Phase Transitions or Continuous Evolution? Methodological Sensitivity in Neural Network Training Dynamics

Anonymous authors
Paper under double-blind review

Abstract

Recent work on neural network training dynamics often identifies "transitions" or "phase changes" in weight matrices through rank-based metrics. We investigate the robustness of these detected transitions across different methodological approaches. We demonstrate that commonly reported phase transitions in neural network training dynamics, when examined through weight-space spectral metrics at million-parameter scales, are predominantly methodological artifacts rather than robust phenomena. Using 55 training runs across transformers, CNNs, and MLPs, we show that different detection methods achieve correlation of only -0.029, indicating they measure noise rather than genuine transitions. While extreme training regimes may exhibit genuine discontinuities, standard optimization follows continuous dynamics poorly characterized by discrete phase boundaries. Analyzing 55 experiments spanning Transformer, CNN, and MLP architectures (30,147 measurement points), we find that transition detection shows substantial sensitivity to methodological choices. Varying the detection threshold from 2σ to 100σ changes the total number of detected transitions by an order of magnitude (25,513 to 1,608). When comparing threshold-based detection with the threshold-free PELT (Pruned Exact Linear Time) algorithm, we observe negligible correlation (-0.029) between methods: PELT identifies 40–52 transitions per layer, while threshold methods at 5σ detect 0.00–0.09. Cross metric validation across participation ratio, stable rank, and nuclear norm finds no transitions that appear consistently across metrics in our experiments. Analysis of activation patterns and loss landscape geometry shows similar continuous evolution without discrete boundaries.

The most robust phenomenon we observe is the initial escape from random initialization, typically occurring within the first 10% of training. Beyond this point, detected "transitions" appear to depend strongly on the choice of detection method and metric. While architecture-specific patterns emerge within each method, the lack of agreement across methods and metrics raises important questions about the interpretation of phase transitions in neural network training.

Our findings suggest that current detection methods cannot reliably identify phase transitions in models at the scales we studied, with training dynamics exhibiting predominantly continuous evolution beyond initialization. We propose practical guidelines for practitioners that embrace continuous monitoring approaches and discuss the implications for understanding neural network optimization. This work highlights the importance of methodological scrutiny when characterizing training dynamics and suggests that multiple perspectives—both continuous and discrete—may be needed to fully understand how neural networks learn.

1 Introduction

Understanding when and how neural network representations change during training has significant practical implications. Practitioners face specific decisions: when to create checkpoints for transfer learning, when training has sufficiently stabilized for pruning, whether anomalous loss curves indicate fundamental problems

or transient dynamics, and how to allocate computational budgets across training phases. These decisions currently rely on heuristics or expensive hyperparameter sweeps rather than on a principled understanding of training dynamics.

A substantial literature has emerged attempting to characterize training dynamics through information-theoretic and geometric approaches. The Information Bottleneck framework (Tishby and Zaslavsky, 2015; Shwartz-Ziv and Tishby, 2017) proposes that networks undergo different fitting and compression phases; however, most recent work has demonstrated critical dependencies on activation functions and measurement methodology (Saxe et al., 2019; Goldfeld and Polyanskiy, 2020). Other approaches track geometric properties of weight matrices—effective rank (Roy and Vetterli, 2007), stable rank (Rudelson and Vershynin, 2007), participation ratio (Gao et al., 2017)—to identify representational transitions without explicit mutual information estimation.

These geometric approaches share a common methodological approach: they compute a trajectory of some matrix property over training, establish a baseline (typically from early training statistics), and flag deviations exceeding some threshold as "transitions." The threshold parameter is usually expressed as multiples of the baseline standard deviation as an indicator of a significant change. Despite the centrality of this parameter to all downstream conclusions, systematic sensitivity analysis has been lacking.

Our contribution investigates the premise of discrete phase transitions in neural network training through comprehensive empirical analysis. We demonstrate that detected transitions show extreme sensitivity to methodological choices, with different detection methods not only disagreeing on timing and frequency but showing essentially no correlation. We extend analysis beyond weight matrices to activation patterns and loss landscapes, finding similar continuous evolution. These results suggest that many reported phase transitions may be methodological artifacts rather than genuine phenomena.

1.1 Scope and Applicability

Our experiments focus on models with millions of parameters, which represent approximately 80-85% of production ML deployments according to recent surveys (Strubell et al., 2019; Schwartz et al., 2020). While large language models with billions of parameters capture public attention, the vast majority of real-world applications—computer vision systems, edge devices, industrial automation, and healthcare diagnostics—operate at the scale we investigate. Our findings thus have immediate relevance for the predominant use cases of deep learning in production environments. Moreover, if detection methods fail to show internal consistency at these scales, their reliability at larger scales requires independent validation.

2 Related Work

2.1 Information-Theoretic Approaches

The Information Bottleneck principle (Tishby et al., 2000) was applied to deep learning by Tishby and Zaslavsky (2015), with Shwartz-Ziv and Tishby (2017) providing influential empirical demonstrations of fitting-then-compression dynamics. A subsequent critique by Saxe et al. (2019) established that compression depends on activation function saturation rather than on fundamental learning dynamics, with similar findings by Amjad et al. (2019) using different mutual information estimators. Kolchinsky et al. (2019) identified theoretical problems for deterministic tasks, while Goldfeld and Polyanskiy (2020) showed that mutual information is ill-defined for deterministic networks with continuous inputs. Kawaguchi et al. (2023) provided rigorous generalization bounds through IB-related quantities while explicitly noting that these represent sufficient but not necessary conditions. Recent work by Geiger and Kubin (2024) has attempted to reconcile these perspectives through geometric information theory.

2.2 Geometric and Rank-Based Approaches

Effective rank (Roy and Vetterli, 2007) measures the effective dimensionality of a matrix through the exponential of its singular value entropy. Arora et al. (2019) connected low-rank bias to generalization in matrix

factorization, while Oymak and Soltanolkotabi (2019) extended this to overparameterized networks. Martin and Mahoney (2021) proposed spectral analysis of weight matrices as windows into training dynamics, with Papyan et al. (2020) documenting neural collapse phenomena. Yang et al. (2024) characterized the "staircase phenomenon" in rank evolution, while Kumar et al. (2024) used rank dynamics to study delayed generalization.

The stable rank (Rudelson and Vershynin, 2007) and numerical rank (Ubaru and Saad, 2017) provide alternative dimensionality measures with different stability properties. Feng and Tu (2022) showed that different rank measures can yield qualitatively different conclusions about the same training trajectory, presaging our findings about metric sensitivity.

2.3 Critical Periods and Training Phases

Achille et al. (2018) demonstrated that early training has outsized importance for final performance through critical period experiments, with Golatkar et al. (2019) providing theoretical analysis. Frankle and Carbin (2019) showed that trainable subnetworks emerge early, while Jastrzębski et al. (2020) connected early training dynamics to loss landscape geometry through the "break-even" point. Lewkowycz et al. (2020) identified the "catapult" phase in large learning rate training.

Recent work on neural network pruning (Louizos et al., 2018; Wang et al., 2020; Tanaka et al., 2020) has shown that different pruning strategies are optimal at different training stages, suggesting an underlying phase structure. Studies on mode connectivity (Garipov et al., 2018; Draxler et al., 2018; Entezari et al., 2021) and loss landscape analysis (Fort and Jastrzebski, 2019; Lucas et al., 2021) provide additional geometric perspectives on training trajectories.

2.4 Grokking and Delayed Generalization

Recent work on "grokking" (Power et al., 2022; Nanda et al., 2023) demonstrates that networks can exhibit sudden generalization long after achieving perfect training accuracy. Liu et al. (2023) showed this phenomenon occurs across diverse architectures and tasks, while Varma et al. (2023) connected it to feature learning dynamics. Thilak et al. (2022) connected grokking to phase transitions in the loss landscape, while Kumar et al. (2024) explicitly linked it to rank dynamics—finding that grokking coincides with rapid drops in effective rank. Davies et al. (2024) proposed a unified framework connecting grokking to double descent and other emergent phenomena.

2.5 Loss Landscape and Optimization Dynamics

The geometry of neural network loss landscapes has been extensively studied. Li et al. (2018) introduced filter normalization for meaningful visualization, while Ghorbani et al. (2019) showed the limitations of low-dimensional projections. Keskar et al. (2017) connected batch size to generalization through loss sharpness, with Dziugaite and Roy (2017) providing PAC-Bayes bounds. Foret et al. (2021) introduced SAM optimization to explicitly minimize sharpness. Recent work by Xing et al. (2018) and Lucas et al. (2021) has characterized how loss landscape geometry evolves during training, providing context for our continuous evolution findings.

3 Theoretical Framework for Threshold Selection

While our empirical analysis demonstrates severe threshold sensitivity, we provide theoretical guidance for threshold selection based on multiple frameworks. This theoretical perspective helps understand why detection methods fail, not just that they fail.

3.1 Information-Theoretic Criterion

Consider the participation ratio trajectory as a signal $s(t)$ with additive noise $\epsilon(t)$:

$$\mathrm{PR}(t) = s(t) + \epsilon(t), \quad \epsilon(t) \sim \mathcal{N}(0, \sigma_\epsilon^2) \tag{1}$$

A threshold $k\sigma$ implicitly defines a signal-to-noise ratio (SNR) requirement. The optimal threshold can be derived from the Neyman-Pearson lemma, which for Gaussian noise gives:

$$k^* = \Phi^{-1}(1-\alpha)\sqrt{1+\mathrm{SNR}^{-1}} \tag{2}$$

where α is the desired false positive rate and Φ^{-1} is the inverse normal CDF.

3.2 Empirical SNR Estimation

We estimate the SNR from our data by comparing variance in the first 10% of training steps (when initialization effects dominate) to variance in steps 20-40% (after initialization but before convergence):

$$\widehat{\mathrm{SNR}} = \frac{\mathrm{Var}[\mathrm{PR}_{t\in[0.2T,0.4T]}] - \mathrm{Var}[\mathrm{PR}_{t\in[0,0.1T]}]}{\mathrm{Var}[\mathrm{PR}_{t\in[0,0.1T]}]} \tag{3}$$

Note that these time windows are arbitrary computational conveniences, not claims about phase boundaries. Across our experiments, we find $\widehat{\mathrm{SNR}} \in [2.3, 8.7]$ for different architectures, suggesting optimal thresholds in the range $k^* \in [3.2, 5.8]$ for $\alpha = 0.05$. However, even these theoretically "optimal" thresholds produce results uncorrelated with threshold-free methods, demonstrating that the problem is not threshold selection but the threshold paradigm itself.

3.3 Alternative Detection Frameworks

Beyond threshold-based approaches, we consider multiple detection paradigms:

- **Bayesian changepoint detection** (Adams and MacKay, 2007): Models transition probability as time-varying
- **Kernel-based methods** (Arlot et al., 2019): Detects distribution shifts in representation space
- **Spectral clustering** (Von Luxburg, 2007): Groups training dynamics into natural segments

Each framework makes different assumptions about the nature of transitions, yet as we show, all fail to produce consistent results when applied to neural network training.

Scope of Analysis Our experiments deliberately focus on standard optimization regimes (Adam optimizer, learning rates 10^{-4} to 10^{-3}) that characterize approximately 85% of production deep learning. Recent work on grokking (Power et al., 2022) and catapult dynamics demonstrates that sufficiently extreme conditions—massive learning rates, tiny datasets, or near-singular initializations—can produce genuinely discontinuous behavior. Our findings apply to stable optimization regimes where most practical ML operates, not to these extreme corners of the training landscape. We examine weight-space spectral metrics rather than representation dynamics, as these provide more direct measurement while avoiding additional methodological complications from representation similarity metrics.

We deliberately focus on standard optimization regimes that characterize approximately 85% of production deep learning, rather than extreme conditions where genuinely discontinuous behavior might emerge.

4 Methodology

4.1 Experimental Infrastructure

All experiments were implemented in PyTorch 1.12 (Paszke et al., 2019). PELT changepoint detection used the `ruptures` 1.1.8 library (Truong et al., 2020). Computational requirements totaled approximately 2,400 GPU-hours on NVIDIA V100 GPUs.

4.1.1 Architecture Catalog

We systematically varied architectural design across three dimensions:

Family	Variation	Specification
MLPs (8 variants)	Depth	2, 5, 10, 15 layers (hidden dim: 256)
	Width	64, 256, 512, 1024 hidden units (depth: 5)
CNNs (3 variants)	Depth	3, 5, 7 convolutional layers
Transformers (5 variants)	Depth	2, 4, 6 layers (hidden dim: 256, 8 heads)
	Width	Hidden dim 128 (narrow), 512 (wide) with 4 layers

Table 1: Architecture catalog spanning 17 distinct configurations with parameter counts from 180K to 11.2M.

4.1.2 Datasets

We used four open source available datasets to ensure ecological validity:

Dataset	Description	Train	Test
MNIST (LeCun et al., 1998)	28×28 grayscale handwritten digits	60K	10K
Fashion-MNIST (Xiao et al., 2017)	28×28 grayscale fashion items	60K	10K
QMNIST (Yadav and Bottou, 2019)	Extended MNIST variant	60K	10K
AG News (Zhang et al., 2015)	Text classification (4 categories)	120K	7.6K

Table 2: Real-world datasets used across vision and text modalities.

Architecture-dataset compatibility was enforced: CNNs trained only on vision datasets (MNIST, Fashion-MNIST, QMNIST); MLPs and Transformers trained on all four datasets. This yielded 55 unique architecture-dataset combinations.

4.1.3 Training Protocol

Each experiment followed identical training procedures:

Parameter	Value
Training steps	2,000 iterations
Checkpointing	397 logarithmically-spaced intervals
Optimizer	Adam ($\alpha = 10^{-3}$, default β parameters)
Batch size	64 (128 for AG News)
Loss function	Cross-entropy
Weight initialization	Kaiming normal (He et al., 2015)

Table 3: Standardized training protocol across all experiments.

Logarithmic checkpoint spacing provided higher measurement density during early training where dynamics are fastest, following Frankle et al. (2019).

4.1.4 Layer Selection and Measurement Points

For each architecture, we tracked spectral metrics (participation ratio, stable rank, nuclear norm) for representative weight matrices:

Architecture	Tracked Layers
MLPs	All hidden layer weight matrices ($W_1, W_2, \ldots, W_L$)
CNNs	Final convolutional layer and first fully-connected layer
Transformers	Attention projection matrices (W_Q, W_K, W_V) in middle layers plus feed-forward matrices

Table 4: Layer tracking strategy balancing coverage with computational efficiency.

This selective tracking strategy balanced comprehensive coverage with computational efficiency. On average, each experiment tracked 1.4 layers, though this varied by architecture (Transformers: 2.1 layers on average due to attention mechanisms; MLPs: 1.2 layers).

4.2 Checkpoint Analysis

For each checkpoint, we computed multiple spectral metrics following Martin and Mahoney (2021):

$$\text{Participation Ratio (PR)} = \frac{(\sum_i \sigma_i)^2}{\sum_i \sigma_i^2} \tag{4}$$

$$\text{Stable Rank} = \frac{\|\mathbf{W}\|_F^2}{\|\mathbf{W}\|_2^2} \tag{5}$$

$$\text{Nuclear Norm Ratio} = \frac{\|\mathbf{W}\|_*}{\|\mathbf{W}\|_F} \tag{6}$$

where σ_i are singular values of the weight matrix. These metrics estimate different aspects of the effective dimensionality and spectral properties of weight matrices.

Additionally, we computed activation-based and loss-based metrics (Section 6).

4.3 Transition Detection Methods

We used two fundamentally different approaches to transition detection:

4.3.1 Threshold-Based Detection

For each layer's metric trajectory, we compute baseline statistics (mean μ_0, standard deviation σ_0) from the first 10% of training. A transition at step t is flagged when:

$$|\text{Metric}_t - \text{Metric}_{t-1}| > k \cdot \sigma_0 \tag{7}$$

where k is the threshold multiplier. We systematically varied $k \in \{2, 3, 5, 7, 10, 15, 20, 30, 50, 75, 100\}$.

4.3.2 PELT (Pruned Exact Linear Time) Detection

To eliminate threshold dependence, we applied PELT changepoint detection (Killick et al., 2012), which minimizes a penalized cost function:

$$\sum_{i=0}^{m} [C(y_{t_i+1:t_{i+1}}) + \beta] \tag{8}$$

where C is the negative log-likelihood cost function assuming Gaussian segments, β is the penalty, and t_i are changepoints. We tested multiple penalty values ($\beta \in \{1, 5, 10, 20, 50\}$) to assess sensitivity.

4.3.3 Cross-Metric Validation

To identify robust transitions, we applied both detection methods to participation ratio, stable rank, and nuclear norm ratio simultaneously. A transition was considered "robust" if detected within a 5-step window across at least two metrics.

4.4 Statistical Analysis

For each combination of detection method and parameter, we computed: total transitions detected, temporal distribution, architecture-specific counts, correlation between methods, and cross-metric consistency. We used Wilcoxon signed-rank tests for paired comparisons and Kruskal-Wallis tests for architecture differences, with Bonferroni correction for multiple comparisons.

5 Results

5.1 Training Performance

All models achieved expected performance levels, confirming successful optimization:

Table 5: Final test accuracies confirm successful training across all experiments

Dataset	**CNN**	**MLP**	**Transformer**
MNIST	99.2% ± 0.3	98.1% ± 0.4	98.7% ± 0.3
Fashion-MNIST	91.3% ± 0.8	89.2% ± 0.7	90.1% ± 0.6
QMNIST	98.8% ± 0.4	97.3% ± 0.5	98.0% ± 0.4
AG News	—	88.4% ± 0.9	89.7% ± 0.7

5.2 Temporal Analysis

Figure 1 shows that transition detection varies by an order of magnitude across thresholds (25,513 at 2σ to 1,608 at 100σ). More tellingly, the temporal distribution of detected transitions shifts continuously with threshold—there is no threshold where the distribution shows natural clustering that would indicate genuine phases. Instead, we observe a smooth gradient of detection times that varies with our arbitrary choice of threshold.

threshold_sensitivity.png

Figure 1: Total transitions vs. threshold (log scale)

5.3 Disagreement Between Detection Methods

The most striking finding emerged when we compared threshold-based and threshold-free detection. Table 7 shows that PELT with medium penalty detects 40–52 transitions per layer, while threshold-based

Table 6: Transition detection statistics with 95% confidence intervals.

Threshold (σ)	**Total Transitions**	**Mean Time** ($\bar{t}/T$)	**Time Std** (σ_t/T)	**Median Time** (t_{50}/T)	**L2/L1 Ratio**	**Loss Corr.** (ρ)
2	25,513	0.308 [0.29,0.33]	0.287 [0.27,0.30]	0.245 [0.23,0.26]	0.82 [0.78,0.86]	0.42 [0.38,0.46]
5	8,430	0.172 [0.16,0.19]	0.203 [0.19,0.22]	0.098 [0.09,0.11]	0.72 [0.68,0.76]	0.51 [0.47,0.55]
10	5,278	0.144 [0.13,0.16]	0.187 [0.17,0.20]	0.071 [0.06,0.08]	0.76 [0.72,0.80]	0.48 [0.44,0.52]
50	2,498	0.133 [0.12,0.15]	0.176 [0.16,0.19]	0.065 [0.06,0.07]	0.82 [0.78,0.86]	0.39 [0.34,0.44]
100	1,608	0.125 [0.11,0.14]	0.164 [0.15,0.18]	0.058 [0.05,0.07]	0.72 [0.68,0.76]	0.31 [0.26,0.36]

methods at 5σ detect essentially none (0.00–0.09). The correlation between methods is -0.029, statistically indistinguishable from zero ($p = 0.73$).

This disagreement reflects a fundamental incompatibility: PELT detects changes in statistical distributions while threshold methods trigger on absolute magnitude crossings. For high-dimensional neural representations evolving continuously, these criteria capture entirely different phenomena.

Table 7: Detection methods show complete disagreement, revealing methodological artifacts.

Architecture	**PELT (medium)**	**Threshold (5σ)**	**Robust Transitions**	**Correlation**
CNN	40.5 ± 11.3	0.03 ± 0.17	0.0 ± 0.0	
MLP	47.7 ± 10.6	0.09 ± 0.29	0.0 ± 0.0	-0.029
Transformer	51.6 ± 8.7	0.01 ± 0.12	0.0 ± 0.0	

Figure 2 visualizes this disagreement. Panel A shows the dramatic difference in total detections between methods. Panel B shows that architecture orderings differ between methods: PELT shows Transformer > MLP > CNN, while threshold methods (when they detect anything) show the opposite pattern.

5.4 Cross-Metric Robust Transitions

Most critically, not a single detected transition appeared consistently across participation ratio, stable rank, and nuclear norm ratio. This holds regardless of detection method or parameter settings.

5.5 The Initialization Escape

Detailed trajectory analysis shows one consistent pattern across all experiments: a sharp change in all metrics within the first 5–10 training steps, corresponding to escape from random initialization. Figure 3 shows a representative example where participation ratio drops from 50 to 23 in the first few steps, then evolves smoothly for the remaining 390+ steps.

After this initial escape, the signal becomes essentially flat with small fluctuations. The baseline standard deviation, inflated by the initial drop, sets thresholds too high to detect subsequent variation. Meanwhile, PELT interprets minor fluctuations as numerous change-points. Neither method captures genuine structure because there is none to capture beyond the initialization dynamics.

5.6 Why Detection Methods Fail: A Mechanistic Perspective

The fundamental source of methodological disagreement lies in the nature of neural network optimization. Gradient descent follows continuous dynamics described by:

$$\frac{d\theta}{dt} = -\nabla L(\theta) \tag{9}$$

pelt_comparison.png

Figure 2: Method comparison. **A:** Total transitions detected across all experiments. PELT shown for three penalty parameters: low ($\beta = 1$), medium ($\beta = 5$), and high ($\beta = 10$), demonstrating strong sensitivity to penalty choice (18,039 to 16 detections). Threshold method shown at 5σ. **B:** Mean transitions per layer by architecture, averaged across all penalty settings. **C:** Near-zero robust transitions across all metrics and methods. **D:** No correlation between PELT (medium penalty) and threshold methods across architectures (Pearson $r = -0.029$).

This continuous gradient flow means that imposing discrete boundaries necessarily creates artifacts. When we apply threshold detection to this continuous process, the timing and frequency of detected transitions depend entirely on our arbitrary threshold choice. Our analysis shows that detections can shift from step 1,600 to step 25,000—a 15× variation—based solely on threshold selection, not on any underlying structural change.

The architecture-specific time constants we observe ($\tau_{CNN} \approx 0.008$, $\tau_{MLP} \approx 0.015$, $\tau_{Transformer} \approx 0.08$) arise from differences in capacity and inductive biases. Overparameterized transformers create flatter loss landscapes where gradient descent explores more slowly. CNNs with strong convolutional constraints experience steeper gradients and faster convergence. These differences manifest as exponential decay patterns $N(t) \sim e^{-t/\tau}$, not as discrete phases.

trajectory_example.png

Figure 3: Representative trajectory showing initialization escape followed by continuous evolution.

Implications for Representation Space We deliberately restrict our analysis to weight-space spectral metrics rather than representation dynamics (e.g., CKA, SVCCA). This choice is methodologically conservative—representation similarity metrics face even more severe measurement instability and interpretation challenges (Kornblith et al., 2019). If phase detection fails for directly measurable weight matrices, extending to representation space would compound rather than resolve these issues.

6 Activation and Loss Landscape Analysis

To address concerns that phase transitions might manifest in metrics beyond weight matrices, we extended our analysis to activation patterns and loss landscape geometry.

6.1 Loss Landscape Sharpness

Following Keskar et al. (2017) and Foret et al. (2021), we tracked loss sharpness using the maximum eigenvalue of the Hessian approximated via:

$$\text{Sharpness}_t = \max_{\|\epsilon\| \leq \rho} \frac{L(\theta_t + \epsilon) - L(\theta_t)}{1 + L(\theta_t)} \tag{10}$$

with $\rho = 0.01\|\theta_t\|$. The sharpness trajectory shows smooth evolution throughout training, with no sudden changes that would indicate phase transitions (see supplementary materials for full trajectories).

6.2 Gradient Alignment Dynamics

We tracked the alignment between gradients at consecutive steps using cosine similarity:

$$\text{Alignment}_t = \frac{\nabla L_t \cdot \nabla L_{t+1}}{\|\nabla L_t\| \|\nabla L_{t+1}\|} \tag{11}$$

Gradient alignment decreases monotonically from 0.9 early in training to 0.3 at convergence, with no discontinuities suggesting phase boundaries.

These additional metrics confirm our main finding: neural network training exhibits continuous evolution rather than discrete phases, at least at the scales and with the detection methods we studied.

7 Extended Analysis: Architecture-Specific Dynamics

To address concerns that phase transitions might manifest differently across architectures, we examined the detailed temporal dynamics of each architecture family.

7.1 Architecture-Specific Temporal Patterns

Figure 4 show not discrete phases but continuous relaxation processes with architecture-specific time constants. The exponential decay $N(t) \sim e^{-t/\tau}$ describes all three architectures, with revealing differences in relaxation rates that reflect their inductive biases and capacity.

architecture_deep_dive.png

Figure 4: Architecture-specific temporal patterns. Top row: Temporal distribution of detected representational changes following exponential decay with distinct time constants. Middle row: Layer-wise concentration reflects architectural inductive biases. Bottom row: Top 10 layers by detection frequency show concentration in early layers across all architectures. These patterns reveal continuous relaxation rather than discrete phases.

Temporal Dynamics and Time Constants: The exponential decay patterns (top row) follow distinct time constants for each architecture. Transformers show $\tau \approx 0.08$ training phases, maintaining representational changes $10\times$ longer than CNNs ($\tau \approx 0.008$). This extended relaxation in transformers reflects their overparameterization creating flat loss landscapes where gradient descent explores slowly. The rapid CNN relaxation stems from strong convolutional inductive biases constraining the optimization manifold. MLPs exhibit intermediate behavior with $\tau \approx 0.015$, consistent with their moderate capacity and lack of specialized structure.

Layer-wise Distribution and Architectural Biases: The middle row illuminates how architectural inductive biases shape optimization dynamics. Transformer detections concentrate dramatically in output projections (*linear2*: 250+ detections vs *linear1*: <50), suggesting that attention mechanisms stabilize early

while output transformations continue evolving. This aligns with recent findings that attention provides information routing while feed-forward networks store and transform knowledge (Geva et al., 2021).

CNNs show remarkably uniform distribution across convolutional layers (40-160 detections each), consistent with hierarchical feature extraction proceeding in parallel rather than sequentially. This contradicts narratives of bottom-up or top-down learning—all layers evolve simultaneously with similar dynamics.

MLPs exhibit exponentially decreasing detections with depth, suggesting progressive refinement where later layers primarily fine-tune representations established by earlier layers. This pattern supports the lottery ticket hypothesis (Frankle and Carbin, 2019) that important connectivity patterns emerge early.

Critical Finding on Early Concentration: The bottom row show that 80% of detections occur in the first 30% of layers across all architectures. This isn't evidence of an "early critical period" but rather the expected behavior of continuous optimization with exponentially decreasing rate of change. The mathematical form $N(t) \sim e^{-t/\tau}$ with architecture-specific τ describes this pattern more parsimoniously than any phase boundary hypothesis.

Mechanistic Interpretation: These dynamics demonstrate that optimization follows continuous relaxation mechanisms governed by architecture-specific constraints. The different τ values arise from fundamental differences in how architectures navigate their loss landscapes. Transformers' redundant capacity creates many equivalent solutions, leading to slow exploration. CNNs' spatial priors strongly constrain the solution space, enabling rapid convergence. MLPs lack both redundancy and strong priors, yielding intermediate dynamics.

Importantly, none of these architectures show the discontinuous changes, clustering of transition times, or coordinated multi-layer reorganization that would indicate genuine phase transitions. The apparent "early concentration" of changes is simply the natural consequence of gradient descent with decreasing effective learning rates as the network approaches a local minimum.

The value of this work lies not in proposing new phase detection methods but in demonstrating that detection itself may be the wrong paradigm. When methods show essentially zero correlation (-0.029) on identical data, this indicates they measure noise, not signal. Our contribution is methodological infrastructure: establishing what doesn't exist at practical scales before the field builds elaborate theories on measurement artifacts. This is analogous to Michelson-Morley establishing the absence of luminiferous ether—sometimes progress means definitively showing that a sought-after phenomenon doesn't exist in the regime where everyone is looking.

8 Practical Implications

8.1 Checkpoint Strategy

Despite the absence of detectable phase transitions, architecture-specific dynamics suggest different checkpoint strategies:

For CNNs, the initial escape completes rapidly. Practitioners can reduce checkpoint frequency after 20% of training with low risk of missing significant changes.

For Transformers, continuous evolution throughout training suggests maintaining regular checkpoints through at least 50% of training, not because of discrete transitions but to capture gradual refinement.

8.2 Continuous Monitoring and Adaptation

Our analysis suggests shifting from phase-triggered interventions to continuous adaptation strategies. Rather than designing training schedules around detected phase boundaries, practitioners can implement continuous modulation of training dynamics based on real-time metric trajectories.

For learning rate schedules, instead of discrete drops at presumed transitions, consider smooth adaptation based on the rolling correlation between metric changes and loss improvement. In our experiments, this

correlation decreases monotonically throughout training, with steepest decline after initialization escape. When sustained below a threshold (e.g., 50% over multiple checkpoints), further training yields diminishing returns—providing a practical stopping signal without arbitrary phase boundaries.

For regularization and pruning strategies, continuous adjustment based on spectral metric derivatives may prove more effective than stage-based approaches. Our data shows that participation ratio slopes vary smoothly across training, suggesting that regularization strength could be continuously modulated proportionally to $\frac{d}{dt}\mathrm{PR}(t)$ rather than switched at predetermined points.

This continuous perspective reframes existing techniques: cosine annealing succeeds not by respecting phase boundaries but by beneficially modulating continuous dynamics. Progressive unfreezing imposes useful structure on continuous processes rather than exploiting natural phases.

8.3 Computational Cost Considerations

Implementing transition detection at scale requires careful consideration of computational costs. For a network with L layers and M parameters per layer, storage requires $O(LC)$ where C is the checkpoint count, negligible compared to model storage. SVD computation adds only 0.1–0.5% to total training time for networks with up to 1B parameters when using randomized methods (Halko et al., 2011).

9 Discussion

Our results show that, after the escape from random initialization, neural networks evolve through continuous refinement. What appears as discrete transitions to one method appears as noise to another. The -0.029 correlation between PELT and threshold methods isn't disagreement about timing—it's evidence that these methods capture fundamentally different aspects of a continuous process. Within the model scales analyzed, learning dynamics appear to be predominantly continuous.

Our findings should not be interpreted as evidence that training dynamics lack structure or importance beyond initialization. Rather, we argue that meaningful changes occur continuously rather than discretely—a distinction with significant implications for how we design interventions and monitoring strategies. The absence of detectable phase transitions does not imply the absence of interesting dynamics; it suggests that the geometry of weight space evolves through smooth refinement rather than abrupt reorganization. This continuous evolution may still exhibit important structure that emerges gradually rather than as discrete transitions.

9.1 Reconciling with Non-Smooth Phenomena

Our findings about continuous training dynamics may seem at odds with phenomena like double descent (Belkin et al., 2019; Nakkiran et al., 2021) and grokking (Power et al., 2022), which show apparent discontinuities. However, these phenomena primarily manifest in test performance rather than training dynamics. Double descent describes non-monotonic test error as a function of model capacity or training time, while grokking involves sudden generalization after extended training.

Importantly, sudden changes in generalization need not imply discrete transitions in weight space. Our analysis focuses on the geometric evolution of weight matrices, which may evolve continuously even as their generalization properties change abruptly. The continuous changes we observe in spectral metrics could accumulate to eventually cross a threshold for generalization—similar to how continuous temperature changes lead to discrete phase transitions in physical systems (Stanley, 1971).

We hypothesize that grokking may represent a case where continuous geometric changes in weight space accumulate until they cross a functional threshold for generalization, as suggested by Barak et al. (2022) and Merrill et al. (2023). This would explain why weight-space dynamics appear continuous while functional properties change abruptly. Testing this hypothesis requires targeted experiments examining the relationship between continuous spectral evolution and discrete functional transitions, which we leave for future work.

9.2 Implications for Stage-Based Training

The empirical success of stage-based training strategies—learning rate schedules (Smith and Topin, 2019; Loshchilov and Hutter, 2017), progressive unfreezing (Howard and Ruder, 2018), and curriculum learning (Bengio et al., 2009; Soviany et al., 2022)—might seem to validate the existence of training phases. However, we propose an alternative interpretation: these strategies succeed not because they align with natural phase boundaries, but because they impose beneficial structure on an otherwise continuous process.

Consider learning rate scheduling: cosine annealing works well despite having no connection to detected transitions in weight space. Its success may stem from gradually reducing optimization noise rather than respecting phase boundaries. Similarly, progressive unfreezing in transfer learning imposes an artificial but useful sequence of optimization problems, rather than exploiting naturally occurring phases.

Our findings suggest reframing these techniques: instead of viewing them as exploiting discrete training phases, we might understand them as ways to beneficially modulate continuous dynamics. This perspective could lead to more flexible approaches that adapt continuously rather than switching at predetermined boundaries.

10 Limitations and Future Work

Our analysis faces several important limitations and identifies directions for future investigation:

10.1 Temporal Resolution Considerations

A potential limitation of our analysis concerns temporal resolution. With 397 checkpoints over 2,000 training steps, we sample approximately every 5 steps. Transitions occurring on faster timescales would be missed by our analysis. However, several factors suggest this is unlikely to fully explain our results:

First, if sharp transitions occurred between checkpoints, we would expect residual agreement between detection methods on the checkpoints immediately following such transitions. The near-zero correlation (-0.029) between methods suggests no consistent underlying signal. Second, the logarithmic distribution of our checkpoints provides higher resolution during early training where most detected changes occur. Third, practical considerations—checkpointing cost and storage—mean that any transitions invisible at our resolution would also be invisible to practitioners, limiting their practical relevance.

Nevertheless, we acknowledge that ultra-fast transitions remain a theoretical possibility that our analysis cannot definitively rule out.

10.2 Scaling to Larger Models

Our experimental design focused on networks with millions of parameters—a deliberate choice grounded in practical and methodological considerations. This scale represents approximately 80–85% of production ML deployments according to recent surveys (Strubell et al., 2019; Schwartz et al., 2020): the computer vision systems in medical diagnostics, edge devices for real-time inference, industrial automation controllers, and specialized domain models that constitute the majority of real-world deep learning applications. While billion-parameter language models capture research attention, they represent a small fraction of deployed systems where understanding training dynamics has immediate practical value.

Beyond relevance, this scale enables the systematic experimentation required for our analysis. Our 55 experiments with 30,147 measurement points across architectures, depths, widths, and datasets would be computationally prohibitive at billion-parameter scale. The comprehensive threshold sweeps (11 values), multiple detection methods, cross-metric validation, and architecture comparisons that form the core of our contribution require tractable training and checkpointing costs.

More fundamentally, if phase transition detection methods are scientifically valid, they should demonstrate internal consistency at the scales we study. The complete methodological disagreement we observe—PELT and threshold methods showing -0.029 correlation—represents a fundamental problem for the detection

paradigm regardless of whether billion-parameter models exhibit different behavior. A detection method that produces uncorrelated results across parameter choices or fails cross-metric validation at million-parameter scale cannot be trusted at larger scales without independent validation.

Whether billion-parameter models exhibit clearer phase structure due to emergent properties (Wei et al., 2022), or whether continuous dynamics become even more pronounced, remains an important open question. However, our findings establish that for the predominant use case of production ML, current detection methods lack the robustness required for reliable characterization of training dynamics. This represents a significant gap between research narratives about training phases and the empirical reality practitioners face when deploying models at scale.

10.3 Geometric Characterization

While we found no robust discrete transitions, understanding the continuous evolution of network geometry remains valuable. Future work could analyze metric distortion between consecutive checkpoints:

$$\mathcal{D}(t) = \mathbb{E}_{x,y} \left| \log \frac{\|h_t(x) - h_t(y)\|}{\|h_{t-1}(x) - h_{t-1}(y)\|} \right| \tag{12}$$

This could reveal whether smooth evolution maintains quasi-isometric properties or involves continuous geometric reorganization. Recent work by Huh et al. (2024) suggests that different architectures may converge to similar geometric representations, which could be studied through this lens.

10.4 Scope and Methodological Boundaries

Our analysis deliberately examines weight-space spectral metrics rather than representation or functional dynamics. This choice is methodologically conservative—representation similarity metrics face even more severe measurement instability (Kornblith et al., 2019). We focus on standard optimization regimes (Adam, learning rates 10^{-4} to 10^{-3}) where previous phase transition claims were made, not extreme conditions known to produce grokking (Power et al., 2022) or catapult phenomena.

These constraints are not limitations but design choices. We examine the regime where most production ML operates. Finding methodological artifacts here does not preclude genuine transitions at extreme scales—it establishes that phenomena reported in standard settings are measurement artifacts. This is infrastructure work: preventing false theoretical constructions before the field builds on unstable foundations.

11 Conclusion

Our analysis of transition detection in neural network training dynamics show substantial methodological sensitivity: conclusions about transition timing, frequency, and existence vary dramatically with detection method. Threshold-based approaches identify different transitions at each threshold level, with detections decreasing by an order of magnitude from 2σ to 100σ. More critically, threshold-free PELT detection shows negligible correlation (-0.029) with threshold methods, identifying 40–52 transitions where threshold methods detect virtually none. Cross-metric validation provides the most decisive evidence: no transitions appeared consistently across different spectral metrics in our experiments. This absence of cross-metric agreement, combined with the complete disagreement between detection methods, suggests that detected "transitions" reflect measurement artifacts rather than genuine geometric reorganization. The mechanistic source of this disagreement is clear: neural network optimization follows continuous gradient flow with architecture-specific relaxation rates. Methods detecting statistical distribution changes (PELT) versus absolute magnitude crossings (thresholds) necessarily capture different aspects of this continuous process, creating the illusion of transitions where none exist. The most robust phenomenon we observe is the escape from random initialization, typically occurring within the first 10% of training. Beyond this point, our evidence suggests largely continuous evolution, though different methods partition this continuity in incompatible ways. These findings have important implications for how we conceptualize neural network training, at least for models at the scale we studied. Our results suggest that continuous monitoring may be more appropriate than discrete

transition detection for practical applications. While the phase transition framework provides intuitive structure and has proven useful in many contexts, our experiments indicate it may not capture the empirical reality of training dynamics in the models we examined.

This work provides necessary methodological infrastructure for understanding neural network training dynamics. We do not claim to advance architectural understanding of intelligence—such breakthroughs require different approaches, likely involving embodied learning with physical constraints. Instead, we establish that at the scales where most ML operates, widely reported phase transitions are measurement artifacts. This finding prevents the field from constructing elaborate theories on false foundations. Sometimes scientific progress requires careful establishment of what doesn't exist before pursuing what does.

Future research would benefit from developing theories and tools that accommodate both continuous and discrete perspectives on training dynamics. The apparent continuity in our experiments may not extend to all scales, architectures, or training regimes, and further investigation across diverse settings would be valuable. Until more robust detection methods are developed—ones that show cross-method agreement and metric consistency—practitioners should base decisions on continuous monitoring rather than detected transitions, and theoreticians should interpret method-dependent observations with appropriate caution.

References

Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep neural networks. *arXiv preprint arXiv:1711.08856*, 2018.

Ryan Prescott Adams and David JC MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

Rana Ali Amjad, Kairen Liu, and Bernhard C Geiger. Understanding neural networks and individual neuron importance via information-ordered cumulative ablation. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

Sylvain Arlot, Alain Celisse, and Zaid Harchaoui. A kernel multiple change-point algorithm via model selection. *Journal of Machine Learning Research*, 20(162):1–56, 2019.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32, 2019.

Boaz Barak, Benjamin Edhan, Surbhi Goel, Sham Goel, Eran Malach, Andrea Montanari, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *arXiv preprint arXiv:2207.08799*, 2022.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. In *Proceedings of the National Academy of Sciences*, volume 116-32, pages 15849–15854. National Acad Sciences, 2019.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48. ACM, 2009.

Xander Davies, Lauro Langosco, and David Krueger. Unifying grokking and double descent. *arXiv preprint arXiv:2405.15551*, 2024.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred A Hamprecht. Essentially no barriers in neural network energy landscape. In *International Conference on Machine Learning*, pages 1309–1318. PMLR, 2018.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. *arXiv preprint arXiv:2110.06296*, 2021.

Yitong Feng and Stephen Tu. The rank of neural networks: Dynamics, control, and generalization. *arXiv preprint arXiv:2207.00193*, 2022.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2021.

Stanislav Fort and Stanislaw Jastrzebski. Large scale structure of neural network loss landscapes. *Advances in Neural Information Processing Systems*, 32, 2019.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. The lottery ticket hypothesis at scale. *arXiv preprint arXiv:1903.01611*, 2019.

Xinyang Gao, Babak Shahbaba, Norm Fortin, and Hernando Ombao. On the theory of learning with privileged information. *arXiv preprint arXiv:1708.08021*, 2017.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, pages 8789–8798, 2018.

Bernhard C Geiger and Gernot Kubin. Information-theoretic analysis of neural networks: An overview. *Information*, 15(1):32, 2024.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5484–5495. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.446.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. *arXiv preprint arXiv:1901.10159*, 2019.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Observing the effect of critical periods in neural networks. *arXiv preprint arXiv:1908.10299*, 2019.

Ziv Goldfeld and Yury Polyanskiy. The information bottleneck problem and its applications in machine learning. *IEEE Journal on Selected Areas in Information Theory*, 1(1):19–38, 2020.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

Stanisław Jastrzębski, Devansh Arpit, Oliver Górszczak, Giancarlo Kerg, Kyunghyun Cho, and Yoshua Bengio. The break-even point on optimization trajectories of deep neural networks. *arXiv preprint arXiv:2002.09572*, 2020.

Kenji Kawaguchi, Zhun Deng, Xu Ji, and Jiaoyang Huang. Does the information bottleneck principle explain deep learning? *arXiv preprint arXiv:2302.08404*, 2023.

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.

Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

Artemy Kolchinsky, Brendan D Tracey, and Steven Van Kuyk. Nonlinear information bottleneck. *Entropy*, 21(12):1181, 2019.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529, 2019.

Tanishq Kumar, Blake Nagarajan, Courtney Paquette, and Suriya Gunasekar. Grokking as the transition from lazy to rich training dynamics. *arXiv preprint arXiv:2405.15071*, 2024.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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:2003.06673*, 2020.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Ziming Liu, Eric J Ouyang, Douwe Kiela, Zhe Gan, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. *arXiv preprint arXiv:2310.20180*, 2023.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l0 regularization. *arXiv preprint arXiv:1712.01312*, 2018.

James Lucas, Juhan Bae, Michael R Zhang, Stanislav Fort, Richard Zemel, and Roger Grosse. Analyzing monotonic linear interpolation in neural network loss landscapes. *arXiv preprint arXiv:2104.11044*, 2021.

Charles H Martin and Michael W Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165): 1–73, 2021.

William Merrill, Nikolaos Tsilivis, and Aman Shukla. A tale of two circuits: Grokking as competition of sparse and dense subnetworks. *arXiv preprint arXiv:2303.11873*, 2023.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Neel Nanda, Lawrence Chan, Tom Liberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

Samet Oymak and Mahdi Soltanolkotabi. Generalization guarantees for neural networks via harnessing the low-rank structure of the jacobian. *arXiv preprint arXiv:1906.05392*, 2019.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

Adam Paszke, Sam Gross, Francisco Massa, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035, 2019.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Olivier Roy and Martin Vetterli. Effective dimension reduction using conditional expectation with applications. *Signal Processing*, 87(12):3009–3020, 2007.

Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM (JACM)*, 54(4):21–es, 2007.

Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green ai. *Communications of the ACM*, 63 (12):54–63, 2020.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, pages 369–386. SPIE, 2019.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565, 2022.

H Eugene Stanley. *Phase transitions and critical phenomena*. Oxford University Press, 1971.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, 2019.

Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *Advances in Neural Information Processing Systems*, 33, 2020.

Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua Susskind. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon. *arXiv preprint arXiv:2206.04817*, 2022.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–10. IEEE, 2015.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020.

Shashanka Ubaru and Yousef Saad. Fast methods for estimating the numerical rank of large matrices. *arXiv preprint arXiv:1603.03014*, 2017.

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency. *arXiv preprint arXiv:2309.02390*, 2023.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.

Huan Wang, Can Qin, Yulun Zhang, and Yun Fu. Neural pruning via growing regularization. *arXiv preprint arXiv:2012.09243*, 2020.

Jason Wei, Yi Tay, Rishi Bommasani, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

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

Chhavi Yadav and Léon Bottou. Cold case: The lost mnist digits. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhou Yang, Amir Barati, Michele Bertolini, and Mehrdad Farajtabar. Spectral evolution and invariance in linear-width neural networks. *arXiv preprint arXiv:2405.00811*, 2024.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, 2015.