The Butterfly Effect: Neural Network Training Trajectories Are Highly Sensitive to Initial Conditions

Code Paper

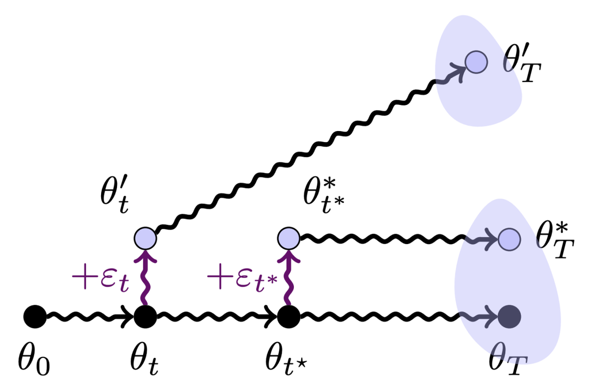
Gül Sena Altıntaş* Devin Kwok* Colin Raffel David Rolnick

University of Toronto McGill University Vector Institute Mila - Quebec AI Institute

🚀 Problem

- Neural network training is *unstable*: even when it succeeds in converging to *a* solution, it may not consistently reach the *same* solution.
- Prior work found that in the early (chaotic) phase of training, training (SGD) noise can cause the same network to diverge to disconnected minima [1, 2], as measured via barriers (Eq 1).
- Knowing whether training and fine-tuning is stable matters in practice: model averaging benefits from connected solutions, while ensembling benefits from diverse solutions.
- ? But how unstable is training, really? Is early-phase training stable to perturbations smaller than training noise, and is late-phase training unstable to perturbations larger than training noise?
- ? How does pre-training affect stability? Does stability depend on the amount of pre-training, and the specific combination of pre-training and fine-tuning tasks?
- ? Are some model architectures, task domains, or hyperparameters more stable than others?

🔧 Experiment



- Choose an initial network θ_0 (pre-trained or randomly initialized).
- Train the network until time t.
- Make two copies of the network θ_t, and perturb one by adding noise (ϵ) with magnitude σ to get $\theta'_t = \theta_t + \sigma\varepsilon$.
- Train both original (θ_t) and perturbed (θ'_t) copies with *identical training noise* to get θ_T and θ'_T.
- Measure similarity of θ_T and θ'_T via weight distance, barriers, barriers mod permutation, and representation similarity (CKA).
- Determine how similarity depends on the choice of θ_0, the perturbation time t, and the perturbation size σ.

Perturbations

Perturbation: $\epsilon = \frac{\hat{\epsilon}\cdot M}{\|\hat{\epsilon}\cdot M\|_2}\sqrt{\mathrm{Var}[\theta_0 \cdot M]}$

🎯 Batch Perturbation

An extra SGD step with newly-sampled data

$$\hat{\epsilon}_{\text{Batch}} = \frac{1}{n}\sum_{i=1}^{b} \nabla l(x_i, y_i; \theta_t)$$

→ Mimics natural training noise

🎲 Gaussian Perturbation

Random noise matching initialization scale

$$\hat{\epsilon}_{\text{Gaus}} = \left[\epsilon_i^{(l)}\right], \epsilon_i^{(l)} \sim \mathcal{N}\left(0, \frac{2}{n_{l-1}}\right)$$

→ Controlled random direction

$\sigma = 0.01$ means perturbation is 1% the size of the network's weights at initialization.

References

[1] S. Fort, G. K. Dziugaite, M. Paul, S. Kharaghani, D. M. Roy, and S. Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. *Advances in Neural Information Processing Systems*, 33:5850--5861, 2020.

[2] J. Frankle, G. K. Dziugaite, D. Roy, and M. Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 3259--3269. PMLR, 2020.

[3] A. H. Williams, E. Kunz, S. Kornblith, and S. Linderman. Generalized shape metrics on neural representations. In *Advances in Neural Information Processing Systems*, volume 34, pages 4738--4750, 2021.

🦋 **Tiny early perturbations** cause neural network training to **diverge**.

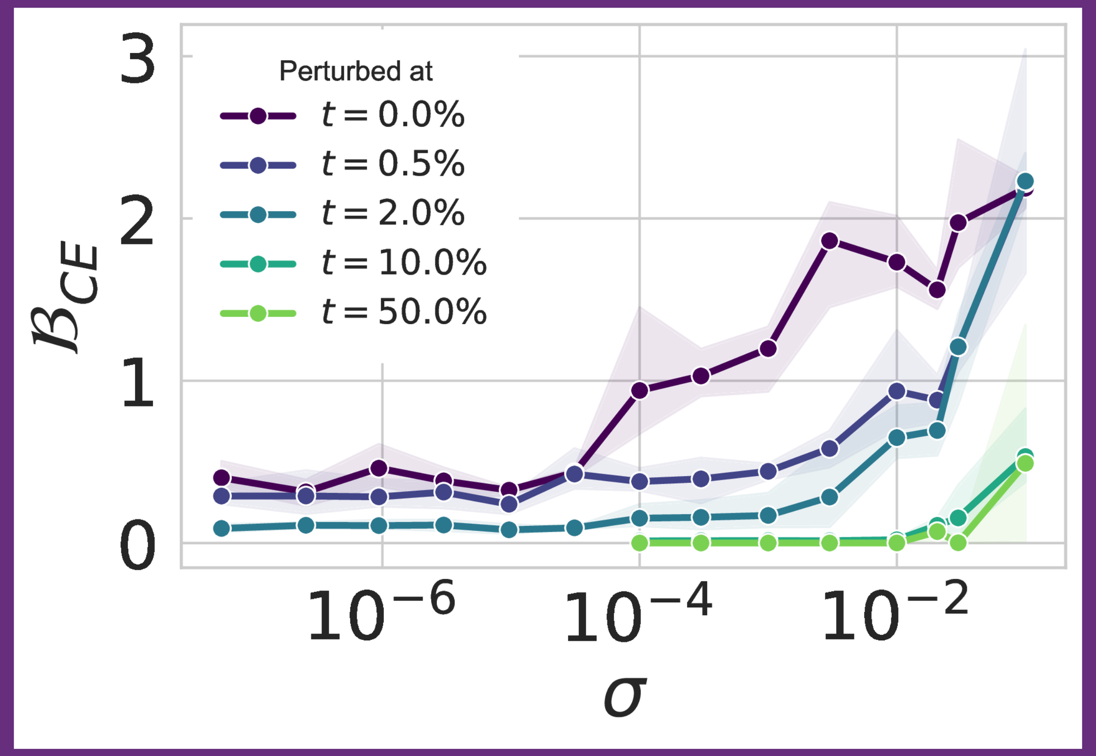


Figure 1. Barriers (cross-entropy loss on training data) at time T after training, versus perturbation magnitude, where $\sigma = 1$ is the network's scale at initialization, and colors indicate the perturbation time t.

- Perturbing as little as a *single weight* at initialization causes large barriers (left points).
- Small perturbations (0.01% of initialization) are sufficient for divergence in early training.
- Instability to small perturbations *drops rapidly* within the first 2% of training time (teal).
- Only very large perturbations (10% of initialization scale) cause networks to diverge after 50% of training time (rightmost points).
- *Direction independence:* networks are equally unstable to Gaussian and batch perturbations early in training, but more stable to Gaussian than batch perturbations late.

📏 Measuring Functional Dissimilarity

- L^2 **divergence**: distance between weights $\|\theta_T - \theta'_T\|_2$
- **Barriers**: maximum increase in loss/error along the linear path between the weights

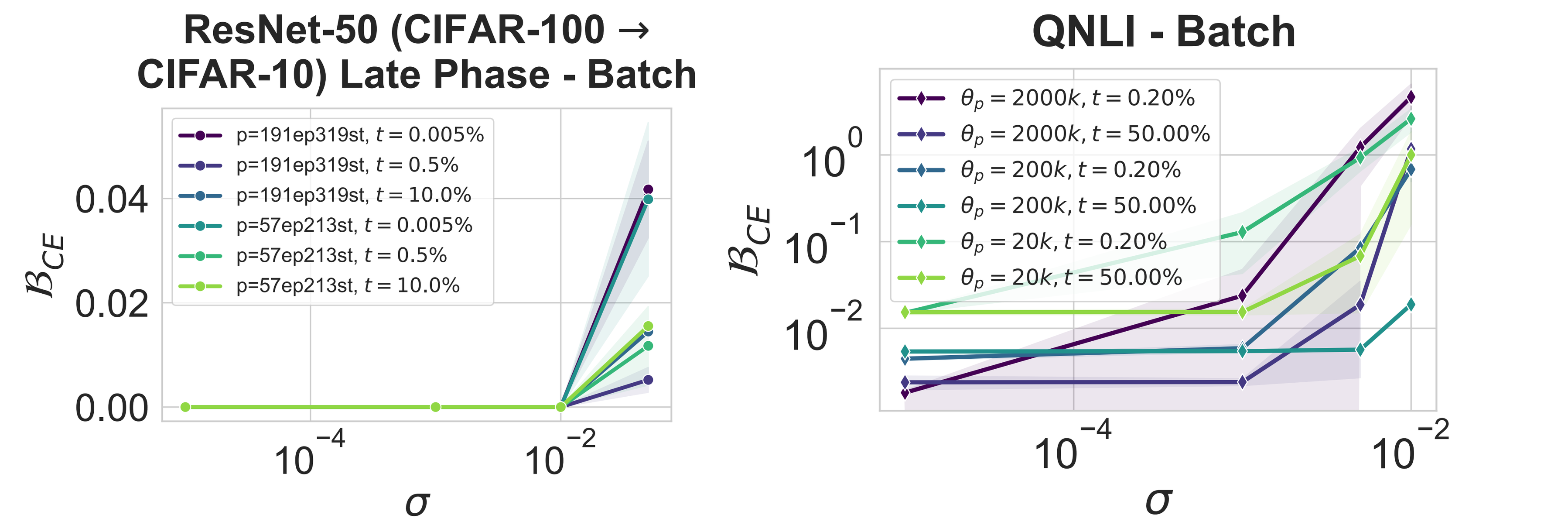
$$B(\theta_T, \theta'_T) \coloneqq \sup_{\alpha\in(0,1)} \ell\left(x, y; \alpha\theta_T + (1-\alpha)\theta'_T\right) - \alpha\ell\left(x, y; \theta_T\right) - (1-\alpha)\ell\left(x, y; \theta'_T\right). \quad (1)$$

- **Barriers mod permutation**: $B(\theta_T, P\theta'_T)$, where P is a permutation minimizing $\|\theta_T - P\theta'_T\|_2$.
- **Representation similarity**: measures cross correlation between the penultimate hidden outputs of two networks using Angular Centered Kernel Alignment (Angular CKA) [3]

$$d_{\text{CKA}}(\theta_T, \theta'_T)) = \text{CKA}\left[f_{L-1}(\theta_T), f_{L-1}(\theta'_T)\right], \qquad \text{CKA}(\mathbf{X}, \mathbf{Y}) = \arccos\left(\frac{\text{HSIC}(\mathbf{X}, \mathbf{Y})}{\text{HSIC}(\mathbf{X}, \mathbf{X})\,\text{HSIC}(\mathbf{Y}, \mathbf{Y})}\right), \quad (2)$$

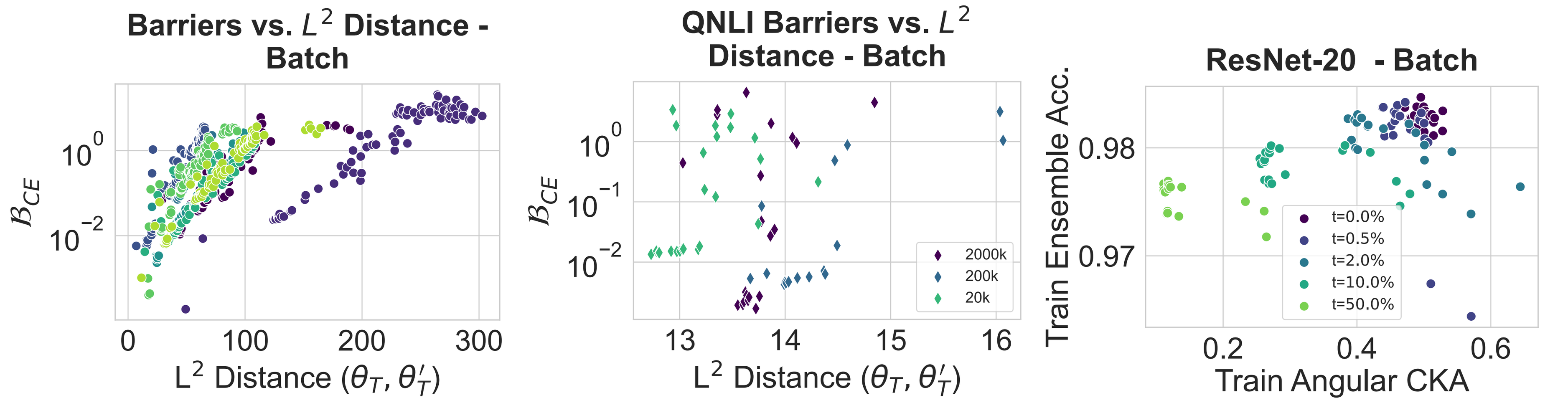
where HSIC is the Hilbert-Schmidt Independence Criterion.

🤔 The Pre-training Paradox



- **Vision—ResNet:** ResNet-50 models trained and fine-tuned from CIFAR-100 to CIFAR-10 (and vice versa) become *more* stable with more pre-training (left, Figure 5 in paper).
- **NLP—Transformer:** on some fine-tuning tasks, BERT & OLMo become *less* stable with more pre-training (right, Figure 5 and Appendices D.3-D.4 in the paper).
- **Vision Transformers:** for pre-trained ViT models, extra pre-training on ImageNet-1K *reduces* CIFAR-100 fine-tuning stability by an order of magnitude (Appendix D.2).
- **Hypothesis:** over-training on pre-training data causes "catastrophic overfitting".

🌀 Divergence & Representation Similarity



- Weight distance and functional dissimilarity are related in some settings but not others (Figure 7): barriers correlate exponentially with L^2 divergence in vision (left) but not NLP (middle).
- Counter to linearized dynamics, L^2 and barriers do not grow exponentially over training (Figure 6).
- Representation similarity correlates with barriers (Figure 3, Appendix C.4) and ensemble accuracy, indicating that instability can *increase* model diversity (right).

🎛️ Hyperparameters Matter

Warm-up, larger batch sizes, and wider networks *enhance* stability, while Adam and weight decay *degrade* it (Figure 4 and Appendix C.3 in the paper).

Even combining the best settings cannot eliminate instability at initialization!

📧 Correspondence to gsaltintas@cs.toronto.edu, devin.kwok@mail.mcgill.ca