Edge of Stability Selectively Shapes Learning Across the Data Distribution

Shauna Kwag[*,†] MIT **Anakha Ganesh**[*,†] MIT **Tomaso Poggio** MIT **Pierfrancesco Beneventano**[†] MIT

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 [1–3], 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 [4–11]. 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 geometry of $P_{X,Y}$ 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

* Core contributors
† Correspondence to {kwags,anakhag,pierb}@mit.edu

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

unevenly, amplifying subsets whose gradients persistently align with the top Hessian eigendirection while suppressing others. This 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 findings connect to a longstanding debate on whether low curvature improves generalization [2, 12–14]. EoS is the regime in which sharpness is actively constrained near the stability threshold, making it a natural setting to test what low curvature actually confers. Our results suggest the answer is not global: the functional benefit depends on which subset dominates the top Hessian eigendirection and shifts with the geometric composition of the training distribution. Flatness, in this view, is a directional property determined by data geometry rather than a scalar property of the solution. More broadly, our results take a step toward connecting two largely separate lines of work: *implicit regularization* in parameter space and *inductive bias* over the data distribution.

Our paper makes the following contributions:

- **EoS is selective, not global** (§3). Using a branching intervention that enters or exits EoS from a shared training trajectory, we find that the stability constraint is not a uniform bottleneck: it selectively benefits some subsets of the training distribution while suppressing others. The trade-off qualitatively replicates across different architectures, optimizers, and datasets (Appendix C, D).
- **Selectivity is governed by alignment $\times$ persistence** (§4). The advantage at EoS is captured by subsets whose aggregate gradient both aligns with the top Hessian eigenvector and remains non-vanishing throughout training. Two controlled counterfactuals isolate each factor: random-direction displacement removes alignment, and cross-entropy saturation removes persistence. In both cases the EoS advantage disappears or is transferred to the subsets that retain the missing factor. Both factors fall out of extending the self-stabilization framework [15] from the global loss to per-subset losses (Section 2.2; full derivation in Appendix H).
- **Geometry shifts the beneficiary** (§5). In our MLP experiments on CIFAR-10 (the standard setting in prior EoS work [7]), the subset that benefits from EoS is the one furthest from the class centroids in the input space. Varying the geometric composition of the training distribution continuously shifts which subset this is. Preliminary evidence suggests the resulting generalization behavior shifts accordingly, with improved adversarial robustness when geometrically distant examples lie near the decision boundary, and improved out-of-distribution generalization when they lie far from the training distribution.

2 Preliminaries

2.1 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 [7]. 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 [7].

2.2 Self-stabilization

The non-monotonic descent at EoS is explained by the *self-stabilization* mechanism of Damian et al. [15]. Once λ_1 exceeds $2/\eta$, gradient descent develops a fast period-2 oscillation along $\mathbf{v}_1$. Higher-order terms convert this oscillation into a slow corrective drift toward lower sharpness, so that the cycle-averaged dynamics remain near the active boundary $\{\lambda_1 \approx 2/\eta\}$.

Under this cycle-averaged description, the EoS drift differs from ordinary gradient descent by an additional sharpness-reducing component,

$$-(\alpha/\beta)\nabla\lambda_1, \qquad \alpha = -\langle \nabla L, \nabla\lambda_1 \rangle, \quad \beta = \|\nabla\lambda_1\|^2.$$

For a subset loss ℓ_k, this extra drift contributes

$$-(\alpha/\beta)\langle \nabla \ell_k, \nabla \lambda_1 \rangle$$

We use this established framework to motivate our experimental design. The selector $\langle \nabla \ell_k, \nabla \lambda_1 \rangle$ depends on both a directional factor (alignment between $\nabla \ell_k$ and $\nabla \lambda_1$) and a magnitude factor (persistence of $\|\nabla \ell_k\|$ over training). These two components motivate complementary controlled interventions in Section 4.

Empirically, we track the directional factor through alignment with the top Hessian eigenvector $\mathbf{v}_1$, measuring how strongly each data subset couples to the unstable mode. Appendix H derives the per-subset loss decomposition and shows that the measured $\cos^2 \theta_k$ serves as an empirical proxy for the theoretical EoS selector $Q_k = \langle \nabla \ell_k, \nabla \lambda_1 \rangle$ under the single-mode approximation. This approximation is supported by the observation that only λ_1 is near-critical at EoS onset (Appendix H.5). These considerations motivate our experiments: branching interventions test the effect of remaining at EoS, random-direction displacement isolates alignment, and cross-entropy saturation isolates gradient persistence.

2.3 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.

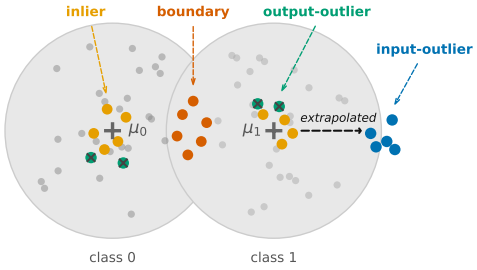

Figure 1: **Conceptual taxonomy of prototypes.** Data samples are categorized based on geometric proximity in input space relative to class-specific cluster centroids (μ_0, μ_1).

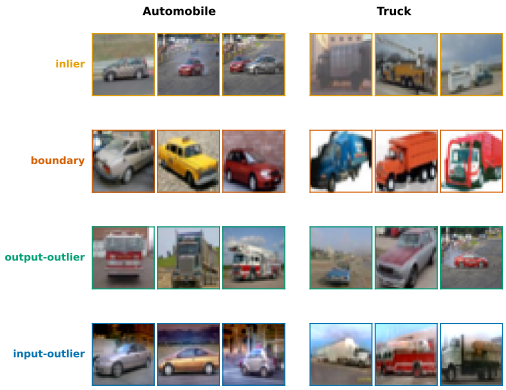

Figure 2: **Input-space visualization of prototype groups for CIFAR-10.** Three representative samples per class (automobile vs. truck) are shown for each group.

Construction. We instantiate the taxonomy on a binary CIFAR-10 task (automobile vs. truck, $n = 10{,}000$) [16]. 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.

2.4 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

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

where $\mathbf{v}_1$ is the top Hessian eigenvector (assumed unit norm). When $\cos^2 \theta_k \approx 1$, the group gradient is nearly aligned with $\mathbf{v}_1$, 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 $\mathbf{v}_1$ [15]. Consequently, loss decreases predominantly for groups whose gradients are aligned with $\mathbf{v}_1$, while groups with orthogonal gradients make limited progress (Figure 3).

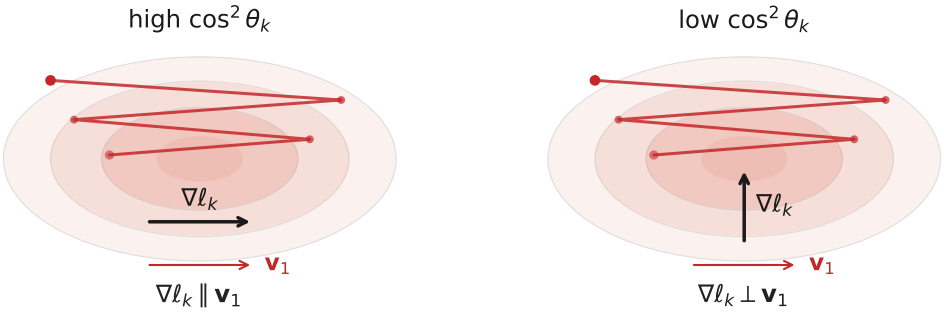

Figure 3: **Directional coupling at EoS.** The optimizer oscillates along $\mathbf{v}_1$ (red zigzag). When a group's gradient $\nabla \ell_k$ aligns with $\mathbf{v}_1$ (left), self-stabilization reduces loss for that group. When $\nabla \ell_k$ is orthogonal to $\mathbf{v}_1$ (right), the group is decoupled from the oscillation and its loss does not benefit.

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

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

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.

3 Selective Learning at the Edge of Stability

3.1 Setup and 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 2.3. 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. Full experimental setup details can be found in Appendix A.

3.2 The Selective Trade-off

Figure 4 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, favoring some groups at the expense of others.

The selective trade-off replicates across alternative architectures (CNN, ResNet), optimizers, and class pairs (Appendix C). It also transfers to binary ImageNet32, though it weakens as the number of classes grows (Appendix D).

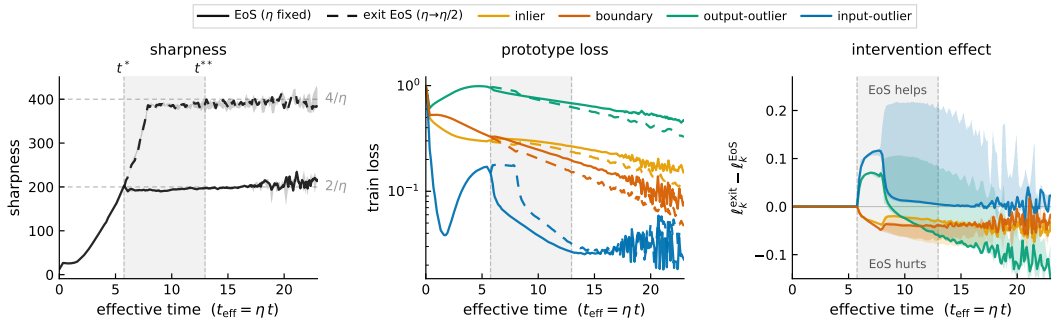

Figure 4: **EoS creates selective trade-offs across prototype groups.** Baseline run (solid) enters EoS at t^*, while the exit branch (dashed) subsequently leaves. **Left:** sharpness confirms the two branches occupy distinct stability regimes. **Middle:** prototype losses diverge post-branch, with input-outliers 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.

3.3 Alignment Resolves Under EoS, Persists Without It

Figure 5 shows directional alignment $\cos^2\theta_k$ for input-outliers and boundary points. During progressive sharpening, alignment for input-outliers increases rapidly, approaching 1 near the onset of EoS, while boundary alignment remains below 0.2. As a result, input-outlier gradients dominate the top eigendirection of the Hessian, and $\mathbf{v}_1$ aligns with them.

Once EoS is reached at t^*, input-outlier $\cos^2\theta_k$ declines under the baseline as self-stabilization reduces their loss and gradient magnitude. As their curvature contribution weakens, boundary alignment rises and $\mathbf{v}_1$ rotates toward it. Under the exit branch, this transition does not occur: input-

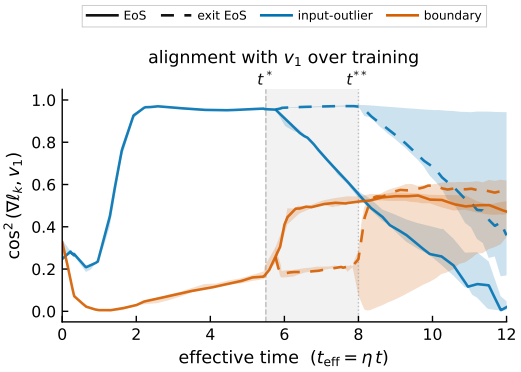

Figure 5: **Alignment dynamics under EoS.** Input-outlier $\cos^2 \theta_k$ rises during progressive sharpening and dominates $\mathbf{v}_1$ at EoS onset. Under the baseline (solid), self-stabilization resolves this alignment and $\mathbf{v}_1$ rotates toward boundary points. Under the exit branch (dashed), alignment remains elevated until the new threshold is reached at t^{**}.

outlier alignment remains elevated through t^{**}, indicating that the resolution is driven specifically by EoS dynamics. This motivates identifying the properties that determine which group dominates $\mathbf{v}_1$.

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.

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 E).

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 6). 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. The group must push the Hessian coherently along a shared direction to dominate $\mathbf{v}_1$ and benefit at the edge of stability.

At the neuron level, our gradient analysis suggests that coherent outliers reinforce updates to shared ReLU units. In a controlled sandbox, this reinforcement accompanies the emergence of rarely active, high-magnitude neurons that selectively fit the subgroup, consistent with the neural-shattering phenomenon of Liang et al. [17, 18]. Appendix F develops this connection and its relationship to subgroup-specific EoS dynamics.

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 7).

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

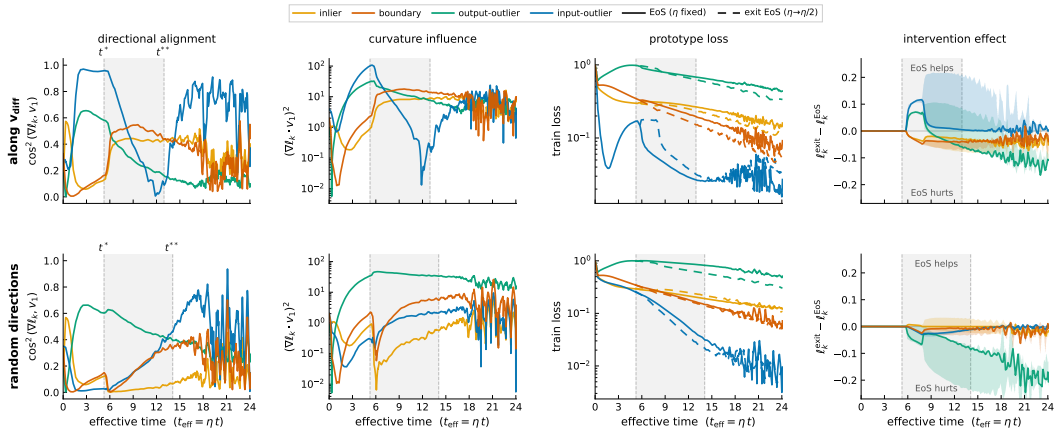

Figure 6: **Directional alignment is necessary for the selective EoS advantage.** Identical seeds and configurations; only the input-outlier displacement direction differs. **Top:** Coherent displacement along v_{diff} yields high alignment and curvature influence for input-outliers, which capture the EoS advantage. **Bottom:** Random orthogonal displacement at equal distance reduces alignment and curvature influence, largely eliminating the trade-off.

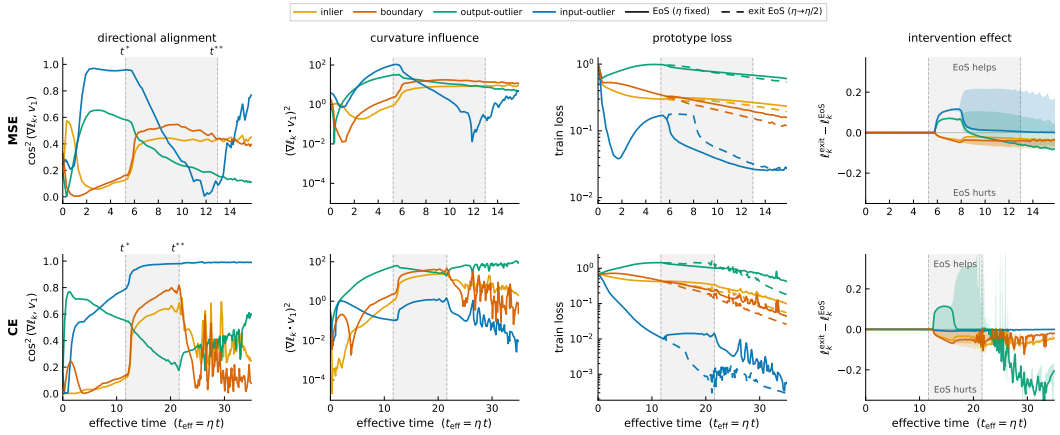

Figure 7: **Gradient persistence determines which group retains curvature influence.** Seeds and configurations are identical, and only the loss differs. **Top:** Input-outliers have elevated alignment and strong curvature influence, and they capture the EoS advantage. **Bottom:** Alignment for input-outliers remains high, but gradient saturation weakens their curvature influence, and the EoS advantage shifts to output-outliers, whose gradients remain active.

shrink. The functional consequence is that output-outliers, with the most elevated curvature influence, become the sole beneficiary at the edge of stability.

4.3 Natural Outliers in the Data Distribution.

To verify that the coupling between centroid distance and $\mathbf{v}_1$ alignment is not an artifact of synthetic prototype construction, we measure per-example directional alignment directly on the natural CIFAR-10 training distribution. For each of the 10,000 training examples, we compute centroid distance $\|x_i - \mu_c\|$ and per-example alignment $\cos^2(\nabla \ell_i, v_1)$ at checkpoints throughout training.

Figure 8 shows the relationship at two timepoints. During progressive sharpening, centroid distance and $\cos^2$ are uncorrelated (Spearman $\rho = -0.11$). At EoS, a positive correlation emerges ($\rho = 0.39$), where examples that are farther from their class centroid align more strongly with $\mathbf{v}_1$. Centroid distance, a purely distributional quantity computed before training, correlates with which examples will dominate the unstable direction at EoS.

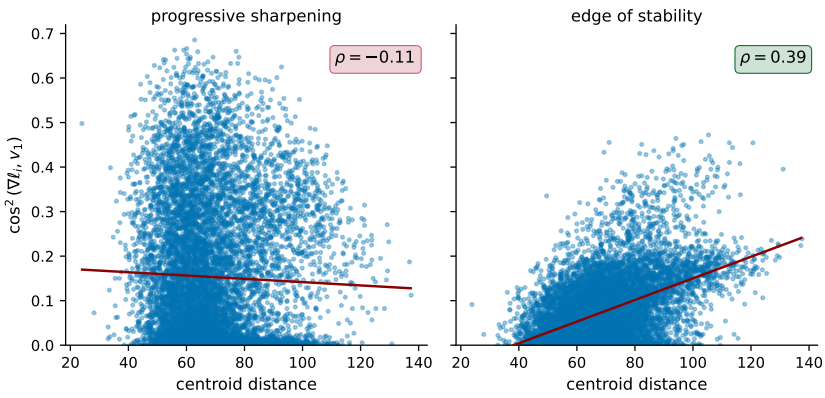

Figure 8: **Centroid distance predicts per-example alignment with $\mathbf{v}_1$ at EoS.** Each dot is one training example; the line indicates the monotonic trend captured by Spearman ρ. **Left:** during progressive sharpening, centroid distance and $\cos^2(\nabla \ell_i, v_1)$ are uncorrelated ($\rho = -0.11$). **Right:** at EoS, correlation emerges ($\rho = 0.39$). Full trajectory of correlation is shown in Appendix G.

5 Generalizing the Alignment Principle

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.

5.1 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, boundary points 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 C.3).

5.2 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 9 reports PGD adversarial accuracy ($\varepsilon = 0.03$, 10 steps, step size $\varepsilon/4$) on 50 test-set boundary points selected by k-NN ambiguity [19]. 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 spending its optimization budget elsewhere.

Out-of-distribution generalization. Figure 10 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}.

6 Discussion

We find that the EoS stability constraint acts as an inductive bias, not merely an implicit regularizer. Rather than selecting among equivalent minimizers, it determines which subsets of the data are

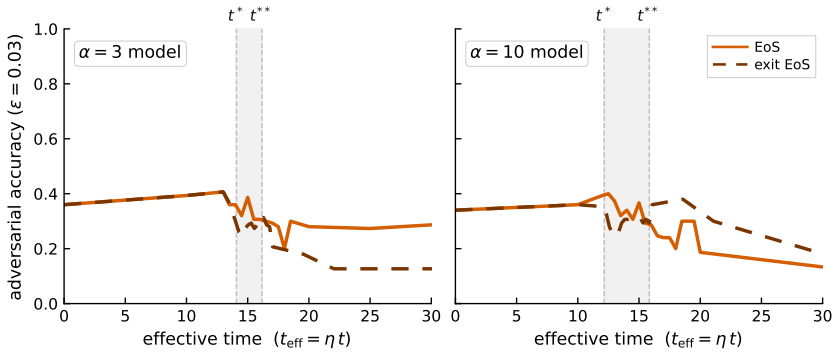

Figure 9: **EoS improves adversarial robustness only when boundary points dominate $\mathbf{v}_1$. Left** ($\alpha = 3$, boundary dominates $\mathbf{v}_1$): the EoS branch (solid) outperforms the exit branch (dashed) after t^{**}. **Right** ($\alpha = 10$, input-outlier dominates $\mathbf{v}_1$): the pattern reverses, and the exit branch performs better. Robustness gains appear only when EoS prioritizes the evaluated subset. Single seed.

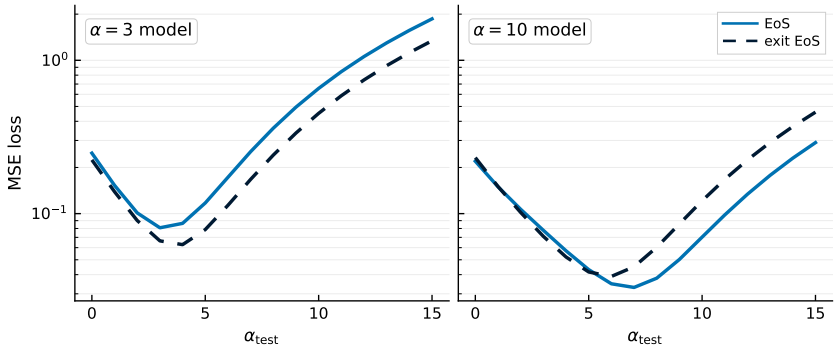

Figure 10: **EoS steers generalization toward the dominant group.** Test MSE on input-outliers across α_{test} (after t^{**}). **Left** ($\alpha = 3$, boundary dominates $\mathbf{v}_1$): no OOD advantage. **Right** ($\alpha = 10$, input-outlier dominates $\mathbf{v}_1$): EoS improves OOD performance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

optimized, as a function of data geometry and loss. Concretely, EoS optimizes on the subset with the largest curvature influence, and the functional consequence depends on which subset dominates.

Comparing the $\alpha = 3$ and $\alpha = 10$ models illustrates how data geometry changes the functional effect of EoS. When boundary points dominate $\mathbf{v}_1$, EoS focuses optimization near the decision boundary and can improve adversarial robustness; when input-outliers dominate, it instead shifts optimization toward distributional tails and can improve extrapolation along the outlier direction (Appendix C.3). Thus, hyperparameters typically viewed as controlling convergence—such as learning rate or loss—may also influence which subset of the data, and hence which functional property of the model, is emphasized.

This perspective offers a possible explanation for divergent findings in the flatness literature. The benefit of low or controlled sharpness is not determined by scalar flatness alone, but by which subset captures the top curvature direction during training. Different empirical settings may therefore induce different functional outcomes, depending on whether boundary points, input-outliers, or other subgroups dominate $\mathbf{v}_1$. This yields a testable prediction: changing the subgroup that dominates $\mathbf{v}_1$ should change the direction of the resulting robustness or generalization effect.

Limitations and scope. Our experiments focus on relatively small models and low-resolution datasets because tracking EoS requires repeated estimation of the top Hessian eigenvalue through Hessian–vector products, with costs increasing with model and dataset size. This limitation is common in the EoS literature [7, 9], whose experimental setups we build upon. Our multiclass experiments extend beyond binary CIFAR-10 to ImageNet32 (Appendix D), but also highlight the limitations of single-mode analysis as the number of classes increases. Extending the mechanism to

multiple interacting curvature directions remains future work. We also focus on full-batch gradient descent, as extending the subset-level self-stabilization analysis [15] to mini-batch training requires a corresponding theoretical treatment of stochastic dynamics. Our prototype taxonomy is defined in pixel space, which relates most directly to gradient geometry in MLPs. In convolutional architectures, the curvature-influence score $(\nabla \ell_k \cdot v_1)^2$ remains informative, but the dominant subgroup can change as learned features alter the mapping from inputs to gradients. Extending the taxonomy to representation space is therefore a natural direction. Finally, our robustness and generalization findings remain preliminary and warrant broader evaluation across architectures, seeds, and distribution shifts.

7 Conclusion

The edge of stability is not a uniform optimization phenomenon, but a selective one: the stability constraint redistributes learning unevenly across the training distribution, accelerating progress on some subsets while suppressing others. We find that two properties jointly determine which subsets benefit: directional alignment with the top Hessian eigendirection and persistence of gradient signal under the loss function. Removing either component—by disrupting directional coherence or by inducing gradient saturation—eliminates or shifts the advantage to different subsets. Importantly, by characterizing how a curvature constraint differentially shapes learning across the data distribution, this work provides a step toward connecting implicit regularization in parameter space with inductive bias over examples.

Acknowledgements

PB and TP acknowledge support from the McGovern Institute for Brain Research, and TP additionally acknowledges support from the McDermott Endowed Chair. The authors thank Arseniy Andreyev, Jeremy Cohen, and Alex Damian for fruitful discussions.

References

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

[2] 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.

[3] 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.

[4] Chen Xing, Devansh Arpit, Christos Tsirigotis, and Yoshua Bengio. A walk with sgd. *arXiv preprint arXiv:1802.08770*, 2018.

[5] 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. arXiv:1807.05031.

[6] 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. arXiv:2002.09572.

[7] 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.

[8] 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.

[9] Arseniy Andreyev and Pierfrancesco Beneventano. Edge of stochastic stability: Revisiting the edge of stability for SGD. *arXiv preprint arXiv:2412.20553*, 2024.

[10] 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.

[11] 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.

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

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

[14] 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.

[15] 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.

[16] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[17] Tongtong Liang, Dan Qiao, Yu-Xiang Wang, and Rahul Parhi. Stable minima of ReLU neural networks suffer from the curse of dimensionality: The neural shattering phenomenon. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[18] Tongtong Liang, Alexander Cloninger, Rahul Parhi, and Yu-Xiang Wang. Generalization below the edge of stability: The role of data geometry. In *International Conference on Learning Representations*, 2026.

[19] 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.

[20] 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.

[21] Arseniy Andreyev and Pierfrancesco Beneventano. Edge of stochastic stability, 2025. Software, Apache 2.0 license.

[22] Marius Saether, Amir Kolic, Tomaso Poggio, and Pierfrancesco Beneventano. Does weight decay enhance training stability? *arXiv preprint arXiv:2605.16622*, 2026.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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. arXiv:2107.07075.

[32] 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.

[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.

[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.

[35] 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.

[36] 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.

[37] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[38] Lei Wu, Zhanxing Zhu, and Weinan E. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

[39] 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.

[40] 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.

[41] 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.

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

[43] Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[44] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

[45] Levent Sagun, Utku Evci, V. Uğur Güney, Yann Dauphin, and Léon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

A Experimental setup

A.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.

A.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.

A.3 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 $\eta \cdot$ step to provide a valid comparison of speed across different learning rates.

A.4 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 $2/\eta$ [20]. 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.

A.5 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 code is publicly available at this repository. Our implementation builds on the open-source codebase of [9], which provides the CIFAR-10 training pipeline and curvature and sharpness logging infrastructure (Apache 2.0; 21). 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 Related Work

Our work connects four threads: training dynamics at EoS, sample-centric analyses of which examples drive learning, curvature-based implicit regularization, and the shatterability of data by ReLU activation regions. 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.

B.1 Training Dynamics at the Edge of Stability

Onset of EoS. EoS occurs when the top Hessian eigenvalue grows along the optimization trajectory until it reaches the stability threshold $2/\eta$ [4–7]. Cohen et al. [7] termed this growth phase progressive sharpening. 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 [7]. Recent work has shown that EoS-like phenomena extend beyond this setting, to adaptive optimizers, stochastic training, and momentum dynamics [8–10], and that saturation levels may be different in case of regularization [22]. 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 et al., where oscillations along the top Hessian eigenvector $\mathbf{v}_1$ bound curvature via self-stabilization [15]. Specifically, these oscillations can feed back through higher-order derivatives to reduce sharpness, yielding an implicit projected dynamics near the stability boundary [15]. Prior work characterizes EoS through global dynamics—fast oscillation along $\mathbf{v}_1$ and slow sharpness-reducing drift along flat directions [23–25]. We instead ask which examples contribute to these dynamics, revealing a selective effect over the data distribution that scalar sharpness obscures.

B.2 Sample-Centric Learning

A parallel literature studies which individual examples drive learning [26–34]. Rare or atypical examples can disproportionately influence generalization: memorizing them is necessary for near-optimal generalization on long-tailed distributions [27, 28]. Toneva et al. [29] also track forgetting events and find rare examples are repeatedly forgotten and relearned. Other works score rare examples by difficulty throughout training and reveal ambiguous examples (those which are flipped throughout training) are key for out-of-distribution generalization [30]. Rare examples can also dominate gradient signal and sharpening dynamics [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 [31] 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.

B.3 Implicit regularization through curvature.

Curvature has been widely linked to generalization: sharp minima correlate with worse performance, while smaller batch or higher learning rates implicitly bias toward flatter solutions [2, 3, 12, 13, 35–38]. Other methods make the flatness objective explicit or directly bias the final iterate toward flatter regions, including Sharpness-Aware Minimization and stochastic weight averaging [13, 39]. 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 [23–25, 40, 41]. We add a data-level perspective: at EoS, curvature acts selectively across examples, with functional consequences for which subsets generalize and which are robust.

B.4 Neural Shattering and Data Geometry

Liang et al. [17] show that stable minima of two-layer ReLU networks can exhibit *neural shattering*, where rarely active, high-magnitude neurons fit small subsets of the data, leading to poor generalization in high dimensions. Follow-up work [18] links this behavior to data geometry: when training examples can be readily isolated by ReLU activation regions, gradient descent below EoS can favor memorization over shared representations. We provide a complementary perspective at EoS, showing that coherent subgroup gradients reinforce one another and accompany the emergence of selectively activated ReLU neurons that fit the subgroup (Appendix F). The neural-shattering and data-shatterability perspectives are complementary to ours: shared geometry can make a subgroup isolatable by localized ReLU regions, while EoS preferentially acts on that subgroup only when its aggregate gradient forms a persistent near-critical mode. A coherent subgroup may be represented through a shared low-rank feature rather than private shattering. Representational availability and EoS selection should therefore not be identified.

B.5 Our positioning

Our work connects these research directions by showing how the dominant curvature direction at EoS shapes which groups of examples benefit from training and which experience suppressed progress. Prior work establishes that small groups of outliers with opposing signals can disproportionately influence sharpening and EoS dynamics [33]. We extend this perspective by introducing a model-independent prototype taxonomy based on initial data geometry and identifying gradient alignment as a key factor in selective optimization. Complementing work on neural shattering [17, 18], our controlled experiments show how coherent subgroup gradients accompany the emergence of selective ReLU neurons before EoS onset, after which the EoS loss oscillations concentrate on that subgroup (Appendix F). Together, these results reveal how a global stability constraint can produce unequal optimization outcomes across training examples.

C Robustness Across Experimental Settings

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 pairs, 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 $\Delta\ell_k = \ell_k^{\text{exit}} - \ell_k^{\text{EoS}}$.

C.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.

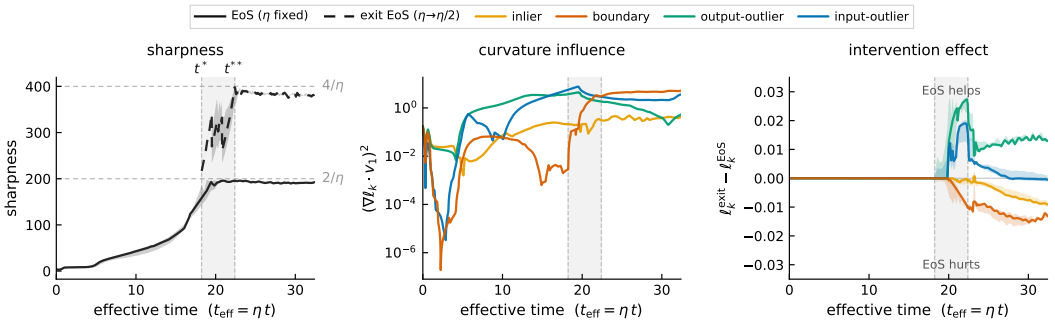

Figure 11: GD CNN MSE. Curvature influence is comparable for output-outliers and input-outliers, both benefit.

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.

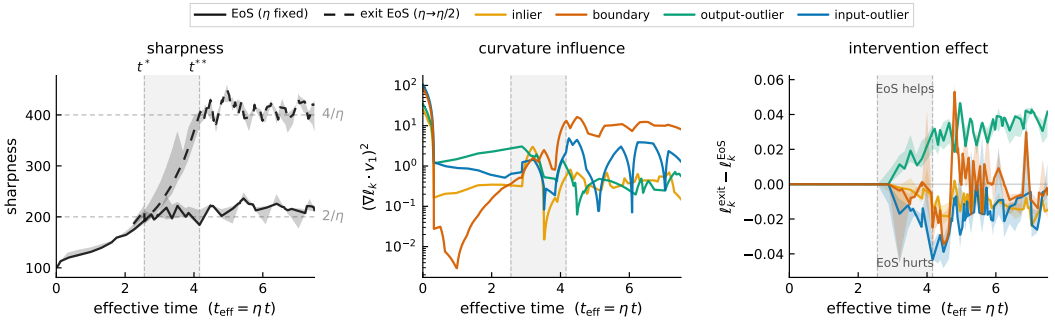

Figure 12: GD ResNet MSE. Curvature influence is highest for output-outliers, and it is the primary beneficiary of EoS.

C.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 [42] 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. With a large batch (Figure 13), the group with the highest curvature influence (input-outliers) is correspondingly the primary beneficiary of EoS. Similarly for Figure 14, input-outliers have the highest curvature influence, and the intervention benefits them.

SGD (batch size=128).

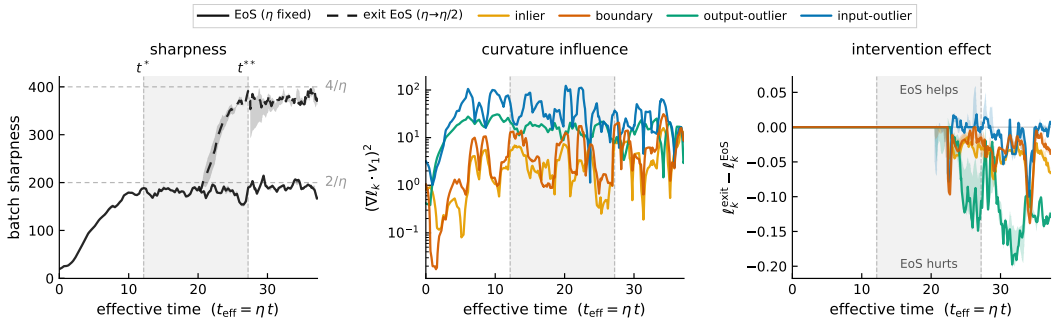

Figure 13: SGD MLP MSE. Curvature influence is highest for input-outliers, and it is the primary beneficiary of EoS.

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

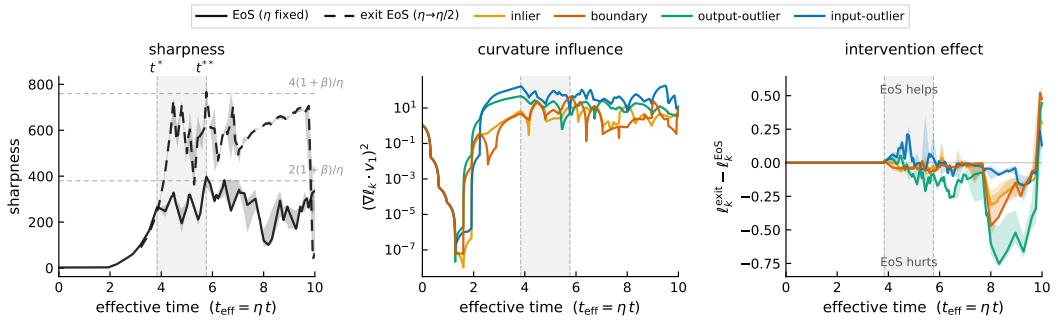

Figure 14: GD Momentum MLP MSE. Curvature influence is highest for input-outliers, and it is the primary beneficiary of EoS. Momentum implemented as in [43] and EoS threshold implemented as described for large-batch momentum in [10].

C.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 15). The alignment principle predicts that boundary points should therefore capture the EoS advantage on this pair, and they do (Figure 16).

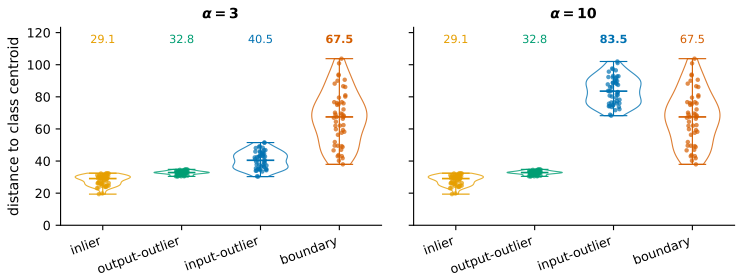

Figure 15: Distribution of centroid distance by prototype subgroup on the (3,5) class pair. Boundary points have the largest median distance under $\alpha = 3$ but input-outliers have the largest median distance under $\alpha = 10$.

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.

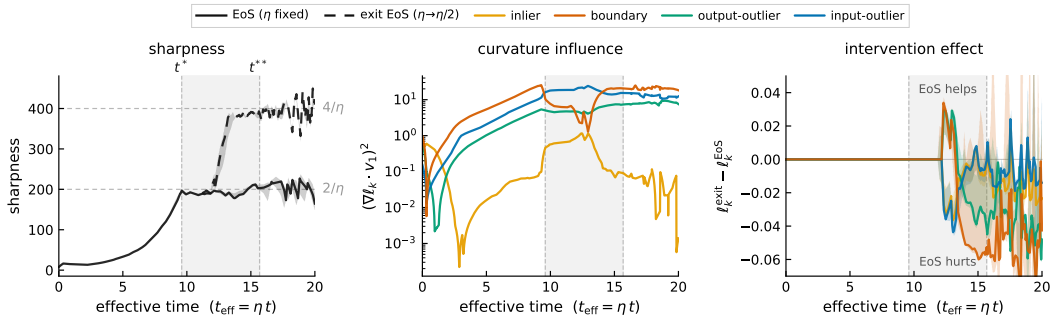

Figure 16: Curvature influence under input-outlier construction ($\alpha = 3$). The boundary group has the highest curvature influence and is the primary beneficiary of EoS.

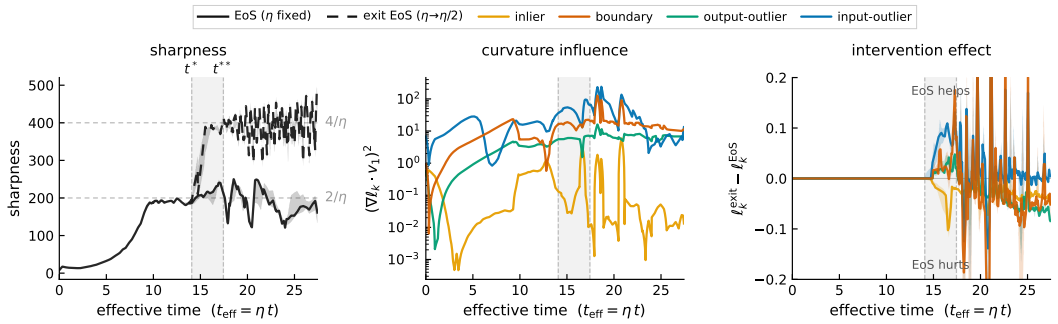

Figure 17: Curvature influence under input-outlier construction ($\alpha = 10$). The input-outlier group has the highest curvature influence and is the primary beneficiary of EoS.

D Extension to ImageNet32 and Multiclass Settings

Experimental setup. We extend the protocol to $K \in \{2, 3, 5, 10\}$ class subsets of ImageNet32 [44] (MLP, MSE loss, full-batch gradient descent, three seeds per K), with the identical EoS to exit-EoS fork: $\eta = 0.01$, reduced to $\eta_{\text{low}} = 0.005$ at the baseline's onset t^*. The prototype construction generalizes per class: inlier and boundary pools are drawn per class, output-outliers are relabeled to the nearest alternative class centroid, and input-outliers are extrapolated away from the nearest alternative class. Both reduce exactly to the binary construction at $K = 2$, so the main-text setting is a special case of this design. All runs exhibit standard EoS dynamics and re-equilibrate after the drop, with onset arriving later as K grows, so the branching intervention is well defined at every class count.

The single-mode trade-off weakens with K. On ImageNet32, the trade-off persists at $K = 2$ and $K = 3$: input-outliers benefit from staying at EoS ($\Delta\ell_k > 0$ within the intervention window), while every other group benefits from exiting. At $K = 5$ and $K = 10$, every group benefits from exiting and the trade-off weakens (Figure 18). As K grows, multiple curvature directions compete for the instability and absolute alignment with $\mathbf{v}_1$ dilutes across class groups: at $K = 10$, output-outliers carry roughly 25% of the gradient norm but only 3% of the $\mathbf{v}_1$-alignment. By the late global onset, input-outliers have largely lost their dominant coupling to $\mathbf{v}_1$ and have already been fit, so alignment no longer coincides with a non-saturated residual, which the mechanism predicts is necessary for the trade-off (Appendix H.4). A multiclass extension should therefore track a near-critical subspace rather than a single $\mathbf{v}_1$.

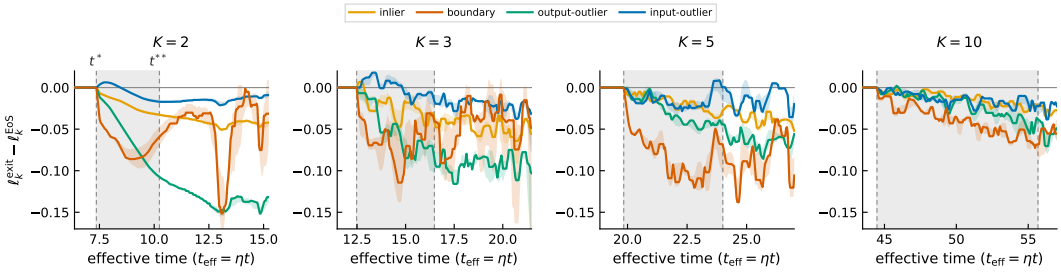

Figure 18: Multiclass ImageNet32 experiments with MSE loss, varying the number of classes K. While input-outliers benefit from staying at EoS at $K = 2$ or $K = 3$, no comparable advantage emerges at $K = 5$ or $K = 10$.

E Input-Outlier Construction Ablation

The conceptual schematic for the directional perturbation is shown in Figure 19. Figure 20 confirms that geometric atypicality of input-outlier is preserved under the random-direction control.

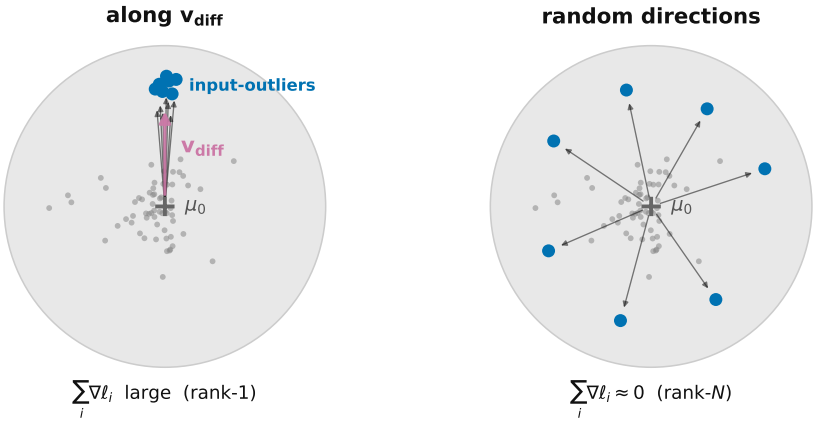

Figure 19: Schematic for coherent vs. incoherent input-outlier construction. **Left:** input-outliers displaced along a shared direction v_{diff}; per-example gradients reinforce, producing a dominant curvature contribution along one direction. **Right:** same displacement distance but in random orthogonal directions; per-example gradients partially cancel, producing a diffuse contribution across many eigenvectors. Seed points, labels, and centroid distances are identical across conditions.

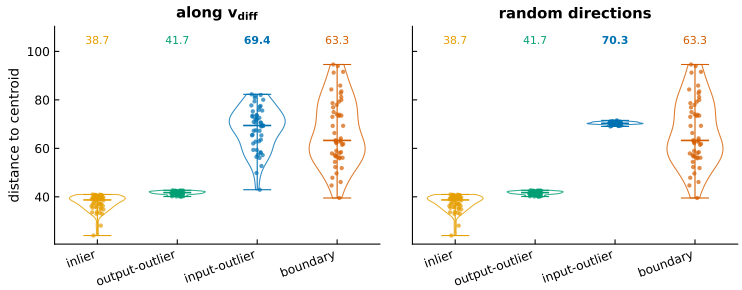

Figure 20: Centroid distance is preserved under the random-direction control (median 69.4 vs. 70.3 for along-v_{diff} and random-direction outliers, respectively). Geometric atypicality is conserved.

F Gradient Mechanism of Data Selection and Connection to Neural Shattering

This appendix examines how coherent input geometry produces reinforcing per-neuron gradients and how these gradients relate to subgroup-selective ReLU activation. In a controlled sandbox, coherent input-outliers exhibit strong gradient alignment and substantially better fitting than equally displaced random-direction outliers, although sparsely activated neurons arise in both conditions. This provides a neuron-level perspective on the coherence mechanism of Lemma 4, connects our findings to neural shattering [17, 18], and complements the full-scale comparison in Appendix E.

Per-neuron gradient analysis. For a two-layer ReLU network $f(x) = \sum_j a_j \sigma(w_j^\top x + b_j)$ trained with MSE, the hidden-weight gradient is

$$\frac{\partial L}{\partial w_j} = \frac{2a_j}{n} \sum_i r_i \, \mathbf{1}\{w_j^\top x_i + b_j > 0\} \, x_i, \qquad r_i = f(x_i) - y_i,$$

so each neuron receives signal only from examples in its active halfspace, weighted by their residuals. Let S contain k coherent input-outliers $x_i = s_i + \alpha u$, where $u = \pm v_{\text{diff}}$ is the per-class outward displacement direction (Section 2.3). Writing $I_{ij} = \mathbf{1}\{w_j^\top x_i + b_j > 0\}$, the subgroup contribution to neuron j decomposes as

$$g_j^S = \frac{2a_j\alpha}{n} \left(\sum_{i \in S} r_i I_{ij} \right) u + \frac{2a_j}{n} \sum_{i \in S} r_i I_{ij} s_i.$$

When gates and residual signs are consistent across the subgroup, the displacement terms reinforce into an $O(k\alpha)$ pull along u, whereas independent random displacements of equal magnitude contribute only $O(\sqrt{k}\,\alpha)$ on average, with no shared direction. Because ReLU gating depends on the inputs, these scalings are heuristic. A neuron whose halfspace isolates S receives nearly pure subgroup signal, and the displacement-dependent part of its Gauss–Newton curvature contribution ($\nabla_{w_j} f = a_j I_{ij} x_i$) scales as α^2 for fixed gates. Whether gradient descent reaches such configurations is the empirical question the sandbox examines.

Sandbox experiment. We test this mechanism on a 2-D two-Gaussian binary MSE task ($n = 208$) with a two-layer ReLU network ($m = 50$) and full-batch GD, displacing $k = 8$ class-0 seeds either coherently along αu with $\alpha = 2$ or along equal-norm independent random directions (the 2-D analogue of the control in Appendix E). In both runs, $\lambda_{\max}$ rises quickly to just below the stability threshold and first crosses $2/\eta$ at t^* (median $\lambda_{\max}/(2/\eta) = 0.999$ and 0.998). Selective activation patterns form before t^*: one neuron is selective from initialization, and the other completes its migration shortly before onset. Pronounced loss oscillations then appear on the already-fitted subgroup.

Figure 21 summarizes the results. In the coherent run, the mean pairwise cosine of subgroup gradient contributions within a neuron's active set approaches 1, and two neurons become selective for the subgroup. They activate on 88% and 75% of the input-outliers but on at most 1.5% of the bulk, whereas every other neuron covering the subgroup fires on at least 6.5% of the bulk (Figure 21, middle). Their halfspaces isolate the subgroup (Figure 21, left), and their received subgroup gradient satisfies $|\cos(g_j^S, u)| = 0.99$. The subgroup is fit to MSE $\approx 10^{-4}$ by t^*, below the bulk loss, and during the subsequent oscillatory phase its loss fluctuates over three orders of magnitude while the bulk loss varies by about a factor of two (Figure 21, right). In the random-direction control, pairwise cosines remain around 0.2–0.5, alignment reaches only 0.60, and subgroup MSE stays above 3×10^{-2}. The control also produces rarely active neurons that capture a few displaced points, but none combines selective activation with reinforced gradients and subgroup fitting.

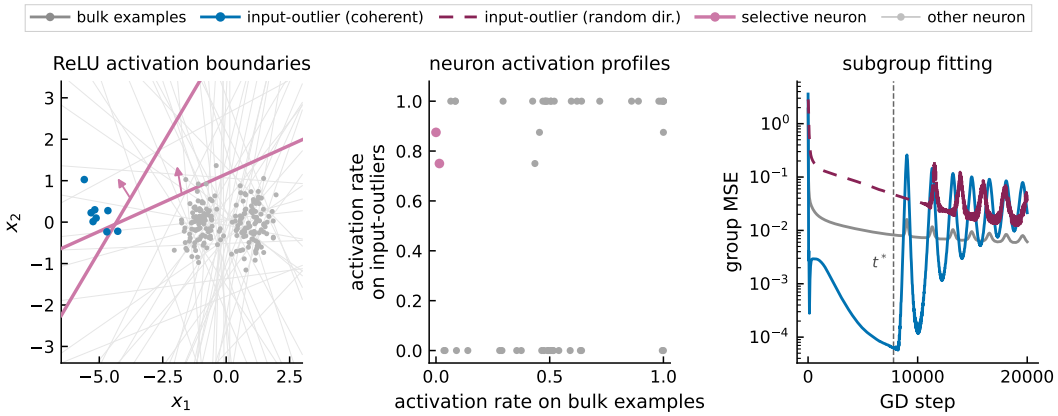

Figure 21: Coherent input-outliers are isolated and fit by subgroup-selective ReLU neurons. **Left:** Activation boundaries of all $m = 50$ neurons, highlighting two whose active halfspaces (arrows) isolate the input-outliers from the bulk. **Middle:** Neuron activation rates on input-outliers versus bulk examples. The two selective neurons activate on most outliers but at most 1.5% of the bulk. **Right:** Per-group training loss. Coherent outliers are fit below the bulk by EoS onset t^*, followed by pronounced subgroup loss oscillations, while random-direction outliers remain at higher loss.

Relation to neural shattering. The selective neurons show the qualitative signatures of neural shattering [17]. They activate rarely (3–4% of all examples), carry effective weight magnitude $\|w_j\||a_j|$ about twice the median, and fit a small group of points locally. The two mechanisms are related but not identical. Neural shattering describes many nearly private neurons that each memorize individual points, whereas here a small number of neurons exhibit activation patterns shared across the coherent subgroup, consistent with group-level specialization rather than pointwise memorization. The perspectives are therefore complementary. Data shatterability [18] describes which localized representations are available to the network, while alignment and persistence (Section 4, Appendix H) determine which subgroup EoS preferentially optimizes.

G Natural points

We verify that the alignment principle holds on natural geometric outliers, with no synthetic displacement applied. Figure 22 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. The Spearman coefficient measures rank agreement; lines show the line of best fit for visual clarity.

Figure 23 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.

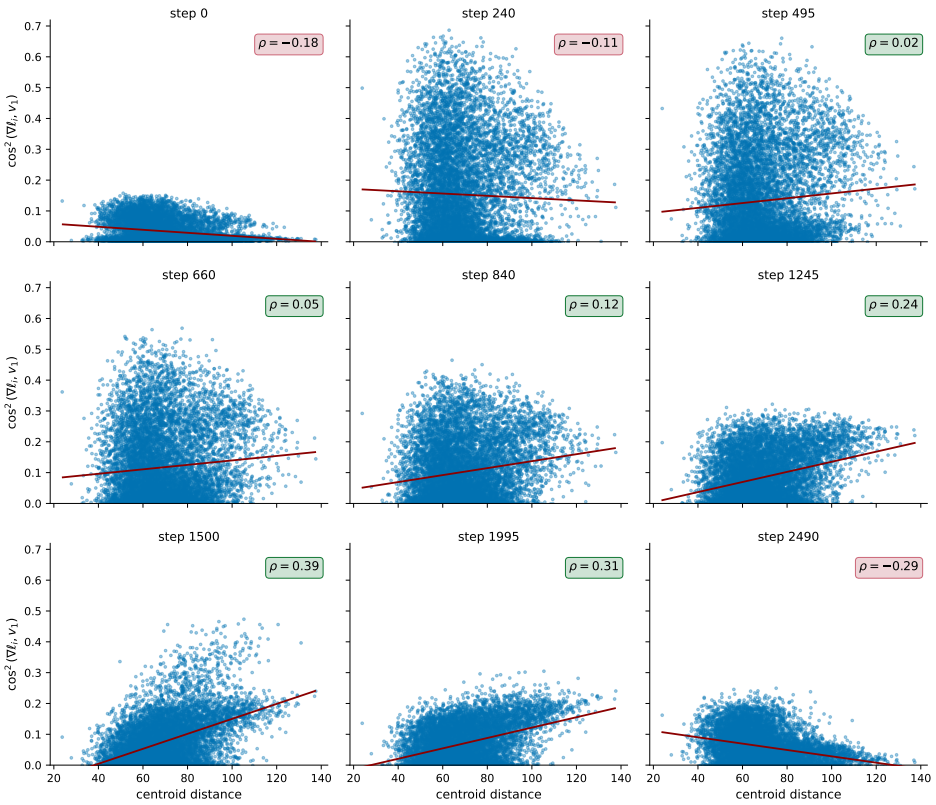

Figure 22: **Top:** early in training, the correlation starts negative and increases. **Middle:** at EoS onset, the correlation increases monotonically. **Bottom:** the correlation peaks when large-amplitude oscillations along $\mathbf{v}_1$ develop, then decreases monotonically and turns negative.

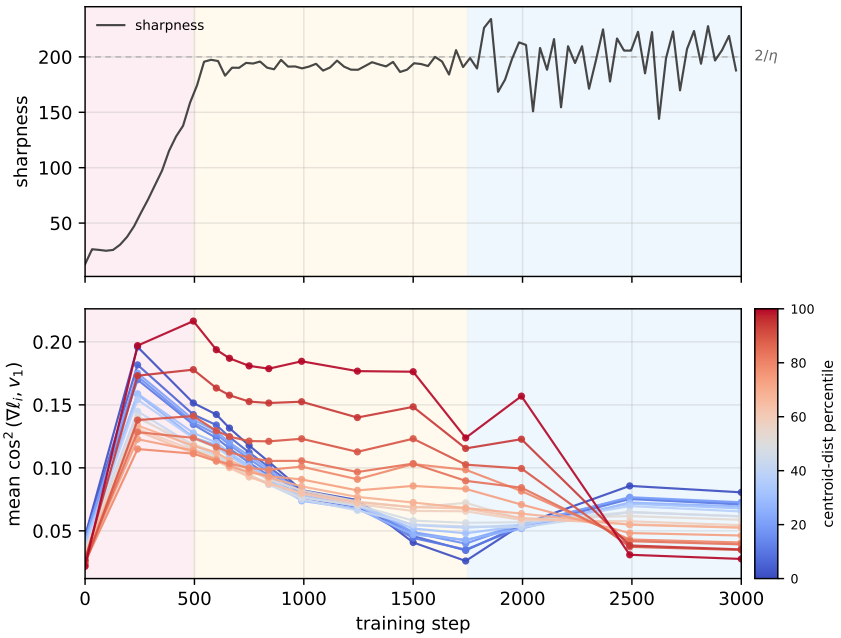

Figure 23: **Top:** correlation rises at EoS onset and peaks with sharpness oscillations. **Bottom:** alignment vs. centroid-distance percentile over training. Atypical points peak near EoS (step 660), then decline, and dominance shifts to typical points later in training.

H 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 [15]: 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.

H.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 [15], 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 [15].

(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.

H.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 $\langle \nabla \ell_k, \nabla S\rangle$.

H.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

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

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

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

Using the local EoS drift from Lemma 1, this becomes

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

Expanding the inner product gives

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

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:

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

and

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

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

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

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.

H.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 $r_\perp$. This splits the selector as

$$Q_k \;=\; \underbrace{\gamma\langle \nabla\ell_k, v_1\rangle}_{T_1^{(k)}} \;+\; \underbrace{\langle \nabla\ell_k, r_\perp\rangle}_{T_2^{(k)}},$$

and C_k functions as a single-mode proxy for the squared selector magnitude when the v_1-component term $T_1^{(k)}$ determines the sign and ranking of Q_k. Appendix H.6 tests this condition by evaluating Q_k directly. 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). Appendices H.7 and H.8 establish that both factors are necessary.

H.5 Single-Mode Approximation at EoS Onset

We now verify the single-mode assumption empirically. Tracking the top three full-batch Hessian eigenvalues confirms that at EoS onset only λ_1 is near the stability boundary: $\lambda_1 = 200.3$ (within 0.2% of $2/\eta$), $\lambda_2 = 99.7$, and $\lambda_3 = 46.0$, giving $\lambda_2/\lambda_1 = 0.498$. The gap narrows later in training (λ_2/λ_1 reaches 0.93), yet even at $\lambda_2/\lambda_1 = 0.90$ the full-batch gradient places 95% of its energy in $\mathbf{v}_1$ versus 3.6% in $\mathbf{v}_2$, so a narrowing eigengap does not by itself produce a two-dimensional update. The low-dimensional dominant spectrum is consistent with prior spectral analyses [45].

Table 1 reports per-group coupling to the top two modes in the early EoS window following onset ($t_{\text{eff}} \in [6, 15]$). The outlier groups driving the instability couple to $\mathbf{v}_1$ by one to two orders of magnitude over $\mathbf{v}_2$, so the single-mode analysis captures the groups on which the mechanism operates. The approximation is therefore sufficient for the onset-centered selection claim, though not for late training, where the spectrum and couplings shift (Figure 24).

H.6 Direct validation of the single-mode proxy

We evaluate the selector $Q_k = \langle \nabla\ell_k, \nabla S\rangle$ directly. For a simple top eigenvalue, the Hellmann–Feynman identity gives $\nabla_\theta\lambda_1 = \nabla_\theta\, v_1^\top H(\theta)\, v_1$ with the normalized eigenvector held fixed, which

Table 1: Per-group coupling to the top two Hessian modes, medians over checkpoints in the early EoS window following onset ($t_{\text{eff}} \in [6, 15]$).

Group	$\cos^2(\nabla \ell_k, \mathbf{v}_1)$	$\cos^2(\nabla \ell_k, \mathbf{v}_2)$
Input-outliers	0.68	0.003
Output-outliers	0.47	0.04
Boundary points	0.09	0.12
Inliers	0.02	0.16

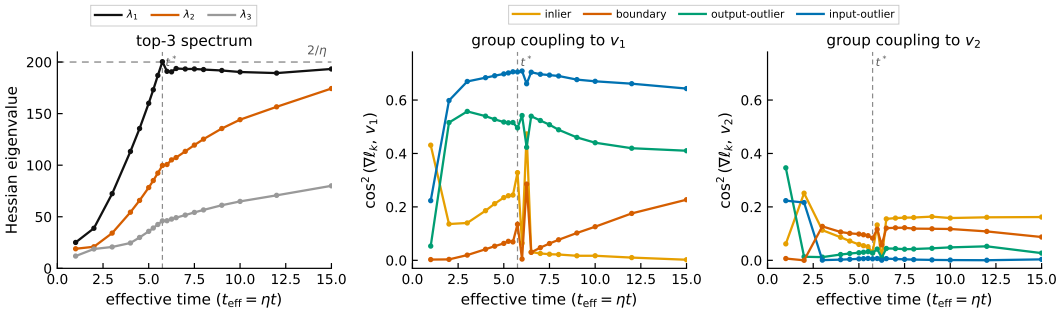

Figure 24: Single-mode structure at EoS onset. **Left:** top-3 full-batch Hessian eigenvalues against the stability boundary $2/\eta$. **Middle:** per-group coupling $\cos^2(\nabla \ell_k, \mathbf{v}_1)$. **Right:** the same quantity on $\mathbf{v}_2$. t^* is marked on all panels. Single representative run.

we evaluate by higher-order automatic differentiation of the top-eigenvalue Rayleigh quotient at the converged LOBPCG eigenvector. This removes finite-difference approximation error. The remaining error comes from eigenvector convergence and floating-point arithmetic, and we control it by verifying a normalized eigenpair residual below 10^{-4} at every checkpoint. As a numerical precaution we restrict to checkpoints with relative eigengap $(\lambda_1 - \lambda_2)/\lambda_1 \geq 0.05$, retaining 84–85 of 97 checkpoints per seed. The cutoff is a chosen criterion rather than a requirement of the identity, and the results do not depend on it (pooled Spearman 0.80 ungated vs. 0.81 gated).

We evaluate the proxy through three complementary comparisons. The first comparison tests magnitude. Figure 25 plots Q_k^2 against the nonnegative C_k across (subgroup, checkpoint) pairs, showing a monotone positive relationship spanning several orders of magnitude, with the dashed line marking the median ratio. The second comparison tests subgroup ranking. Comparing the normalized signed selector $\cos(\nabla \ell_k, \nabla S) = Q_k/(\|\nabla \ell_k\| \, \|\nabla S\|)$ with the normalized proxy $\cos^2(\nabla \ell_k, v_1)$ across three seeds gives pooled Spearman $\rho = 0.81$, median per-checkpoint Kendall $\tau = 0.67$ across the four groups, and the same dominant subgroup at 77% of checkpoints (25% under uniform random selection). The dominant-subgroup agreement is unchanged (77%) when computed on the unnormalized Q_k and C_k. Because C_k is nonnegative, these two comparisons test magnitude and ranking, not the selector's sign.

The third comparison tests the sign. Although ∇S is only weakly aligned with v_1 (median $\cos^2(\nabla S, v_1) = 0.05$), Table 2 shows that the top-mode term $T_1^{(k)}$ closely tracks the signed selector's ranking for the non-inlier groups ($\rho(Q_k, T_1^{(k)}) = 0.91$–$0.96$), with sign agreement strongest for output- and input-outliers (0.97 and 0.99) and weaker for boundary points (0.76). For input-outliers the orthogonal residual is comparable in magnitude to the top-mode term, yet the top-mode term remains highly predictive of the selector's sign and ranking. For inliers the v_1-component is roughly an order of magnitude smaller than for the outlier groups and C_k sits near its floor, so the proxy carries little information for this group, consistent with the selector predicting no EoS advantage there.

Table 2: Decomposition of the directly evaluated selector, medians over three seeds. $T_1^{(k)} = \gamma\langle\nabla\ell_k, v_1\rangle$ is the v_1-component term, $T_2^{(k)} = \langle\nabla\ell_k, r_\perp\rangle$ the orthogonal residual term.

Subgroup	med $\lvert T_2\rvert/\lvert T_1\rvert$	$\rho(Q_k, T_1^{(k)})$	sign agreement
Output-outliers	0.31	0.96	0.97
Input-outliers	1.05	0.96	0.99
Boundary points	0.53	0.91	0.76
Inliers	3.54	0.78	0.81

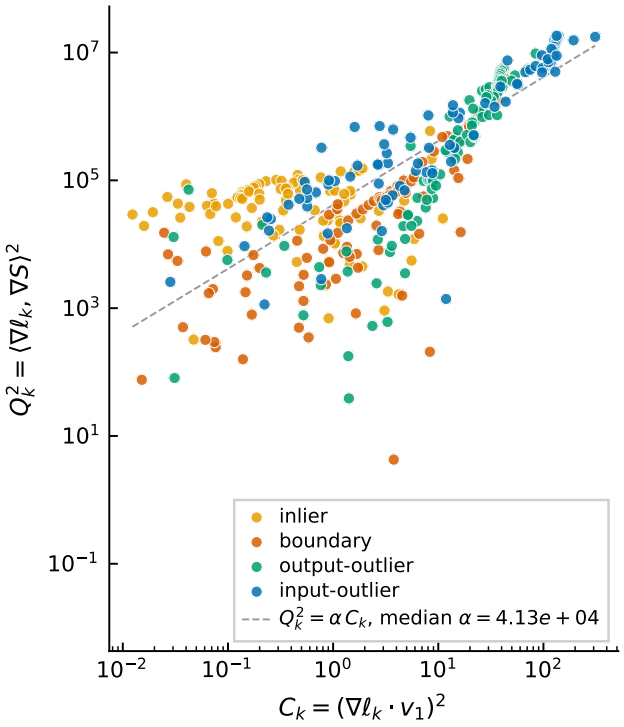

Figure 25: **Empirical validation of the single-mode proxy.** Scatter of $Q_k^2 = \langle\nabla\ell_k, \nabla S\rangle^2$, evaluated directly by higher-order automatic differentiation, versus $C_k = (\nabla\ell_k \cdot v_1)^2$ across (subgroup, checkpoint) pairs in training. The dashed line shows the median ratio $Q_k^2/C_k = 4.1 \times 10^4$. The relationship is monotone for the outlier groups. For inliers, which are nearly orthogonal to v_1, C_k is small and the proxy is loosest, in the regime where the selector predicts no EoS advantage.

H.7 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):

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

Then

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

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 3.3 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.

H.8 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, $\|J_i\|_{op} \le B$. 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 $|\langle \nabla \ell_k, u\rangle| \le \|\nabla \ell_k\|$ 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.

H.9 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, the directly evaluated selector tracks the proxy in both magnitude and ranking across (subgroup, checkpoint) pairs in the EoS regime (Appendix H.6 and Figure 25), 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.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and the introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations of the work are discussed in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Paper provides full set of assumptions and complete proof for theoretical results in Appendix H.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We introduce the experimental setup in Section 3.1, with implementation details provided in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to the data and code with instructions included for reproduction.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We specify the optimizer, learning rate, batch size, model architecture, and number of training steps in Section 3.1. CIFAR-2 data splits and prototype construction are described in Section 2.3. Additional hyperparameter and robustness experiments are reported in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Figures report the median across seeds as the central line. For intervention-effect plots, shaded bands indicate the interquartile range (25th–75th percentile), computed pointwise at each logged training step; other plots show only the median for visual clarity. We use median and IQR rather than mean and standard deviation because the retained-seed distribution is not assumed to be Gaussian and sample sizes are small.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A discusses compute resources needed for reproducing experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to NeurIPS Code of Ethics in all respects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our work studies the theoretical behavior of optimization dynamics, so we anticipate minimal direct societal impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no safety risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We provide a citation of the original authors of the codebase in Appendix A, including the license and credit.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not introduce new datasets or pretrained models.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This study does not involve any human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: This study involves no participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Core method development in this work does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.