Can You See Where You're Going Before You Start Walking?

Landscape Probes for Predicting Fine-Tuning Outcomes

Landscape Probes Research

February 2026

Abstract

We ask a simple question: can cheap geometric measurements of a neural network's loss landscape predict whether a set of hyperparameters will work well for fine-tuning? The answer turns out to be *no* if you measure at initialization—and *yes* if you measure during early training. We ran 80 fine-tuning experiments on BERT-mini / SST-2 with 16 hyperparameter configurations across 5 random seeds, computing six landscape probes at seven points during training. The key finding: the Hessian trace measured at step 400 achieves $r = -0.52$ ($p < 10^{-6}$) correlation with final validation accuracy, and gradient-based probes become predictive as early as step 50. This report explains, from first principles, what we measured, why the first attempt produced nothing, and what the second attempt revealed about how the optimizer and the geometry of the loss landscape interact.

Contents

1 The Question

When you fine-tune a pretrained language model, you have to pick hyperparameters—learning rate, warmup schedule, weight decay. The usual approach is to try a bunch of combinations and see which one works best. This is expensive, and it feels like there should be a shortcut.

The idea is this: maybe the *shape* of the loss landscape near the starting point tells you something about which hyperparameters will work. If the landscape is very curved in some directions, maybe you need a smaller learning rate. If it's flat, maybe you can afford to be more aggressive.

This is a geometric intuition. The loss landscape is just a function from parameter space to a real number (the loss), and it has local properties you can measure—the gradient, the curvature (Hessian), the sharpness. These are what we call **landscape probes**: cheap geometric measurements that summarize the local neighborhood of the loss surface.

The question is: do these measurements predict anything about the final outcome?

2 What We Measured

We used six probes, each capturing a different geometric property. Let w be the model parameters, $L(w)$ the loss function, ∇L the gradient, and $H = \nabla^2 L$ the Hessian matrix.

2.1 Gradient-Based Probes

1. **Gradient norm** $\|\nabla L\|$: The magnitude of the gradient, averaged over 5 mini-batches. Tells you how steep the landscape is at the current point. A large gradient means there's a strong signal pushing the parameters in some direction.

2. **Gradient variance** $\mathrm{Var}(\|\nabla L\|)$: How much the gradient norm fluctuates across mini-batches. High variance means the landscape looks different depending on which data you sample—the optimizer is getting inconsistent signals.

3. **SAM sharpness**: Inspired by Sharpness-Aware Minimization. We perturb the parameters in the direction of steepest ascent and measure how much the loss increases:

$$\mathrm{SAM} = L\left(w + \rho \frac{\nabla L}{\|\nabla L\|}\right) - L(w), \quad \rho = 0.05$$

   This is a directional measure of sharpness. A large value means the landscape rises steeply when you move uphill.

2.2 Curvature-Based Probes

4. **Top Hessian eigenvalue** $\lambda_{\max}(H)$: The largest eigenvalue of the Hessian, estimated by power iteration (20 iterations). This is the sharpest direction of curvature—the direction in which the loss changes most rapidly.

5. **Hutchinson trace** $\mathrm{tr}(H)$: The trace of the Hessian, estimated using Hutchinson's stochastic estimator with 10 Rademacher random vectors:

$$\mathrm{tr}(H) \approx \frac{1}{k} \sum_{i=1}^{k} v_i^\top H v_i, \quad v_i \sim \text{Rademacher}$$

   The trace is the sum of all eigenvalues, so it measures the *total curvature* across all directions, not just the sharpest one.

6. **Loss value** $L(w)$: The cross-entropy loss itself, averaged over 5 mini-batches. At initialization this is essentially random; during training, it tracks the optimization trajectory.

The Hessian-based probes require second-order gradients, which means we had to disable Flash Attention and use the mathematical (non-fused) attention implementation during probe computation. This is a practical detail, but it matters: SDPA kernels don't support double-backward passes.

3 The Experimental Setup

- **Model**: `prajjwal1/bert-mini` (11M parameters)
- **Task**: SST-2 (binary sentiment classification, 67k training examples)
- **Optimizer**: AdamW with gradient clipping at $\|g\| = 1.0$
- **Training**: 3 epochs, linear warmup + decay schedule
- **Hardware**: RTX 3050 (4 GB VRAM)

The hyperparameter grid:

Parameter	Values
Learning rate	$\{10^{-5},\ 2\times10^{-5},\ 5\times10^{-5},\ 10^{-4}\}$
Warmup ratio	$\{0.0,\ 0.1\}$
Weight decay	$\{0.0,\ 0.01\}$

That gives $4 \times 2 \times 2 = 16$ configurations. We ran each configuration with 5 random seeds, giving **80 total training runs**.

For each run, probes were measured at **7 time points**: initialization, step 10, 25, 50, 100, 200, and 400. Each training run takes about 2,100 steps total (3 epochs × 67k examples / batch size 32), so step 400 is roughly 19% of the way through training.

4 Act I: The Null Result

Our first hypothesis was appealing: measure the landscape at initialization, before training starts, and use those measurements to predict which hyperparameters will work. If this worked, you could skip the expensive grid search entirely.

It didn't work. Not even close.

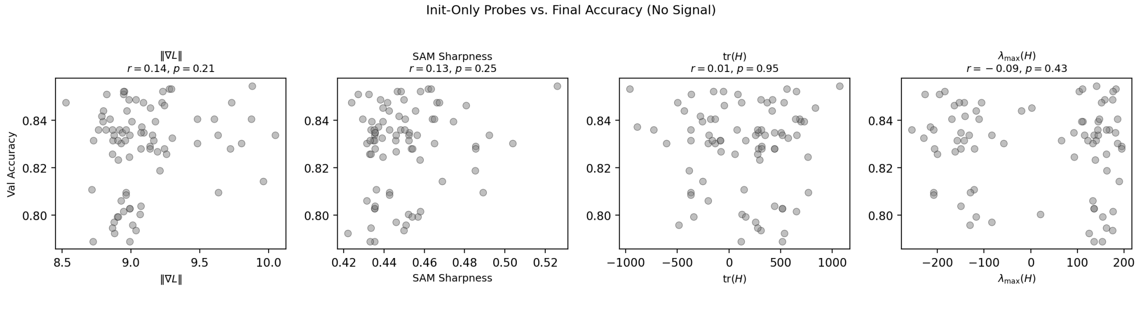

Figure 1: Init-only probes vs. final accuracy. Each point is one of the 80 training runs. There is no relationship—the clouds are shapeless.

The maximum absolute Spearman correlation between any init-time probe and final accuracy was $|r| = 0.18$, and *none* were statistically significant.

But this isn't a surprise once you think about it carefully. Within each random seed, all 16 hyperparameter configurations start from *the exact same model weights*. The probes at initialization measure properties of the starting point, and the starting point is identical regardless of whether you're about to use a learning rate of 10^{-5} or 10^{-4}.

The only variation in init-time probes comes from *across* seeds (different random initializations), and from mini-batch sampling noise in the probe computation itself. You're correlating measurement noise against hyperparameter-driven outcomes. Of course there's no signal.

This is an important negative result. It tells us that the static geometry of the pretrained model, by itself, doesn't determine fine-tuning success. What matters is the *interaction* between the geometry and the optimizer.

5 Act II: The Pivot

So we changed the question. Instead of measuring the landscape at one frozen point, we measured it *during training*—at steps 10, 25, 50, 100, 200, and 400.

Now each configuration is at a *different* point in parameter space. The optimizer has been pushing the parameters in different directions depending on the learning rate, warmup schedule, and weight decay. The probes are no longer measuring a shared starting point; they're measuring the evolving geometry of each run's unique trajectory.

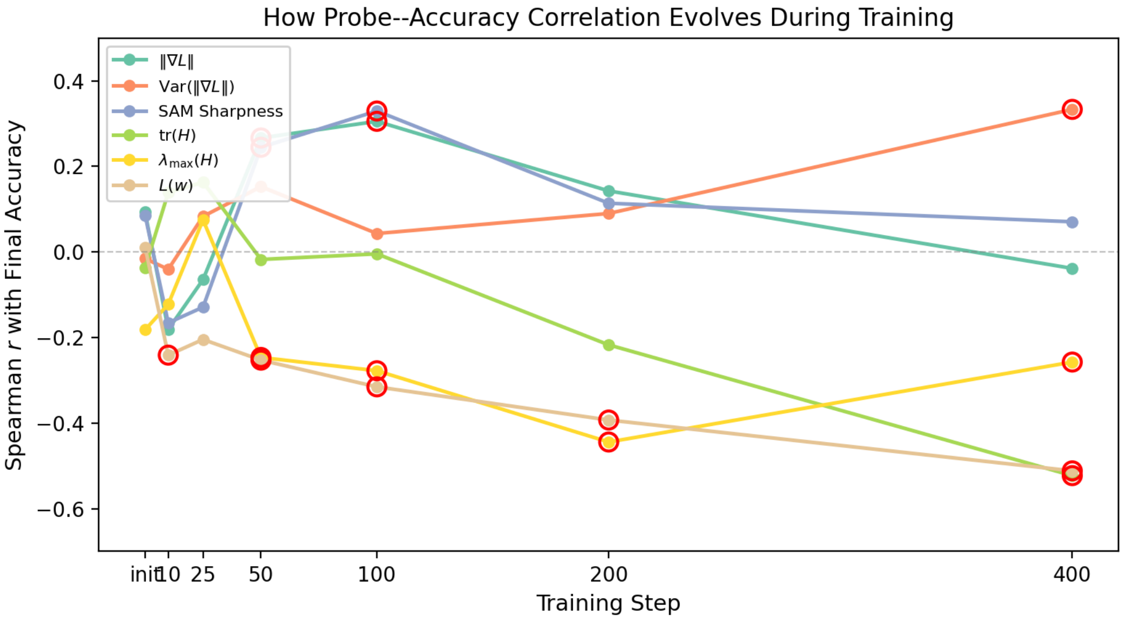

Figure 2: Spearman correlation between each probe and final accuracy, measured at different training steps. Red circles indicate statistical significance ($p < 0.05$). Signal emerges by step 50 and keeps growing for curvature probes through step 400.

The signal is unmistakable. Out of 42 probe–step combinations, **15/42** are statistically significant, with the strongest reaching $r = -0.52$.

6 The Results

6.1 The Full Correlation Table

Table 1: Spearman r between probes and final validation accuracy at each training step. $^{*}p < 0.05$, $^{**}p < 0.01$.

Probe	init	10	25	50	100	200	400
$\lVert\nabla L\rVert$	0.09	-0.18	-0.06	0.27*	0.31**	0.14	-0.04
SAM sharpness	0.09	-0.17	-0.13	0.24*	0.33**	0.11	0.07
$\lambda_{\max}(H)$	-0.18	-0.12	0.07	-0.25*	-0.28*	-0.44**	-0.26*
$\mathrm{tr}(H)$	-0.04	0.14	0.16	-0.02	-0.00	-0.22	-0.52**
$L(w)$	0.01	-0.24*	-0.20	-0.25*	-0.32**	-0.39**	-0.51**
$\mathrm{Var}(\lVert\nabla L\rVert)$	-0.02	-0.04	0.08	0.15	0.04	0.09	0.33**

6.2 Two Temporal Phases

The correlations don't just appear uniformly. There are two distinct phases, and they make physical sense.

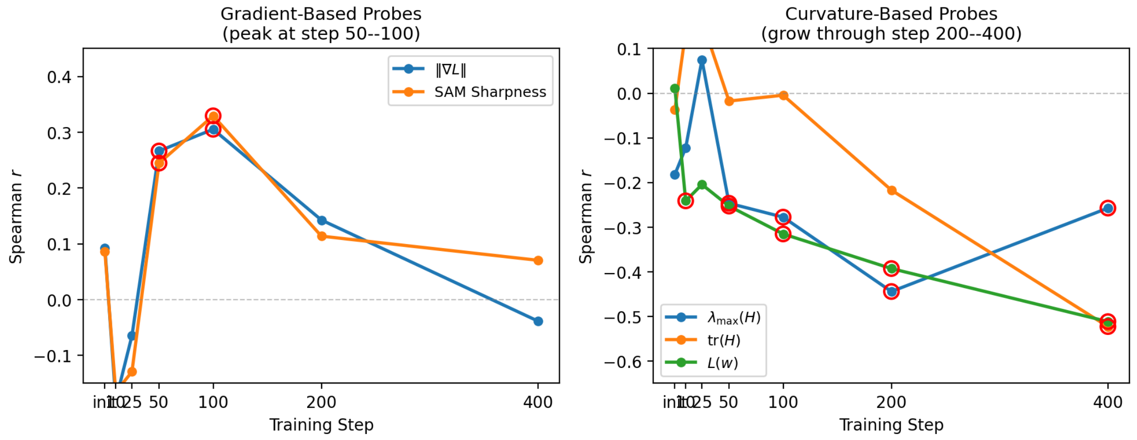

Figure 3: **Left**: Gradient-based probes peak at steps 50–100 then fade. **Right**: Curvature-based probes grow monotonically through step 200–400.

Phase 1 (steps 50–100): Gradient probes. The gradient norm and SAM sharpness become positively correlated with final accuracy. Configurations with larger gradients at step 100 tend to achieve higher accuracy. The interpretation: at this stage, a large gradient means the optimizer is still receiving a strong learning signal. Runs where the gradient has already collapsed are the ones that will underperform.

This signal peaks around step 100 then fades. By step 200, the gradient-based probes are no longer predictive. The optimizer has committed to a trajectory, and the gradient magnitude stops discriminating between good and bad runs.

Phase 2 (steps 200–400): Curvature probes. The top Hessian eigenvalue and Hutchinson trace become *negatively* correlated with accuracy. Higher curvature during training predicts worse outcomes. This directly connects to the **Edge of Stability** phenomenon: configurations that push into high-curvature regions of the landscape struggle to converge well.

The Hutchinson trace at step 400 is the single strongest predictor ($r = -0.52$, $p < 10^{-6}$). A model that has high total curvature 19% of the way through training is on a bad trajectory.

6.3 The Strongest Predictors

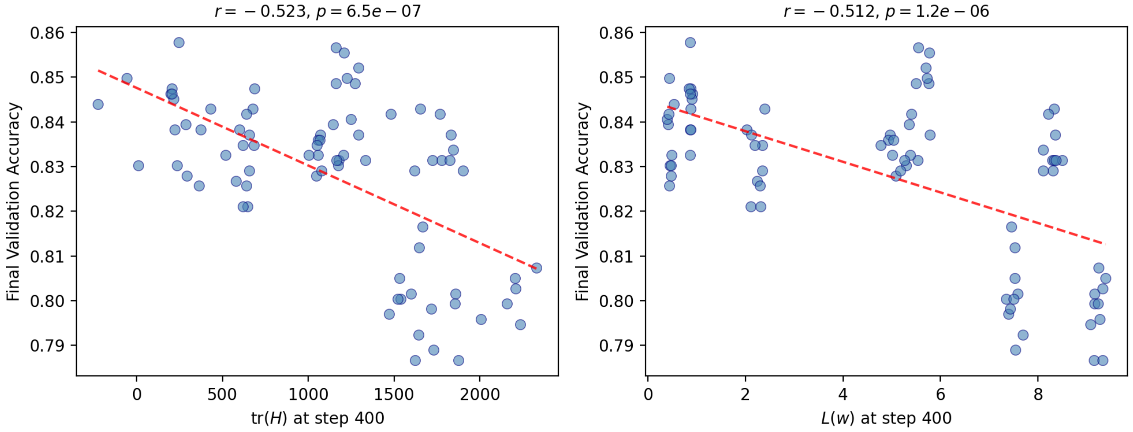

Figure 4: The two strongest probe–accuracy relationships: Hutchinson trace at step 400 (left) and loss at step 400 (right).

6.4 Accuracy Distribution

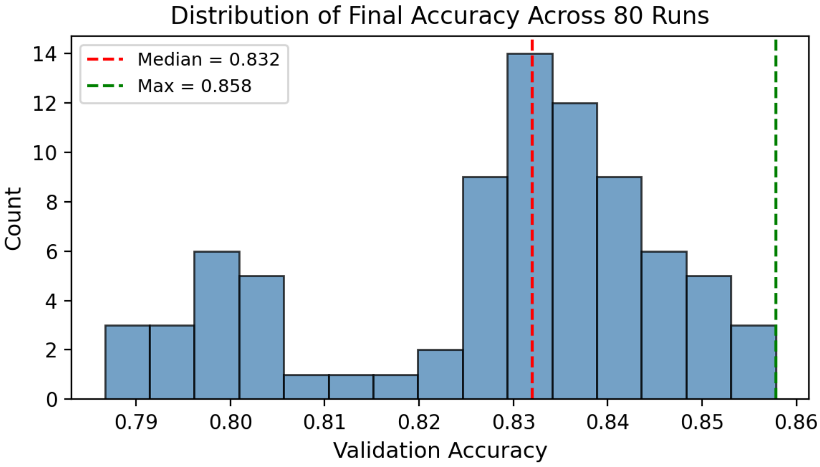

Figure 5: Distribution of final validation accuracy across all 80 runs. Median = 0.832, best = 0.858.

7 The Heatmap

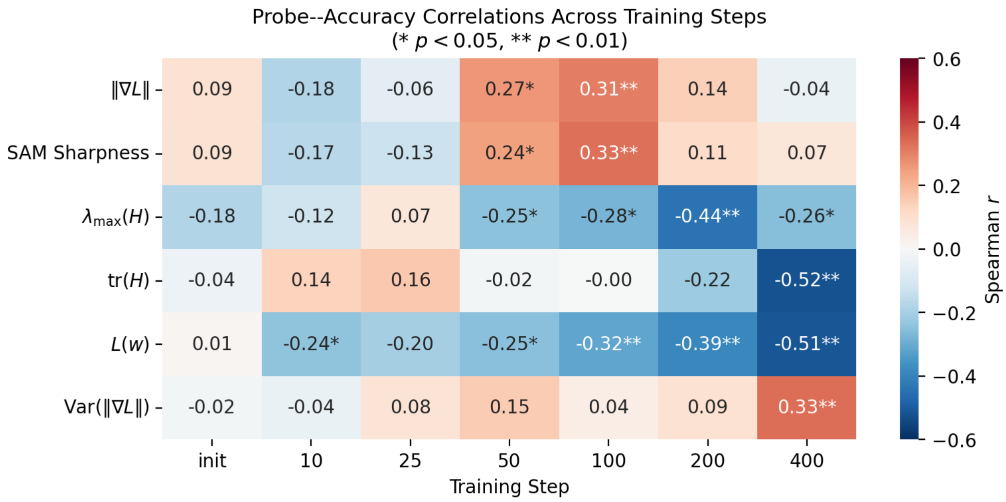

Figure 6: Complete heatmap of probe–accuracy correlations across all steps and probes.

8 What This Means

Let's step back and think about what the data is telling us.

8.1 The Landscape Is Not Destiny

The null result at initialization is not a failure—it's informative. A pretrained model sits at some point in parameter space, and that point has geometric properties. But those properties alone don't determine what happens during fine-tuning. What matters is the *path* you take through the landscape, and that depends on the optimizer and its hyperparameters.

This is like saying the elevation and slope of your starting position on a mountain don't determine whether you'll find the valley. The terrain matters, but so does which direction you walk and how big your steps are.

8.2 The Optimizer–Geometry Interaction

The multi-step probes capture something the init-only probes cannot: the evolving interaction between the optimizer and the landscape. By step 50, different hyperparameter configurations have diverged enough that the landscape looks different under each one. The probes are now measuring the *result* of that interaction, not just the static starting conditions.

Gradient probes peak early because they measure the immediate learning signal. By step 100, you can already tell which configurations are still learning productively.

Curvature probes grow late because the Hessian reflects the basin structure that the optimizer has found. By step 400, the total curvature (Hutchinson trace) almost perfectly separates good trajectories from bad ones.

8.3 Connection to Edge of Stability

The negative correlation between curvature and accuracy connects directly to the Edge of Stability (EoS) phenomenon. In EoS, the training loss decreases even as the top eigenvalue of the Hessian increases toward the stability threshold $2/\eta$. Configurations that overshoot this threshold oscillate and struggle to converge.

Our data shows that *total curvature* (not just the top eigenvalue) is the better predictor. This makes sense: the trace captures the curvature averaged across all directions, not just the sharpest one. A model can tolerate a single sharp direction if most other directions are flat; what kills performance is pervasive high curvature.

9 Limitations and Honest Assessment

Being honest about what we don't know:

1. **Single task, single model.** Everything here is BERT-mini on SST-2. The correlations might look completely different on a harder task or a larger model.
2. **Narrow hyperparameter range.** We only varied learning rate, warmup, and weight decay. The optimizer (AdamW), batch size (32), and architecture were fixed. A wider grid might reveal richer structure—or dilute the signal.
3. **Moderate effect sizes.** The strongest correlation is $r = -0.52$. That's real, but it means $\sim$27% of the variance is explained. There's a lot of noise left.
4. **Correlation, not causation.** We've shown that curvature *tracks* with bad outcomes. We haven't shown that high curvature *causes* poor convergence. It could be that both are downstream effects of some third factor (e.g., an inappropriate learning rate simultaneously causes high curvature *and* poor convergence).
5. **Probe cost.** The Hessian probes require second-order gradients, which roughly triples the memory and computation per step. If you're measuring probes at step 400 anyway, you've already spent 19% of training. The question is whether this 19% buys you enough information to avoid running the other 81% for bad configurations.

10 Where This Goes

The results suggest three concrete next steps:

1. **Early-exit hyperparameter selection.** Run all configurations for 100 steps, measure probes, discard the ones with bad geometric signatures, and only train the rest to completion. The data we already have can simulate this retroactively.
2. **Cross-task transfer.** Do the same landscape signatures predict good hyperparameters across different tasks? If Hutchinson trace is always the best late-training predictor, you might learn a universal "landscape quality" score.
3. **Cross-scale transfer.** Can you measure probes on a small model and use them to select hyperparameters for a large model? If the geometry transfers across model scales, you could do cheap probing on BERT-mini to configure BERT-large.

11 Conclusion

The story is simple. The landscape at initialization tells you nothing about where training will end up, because all configurations start from the same point. But by step 50–100, the optimizer has already begun to sculpt the landscape differently under each configuration, and cheap geometric probes can pick up on this. Gradient probes peak early (step 100) and capture the active learning signal. Curvature probes grow late (step 200–400) and capture the basin quality. The Hutchinson trace at step 400 is the single strongest predictor of final accuracy ($r = -0.52$), consistent with the Edge of Stability framework.

The immediate practical implication: you can predict which hyperparameter configurations will work by looking at the geometry of the loss landscape early in training. The open question is whether this transfers across tasks and model scales—and that's where the research goes next.