

Project track template*

Ivan Lin
Apart Research

Abstract

Write a few paragraphs that concisely and accurately summarize the content of your research, including your physics approach, key findings, and implications for AI safety. Keep it under 250 words. You are welcome to modify the format and content of this template as needed.

Describe your physics methodology (e.g., statistical mechanics, renormalization, control theory) and how it addresses specific AI safety challenges. Summarize your theoretical insights, empirical results, or novel frameworks that bridge physics and AI alignment research.

Keywords: Physics-informed AI safety, [your specific physics approach], [AI safety problem area], [your methodology]

1 Introduction

A central challenge in AI safety is understanding the internal dynamics and developmental transitions of neural networks during training. I target Problem Area 2 — Learning, Inductive Biases, and Generalization — by investigating how sensitive a model is to weight perturbations at various training stages. I focus particularly on the robustness of neural network parameters to perturbations, motivated by a core insight from Singular Learning Theory (SLT): neural networks converge to local minima in the loss landscape, some of which are significantly flatter and thus I assume more robust to perturbations than others.

Singular Learning Theory provides a rigorous theoretical framework showing that not all minima are equally stable or generalizable; flatter minima typically correspond to simpler, more robust internal representations. This insight prompts a critical research question: *Can we quantitatively measure how robust a neural network is to perturbations by characterizing the geometric complexity of its loss minima?*

I address this question through the physics-inspired complexity measure known as the Local Learning Coefficient (LLC). The LLC quantifies the geometric complexity of a neural network’s loss landscape around a local minimum, capturing how ”flat” or ”sharp” the minimum is, independently of parameter count. Models located in flatter minima exhibit smaller changes in loss when parameters are perturbed, indicating greater robustness.

By estimating the Local Learning Coefficient (LLC) across a series of training checkpoints from the Tetrahedron 3M transformer model released in the Loss Landscape Degeneracy study [0], we retrospectively analyze how the degeneracy of the loss landscape evolves and how it relates

*Research conducted at the AI Safety x Physics Grand Challenge, 2025

to robustness under perturbation. Although we evaluate model robustness on a disjoint test set not used during training or model selection, this approach reveals how certain stages of training lead to flatter, more robust minima. Crucially, this insight opens the door to steering model behavior by applying perturbations along degenerate directions—directions that do not increase loss, but can still alter internal computation. In this way, the LLC not only enhances our theoretical understanding of learning dynamics, but also contributes to practical AI safety tools by identifying opportunities for safe model intervention without retraining or loss degradation.

Our codebase and detailed methodology are publicly available to encourage further exploration and validation of these findings: <https://github.com/coolmannine/part>.

The LLC provides a quantitative measure of the local degeneracy of a neural network’s loss landscape near a parameter configuration. Intuitively, it captures how sharply the loss increases as one moves away from a local minimum: lower LLC values indicate flatter minima, meaning small perturbations to model weights should cause minimal increases in loss, whereas higher LLC values suggest sharper, less robust minima.

2 Methods

Our analysis applies the Local Learning Coefficient (LLC) from Singular Learning Theory (SLT) to track how the geometric degeneracy of the loss landscape evolves during training and how it relates to robustness under weight perturbations [0].

2.1 Local Learning Coefficient (LLC)

Let $L(w)$ denote the population loss of a trained network, and let w^* be a local minimum. Define the ε -sublevel set around w^* by

$$S(\varepsilon) = \{ w : L(w) \leq L(w^*) + \varepsilon \}.$$

As $\varepsilon \rightarrow 0$, SLT shows that its volume scales like

$$\text{Vol } S(\varepsilon) \propto \varepsilon^{\lambda(w^*)} (-\log \varepsilon)^{m-1},$$

where $\lambda(w^*)$ is the LLC and m the local multiplicity. In practice we focus on settings with $m = 1$, so

$$\text{Vol } S(\varepsilon) \propto \varepsilon^{\lambda(w^*)}.$$

A smaller λ means a flatter (more degenerate) basin and—per the theory—greater robustness to small weight perturbations; a larger λ indicates a sharper minimum.

2.2 Estimating the LLC with SGLD

Directly computing $\text{Vol } S(\varepsilon)$ is infeasible in high dimensions. Instead, following Hoogland *et al.*, we estimate $\lambda(w^*)$ via the local free-energy expansion. For a dataset of size n , one defines the localized posterior

$$p(w) \propto \exp \left\{ -n L_n(w) - \frac{\gamma}{2} \|w - w^*\|^2 \right\},$$

where L_n is the empirical (training) loss, and γ a Gaussian-prior strength. SLT predicts the free energy

$$F(\gamma, n) = -\log \int p(w) dw = n L_n(w^*) + \lambda(w^*) \log n + O(\log \log n).$$

Rearranging gives the estimator

$$\hat{\lambda}(w^*) = n \beta [\mathbb{E}_p L_n(w) - L_n(w^*)], \quad \beta = \frac{1}{\log n}.$$

I sample $p(w)$ via minibatch SGLD (as in [0]) and compute the expectation by averaging over the chain.

2.3 Experimental Design and Weight Perturbation

I apply the SGLD-based LLC estimator to checkpoints of the 3M-parameter Tetrahedron model, recorded every 200 steps up to step 11,800. At each checkpoint w^* , I:

1. Load the model weights from the checkpoint,
2. Evaluate the baseline loss on a held-out validation set,
3. Run SGLD (20 chains $\times$ 200 draws, no burn-in, one step between draws, $\gamma = 200$, $\beta = 1/\log n$),
4. Estimate the Local Learning Coefficient $\lambda(w^*)$ using the minibatch expectation of $L_n(w)$,
5. Generate a single random noise tensor $\xi \sim \mathcal{N}(0, I)$, and apply it to the weights:

$$w_{\text{perturbed}} = w^* + \epsilon \cdot \xi,$$

6. Evaluate the loss of the perturbed model on the same validation set,
7. Compute the robustness metric as the change in validation loss due to perturbation.

The same random noise tensor ξ is generated once and reused across all checkpoints to ensure fair comparisons of robustness over training. The scale ϵ is chosen to keep the perturbation within the local neighborhood defined by the LLC. Flatter minima (lower λ) are expected to show smaller increases in loss under such perturbations, whereas sharper minima yield larger changes.

All SGLD hyperparameters follow the recommendations of Hoogland *et al.* and Wang *et al.* [0, ?].

3 Results

Our empirical investigation explores how the Local Learning Coefficient (LLC), as estimated through Singular Learning Theory, correlates with sensitivity to weight perturbations across the training trajectory of the Tetrahedron 3M transformer.

Empirical Findings

For each saved checkpoint, I estimated the LLC via SGLD sampling and then applied a small isotropic Gaussian perturbation to the model weights. I measured the change in validation loss before and after perturbation. As shown in Figure 1, I observe a weak positive correlation between the LLC and the change in loss. That is, checkpoints with higher LLC values tend to exhibit a greater increase in loss under perturbation.

This result supports the theoretical prediction that models converged to flatter minima are more robust to small perturbations, whereas those in sharper basins show greater sensitivity.

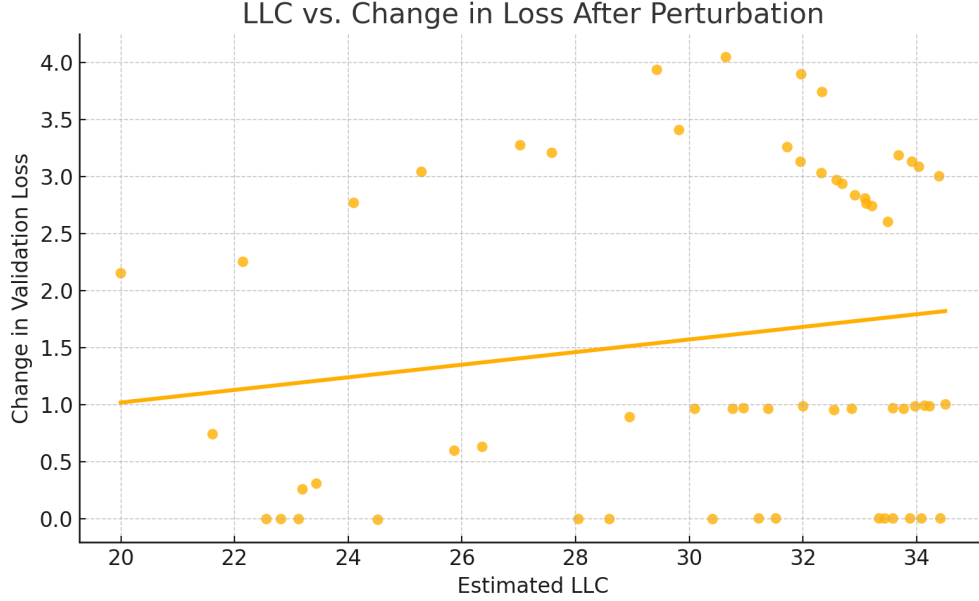


Figure 1: Estimated Local Learning Coefficient (LLC) plotted against the change in validation loss after a small Gaussian weight perturbation.

4 Discussion and Conclusion

Our results captures some meaningful information about the geometry of the loss landscape and the robustness of neural networks under perturbation. This could reinforce the view that certain theoretical tools from physics in this case, complexity measures derived from SLT can provide practical insight into neural network behavior, particularly in understanding phase transitions, specialization, and inductive bias formation during training.

Limitations: The scope of our current analysis is narrow. We evaluated only a single model architecture (Tetrahedron 3M) and used relatively small-scale SGLD sampling to estimate the LLC. Additionally, the perturbations applied were isotropic and uninformed by the model’s structure. More precise geometric analysis—such as directly estimating the Hessian and identifying its null space—was not performed here but may reveal more fine-grained steering directions and degeneracy structure.

Future Directions: A promising next step would be to combine LLC estimation with Hessian-based approaches. By identifying low-LLC checkpoints and then performing explicit null-space vector steering, one could explore the extent to which model behavior can be altered without affecting loss. This line of research could enable robust, post-training interventions in model behavior that preserve performance while introducing desired inductive biases or safety guarantees. With further development, such tools might enable safe manipulation of neural networks at inference time or offer diagnostic signals of instability during training, contributing meaningfully to the broader AI alignment effort.

5 References

Hoogland, J., Wang, G., Farrugia-Roberts, M., Carroll, L., Wei, S., & Murfet, D. (2024). *Loss Landscape Degeneracy Drives Stageswise Development in Transformers*. arXiv. <https://arxiv.org/abs/2405.14001>.

org/abs/2402.02364

Wang, G., Hoogland, J., van Wingerden, S., Furman, Z., Murfet, D. (2024). *Differentiation and Specialization of Attention Heads via the Refined Local Learning Coefficient*. arXiv. <https://arxiv.org/abs/2410.02984>

Hoogland, J. (2025). *Open Problems in Mechanistic Interpretability*. arXiv. <https://arxiv.org/abs/2501.16496>

6 Appendix

A.1 Supplementary Material

- **Code Repository:** The full implementation of LLC estimation, SGLD sampling, and perturbation evaluation is available at <https://github.com/coolmannine/part>.
- **SGLD Sampling Details:** For each checkpoint, we ran 20 chains of minibatch SGLD with 200 draws per chain. No burn-in or thinning was applied. The learning rate was fixed at 10^{-4} , with $\gamma = 200$, and $\beta = 1/\log n$. Sampling was performed over the full model parameter vector using PyTorch and Hugging Face-compatible transformer code.
- **Perturbation Code:** At each checkpoint w^* , we applied an isotropic Gaussian perturbation:

$$w_{\text{perturbed}} = w^* + \epsilon \cdot \xi, \quad \xi \sim \mathcal{N}(0, I),$$

with the same sampled ξ reused across checkpoints. The loss difference was computed on a 50k-token held-out validation split disjoint from the training set.

- **Mathematical Derivation of LLC Estimator:** From the SLT-predicted form of the free energy:

$$F(\gamma, n) = nL_n(w^*) + \lambda(w^*) \log n + O(\log \log n),$$

and the definition of free energy as $F = -\log \int p(w)dw$, we rearranged to get the estimator:

$$\hat{\lambda}(w^*) = n\beta [\mathbb{E}_p[L_n(w)] - L_n(w^*)], \quad \beta = \frac{1}{\log n}.$$

A.2 Use of Language Models

Large language models (LLMs) were used throughout this project to assist with:

- **Writing Assistance:** Drafting the Introduction, Methods, Results, and Discussion sections.
- **Code Debugging and Refactoring:** Spot-checking bugs in PyTorch code related to SGLD sampling and loss evaluation, and validating experiment logic in Colab notebooks.

All generated content was reviewed and adapted by the author, and all code and analysis were executed and verified manually.