Training Trajectory Structure: Regime Detection, Cross-Seed Convergence, and Speculative Weight Prediction

Jeremy McEntire
jmc@cageandmirror.com

Abstract

Training a neural network is not a smooth descent. The optimization trajectory alternates between stable basins where gradients are predictable and chaotic transitions where they are not. We show that these regimes are detectable in real time using activation fingerprints, synchronized across independent training runs, and exploitable for speculative weight prediction.

Using 100 fixed probe inputs to capture activation snapshots at 40 checkpoints across GPT-2 124M (5 seeds), Qwen 2.5-1.5B (4–5 seeds), and Qwen 2.5-7B (5 seeds) trained on WikiText-103, we establish three results. First, a three-regime taxonomy (chaotic, transition, stable) based on thresholded cosine similarity of consecutive activation fingerprints correlates with loss curve features and generalizes across two orders of magnitude in model scale. Second, independently initialized seeds converge to nearly identical final states—validation loss coefficient of variation is 0.41% at 124M, 2.43% at 1.5B, and 1.53% at 7B—with phase boundaries synchronized to within ±50 gradient steps at most scales. Third, speculative weight prediction using finite-difference extrapolation achieves 24% strict acceptance at 124M, 37% at 1.5B, and 60–90% at 7B in favorable regimes, while momentum-based prediction (Adam $m/\sqrt{v}$ extrapolation) fails catastrophically at all scales with 100–10,000× loss inflation.

Together, these findings suggest that training trajectories contain exploitable structure that current methods ignore. The regimes are landscape properties—not initialization artifacts—and prediction viability is regime-dependent, not scale-dependent. We do not measure wall-clock speedup; whether prediction-plus-validation is cheaper than K gradient steps remains open.

1 Introduction

Every gradient step in neural network training depends on the result of the previous one. This sequential bottleneck consumes enormous resources: training a frontier language model costs millions of dollars in GPU-hours [Kaplan et al., 2020], and standard practice multiplies that cost by 3–5× through multi-seed training to assess robustness. Yet the sequential nature of gradient descent is not as rigid as it appears. Weight trajectories exhibit substantial regularity—particularly in late training—suggesting that future parameter states may be predictable from past trajectory information, and that independent training runs may be exploring the same landscape features rather than genuinely different solutions.

This paper presents evidence for both claims. We show that training trajectories have a detectable phase structure—alternating between chaotic regimes where representations change rapidly and stable regimes where they barely move—and that this structure is synchronized across independent random initializations. The synchronization implies that regime boundaries are properties of the optimization landscape (the interaction of architecture, data distribution, and learning rate schedule) rather than artifacts of any particular initialization. And in the stable regimes, we show that simple finite-difference extrapolation can predict future weight states with meaningful acceptance rates, while optimizer-state extrapolation fails catastrophically.

Speculative execution as a framework. The conceptual framework comes from speculative execution in computer architecture. The Automatically Scalable Computation (ASC) architecture [Waterland et al., 2014] accelerates sequential program execution by viewing it as a trajectory through state space: a *recognizer* identifies states amenable to prediction, *predictors* forecast future states from observed trajectory structure, and a *validator* confirms correctness before fast-forwarding. ASC achieved 256× speedup on 1024 cores for

unmodified sequential binaries—super-linear, because the trajectory had enough structure that speculative exploration discovered shortcuts the sequential path could not see.

We adapt this architecture for neural network training. The recognizer becomes a regime detector based on activation fingerprint similarity. The predictors become analytic weight extrapolators. The validator becomes a held-out loss comparison. The key observation motivating this transplant is that training trajectories should be *more* structured than arbitrary program execution, because gradient descent follows smooth loss landscapes with basins of attraction and predictable curvature.

Mode connectivity and the landscape. Prior work establishes that the loss landscape has more structure than its astronomical dimensionality suggests. Garipov et al. [2018] and Draxler et al. [2018] showed that independently trained networks can be connected by smooth, low-loss paths in weight space. Ainsworth et al. [2023] demonstrated that after permutation alignment, networks from different seeds are often linearly mode connected. Frankle and Carlin [2019] showed that sparse subnetworks ("winning tickets") can match full-network performance, and Frankle et al. [2020] showed that networks become "stable to SGD noise" early in training, after which models sharing a common prefix converge to linearly connected solutions.

These results operate in weight space. We complement them with evidence from activation space—showing that models converge not just to functionally equivalent parameterizations, but to near-identical internal representations, and that the convergence has a characteristic temporal structure tied to detectable training regimes.

Contributions.

1. **Three-regime taxonomy**: Chaotic, transition, and stable training regimes detected via thresholded cosine similarity of activation fingerprints, validated across GPT-2 124M, Qwen 1.5B, and Qwen 7B.
2. **Cross-seed convergence**: Independent seeds converge to <2.5% validation loss CV at all scales, with phase boundaries synchronized to ±50 steps. We measure temporal stability within each seed and final-state convergence across seeds; direct cross-seed fingerprint comparison at matched checkpoints remains open.
3. **Universal momentum catastrophe**: Adam moment extrapolation produces 100–10,000× loss inflation at all three scales—a qualitative, not marginal, failure.
4. **Regime-dependent prediction**: Finite-difference extrapolation achieves 60–90% strict acceptance at 7B in stable regime, with acceptance rates that *increase* with model scale within comparable regimes.
5. **Honest negative results**: We report what does not work (momentum prediction, prediction during chaotic regimes) as prominently as what does.

2 Methods

2.1 Models, Data, and Training

We evaluate three language models spanning two orders of magnitude in parameter count:

- **GPT-2 124M**: 12 layers, 768 hidden dimensions, 12 attention heads. 5 seeds (42–46).
- **Qwen 2.5-1.5B** [Team, 2025]: 28 layers, 1536 hidden dimensions, 12 attention heads. 4 seeds (42–45) for cross-seed analysis; 5 seeds for regime detection.
- **Qwen 2.5-7B** [Team, 2025]: 28 layers, 3584 hidden dimensions, 28 attention heads. 5 seeds (42–46).

All models are randomly initialized and trained from scratch on WikiText-103 [Merity et al., 2017] for 2000 steps with AdamW ($\eta = 5 \times 10^{-5}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0.01), cosine learning rate schedule with 100-step warmup, batch size 4, and sequence length 256. Checkpoints are saved every 50 steps (40 per seed), yielding 560 total fingerprint observations across all models. GPT-2 and Qwen 1.5B were trained on NVIDIA A100 40GB; Qwen 7B on A100 80GB.

2.2 Activation Fingerprinting

At each checkpoint, we compute an *activation fingerprint* by forward-passing 100 fixed probe sentences through the model and concatenating the mean-pooled final hidden states. The probe set spans arithmetic, linguistic, factual, and random inputs, and remains identical across all checkpoints, seeds, and model scales.

The cosine similarity between consecutive fingerprints serves as a proxy for representational stability:

$$s_t = \frac{\mathbf{a}_t \cdot \mathbf{a}_{t-\Delta}}{\|\mathbf{a}_t\| \|\mathbf{a}_{t-\Delta}\|} \tag{1}$$

where $\mathbf{a}_t$ is the activation fingerprint at step t and $\Delta = 50$ steps (the checkpoint interval).

2.3 Regime Classification

We classify training into three regimes using thresholds calibrated on initial training runs:

$$\text{regime}(t) = \begin{cases} \text{stable} & \text{if } s_t > \tau_{\text{high}} \\ \text{chaotic} & \text{if } s_t < \tau_{\text{low}} \\ \text{transition} & \text{otherwise} \end{cases} \tag{2}$$

The naming convention draws an analogy to dynamical systems. In chaotic dynamical systems, the Lyapunov exponent measures the rate at which nearby trajectories diverge; a positive exponent indicates chaos, while a negative exponent indicates stability. Our cosine similarity signal is *analogous*—low similarity between consecutive checkpoints indicates rapid representational change ("chaotic"), while high similarity indicates near-stationarity ("stable"). We emphasize that this is an analogy, not a formal Lyapunov exponent computation: we measure representational change over a fixed step interval, not the infinitesimal divergence rate of perturbed trajectories. The analogy is useful because it predicts correctly—prediction succeeds in stable regimes and fails in chaotic ones—but it should not be taken as a rigorous dynamical-systems proof.

Threshold calibration. The thresholds τ_{high} and τ_{low} were calibrated on GPT-2 124M initial runs and carried forward to larger scales without recalibration. The threshold values are not reported—a reproducibility gap we acknowledge in Section 7. Future work should include sensitivity analysis across threshold choices and an adaptive thresholding scheme that recalibrates with model scale.

2.4 Cross-Seed Measurements

For cross-seed analysis, the key measurements are:

- **Cross-seed loss CV**: Coefficient of variation of final validation loss across seeds.
- **Phase boundary synchronization**: Standard deviation of the training step at which each seed first enters each regime.
- **Per-regime cosine similarity**: Mean consecutive-checkpoint similarity within each regime, computed per seed and averaged.
- **Cross-seed regime agreement**: Fraction of checkpoints where all seeds are classified into the same regime.

An important clarification. The similarity measurements above are *within-seed temporal comparisons*: how much does seed i's fingerprint change from step t to step $t + \Delta$? They are *not* cross-seed comparisons at matched checkpoints (i.e., how similar is seed i's fingerprint to seed j's fingerprint at the same step t). The cross-seed evidence comes from two sources: (1) all seeds exhibit the same temporal stability dynamics (identical regime classifications at the same steps), and (2) all seeds converge to the same final validation loss. Direct cross-seed fingerprint comparison at matched checkpoints—which would establish whether seeds traverse identical intermediate representations, not merely arrive at the same destination—remains open.

2.5 Speculative Weight Prediction

Given a checkpoint at step t with parameters θ_t, we predict parameters θ_{t+K} using three analytic predictors:

Momentum prediction. Extrapolates using Adam's exponential moving averages:

$$\hat{\theta}^{\text{mom}}_{t+K} = \theta_t + K \cdot \frac{m_t}{\sqrt{v_t} + \epsilon} \tag{3}$$

where m_t and v_t are Adam's first and second moment estimates.

Linear prediction. Extrapolates from two consecutive checkpoints:

$$\hat{\theta}^{\text{lin}}_{t+K} = \theta_t + \frac{K}{\Delta}(\theta_t - \theta_{t-\Delta}) \tag{4}$$

Quadratic prediction. Fits a parabola through three consecutive checkpoints:

$$\hat{\theta}^{\text{quad}}_{t+K} = \theta_t + \frac{K}{\Delta}(\theta_t - \theta_{t-\Delta}) + \frac{K(K-\Delta)}{2\Delta^2}(\theta_t - 2\theta_{t-\Delta} + \theta_{t-2\Delta}) \tag{5}$$

After computing $\hat{\theta}_{t+K}$, we load the predicted weights and evaluate validation loss $\hat{L}_{t+K}$. We compare against the current validation loss L_t using three acceptance criteria:

- **Strict**: Accept if $\hat{L}_{t+K} < L_t$ (predicted improvement).
- **Adaptive**: Accept if $\hat{L}_{t+K} < L_t + \sigma_L$ (within one standard deviation of recent losses).
- **Proximity**: Accept if $|\hat{L}_{t+K} - L_t| < \epsilon \cdot L_t$ (within a percentage of current loss).

If accepted, the model would fast-forward to step $t+K$, skipping K gradient updates. If rejected, training continues unmodified—the prediction is purely speculative with no side effects. Predictions are evaluated only on non-chaotic checkpoints, since chaotic regimes yield near-zero acceptance by design.

2.6 Evaluation Protocol

For each model and seed, we run up to three passes:

1. **Training**: 2000 steps, saving checkpoints every 50 steps. Record activation fingerprints and regime classifications at each checkpoint.
2. **K-Sweep**: For each non-chaotic checkpoint, evaluate all three predictors at $K \in \{5, 10, 25, 50, 75, 100\}$ with all three acceptance criteria.
3. **Cascades** (124M and 1.5B only): From stable-regime checkpoints, evaluate chained predictions at configurations $(D, K) \in \{(4, 25), (2, 50), (10, 10)\}$. Skipped at 7B due to memory constraints.

3 Regime Detection

The loss landscape of neural network training is not uniform. Early in training, the model undergoes rapid representational reorganization—activation fingerprints change substantially between consecutive checkpoints, gradient directions are noisy, and the trajectory is hard to predict. Later, the model settles into a basin where representations change incrementally, gradients are consistent, and the trajectory becomes smooth. Between these extremes lies a transition zone.

Table 1: Regime breakdown by model scale (mean ± std across seeds). Checkpoint counts out of 40 total.

Model	Chaotic	Transition	Stable	Unknown
GPT-2 124M	1.6 ± 1.7	24.0 ± 2.7	13.4 ± 3.2	1.0 ± 0.0
Qwen 2.5-1.5B	25.6 ± 0.5	12.4 ± 1.1	1.0 ± 1.0	1.0 ± 0.0
Qwen 2.5-7B	19.2 ± 1.8	6.4 ± 1.1	13.4 ± 1.5	1.0 ± 0.0

3.1 Three-Regime Taxonomy

Table 1 summarizes the regime breakdown across all three model scales.

The regime distribution changes non-monotonically with model scale. GPT-2 124M spends 34% of training in stable regime and only 4% in chaotic—the loss landscape of this small model is relatively benign. Qwen 1.5B reverses this: 64% chaotic, reaching stable in only 0–2 of 40 checkpoints. This is the most surprising result at this scale—the model is large enough to have complex optimization dynamics (prolonged chaotic phase) but not large enough for those dynamics to settle within 2000 steps. Qwen 7B presents a reversal: despite being 4.7× larger than 1.5B, it spends 34% of training in stable regime—matching GPT-2 124M—while maintaining 48% chaotic. The 7B model exhibits the clearest three-phase structure: sustained chaos, brief transition, then sustained stability.

3.2 Activation Similarity by Regime

Table 2: Mean cosine similarity between consecutive activation fingerprints, by regime. The opposing trends—chaotic similarity *decreasing* and stable similarity *increasing* with scale—are the signature finding.

Model	Chaotic	Transition	Stable
GPT-2 124M	0.9942	0.9987	0.9997
Qwen 2.5-1.5B	0.9851	0.9988	0.9995
Qwen 2.5-7B	0.9764	0.9982	0.9999

Two opposing trends emerge from Table 2:

- **Chaotic regime**: Similarity *decreases* with scale ($0.994 \rightarrow 0.985 \rightarrow 0.976$). Larger models undergo more violent representational reorganization during early training.
- **Stable regime**: Similarity *increases* with scale ($0.9997 \rightarrow 0.9995 \rightarrow 0.9999$). Larger models achieve tighter representational convergence once settled.
- **Transition regime**: Remarkably stable across scales ($\sim$0.998), serving as a natural boundary between the two extremes.

This pattern is consistent with a funnel-shaped loss landscape: larger models explore more broadly during chaotic training but converge more tightly to a narrow solution manifold.

3.3 The Lyapunov Analogy

The cosine similarity signal s_t behaves like a smoothed, inverted Lyapunov exponent: when s_t is low, the training trajectory is diverging rapidly from its recent history (positive effective Lyapunov exponent); when s_t is near 1.0, the trajectory is nearly stationary (negative effective exponent). This analogy correctly predicts that prediction should succeed in stable regimes and fail in chaotic ones.

We stress that this is an analogy, not a proof. A true Lyapunov exponent requires measuring the divergence of infinitesimally perturbed trajectories, which we do not compute. Our signal is a finite-difference approximation over 50-step intervals in activation space, not weight space. The analogy is useful for building intuition and for explaining why regime-conditional prediction works, but the formal relationship between our cosine similarity signal and the Lyapunov spectrum of the training dynamics remains to be established.

4 Cross-Seed Convergence

If training regimes are landscape properties rather than initialization artifacts, then independently initialized seeds should exhibit the same regime dynamics. We test this prediction by training multiple seeds per model scale and comparing their trajectories.

4.1 Loss Convergence

Table 3 summarizes final validation losses across seeds.

Table 3: Final validation loss across seeds. CV = coefficient of variation.

Model	Seeds	Mean	Std	Range	CV (%)
GPT-2 124M	5	1.616	0.007	1.609–1.625	**0.41**
Qwen 2.5-1.5B	4	0.814	0.020	0.801–0.843	2.43
Qwen 2.5-7B	5	0.985	0.015	0.970–1.002	1.53

All three scales exhibit low cross-seed variance. GPT-2 achieves the tightest convergence (CV = 0.41%), likely because its small parameter count constrains the solution space. Qwen 1.5B shows the highest CV (2.43%), which is non-monotonic with model size—the 7B model (1.53%) is tighter than 1.5B. We discuss this anomaly in Section 7.

4.2 Phase Boundary Synchronization

Table 4: Phase boundary synchronization: training step at which each seed first enters each regime. Std computed across seeds.

Model	Transition	Mean Step	Std	Range
GPT-2 124M	First transition	100	0	[100, 100]
	First stable	160	22	[150, 200]
Qwen 2.5-1.5B	First transition	1375	29	[1350, 1400]
	First stable	1775	35	[1750, 1800]
Qwen 2.5-7B	First transition	1060	108	[1000, 1250]
	First stable	1340	42	[1300, 1400]

The synchronization is striking. All five GPT-2 seeds enter transition at exactly step 100—zero variance. At 1.5B, transition onset varies by only ±29 steps across 4 seeds. At 7B, transition onset is wider (±108 steps, driven by seed 42 entering late at step 1250), but stable entry is tight (±42 steps).

Given that these models start from entirely independent random initializations, this synchronization implies that regime boundaries are deterministic properties of the optimization landscape—the learning rate schedule, data distribution, and architecture jointly determine *when* the model transitions, regardless of initialization.

4.3 Regime Agreement Across Seeds

Cross-seed regime agreement is lowest at 124M (74.5%), not because GPT-2 seeds disagree on the landscape, but because GPT-2 oscillates between transition and stable rather than cleanly separating regimes. The larger models show cleaner, more deterministic regime progressions (96.9% agreement at 1.5B, 93.0% at 7B).

Table 5: Fraction of training spent in each regime (mean across seeds) and cross-seed regime agreement.

Model	Chaotic	Transition	Stable	Regime Agreement
GPT-2 124M	4.0%	60.0%	33.5%	74.5%
Qwen 2.5-1.5B	63.7%	31.9%	1.9%	96.9%
Qwen 2.5-7B	48.0%	16.5%	33.0%	93.0%

4.4 Ensemble Collapse in Validation Loss Trajectory (7B)

To visualize convergence dynamics directly, we trace the cross-seed validation loss CV during 7B training. At step 50, CV is 0.73%—baseline diversity from random initialization. At step 250, CV spikes to 35.8% as one seed (seed 44) hits a catastrophic early loss spike (validation loss 2.01 vs. mean $\sim$1.0). From step 500 onward, CV monotonically decreases: 3.1% $\rightarrow$ 2.1% $\rightarrow$ 1.6% $\rightarrow$ 1.53% at convergence.

This is the clearest quantitative signature of what we call *ensemble collapse*: seeds that wildly disagree early in training progressively converge to the same loss basin. The catastrophic outlier (seed 44) is fully absorbed by step 500—the optimization landscape's attractor basin is strong enough to recover from 2$\times$ loss spikes.

4.5 What Cross-Seed Convergence Does and Does Not Show

The evidence establishes that:

1. All seeds reach the same final validation loss (to within noise).
2. All seeds traverse the same regime sequence at approximately the same training steps.
3. Within each seed, the temporal stability dynamics (how much the fingerprint changes per step) are consistent across seeds.

The evidence does *not* establish that seeds pass through identical intermediate representations. We measure within-seed consecutive similarity (how much does seed i's fingerprint change from step t to step $t + \Delta$?), not cross-seed similarity at matched checkpoints (how similar is seed i's fingerprint to seed j's fingerprint at step t?). The former tells us that all seeds undergo the same *dynamics*; the latter would tell us they occupy the same *states*. Direct cross-seed fingerprint comparison at matched checkpoints remains open work.

5 Speculative Weight Prediction

The regime detection framework from Sections 3 and 4 identifies *when* the training trajectory is predictable. This section tests whether that predictability can be exploited: given a checkpoint in a stable or transition regime, can we predict the model's weights K steps into the future?

5.1 The Universal Momentum Catastrophe

Our most striking finding is negative. Momentum-based weight prediction—extrapolating Adam's exponential moving averages (Equation 3)—fails catastrophically at every scale tested.

This is not a marginal degradation. At K=5—the shortest prediction horizon—momentum-extrapolated weights produce losses 122–227$\times$ higher than the current checkpoint. At K=100, the inflation reaches 2,678–10,764$\times$. The predicted weights lie so far outside the region of the loss landscape explored during normal training that the model's outputs are essentially random.

Table 6: Momentum predictor failure across all three model scales. "Predicted loss" is the validation loss of the model with momentum-extrapolated weights; "Actual loss" is the current checkpoint's validation loss. The catastrophe is universal.

	GPT-2 124M			**Qwen 1.5B**			**Qwen 7B**		
K	Pred.	Act.	Ratio	Pred.	Act.	Ratio	Pred.	Act.	Ratio
5	201	1.64	122×	142	0.82	173×	222	0.97	227×
10	360	1.64	219×	250	0.82	305×	347	0.97	357×
25	1284	1.64	782×	581	0.82	709×	681	0.97	700×
50	4505	1.64	2742×	1153	0.82	1407×	1259	0.97	1296×
75	9988	1.64	6077×	1819	0.82	2218×	1901	0.97	1957×
100	17691	1.64	10764×	2468	0.82	3009×	2601	0.97	2678×

Root cause: norm explosion. The displacement $K \cdot m_t/(\sqrt{v_t}+\epsilon)$ grows linearly with K, but the region of validity around the current point on the loss surface does not. Adam's moment estimates accumulate gradient information across the entire training history with exponential decay; extrapolating them forward projects the weights far beyond any basin the model has visited. The finite-difference predictors avoid this failure because they extrapolate from actually observed weight deltas, which are inherently bounded by the step sizes taken during training.

Practical implication. This finding has implications for weight nowcasting methods such as WNN [Jang and Han, 2023], NiNo [Knyazev et al., 2025], and XGrad [Guan et al., 2024]. Optimizer-state extrapolation is fundamentally unsuitable for speculative weight prediction at any scale. Methods must use trajectory-bounded extrapolation—extrapolating from observed weight deltas rather than accumulated gradient moments.

5.2 Finite-Difference Prediction: Regime-Dependent Success

Where momentum fails, finite-difference predictors succeed—but only in favorable regimes.

Table 7: Strict acceptance rate (%) for finite-difference predictors. Momentum achieves 0% in transition regimes at all scales and is omitted. Values are mean ± std across seeds.

	Transition		**Stable**	
K	Linear	Quadratic	Linear	Quadratic
GPT-2 124M				
5	9.3 ± 3.8	7.9 ± 3.0	24.3 ± 6.8	22.1 ± 10.8
10	3.3 ± 3.2	3.4 ± 3.3	18.8 ± 8.0	13.9 ± 8.4
25	0.7 ± 1.7	0.0 ± 0.0	5.8 ± 3.5	9.1 ± 2.0
Qwen 2.5-1.5B (transition only—stable N too small)				
5	37.2 ± 11.0	36.5 ± 12.5	—	—
10	31.3 ± 4.7	39.4 ± 9.1	—	—
25	22.9 ± 8.1	20.5 ± 9.2	—	—
50	17.8 ± 9.0	13.2 ± 6.3	—	—

At 124M, linear prediction achieves 24% strict acceptance at K=5 in stable regime—roughly one in four predictions successfully skips 5 gradient steps. At 1.5B, the same predictor achieves 37% strict in the transition regime (stable regime is essentially unavailable at this scale). Within comparable regimes, accuracy improves with scale: larger models produce smoother weight trajectories.

5.2.1 The 7B Scale: Linear Prediction Dominates

The 7B model reliably reaches stable regime (mean 13.4 stable checkpoints per seed), enabling direct evaluation of all predictors in the most favorable conditions.

Table 8: Strict acceptance rate (%) in stable regime, Qwen 2.5-7B. Momentum achieves 0% at all K values across all 5 seeds. Linear prediction achieves 60–90% acceptance with no degradation at large K. *Interpretation:* Higher acceptance rates indicate the predictor successfully anticipated where the model would be after K gradient steps. Rates above 50% mean prediction is better than chance; rates near 0% indicate the predictor fails in that regime.

K	Momentum	Linear	Quadratic
5	0.0 ± 0.0	66.7 ± 22.4	60.0 ± 41.8
10	0.0 ± 0.0	73.3 ± 27.5	60.0 ± 41.8
25	0.0 ± 0.0	73.3 ± 27.5	70.0 ± 44.7
50	0.0 ± 0.0	60.0 ± 34.6	80.0 ± 27.4
75	0.0 ± 0.0	90.0 ± 22.4	60.0 ± 54.8
100	0.0 ± 0.0	90.0 ± 22.4	80.0 ± 44.7

Three findings distinguish the 7B results:

Linear prediction achieves 60–90% strict acceptance. This is dramatically higher than GPT-2's 24% (stable) or Qwen 1.5B's 37% (transition). Remarkably, acceptance does *not degrade* with K: K=100 achieves 90% strict acceptance vs. K=5's 67%. The 7B model near convergence has an extremely flat loss landscape—the linear predictor's extrapolation remains within the loss basin regardless of prediction horizon. Adaptive acceptance (within one standard deviation) is 100% across all K values and seeds.

Linear outperforms quadratic. Quadratic prediction (60–80% strict) has substantially higher cross-seed variance (std 27–55%) than linear (std 22–35%). The extra degree of freedom in quadratic fitting sometimes overshoots. The linear predictor's simplicity and stability make it the preferred method at 7B.

Seed 46 as an informative outlier. Seed 46 achieves the lowest final validation loss (0.970) but the worst predictor acceptance at small K (50% linear at K=5). The cause is not regime distribution—seed 46 has 13 stable checkpoints, exactly at the mean. Rather, the loss curve exhibits non-monotone micro-oscillations near convergence: the model has converged so tightly that validation loss fluctuates stochastically, causing strict-criterion rejections. This suggests that at 7B, the strict criterion becomes unnecessarily conservative near convergence.

5.3 Proximity Acceptance: Graceful Degradation

Proximity-based acceptance (predicted loss within 5% of current) reveals that even when strict acceptance fails, the predictions are often close. At K=5, proximity acceptance is ∼99–100% at both 124M and 1.5B—the predicted weights produce nearly the same loss as the current checkpoint. The degradation is graceful: at 1.5B, proximity stays above 90% through K=25 and above 60% through K=50.

5.4 Cross-Seed Consistency of Predictions

At short horizons ($K \leq 10$), prediction results are perfectly consistent across seeds (CoV = 0%). Variance increases with K, as longer predictions depend on seed-specific trajectory details. Quadratic prediction shows higher variance than linear at large K, consistent with its sensitivity to local curvature.

Table 9: Proximity acceptance rate (%): predicted loss within 5% of current loss. Momentum remains at 0% in transition regimes. Proximity acceptance reveals graceful degradation with prediction depth.

	GPT-2 124M Trans.		**GPT-2 124M Stable**	
K	Linear	Quadratic	Linear	Quadratic
5	99.1 ± 1.9	100.0 ± 0.0	98.9 ± 2.5	98.9 ± 2.5
10	97.3 ± 2.5	93.4 ± 4.2	95.3 ± 4.7	97.6 ± 3.4
25	44.0 ± 9.8	37.8 ± 12.0	74.6 ± 2.6	70.6 ± 7.7
50	14.1 ± 6.8	9.3 ± 7.1	54.1 ± 16.0	48.2 ± 20.2

	Qwen 2.5-1.5B Transition	
K	Linear	Quadratic
5	100.0 ± 0.0	100.0 ± 0.0
10	100.0 ± 0.0	100.0 ± 0.0
25	94.6 ± 5.0	90.5 ± 7.1
50	66.2 ± 14.4	54.2 ± 10.1
75	38.3 ± 10.5	20.2 ± 12.6
100	22.9 ± 14.5	6.7 ± 9.9

Table 10: Coefficient of variation (%) for proximity acceptance rates across seeds, Qwen 2.5-1.5B transition regime.

K	Linear CoV	Quadratic CoV
5	0.0%	0.0%
10	0.0%	0.0%
25	5.2%	7.9%
50	21.7%	18.6%
75	27.3%	62.5%
100	63.3%	149.1%

5.5 What We Do Not Claim

We do not claim wall-clock speedup. The acceptance rates reported here measure how often a predicted weight state passes a loss-based validation criterion. Whether prediction-plus-validation is cheaper than computing K gradient steps depends on the cost of the validation forward pass relative to K training steps, which we do not measure. At K=5, the validation cost (one forward pass) is comparable to the cost saved (5 forward-backward passes), so net speedup is marginal even at 100% acceptance. At K=100 with 90% acceptance (as observed at 7B), the ratio is much more favorable—but we have not measured it. We report acceptance rates as evidence that the trajectory has exploitable structure, not as a speedup claim.

6 Discussion

6.1 The Merged Argument

The three results of this paper form a coherent picture:

1. Training trajectories have a detectable phase structure (Section 3).
2. That structure is synchronized across independent seeds—it is a landscape property, not an initialization artifact (Section 4).
3. The structure is exploitable: in stable regimes, simple extrapolation predicts future weight states with high acceptance (Section 5).

The connection to ASC [Waterland et al., 2014] runs deeper than analogy. In ASC, the recognizer identifies states from which prediction is tractable—precisely the role of our regime detector. ASC's predictors learn from observed trajectory structure, as do our weight extrapolators. The trajectory cache stores verified state transitions; our acceptance criteria serve the same gatekeeping function. Even the scaling behavior parallels: ASC found that prediction accuracy varies by program structure, just as we find it varies by model scale and training regime.

The key asymmetry favoring training over general program execution is verification cost. In ASC, verification requires executing the speculative segment and comparing end states—potentially as expensive as the original computation. In our framework, verification requires only a single forward pass on held-out data, which is $O(1)$ relative to the K gradient steps being predicted.

6.2 Landscape Implications

The cross-seed convergence results are consistent with a dominant solution manifold: a set of weight configurations that implement approximately the same function, toward which all initializations converge. The manifold has a characteristic temporal structure—chaotic $\rightarrow$ transition $\rightarrow$ stable—that all initializations traverse in synchrony.

Models trained from different seeds may have substantially different weights (weight-space divergence) yet produce nearly identical activations and losses (function-space convergence). This is consistent with the mode connectivity results of Garipov et al. [2018] and Ainsworth et al. [2023], but our evidence comes from activation space rather than weight space, and it captures the *dynamics* of convergence rather than only the endpoint.

The non-monotonic regime distribution—124M mostly stable, 1.5B mostly chaotic, 7B recovering to stable—suggests that the relationship between model capacity and landscape complexity is not simple. One hypothesis: 1.5B has enough parameters for complex dynamics but not enough for the landscape to "self-organize" into a clear funnel, while 7B's overparameterization creates a smoother landscape. Testing this hypothesis requires more data points in the 1–10B range.

6.3 Scaling Prediction

The trends across 124M, 1.5B, and 7B suggest predictions for larger scales:

1. Stable-regime similarity will approach 1.0000 more closely.
2. Linear prediction acceptance rates will continue to improve in stable regimes.
3. The regime availability bottleneck observed at 1.5B may not recur at larger scales—the 7B recovery suggests it is a mid-scale phenomenon.

Whether these trends continue or saturate at 13B, 70B, or larger scales is an empirical question that requires substantially more compute to answer.

6.4 Practical Implications

Multi-seed training. If models from different seeds converge to the same functional behavior with synchronized phase boundaries, multi-seed training provides diminishing returns at scale. At 7B, a single training run verified via regime detection—confirming that the model has reached stable regime—may capture the same information as a 5-seed ensemble at $\frac{1}{5}$ the compute cost.

Regime-conditional optimization. The regime detector is cheap (100 forward passes per checkpoint) and provides actionable information. During stable regimes, compute can be reduced (lower precision, larger batch sizes, speculative skipping). During chaotic regimes, the trajectory is unpredictable and must be followed step by step. The detector tells you which phase you are in.

Prediction as a research tool. Even without wall-clock speedup, the acceptance rate curves serve as a diagnostic: they measure how "smooth" the training trajectory is at each point, providing a more informative signal than loss alone.

7 Limitations

Scale and duration. All experiments use 2000 training steps on WikiText-103. This is a toy scale—far below the millions of steps used for production training. Whether the regime structure, cross-seed convergence, and prediction viability observed here generalize to longer training runs and larger datasets is an open question. The three-scale curve (124M, 1.5B, 7B) is suggestive but not definitive; more data points between 1B and 100B are needed.

Scale extrapolation. Our scaling claims rest on three data points (124M, 1.5B, 7B) from two architecture families (GPT-2, Qwen). We do not claim these trends extrapolate to frontier-scale models. The evidence supports the existence of regime structure at the tested scales, not universal scaling laws.

No wall-clock speedup measurement. We measure acceptance rates, not wall-clock time. The acceptance rates suggest potential speedup, but we do not demonstrate it. Whether prediction-plus-validation is cheaper than K gradient steps depends on implementation details (checkpoint loading overhead, validation batch size, GPU memory constraints) that we do not measure.

Cross-seed similarity is inferred, not directly measured. We measure within-seed temporal stability and cross-seed final-state convergence. We do not directly compare activation fingerprints across seeds at matched checkpoints. The evidence that seeds traverse similar intermediate representations is indirect—they exhibit the same dynamics and reach the same endpoint—not direct.

Regime thresholds not numerically specified. The thresholds τ_{high} and τ_{low} were calibrated on GPT-2 124M and carried forward. Their specific values are not reported in this paper, which hurts reproducibility. An adaptive thresholding scheme would be more principled.

1.5B anomaly unexplained. Qwen 1.5B shows higher cross-seed loss CV (2.43%) than both the smaller GPT-2 (0.41%) and the larger Qwen 7B (1.53%). It also barely reaches stable regime within 2000 steps. One possible explanation is that 1.5B is in a "critical zone" where the model has enough capacity for complex optimization dynamics but not enough for those dynamics to self-organize quickly. But this is speculation—we lack the data to distinguish it from an architecture-specific effect (GPT-2 vs. Qwen) or from insufficient training duration. The Qwen 1.5B ensemble also has only 4 seeds (one was lost to infrastructure failure), giving it the weakest statistical power.

Cascade evaluation is limited. Cascaded (chained) predictions were evaluated only at 124M and 1.5B, and the limited availability of stable checkpoints at 1.5B prevents drawing strong conclusions. Cascades were infeasible at 7B due to memory constraints.

Architecture confound. GPT-2 and Qwen have different architectures. The regime differences across scales may partly reflect architectural differences rather than pure scaling effects. An ideal experiment would use the same architecture family across all scales.

8 Related Work

Weight prediction during training. Kamarthi and Pittner [1999] used Taylor series extrapolation of weight trajectories. Sinha et al. [2017] trained neural predictors on weight histories. Jang and Han [2023] designed the Weight Nowcaster Network (WNN) for 5-epoch-ahead prediction. Knyazev et al. [2025] used graph neural networks (NiNo) modeling neuron connectivity, achieving up to 50% acceleration. Guan et al.

[2024] proposed XGrad using optimizer update rules. Anonymous [2024] applied linear prediction every 3 iterations. All these methods apply predictions unconditionally—none verify before acceptance, and none condition on detected training regimes.

Training dynamics and phase transitions. Cohen et al. [2021] identified progressive sharpening and edge-of-stability via Hessian eigenvalues. Lewkowycz et al. [2020] documented the catapult phase. Frankle et al. [2020] detected stability via weight-space linear interpolation. Nanda et al. [2023] identified three mechanistic phases in grokking. Saxe et al. [2014] derived exact training dynamics for deep linear networks with phase-transition-like behavior. Geiger et al. [2022] showed chaos is intrinsic to SGD. Our activation-space cosine similarity provides a computationally cheap alternative to these methods that operates in real time.

Loss landscape geometry and mode connectivity. Garipov et al. [2018] and Draxler et al. [2018] demonstrated smooth low-loss paths connecting independently trained networks. Ainsworth et al. [2023] showed linear mode connectivity after permutation alignment. Li et al. [2018] introduced filter-normalized loss landscape visualization. Keskar et al. [2017] linked batch size to basin sharpness. Foret et al. [2021] operationalized flat-minima seeking via SAM. Fort et al. [2019] analyzed deep ensembles from a loss landscape perspective. Wortsman et al. [2022] showed that averaging weights of independently fine-tuned models improves accuracy (model soups). Our cross-seed convergence results complement these weight-space findings with activation-space evidence.

Lottery tickets and subnetwork convergence. Frankle and Carlin [2019] showed that sparse subnetworks ("winning tickets") can match full-network performance. Frankle et al. [2020] showed stability to SGD noise after an early training prefix. Our work extends this: even without a shared prefix, models converge to the same activation dynamics.

Neural scaling laws. Kaplan et al. [2020] established power-law relationships between model size and loss. Hoffmann et al. [2022] refined compute-optimal scaling. Our finding that cross-seed variance decreases with scale (at least from 1.5B to 7B) is consistent with larger models exhibiting more predictable training dynamics.

Speculative execution. Speculative decoding [Leviathan et al., 2023, Chen et al., 2023, Stern et al., 2018] provides the predict-then-verify template for inference. ASC [Waterland et al., 2014, 2013] generalizes speculative execution to arbitrary sequential computation. We adapt the ASC framework for training trajectories.

Adaptive optimization. The Lookahead optimizer [Zhang et al., 2019] maintains fast/slow weights with k-step lookahead but always interpolates back (no acceptance criterion) and runs all k inner steps. Smith [2017] showed cyclical learning rates implicitly traverse different curvature regimes. RAdam [Liu et al., 2020] automatically transitions between SGD-like and Adam-like behavior.

9 Conclusion

Training trajectories have structure. We have shown that this structure manifests in three ways: as detectable regimes with characteristic stability properties, as synchronized phase boundaries across independent seeds, and as exploitable predictability in stable regimes.

The negative results are as informative as the positive ones. Momentum-based weight prediction fails catastrophically at every scale—a universal norm explosion in optimizer-state extrapolation that should caution any method relying on accumulated gradient moments for prediction. Prediction during chaotic regimes produces near-zero acceptance regardless of the method used. And cross-seed "convergence" in activation space, while strongly suggested by the data, has not been directly measured at matched checkpoints.

The positive results are concrete. At 7B parameters, linear prediction achieves 60–90% strict acceptance in stable regimes with no degradation at prediction depths up to 100 steps. Phase boundaries are synchronized

across 5 independent seeds to within ±50 steps. Final validation losses agree to within 1.53% coefficient of variation. These are properties of the optimization landscape, not of any particular training run.

Whether this structure can be exploited for practical speedup remains open. We have shown that the trajectory is predictable; we have not shown that predicting is cheaper than computing. The gap between acceptance rate and wall-clock speedup depends on implementation details we do not measure. But the existence of the structure is itself a contribution: it constrains theories of training dynamics and points toward a class of methods—regime-adaptive, verify-before-accept—that current training pipelines do not exploit.

Acknowledgments

The author thanks Amos Waterland for creating the ASC architecture and for formative discussions on trajectory-based speculative computation that seeded this research direction over a decade ago. The author contributed to the original ASC work at Harvard and is acknowledged in Waterland et al. [2014]. Brian Cremeans is acknowledged for discussions on eigenvalue sign toggling in Lorenz system dynamics and its analogy to training regime transitions. Claude (Anthropic) provided extensive assistance with experimental implementation, GPU infrastructure management, data analysis, and manuscript preparation. This work used GPU compute from vast.ai on NVIDIA A100 40GB and 80GB GPUs.

Code availability. All code, experiment scripts, and paper source are available at `https://github.com/jmcentire/leap-verify` (DOI: 10.5281/zenodo.18739387).

References

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations (ICLR)*, 2023.

Anonymous. Enhancing deep neural network training efficiency and performance through linear prediction. *Scientific Reports*, 14:15443, 2024.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

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 (ICLR)*, 2021.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred A Hamprecht. Essentially no barriers in neural network energy landscape. In *International Conference on Machine Learning (ICML)*, 2018.

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

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. In *arXiv preprint arXiv:1912.02757*, 2019.

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

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning (ICML)*, 2020.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Mario Geiger, Leonardo Petrini, and Matthieu Wyart. Chaotic dynamics are intrinsic to neural network training with SGD. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Lei Guan, Dongsheng Chen, et al. XGrad: Boosting gradient-based optimizers with weight prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Jinhyeok Jang and Woo-hun Han. Learning to boost training by periodic nowcasting near future weights. In *International Conference on Machine Learning (ICML)*, 2023.

Sagar V Kamarthi and Stefan Pittner. Accelerating neural network training using weight extrapolations. *Neural Networks*, 12(9):1285–1299, 1999.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

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 (ICLR)*, 2017.

Boris Knyazev, Maude Drozdzal, Graham W Taylor, and Adriana Romero. Accelerating training with neuron interaction and nowcasting networks. In *International Conference on Learning Representations (ICLR)*, 2025.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning (ICML)*, 2023.

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.

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 (NeurIPS)*, 2018.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations (ICLR)*, 2020.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations (ICLR)*, 2023.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Abhishek Sinha, Mausoom Sarkar, Avisek Mukherjee, and Balaji Krishnamurthy. Introspection: Accelerating neural network training by learning weight evolution. In *ICLR Workshop*, 2017.

Leslie N Smith. Cyclical learning rates for training neural networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. Blockwise parallel decoding for deep autoregressive models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Amos Waterland, Elaine Angelino, Ekin D Cubuk, Efthimios Kaxiras, Ryan P Adams, Jonathan Appavoo, and Margo Seltzer. Computational caches. In *Proceedings of the 6th International Systems and Storage Conference (SYSTOR)*. ACM, 2013. doi: 10.1145/2485732.2485750.

Amos Waterland, Elaine Angelino, Ryan P Adams, Jonathan Appavoo, and Margo Seltzer. ASC: Automatically scalable computation. In *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 575–590. ACM, 2014. doi: 10.1145/2541940.2541985.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, et al. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022.

Michael R Zhang, James Lucas, Geoffrey Hinton, and Jimmy Ba. Lookahead optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.