by the EoS-specific selector $(\alpha/\beta)Q_k$. Higher-order differences due to the changed trajectory and sharpness threshold are absorbed into the residual terms.

I.4. Curvature influence as a single-mode proxy

The main text measures

$$C_k := (\nabla\ell_k \cdot v_1)^2.$$

Proposition 2 states the exact local selector as $Q_k = \langle \nabla\ell_k, \nabla S\rangle$. Under standard eigenvalue perturbation, $\nabla S = \nabla^3 L[v_1, v_1]$, so ∇S has a component along $\mathbf{v}_1$ of magnitude $\gamma := \nabla^3 L[v_1, v_1, v_1]$ plus an orthogonal residual. When ∇S is dominated by its top-mode component, $|Q_k|^2 \propto C_k$, and C_k functions as a single-mode proxy for the squared selector magnitude. In what follows, C_k is the measured statistic. The factorization

$$C_k = \|\nabla\ell_k\|^2 \cos^2\theta_k, \qquad \cos^2\theta_k := \frac{(\nabla\ell_k \cdot v_1)^2}{\|\nabla\ell_k\|^2},$$

separates direction (alignment) from magnitude (persistence). The next two subsections establish that both factors are necessary.

I.5. Alignment: coherent vs. random gradients

The random-direction outlier ablation tests whether large geometric distance is sufficient for curvature dominance. The theory predicts that it is not: what matters is whether per-example gradients add coherently in a shared direction.

Let $g_k := \nabla\ell_k = m^{-1}\sum_{i=1}^m g_i$ be the group gradient, and let u be a unit direction (interpreted as the current top Hessian eigenvector).

Lemma 4 (Coherence amplifies curvature influence) *Suppose first that per-example gradients share a coherent component:*

$$g_i = a_i q + \varepsilon_i, \qquad m^{-1}\sum_i \langle \varepsilon_i, u\rangle \approx 0.$$

Then $\langle g_k, u\rangle \approx \bar{a}\,\langle q, u\rangle$ with $\bar{a} := m^{-1}\sum_i a_i$.

Suppose instead that per-example directions are independent, mean-zero, and isotropic in an effective d_{eff}-dimensional subspace (i.e. incoherent directions):

$$g_i = aq_i, \qquad \mathbb{E}[q_i] = 0, \qquad \mathbb{E}\langle q_i, u\rangle^2 = 1/d_{\mathit{eff}}.$$

Then

$$\mathbb{E}\big[\langle g_k, u\rangle^2\big] = \frac{a^2}{m\, d_{\mathit{eff}}}.$$

Proof In the coherent case,

$$\langle g_k, u\rangle = \bar{a}\langle q, u\rangle + m^{-1}\sum_i \langle \varepsilon_i, u\rangle,$$

and the residual average is negligible by assumption. In the random case, the terms $\langle q_i, u\rangle$ are independent with mean zero, so

$$\mathbb{E}\big[\langle g_k, u\rangle^2\big] = \frac{a^2}{m^2}\sum_i \mathbb{E}\langle q_i, u\rangle^2 = \frac{a^2}{m\, d_{\text{eff}}}.$$

■

Lemma 4 justifies the coherent-vs-random ablation in Section **??** as a test of the alignment mechanism. If EoS selectivity is driven by directional alignment rather than distance or gradient magnitude alone, then preserving displacement size while randomizing directions should collapse the group-level projection onto $\mathbf{v}_1$ and eliminate the EoS advantage. In the coherent case, the projection remains order $\bar{a}\langle q, \mathbf{v}_1\rangle$, and hence C_k remains order $\bar{a}^2\langle q, \mathbf{v}_1\rangle^2$. In the random-direction case, the expected squared projection is only $a^2/(md_{\text{eff}})$, so curvature influence averages away with group size and effective dimension. A norm-only account predicts no difference between the two conditions; the observed collapse of the EoS advantage under random-direction displacement is therefore evidence for alignment as the operative mechanism, not magnitude alone.

I.6. Persistence: gradient saturation removes curvature influence

The MSE-vs-CE comparison tests whether alignment is sufficient when gradient magnitude collapses. Write $\ell_i(\theta) = \phi(f_\theta(x_i), y_i)$ and decompose

$$\nabla_\theta \ell_i = J_i^\top r_i, \qquad J_i := \nabla_\theta f_\theta(x_i), \qquad r_i := \nabla_f \phi(f_\theta(x_i), y_i).$$

Lemma 5 (Saturation removes curvature influence) *Suppose $\|r_i\| \to 0$ for all $i \in P_k$ and the Jacobians are uniformly bounded, $\|J_i\|_{\mathit{op}} \le B$. Then $\|\nabla \ell_k\| \to 0$, and consequently*

$$(\nabla \ell_k \cdot u)^2 \to 0$$

for every unit vector u, regardless of the alignment $\cos^2\theta_k$.

Proof Per example, $\|\nabla_\theta \ell_i\| \le \|J_i\|_{\text{op}}\|r_i\| \le B\|r_i\|$. Hence

$$\|\nabla \ell_k\| \le |P_k|^{-1}\sum_{i\in P_k}\|\nabla_\theta \ell_i\| \to 0,$$

and $|\langle \nabla \ell_k, u \rangle| \leq \|\nabla \ell_k\|$ then gives the second claim. ■

For softmax cross-entropy, $r_i = p_i - y_i$, so confidently correctly classified examples ($p_i \to y_i$) drive $r_i \to 0$ and the corresponding subset gradient norm collapses. Output-outliers, whose assigned labels are inconsistent with their input, retain $\|p_i - y_i\|$ bounded away from zero as long as the model continues to predict the input-consistent class, so their gradients persist. Lemma 5 thus predicts that under CE, a confidently classified subset can retain high $\cos^2 \theta_k$ while losing all curvature influence, and that the EoS advantage transfers to whichever subset retains non-vanishing gradients—empirically, output-outliers.

The MSE comparison sharpens the test. MSE residuals also vanish on perfectly fit examples, but in the observed regime the coherent input-outliers retain large residuals throughout training, preserving gradient magnitude. The MSE-vs-CE contrast therefore isolates persistence: a subset that remains aligned with $\mathbf{v}_1$ but loses gradient norm loses EoS influence, while a subset that retains both keeps it.

I.7. Scope of the results

The results above are local and conditional. Under the Damian–Nichani–Lee self-stabilization approximation (A2), Proposition 2 establishes that the cycle-averaged branch differential decomposes additively into a learning-rate confounder controlled by R_k and an EoS-specific selector controlled by Q_k. Lemmas 4 and 5 explain why directional coherence and gradient persistence are each individually necessary for a subset to feel the selector. The results do not claim that the decomposition holds globally, that the single-mode proxy C_k exactly equals $|Q_k|^2$, or that flatness has a universal functional meaning. Empirically, C_k tracks Q_k^2 across (subgroup, checkpoint) pairs in the EoS regime up to a roughly constant proportionality (Figure 20), validating its use as a single-mode proxy. The functional consequence of EoS depends on which subset dominates the selector at training time—a point developed empirically in Section 4.

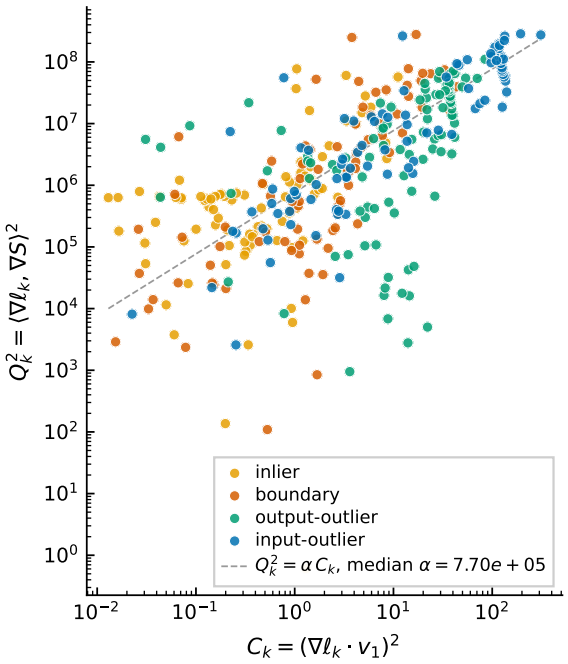

Figure 20: **Empirical validation of the single-mode proxy.** Scatter of $Q_k^2 = \langle \nabla \ell_k, \nabla S \rangle^2$ versus $C_k = (\nabla \ell_k \cdot v_1)^2$ across (subgroup, checkpoint) pairs in training. The dashed line shows the median proportionality $Q_k^2 = \alpha\, C_k$, $\alpha = 7.7 \times 10^5$. The relationship holds across the trajectory; for inliers (nearly orthogonal to v_1), C_k is small and the proxy is loosest, in the regime where the selector predicts no EoS advantage.