HIDDEN SENSITIVITY IN LLM PRE-TRAINING: A FINITE-TIME LYAPUNOV VIEW

Anonymous authors
Paper under double-blind review

ABSTRACT

Subtle differences in numerical precision, parameters, or optimizer state are unavoidable in LLM pre-training, yet loss curves reveal little about whether training will suppress or amplify them. We model AdamW pre-training as an extended-state dynamical system and introduce a directional, parameter-seeded normalized Adam-state finite-time Lyapunov exponent (FTLE). Batch-order changes are evaluated separately as exogenous forcing. Across a 60M LLaMA-style model and OLMo-1B continual pre-training, sensitive windows identified by FTLE produce persistent separation in optimizer state, gradients, logits, token predictions, generations, and individual benchmark decisions, even when aggregate loss and accuracy remain nearly unchanged. FTLE complements learning rate and update norm by adding information about the rate of trajectory separation. The pattern recurs across independent runs and model scales. Offline FTLE profiling further identifies intervention windows that reduce perturbation separation at a measurable optimization cost. At matched scheduled budget, FTLE-selected windows reduce separation by 7.0% and 7.3% relative to gradient- and update-norm-selected windows, respectively, with a 0.002 loss difference from the gradient baseline and lower loss than the update baseline. Finite-time perturbation amplification therefore provides a practical framework for measuring and controlling hidden sensitivity in LLM pre-training.

1 INTRODUCTION

Subtle discrepancies are an unavoidable part in LLM pre-training. A changed batch order, checkpoint resume, reduction order, or rounding pattern can perturb parameters or optimizer memory without visibly changing the loss curve. Whether that difference vanishes or grows is a property of the subsequent training dynamics, with direct consequences for checkpoint reproducibility and intervention comparisons. Nominally identical runs can reach different solutions after minute implementation-level changes (Summers & Dinneen, 2021; Shanmugavelu et al., 2024). We therefore ask: **when does LLM pre-training amplify subtle perturbations, and where does the resulting separation become observable?**

This question separates three notions often conflated by aggregate evaluation. Two runs may have similar validation loss and benchmark accuracy while their parameters and Adam moments follow different trajectories, and while their logits, token choices, and generations cease to be reproducible. We call these aggregate, dynamical, and behavioral stability, respectively. Measuring the latter two requires a local diagnostic alongside scalar performance summaries. This pathwise notion differs from algorithmic stability, which relates data changes to generalization (Hardt et al., 2016): here the data stream may be fixed while we measure amplification along one realized optimization trajectory.

Training-stability diagnostics commonly target visible loss spikes, divergent logits, or large gradient and update norms. These signals support controlled studies of Transformer instability and effective optimizer safeguards (Wortsman et al., 2024; Molybog et al., 2023; Huang et al., 2025; Bai et al., 2025). We study the complementary regime in which optimization appears healthy while nearby states separate rapidly, revealing sensitivity that remains hidden and requiring the local response of the update map rather than only the magnitude of its realized step.

We treat AdamW pre-training as a stochastic, non-autonomous dynamical system whose endogenous state contains parameters and optimizer memory (Kingma & Ba, 2015; Loshchilov & Hutter,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2019). Parameters alone do not determine the next iterate because stored moments retain gradient history and shape the preconditioned update. The extended state (θ, m, v) restores a one-step description; the schedule and sampled batch provide controls and forcing.

An FTLE then measures whether a subtle displacement contracts or expands over a specified training window. Because parameters and the two moments have incompatible scales, we introduce an optimizer-relative normalized Adam-state FTLE and validate it against paired nonlinear trajectories. It measures local neighborhood deformation over training, complementing learning rate, gradient norm, and update norm as a distinct view of trajectory sensitivity.

Our study connects measurement, consequence, generality, and control. On a dense 60M trajectory, FTLE reveals heterogeneous sensitivity beneath a smooth loss curve and adds information about log-distance growth beyond learning rate and update norm. Perturbations to parameters or either Adam moment persist for up to 2000 steps and propagate to token predictions, generations, and sample-level benchmark decisions despite stable aggregate performance. Independent runs and OLMo-1B continual pre-training reproduce this separation between performance and behavioral stability. Finally, FTLE-guided interventions expose a controllable stability–optimization frontier.

Our contributions are:

1. **Hidden sensitivity revelation.** We systematically characterize how small perturbations evolve during LLM pre-training, showing that sensitivity varies substantially across training windows and can persist into trajectory and behavioral differences even when aggregate performance remains close.
2. **Finite-time sensitivity measurement.** We formulate LLM pre-training as a stochastic, non-autonomous dynamical system with extended AdamW state (θ, m, v), and introduce an optimizer-aware normalized FTLE metric that identifies locally sensitive training windows and captures trajectory separation beyond conventional step statistics.
3. **Sensitivity-guided control.** We use an offline FTLE profile to schedule targeted AdamW stabilization in sensitive windows. These experiments expose a stability–optimization frontier and improve contraction over budget-matched gradient- and update-norm schedules. Extensive experiments across diverse experimental conditions consistently validate the effectiveness and robustness of our approach.

2 Related Work

Scaling laws and open checkpoint suites have made the trajectory of LLM pre-training an object of study in its own right (Kaplan et al., 2020; Brown et al., 2020; Biderman et al., 2023; Groeneveld et al., 2024). Work on optimization stability studies edge-of-stability behavior, Adam loss spikes, adaptive-preconditioner failures, and stabilization mechanisms (Cohen et al., 2021; Molybog et al., 2023; Huang et al., 2025; Bai et al., 2025). Numerical non-determinism and reproducibility studies further document that nominally matched runs can diverge (Henderson et al., 2018; Shanmugavelu et al., 2024; Fu et al., 2026; Yuan et al., 2025). We study local sensitivity inside otherwise normally optimizing runs and connect an infinitesimal diagnostic to trajectory and behavioral outcomes.

Dynamical-systems perspectives relate residual networks to discretized flows, motivate stable architectures and Neural ODEs, and analyze stochastic optimization (E, 2017; He et al., 2016; Haber & Ruthotto, 2017; Chen et al., 2018; Li et al., 2017). FTLE has also appeared in two distinct settings. For forward dynamics, it measures how input perturbations propagate through network depth (Storm et al., 2024). For training dynamics, prior work connects curvature to chaos in SGD, derives the tangent operator and training-time FTLE for small regression networks, and studies maximal Lyapunov exponents in large-learning-rate GD (Herrmann et al., 2022; Sharafi et al., 2025; Jiménez-González et al., 2026).

Our work studies stochastic, non-autonomous AdamW LLM pre-training in the extended state (θ, m, v), with optimizer-relative geometry and consequences traced to model behavior. Lyapunov functions have separately been used to establish descent and convergence properties of cautious Adam-style optimizers (Liang et al., 2026). We evaluate SPAM and cautious AdamW as optimizer-axis controls, then use scheduled control as an intervention test of FTLE-identified windows. The diagnostic is connected to nonlinear paired trajectories and behavioral reproducibility.

3 THEORY AND METHOD

3.1 LLM PRE-TRAINING AS AN EXTENDED-STATE DYNAMICAL SYSTEM

We use a dynamical-system formulation to describe the training process. A discrete-time dynamical system consists of a state s_t and an update map T_t that sends the current state to the next state:

$$s_{t+1} = T_t(s_t). \tag{1}$$

The sequence $(s_0, s_1, \ldots)$ is the trajectory of the system. In optimization, the state is often taken to be the parameter vector, and the update map is the optimizer step. For example, for full-batch gradient descent with parameters θ_t, learning rate η_t, and objective L, one may write

$$\theta_{t+1} = \theta_t - \eta_t \nabla L(\theta_t). \tag{2}$$

LLM pre-training is richer for two reasons. First, the gradient is computed on a mini-batch rather than the full data distribution. We denote this mini-batch gradient by $g_t = \nabla_\theta \ell(\theta_t; B_t)$, where B_t is the batch at step t. Second, AdamW carries optimizer memory. Let m_t be the first moment, v_t the second moment, β_1, β_2 the moment coefficients, λ_w the decoupled weight decay, and ϵ the numerical stabilizer. AdamW evolves as

$$m_{t+1} = \beta_1 m_t + (1 - \beta_1) g_t, \tag{3}$$

$$v_{t+1} = \beta_2 v_t + (1 - \beta_2)(g_t \odot g_t), \tag{4}$$

$$\theta_{t+1} = (1 - \eta_t \lambda_w)\theta_t - \eta_t \frac{\hat{m}_{t+1}}{\sqrt{\hat{v}_{t+1}} + \epsilon}, \tag{5}$$

where $\hat{m}_{t+1}$ and $\hat{v}_{t+1}$ are bias-corrected moments. The natural training state is the extended state

$$x_t = (\theta_t, m_t, v_t), \tag{6}$$

with deterministic controls α_t collecting the learning-rate schedule and optimizer hyperparameters, and exogenous forcing ξ_t collecting the realized batch stream, data order, numerical precision, and hardware/kernel behavior. A realized pre-training run is therefore the trajectory of the stochastic, non-autonomous map

$$x_{t+1} = T(x_t; \alpha_t, \xi_t). \tag{7}$$

3.2 OPTIMIZER-AWARE FTLE METRICS AND CONTROL

Given a perturbation δx_t near a training state, the local linearized dynamics over K steps gives

$$\delta x_{t+K} \approx J_{t+K-1} \cdots J_t \delta x_t. \tag{8}$$

Here $J_t = \frac{\partial T}{\partial x}(x_t; \alpha_t, \xi_t)$ is the Jacobian of one optimizer step with respect to the endogenous extended state. The finite-time Lyapunov exponent (FTLE) is

$$\lambda_{t,K} = \frac{1}{K} \log \frac{\|\delta x_{t+K}\|}{\|\delta x_t\|}. \tag{9}$$

Positive $\lambda_{t,K}$ indicates amplification over the finite window.

AdamW requires a scale-aware norm. Raw distances over (θ, m, v) mix quantities with different units and can be dominated by the largest-scale state variable. We use a normalized Adam-state FTLE as our main metric.

Normalized Adam-state FTLE. We measure perturbation growth in a relative extended-state norm expressed in coordinates that govern the AdamW step: relative parameter displacement, relative preconditioned-update displacement, and the logarithmic perturbation of the diagonal preconditioner. Schematically,

$$\|\delta x\|^2_{\mathrm{rel}} = w_\theta \left\| \frac{\delta\theta}{\theta} \right\|^2_{\mathrm{layer}} + w_u \left\| \frac{\delta u}{u} \right\|^2_{\mathrm{layer}} + w_\rho \|\delta \log(\sqrt{\hat{v}} + \epsilon)\|^2_{\mathrm{RMS}}, \tag{10}$$

where ratios are implemented with layer-wise RMS values and numerical floors; Appendix G gives the coordinate map explicitly. The reported diagnostic initializes a normalized Gaussian direction in

the parameter block, $q_t = (q_{\theta,t}, 0, 0)$, and propagates it through the full AdamW tangent map. The normalized FTLE is then

$$\lambda_{t,K}^{\text{state}} = \frac{1}{K} \log \frac{\|\delta(\theta, m, v)_{t+K}\|_{\text{rel}}}{\|\delta(\theta, m, v)_t\|_{\text{rel}}}. \tag{11}$$

Intuitively, $\lambda_{t,K}^{\text{state}} > 0$ means that a small relative perturbation to the training state is amplified over the next K optimizer steps. We also compute raw Adam-state FTLE and Adam-update FTLE as ablations in the experiments, but the normalized Adam-state FTLE is the metric used for ranking sensitive checkpoints and for driving control.

Lyapunov-aware AdamW control. The same quantities can be used as control signals. Let $a_t = \max(0, \lambda_{t,K}^{\text{state}} - \tau)$ denote the excess FTLE above a threshold. Our mild controller rescales the effective learning rate and damps the first moment in sensitive windows,

$$\eta_t^{\text{eff}} = \eta_t \max\left(c_{\min}, \frac{1}{1+\gamma a_t}\right), \qquad m_t \leftarrow \max(0.05, 1-\gamma_m a_t) m_t. \tag{12}$$

Our internal controller acts more directly on AdamW's preconditioned update. In high-FTLE windows it slows the adaptation of the second moment, damps the preconditioner, and caps the update-to-weight ratio:

$$1 - \beta_{2,t}^{\text{eff}} = \frac{1-\beta_2}{1+\gamma_v a_t}, \tag{13}$$

$$P_t^{\text{eff}} = \frac{1}{1+\gamma_p a_t} \operatorname{diag}\left((\sqrt{\hat{v}_{t+1}} + \epsilon_t)^{-1}\right), \tag{14}$$

$$\frac{\|\eta_t^{\text{eff}} P_t^{\text{eff}} \hat{m}_{t+1}\|}{\|\theta_t\|} \le c_u. \tag{15}$$

The controller holds the loss, architecture, and data fixed while modifying the optimizer dynamics in FTLE-selected windows. It directly tests whether the diagnosed sensitivity is manipulable.

4 Experiments

4.1 Settings

We train a 60M LLaMA-style model on C4 for 5000 AdamW steps and continue pre-training OLMo-1B (Groeneveld et al., 2024) on Dolma (Soldaini et al., 2024) for 2000 steps. The 60M run supports dense checkpoint, perturbation-channel, amplitude, horizon, replication, and control studies; OLMo tests transfer to a second model/data system. We perturb parameters, Adam first or second moments, and batch order, then measure relative distances in optimizer state, updates, parameters, and gradients. Functional readouts include loss, logits KL, token disagreement, generation, and PIQA, HellaSwag, and ARC-Easy. Normalized FTLE uses a three-step frozen-local directional probe for 60M and a one-step instantaneous directional probe for OLMo-1B. Appendix **??** gives the hyperparameters, computational details, and all 348 paired trajectories.

Diagnostic sweeps leave the baseline trajectory unchanged. Response sweeps then restart paired branches from one checkpoint, perturb a single state component, and replay the same batches, isolating amplification by the training dynamics. Control runs replay piecewise-constant schedules constructed from an offline baseline profile and are compared at matched intervention budget. High and low checkpoints are selected within each run; independently trained trajectories provide the replications.

4.2 FTLE Reveals Hidden Sensitive Windows

The 60M baseline optimizes normally: validation loss falls from 5.48 to 3.92. Along the same trajectory, normalized Adam-state FTLE varies nearly sevenfold, from 0.163 at step 4000 to 1.116 at step 2000 (Figure 1). Update-space FTLE follows the same temporal structure, whereas raw-state FTLE is dominated by incompatible parameter and moment scales. The empirical response also changes sharply with checkpoint: smooth loss therefore coexists with heterogeneous local neighborhoods.

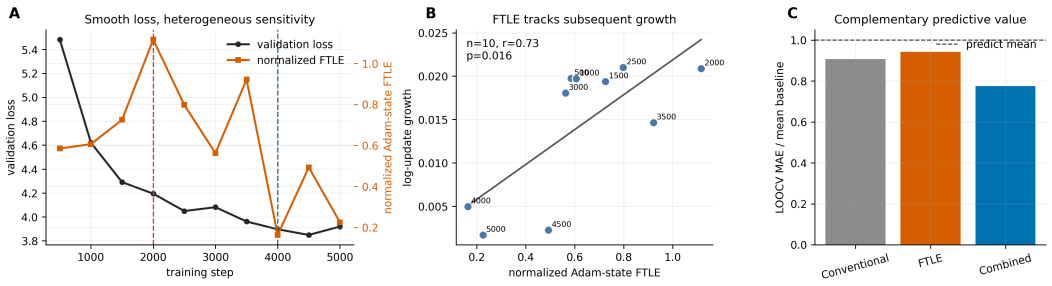

Figure 1: FTLE reveals information hidden by the loss curve. **A**: validation loss varies smoothly while normalized Adam-state FTLE identifies high- and low-sensitivity windows. **B**: FTLE tracks subsequent empirical log-update growth. **C**: adding FTLE to learning rate, update norm, gradient norm, and curvature improves nested-ridge leave-one-checkpoint-out growth prediction.

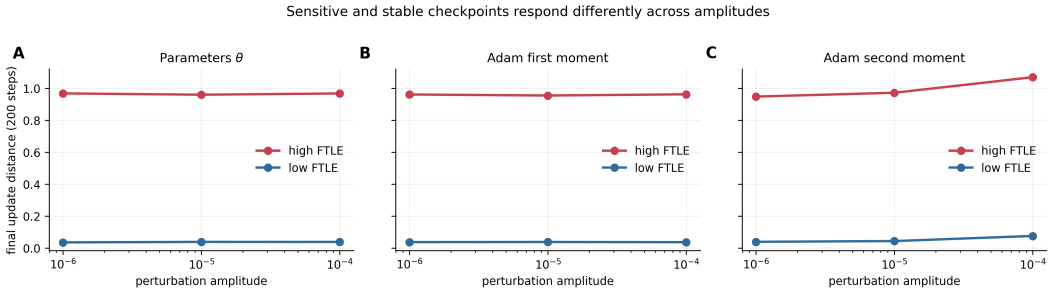

Figure 2: High- and low-FTLE checkpoints respond differently across perturbation amplitudes. Curves report 200-step update separation after perturbing **A** parameters, **B** Adam first moments, or **C** Adam second moments. The sensitive-window contrast persists across three orders of magnitude.

Learning rate and update norm strongly predict both final response magnitude and log-update growth. FTLE has $r = 0.731$ with log-update growth and a partial correlation of $r = 0.692$ after removing learning rate and update norm (two-sided $p = 0.057$, six degrees of freedom). Nested leave-one-checkpoint-out ridge regression reduces growth MAE by 14.5% after adding FTLE (bootstrap 95% CI: -3.3% to 30.7%; paired sign-flip $p = 0.143$) and raises descriptive R^2 from 0.10 to 0.33. These ten checkpoints support incremental predictive evidence with substantial uncertainty (Figure 1C). Normalized state and update-space FTLE share the same temporal structure; the raw-state ablation fails because a Euclidean norm mixes parameters, gradients, and squared gradients. Complete metric and channel comparisons appear in Appendix E.

4.3 Trajectory Separation Becomes Behavioral

Optimizer memory is an active perturbation channel. At $\epsilon = 10^{-5}$, the mean 200-step update distances are 0.505, 0.507, and 0.529 after perturbing θ, m, and v, respectively. At 1000 steps, perturbations launched from the high-FTLE checkpoint reach parameter distances of 0.39–0.40 and gradient distances near 0.85; from the low-FTLE checkpoint they remain below 0.015 and near 0.03. The contrast persists through 2000 steps.

The response is also nonlinear in perturbation amplitude (Figure 2). Early growth retains checkpoint and channel structure, while later distances show threshold transitions and saturation. Parameter and first-moment responses often saturate near unit update distance; for Adam-v, mean parameter distance rises from 0.113 at 10^{-6} to 0.171 at 10^{-5} and 0.818 at 10^{-4}. FTLE therefore characterizes a local growth regime, while finite perturbations reveal how that regime transitions into nonlinear trajectory separation.

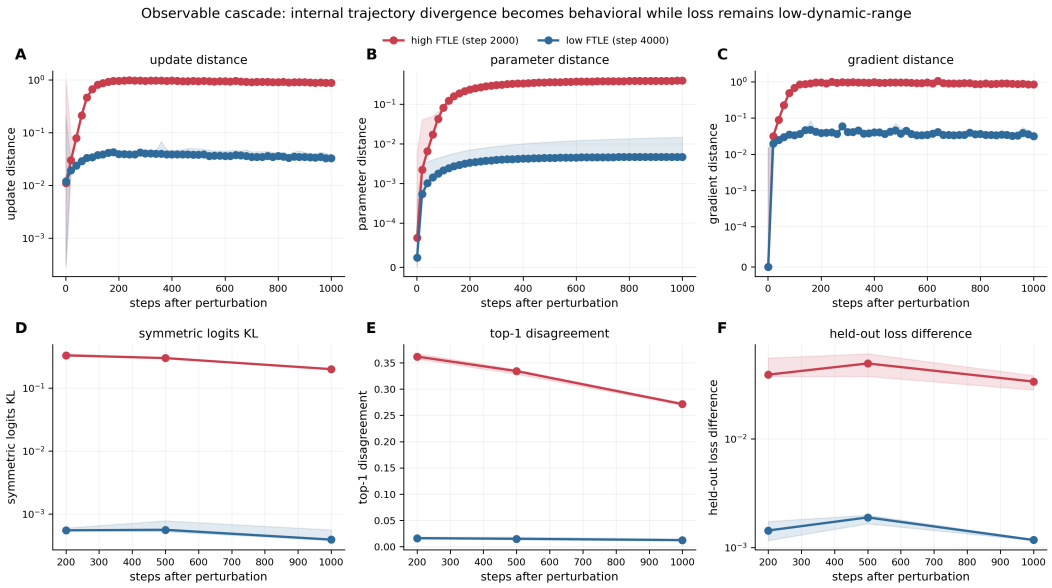

Figure 3: Perturbation amplification becomes behavioral while aggregate performance remains stable. **A–C**: update, parameter, and gradient separation from high- and low-FTLE checkpoints. **D–F**: logits KL and token disagreement remain high in the sensitive regime even as held-out loss changes little. Appendix E reports generation and benchmark outcomes for every condition.

Across the full 90-run endogenous response atlas, update and parameter distances share this checkpoint structure while loss differences occupy a much narrower dynamic range. Batch-order changes are stronger because they continually alter the exogenous forcing. The complete matrix is shown in Figure 6.

Figure 3 traces this separation into function space. At 1000 steps, high-FTLE perturbations produce symmetric logits KL near 0.20, top-1 disagreement near 27%, and top-5 overlap near 0.75. Low-FTLE perturbations remain near $4\text{–}6 \times 10^{-4}$ KL, 1–1.5% disagreement, and 0.987 overlap. Yet the aggregate held-out loss difference stays small. Across all systems, token-choice, generation, and sample-level benchmark decisions change much more than average loss or accuracy. Performance stability therefore hides behaviorally distinct models with distinct behaviors.

The functional readouts expose complementary consequences: KL measures vocabulary redistribution, top-k overlap candidate-set churn, and generation compounded token changes. Per-example benchmark disagreement is kept separate from average accuracy because two models can exchange which examples they solve while retaining the same score, concealing a change in the learned input–output function.

4.4 Replication and Scale Transfer

Sensitive windows recur without occupying a universal training step. Two independent 60M runs yield high/low median update-distance contrasts of 0.957/0.021 and 0.236/0.022. A $0.8\times$-learning-rate AdamW run also retains a 0.783/0.160 contrast, showing that temporal sensitivity remains under a modest global step reduction (Appendix Figure 9). Across three seeds, the location and magnitude of peak sensitivity change, but the within-run high/low contrast is preserved (Table 1). The diagnostic is stable across tangent seeds conditional on a batch; changing the diagnostic batch produces larger variation, as expected for a local map driven by stochastic data.

OLMo-1B provides a qualitative transfer test in a second model/data/training regime. Instantaneous model-parallel directional FTLE varies across eight checkpoints, and 28 perturbation conditions evaluated at both 200 and 1000 steps reproduce the gap between aggregate and behavioral stability (Figure 4). Mean loss difference remains below 6×10^{-4} while top-1 disagreement is about

Table 1: Replication across independent 60M runs. Responses summarize θ, m, v perturbations at run-specific high/low-FTLE checkpoints.

Run	High step	Low step	High resp.	Low resp.	Ratio
Seed 1	2000	4000	0.960	0.039	24.7
Seed 2	1500	5000	0.957	0.021	46.0
Seed 3	3500	5000	0.236	0.022	11.0
Mean ± SD	–	–	0.718±.417	0.027±.010	27.2±17.7

Table 2: OLMo-1B behavior over 28 perturbation conditions. Benchmark entries are prediction disagreement.

Observable	200 steps	1000 steps
Loss difference	5.61×10^{-4}	2.30×10^{-4}
Symmetric KL	8.75×10^{-4}	8.32×10^{-4}
Top-1 disagreement	2.27%	2.30%
Generation match	65.7%	63.6%
PIQA disagreement	0.23%	0.45%
HellaSwag disagreement	1.48%	1.25%
ARC-Easy disagreement	0.91%	1.02%

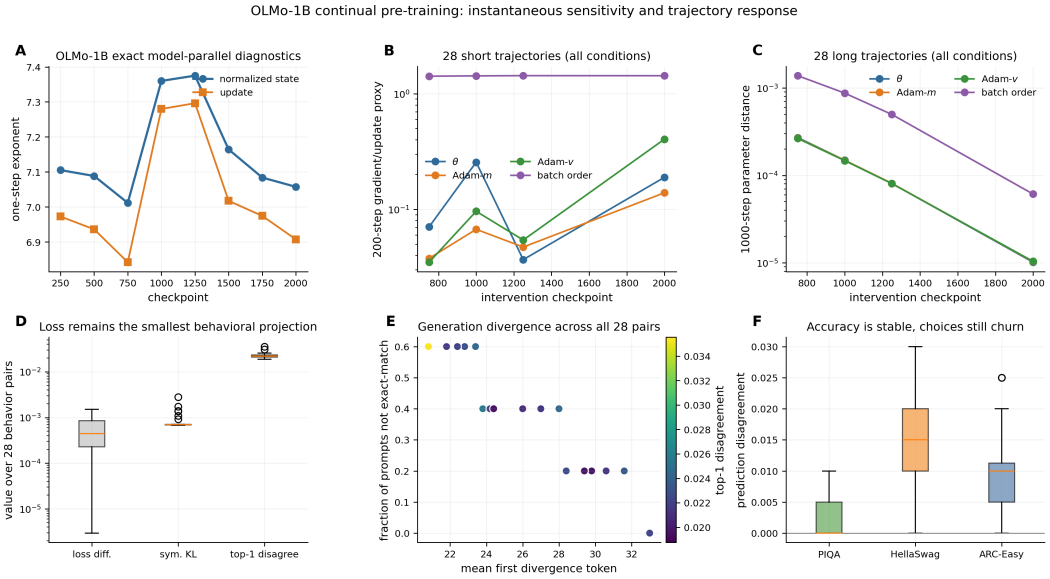

Figure 4: Scale transfer to OLMo-1B. **A**: exact model-parallel FTLE across eight checkpoints. **B–C**: short- and long-horizon trajectory response. **D–F**: token, generation, and benchmark decisions diverge while aggregate performance remains close.

2.3%. Generation exact match falls from 65.7% to 63.6%, and long-horizon prediction disagreement reaches 0.45%, 1.25%, and 1.02% on PIQA, HellaSwag, and ARC-Easy, respectively, while mean absolute accuracy changes remain below 0.75 percentage points. Parameter distance and continuous logit divergence contract for several channels between 200 and 1000 steps, while discrete token, generation, and benchmark disagreements persist. This later contractive regime coexists with decisions changed earlier in the continuation.

4.5 State-Dependent Stability Control

We next use an offline FTLE profile to schedule when AdamW stabilization is activated. Three variants act on effective step size, first-moment memory, second-moment adaptation, preconditioning, and update-to-weight ratio. Basic control reduces separation by 10–13% with little loss cost; stronger internal control reduces mean update distance from about 0.52 to 0.14 while increasing validation loss from 3.919 to 4.143. The intervening operating points form a clear stability–optimization frontier (Figure 5).

Table 3: Functional stability 200 steps after a 10^{-5} parameter perturbation in a high-sensitivity window. sKL, T1 Dis, Gen. EM, Bench. Dis denote symmetric KL, top-1 disagreement, generation exact match, and benchmark disagreement.

Method	sKL ↓	T1 Dis. ↓	Gen. EM ↑	Bench. Dis. ↓
AdamW	0.319993	35.71%	0.8%	16.21%
C-AdamW	0.372854	38.55%	0.0%	18.45%
SPAM	0.307452	34.89%	0.0%	15.31%
Gradient schedule	**0.000153**	**0.76%**	80.5%	0.13%
FTLE schedule	0.000167	0.83%	**88.3%**	**0.07%**

Gradient- and update-norm schedules isolate window selection. Each receives the same four-interval activation-strength multiset as the FTLE schedule (scheduled mean activation 0.0759; mean learning-rate scale 0.976). Their mean update distances are 0.244 and 0.245, versus 0.227 for FTLE—7.0% and 7.3% additional contraction. FTLE and gradient selection give final losses of 4.030 and 4.028, respectively, defining neighboring trade-off points. FTLE selection has both lower separation and lower loss than update-norm selection (4.088).

The contraction reaches model outputs. Table 3 compares the same 10^{-5} parameter-perturbation protocol after 200 steps across AdamW, two Adam-family stabilizers, and our matched-budget controllers. The baseline rows use each optimizer's highest-FTLE checkpoint (step 2000 for AdamW and C-AdamW; 1750 for SPAM), while both controllers are evaluated at the shared AdamW-sensitive step 2000. SPAM and C-AdamW retain 34.9–38.6% token churn, 15.3–18.5% benchmark-decision churn, and no exact 64-token continuation among 128 prompts. Both targeted schedules nearly eliminate token and benchmark churn; FTLE selection further raises greedy exact match from 80.5% to 88.3% relative to gradient selection. Generation exact match compounds correlated rollout errors, so its ordering need not match averaged one-step metrics.

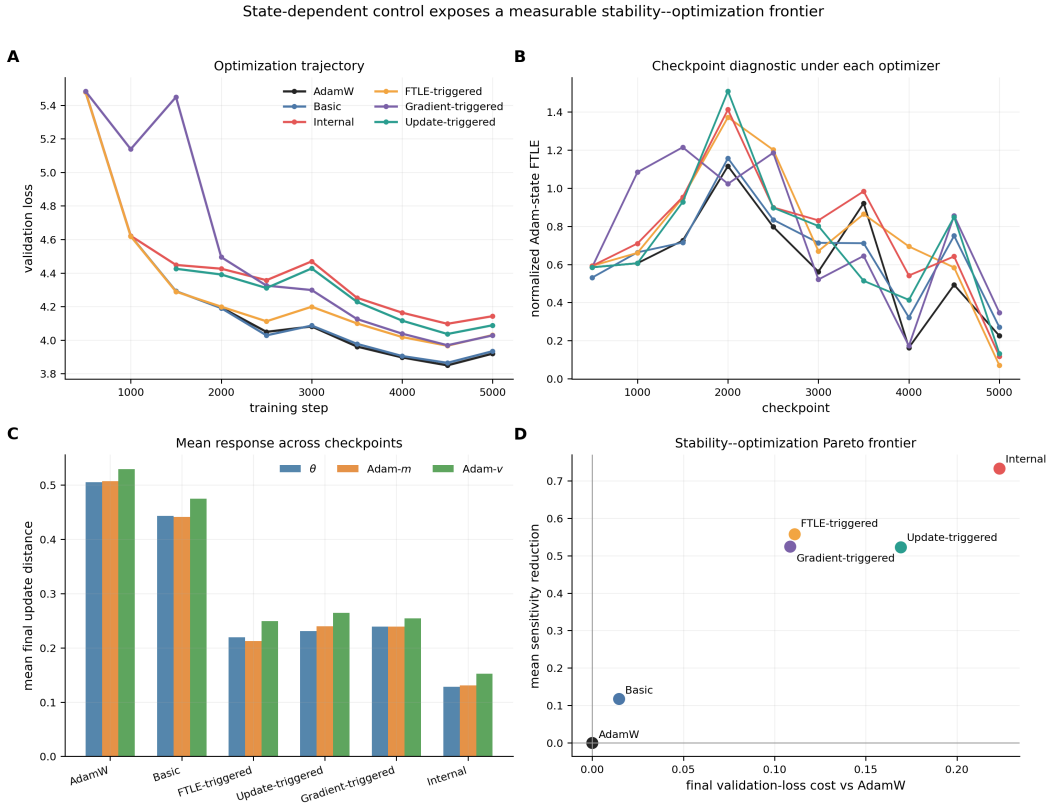

Figure 5: FTLE-guided control exposes a stability–optimization frontier. **A–B**: optimization and FTLE trajectories under each controller. **C**: mean response across perturbation channels. **D**: sensitivity reduction versus validation-loss cost; FTLE selection yields greater contraction than both matched schedules at nearly the same loss as gradient selection.

5 Discussion

Adam moments are part of the training state: perturbing them can alter subsequent updates as strongly as perturbing parameters, then change logits, predictions, and continuations while averaging out in loss or accuracy. SPAM and C-AdamW retain high-window update distances of 1.045

and 1.190 (Table 4). The control frontier makes stability a measurable design objective with an optimization cost.

Table 4: Matched optimizer and controller comparisons. Panel A reports 5k-step runs and mean θ, m, v response at optimizer-specific high/low-FTLE checkpoints. Panel B reports ten-checkpoint mean response for AdamW and budget-matched controllers. Lower is better.

A. Adam-family optimizer baselines				
Method	Val. loss	Grad. norm	Normalized FTLE range	Response (high/low)
AdamW	3.919	0.287	$[0.163, 1.116]$	$0.962/0.040$
C-AdamW	3.932	0.313	$[0.596, 1.524]$	$1.190/0.088$
SPAM	**3.884**	**0.187**	$[0.024, 0.839]$	$1.045/0.594$
B. Matched-budget window-selection controls				
Method	Val. loss	θ response	Adam-m response	Adam-v response
AdamW	**3.919**	0.505	0.507	0.529
Gradient schedule	4.028	0.239	0.239	0.254
Update schedule	4.088	0.231	0.240	0.264
FTLE schedule	4.030	**0.219**	**0.213**	**0.249**

SPAM lowers the post-warmup mean gradient norm by 35%, yet its high-FTLE θ continuation yields 34.9% token disagreement and 15.3% task-mean benchmark disagreement. C-AdamW yields 38.6% and 18.5%, respectively. All 128 greedy generations diverge for both optimizers, while per-task accuracy changes by at most 0.22 and 0.60 percentage points. Spike control and trajectory reproducibility are therefore distinct optimizer properties.

Computational cost. On one RTX 3090, an AdamW step takes 71.54 ± 0.55 ms versus 86.56 ± 0.10 ms with the controller (21.0% overhead); both allocate 5.80 GB. A three-step FTLE probe takes 1.002 ± 0.007 s and 11.61 GB, or 14.0 AdamW steps. At a 500-step measurement interval this adds 2.8% amortized time. The comparisons match intervention budget; compute matching would also charge probe cost. Micro-batching and rematerialization can reduce peak memory at larger scale.

Matched-loss checkpoints can disagree on low-margin tokens, continuations, or benchmark examples. Such cancellations preserve aggregate scores while altering deployment behavior, model comparison, and mechanistic analysis, so trajectory sensitivity measures learned-function reproducibility under routine perturbations. Learning rate and update norm summarize realized movement and predict much of the final separation; FTLE instead probes local propagation through curvature, moment memory, and preconditioner geometry. Step statistics measure movement, while FTLE measures how a neighborhood deforms. Diagnostic scope and limitations are detailed in Appendix B.

6 Conclusion

Optimizer-aware FTLE identifies sensitive windows before scalar loss does. Across seeds, scales, and three Adam-family optimizers, extended-state expansion predicts finite separation in updates and behavior. Parameters and optimizer moments are jointly dynamical: small perturbations can change tokens, continuations, and benchmark decisions while aggregate performance stays close. FTLE-guided interventions reduce this growth at a measured optimization cost, making trajectory reproducibility measurable and controllable. Settings, coverage, derivations, and complete experimental code are provided in the appendices and supplement.

Reproducibility. The anonymous supplement contains all project-owned source code, paper configurations, fixed prompt sets, environment versions, tests, and commands for data preparation, training, FTLE measurement, perturbation, control, behavioral evaluation, and figure generation. It excludes licensed datasets and generated checkpoints; both are recreated through documented entrypoints using relative paths. JSONL/CSV intermediates expose the quantities used in every aggregate.

REFERENCES

Zhiwei Bai, Zhangchen Zhou, Jiajie Zhao, Xiaolong Li, Zhiyu Li, Feiyu Xiong, Hongkang Yang, Yaoyu Zhang, and Zhi-Qin John Xu. Adaptive preconditioners trigger loss spikes in adam. *arXiv preprint arXiv:2506.04805*, 2025.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. *International Conference on Machine Learning*, 2023.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020.

Ricky T Q Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018.

Jeremy M Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. *International Conference on Learning Representations*, 2021.

Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 2017.

Tairan Fu, Gonzalo Martínez, Javier Conde, Carlos Arriaga, Pedro Reviriego, Xiuyuan Qi, and Shanshan Liu. Beyond reproducibility: Token probabilities expose large language model nondeterminism. *arXiv preprint arXiv:2601.06118*, 2026.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*, 2024.

Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2017.

Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pp. 1225–1234, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. *AAAI Conference on Artificial Intelligence*, 2018.

Luis M. Herrmann, Maximilian Granz, and Tim Landgraf. Chaotic dynamics are intrinsic to neural network training with SGD. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Tianjin Huang, Ziquan Zhu, Gaojie Jin, Lu Liu, Zhangyang Wang, and Shiwei Liu. Spam: Spike-aware adam with momentum reset for stable llm training. *arXiv preprint arXiv:2501.06842*, 2025.

Pedro Jiménez-González, Miguel C. Soriano, and Lucas Lacasa. Leveraging chaotic transients in the training of artificial neural networks. *Physical Review Research*, 8:L022032, 2026. doi: 10.1103/t5p9-kv5w.

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.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, pp. 2101–2110, 2017.

Kaizhao Liang, Lizhang Chen, Bo Liu, and Qiang Liu. Cautious optimizers: Improving training with one line of code. In *International Conference on Learning Representations*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations*, 2019.

Igor Molybog, Peter Albert, Moya Chen, Zachary DeVito, David Esiobu, Naman Goyal, Punit Singh Koura, Sharan Narang, Andrew Poulton, Ruan Silva, et al. A theory on adam instability in large-scale machine learning. *arXiv preprint arXiv:2304.09871*, 2023.

Sanjif Shanmugavelu, Mathieu Taillefumier, Christopher Culver, Oscar Hernandez, Mark Coletti, and Ada Sedova. Impacts of floating-point non-associativity on reproducibility for hpc and deep learning applications. *arXiv preprint arXiv:2408.05148*, 2024.

Nahal Sharafi, Christoph Martin, and Sarah Hallerberg. Weight dynamics of learning networks. *Physical Review E*, 111:054208, 2025. doi: 10.1103/PhysRevE.111.054208.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Raghavi Chandu, Jennifer Dumas, Yanai Elazar, et al. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*, 2024.

L. Storm, H. Linander, J. Bec, K. Gustavsson, and B. Mehlig. Finite-time lyapunov exponents of deep neural networks. *Physical Review Letters*, 132:057301, 2024. doi: 10.1103/PhysRevLett.132.057301.

Cecilia Summers and Michael J. Dinneen. Nondeterminism and instability in neural network optimization. In *International Conference on Machine Learning*, pp. 9913–9922, 2021.

Mitchell Wortsman, Peter J. Liu, Lechao Xiao, Katie E. Everett, Alexander A. Alemi, Ben Adlam, John D. Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, Jeffrey Pennington, Jascha Sohl-Dickstein, Kelvin Xu, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Small-scale proxies for large-scale transformer training instabilities. In *International Conference on Learning Representations*, 2024.

Jiayi Yuan, Hao Li, et al. Understanding and mitigating numerical sources of non-determinism in large language model inference. *arXiv preprint arXiv:2506.09501*, 2025.

A Experimental Settings

For the expanded control evaluation, generation prompts are a seed-2027 stratified sample of 43 PIQA contexts, 43 HellaSwag contexts, and 42 ARC-Easy contexts with answer choices and labels removed. Greedy decoding produces 64 new tokens. Multiple-choice evaluation uses length-normalized candidate sequence log-likelihood on every validation example. We report exact flip counts and Wilson intervals in the accompanying JSONL files.

B Scope and Limitations

FTLE is a parameter-seeded directional diagnostic conditional on a local batch; ten dense 60M checkpoints yield wide predictive uncertainty. The 60M $K = 3$ frozen-local probe measures repeated checkpoint-map expansion, whereas OLMo-1B $K = 1$ measures instantaneous expansion during mature BF16/AdamW8bit continual pre-training and establishes qualitative cross-system transfer. Batch-order interventions receive a discrete exogenous-forcing analysis, and the offline-profiled 60M schedules establish manipulability of diagnosed windows.

Table 5: Model and training configurations. Effective batches include gradient accumulation. Both schedules use linear warmup followed by cosine decay to 0.1 of the peak learning rate.

Configuration	60M trajectory	OLMo-1B continual pre-training
Checkpoint / architecture	8 layers, $d = 512$, 8 heads, MLP 1376	OLMo-2-0425-1B; 16 layers, $d = 2048$, 16 heads, MLP 8192
Implemented parameters	76.8M	1.485B
Data / context	C4 / 256	Dolma v1.6 sample / 1024
Microbatch × accumulation	16×32 (131,072 tokens/step)	1×32 (32,768 tokens/step)
Steps / warmup	5000 / 1000	2000 / 100
Peak LR / minimum ratio	8×10^{-4} / 0.1	2×10^{-5} / 0.1
Adam $(\beta_1, \beta_2, \epsilon, \lambda_w)$	$(0.9, 0.999, 10^{-8}, 0.1)$	$(0.9, 0.95, 10^{-8}, 0.1)$
Precision / optimizer	BF16 AMP / AdamW	BF16 / bitsandbytes AdamW8bit
Gradient clipping	1.0	1.0

Table 6: Controller constants. The score at checkpoint t is held over the following 500-step interval. The FTLE schedule is active on intervals beginning at steps 1500, 2000, 2500, and 3500; matched baselines permute the same effective score multiset according to gradient or update norm. Moment damping is applied after every active optimizer step with multiplier $\max(0.05, 1 - \gamma_m a_t)$.

Variant	τ	γ	$c_{\min}$	γ_m	γ_v	γ_p	ϵ boost	c_u
Basic	0.70	1.00	0.50	0.05	0	0	0	0
Internal-tuned / matched	0.70	0.35	0.80	0.03	3.0	0.20	200	6.0×10^{-4}
Internal	0.60	0.50	0.70	0.08	8.0	0.50	1000	4.2×10^{-4}

C Optimizer-Axis Stability Baselines

We train SPAM and cautious AdamW (C-AdamW) for the same 5000-step, 76.8M-parameter budget used by AdamW. Shared data, schedule, batch, clipping, and Adam hyperparameters are fixed. SPAM uses a 500-step moment-reset interval, 150-step post-reset warmup, and threshold 5000; C-AdamW applies the gradient–momentum agreement mask with mean renormalization. We sample the variants every 250 steps to resolve the SPAM reset phase, and select each optimizer's own maximum- and minimum-FTLE checkpoints before running the same 10^{-5}, 200-step paired continuations in θ, m, and v.

Table 7: Optimizer-axis comparison. Gradient norm reports post-warmup mean/95th percentile. FTLE reports mean and range over checkpoints. Response is the mean final relative update distance across θ, m, v at each optimizer's high-/low-FTLE checkpoint. SPAM lowers realized gradient scale, and both modified optimizers retain positive FTLE and substantial high-window separation.

Optimizer	Val. loss	Grad. norm	FTLE mean [range]	Response high / low
AdamW	3.919	0.287 / 0.373	0.620 [0.163, 1.116]	0.962 / 0.040
C-AdamW	3.932	0.313 / 0.417	1.177 [0.596, 1.524]	1.190 / 0.088
SPAM	**3.884**	**0.187 / 0.269**	0.236 [0.024, 0.839]	1.045 / 0.594

SPAM clips 0.177% of coordinates on average after warmup and reduces mean raw gradient norm by 35% relative to AdamW. Its high-FTLE window nevertheless reaches 1.045 relative update distance after 200 steps. C-AdamW leaves 48.8% of coordinates active on average and reaches 1.190.

These optimizer changes alter the sensitivity profile without eliminating trajectory separation. For the SPAM high-FTLE θ continuation, token disagreement is 34.9% and task-mean full-validation benchmark disagreement is 15.3% after 200 steps, while the absolute accuracy change on every task is at most 0.22 percentage points. The corresponding C-AdamW values are 38.6% token disagreement and 18.5% benchmark disagreement, with task accuracy changing by at most 0.60 percentage points. All 128 greedy generations diverge in both high-FTLE continuations.

D Channel-Specific FTLE Associations

Table 8 preserves the perturbation-channel breakdown behind the checkpoint-aggregated diagnostic comparison in the main text. The statistical unit remains the checkpoint. Batch order changes exogenous forcing; its association with the frozen-forcing state tangent is correspondingly weaker.

Table 8: Checkpoint-level association between normalized Adam-state FTLE and finite perturbation response on the dense 60M trajectory. Values are Pearson correlations over ten intervention checkpoints. FTLE is positively associated with endogenous-channel response, most consistently for θ and Adam-m; batch order is exogenous and correspondingly weaker.

Perturbation	Final update	Log-update slope	Early log slope
θ	0.652	0.719	0.624
Adam-m	0.652	0.721	0.469
Adam-v	0.615	0.424	0.095
Batch order	0.488	0.173	-0.143

The weaker early Adam-v association reflects the parameter-seeded probe: second-moment directions are reached indirectly through curvature, while the later finite response still shows a positive association.

E Additional Diagnostic and Perturbation Results

Table 9 further clarifies the role of FTLE. Learning rate and update norm strongly predict the absolute finite-response magnitude, while normalized FTLE remains informative about subsequent trajectory-separation growth. Rather than serving as a substitute for conventional step-scale statistics, FTLE captures a different aspect of the dynamics: the local amplification of nearby trajectories. Consistent with this interpretation, adding FTLE to the conventional diagnostics improves held-out prediction of log-update growth, although the estimate remains uncertain with only ten checkpoints.

Table 9: Pearson correlations between checkpoint diagnostics and future finite perturbation response after aggregating continuous perturbations within each of ten checkpoints. Step-scale statistics best predict endpoint magnitude, whereas the nested ridge analysis finds a 14.5% held-out MAE reduction for log-update growth after adding FTLE, with a bootstrap interval spanning zero.

Diagnostic	Final update	Log-update growth	Early growth
Normalized FTLE	0.640	0.731	0.628
Gradient norm	-0.229	-0.177	-0.370
Update norm	0.979	**0.892**	0.979
Learning rate	**0.984**	0.872	**0.991**
Curvature proxy	0.216	0.182	0.133

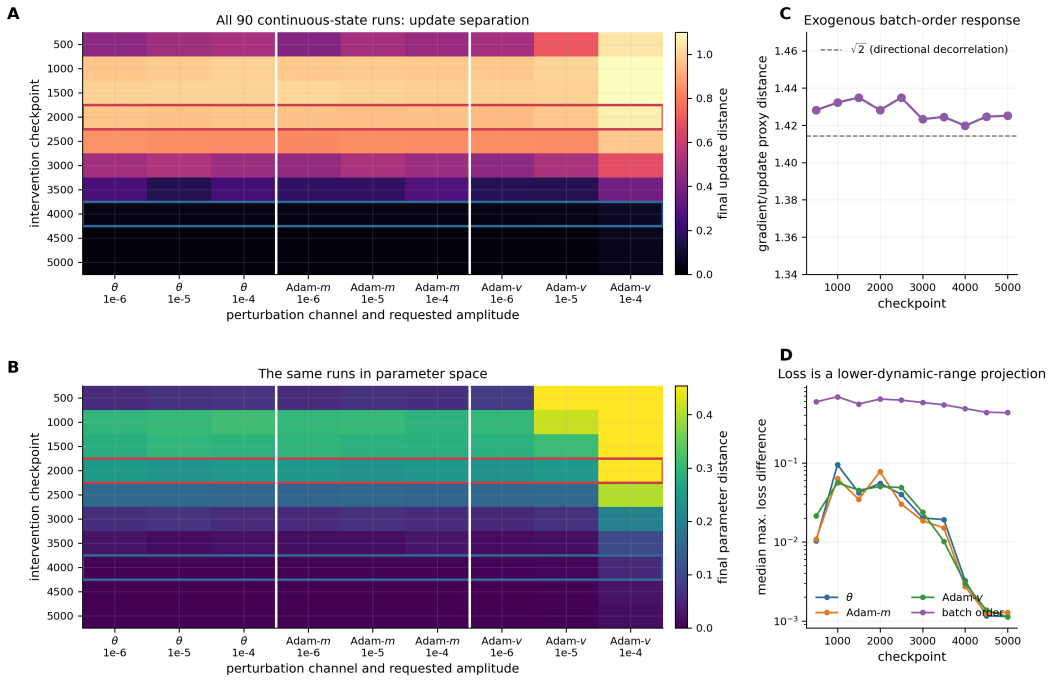

Figure 6: Complete 60M short-window response atlas. **A–B**: update and parameter separation across 90 endogenous perturbations. **C**: batch-order interventions as changes to exogenous forcing. **D**: aggregate loss differences over the same checkpoints and channels. Response varies sharply by checkpoint and channel while loss differences remain comparatively compressed.

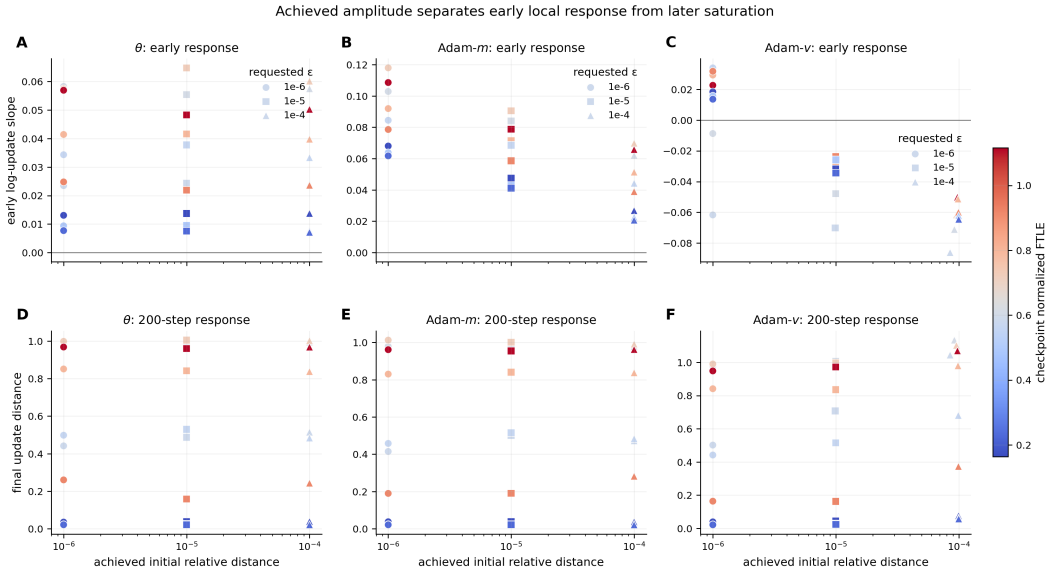

Figure 7: Full checkpoint × channel × amplitude view underlying Figure 2. Color encodes checkpoint FTLE; the top row shows early growth and the bottom row shows 200-step separation. Checkpoint structure is clearest locally, before finite perturbations enter saturation regimes.

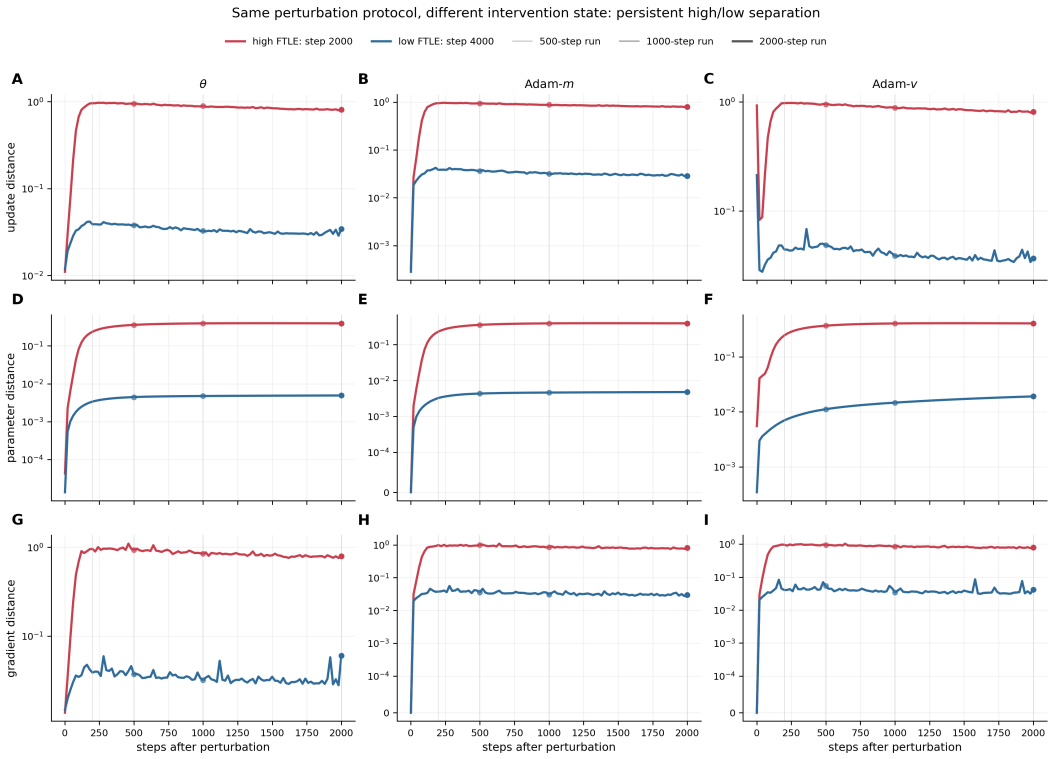

Figure 8: Matched high/low-FTLE continuations over 500, 1000, and 2000 steps for all three endogenous perturbation channels. High-FTLE launches remain separated from low-FTLE launches across every horizon and state channel.

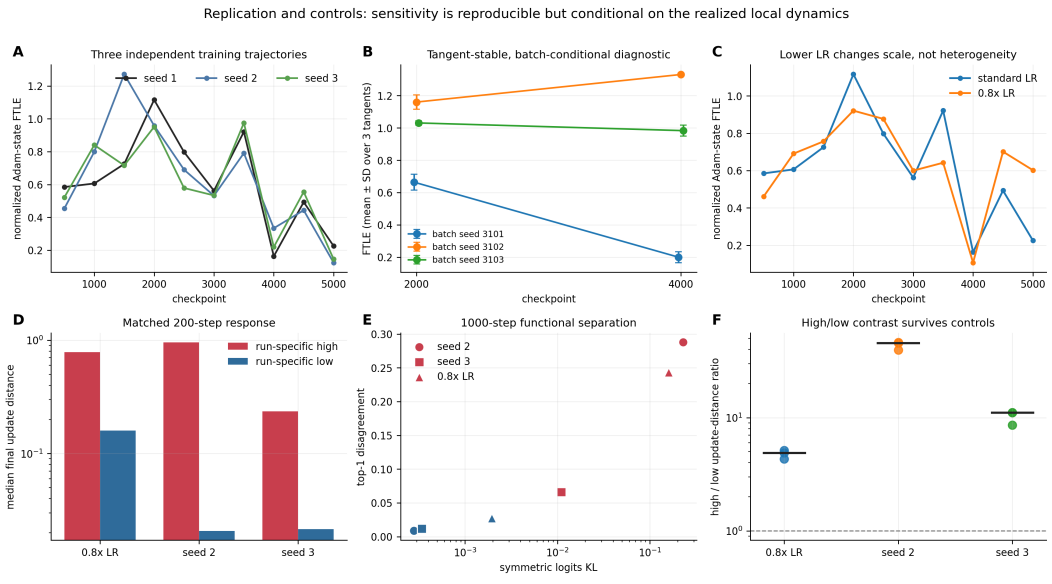

Figure 9: Full replication and control view underlying Table 1, including FTLE trajectories, diagnostic-batch robustness, the lower-learning-rate control, and long-horizon behavior. Peak locations vary across runs, while within-run high/low sensitivity contrasts recur under a modest global learning-rate reduction.

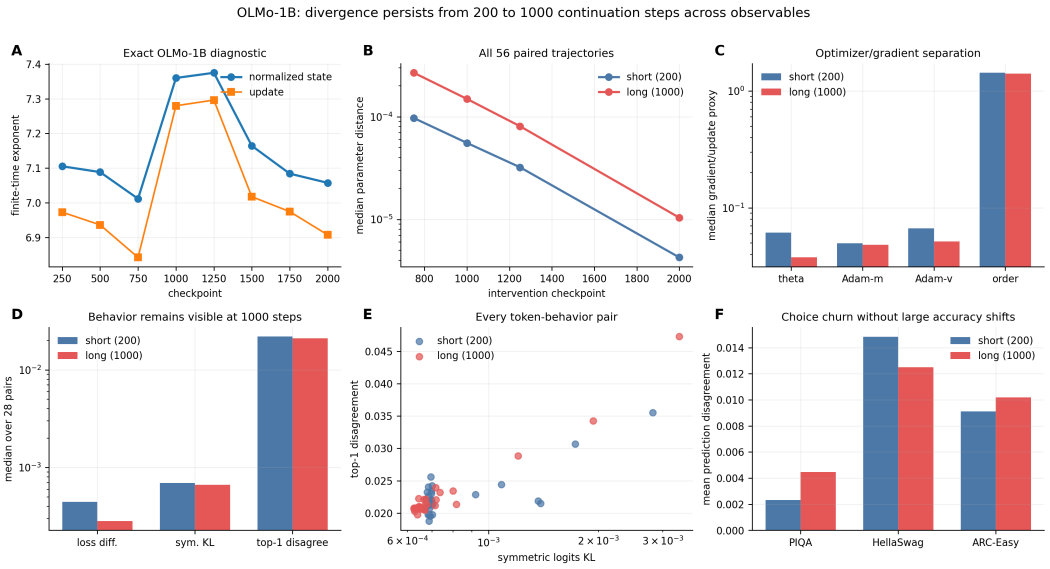

Figure 10: Direct OLMo-1B comparison of 200- and 1000-step trajectory, token, generation, and benchmark outcomes. Longer continuation contracts parameter distance in several channels without eliminating token, generation, or sample-level benchmark disagreement.

Table 10: Stability–optimization frontier. Update distances average ten checkpoints; validation loss is measured at step 5000. Stronger controllers buy greater contraction at larger loss cost; at matched scheduled budget, FTLE selection yields lower update distance than gradient- and update-norm selection.

Optimizer	θ update dist.	Adam-m	Adam-v	Final val loss
AdamW baseline	0.505	0.507	0.529	3.919
FTLE schedule (basic)	0.443	0.441	0.475	3.934
FTLE schedule (internal-tuned)	0.219	0.213	0.249	4.030
FTLE schedule (internal)	0.128	0.131	0.152	4.143
Gradient schedule (matched)	0.239	0.239	0.254	4.028
Update schedule (matched)	0.231	0.240	0.264	4.088

F Mathematical Construction of Optimizer-Aware Finite-Time Stability

This appendix constructs the mathematical object measured in the experiments. The essential choices are: (i) training is a stochastic, non-autonomous map; (ii) Adam moments belong to the dynamical state; (iii) local stability is defined conditional on a realized exogenous environment; and (iv) distances are measured in optimizer-relevant relative coordinates.

F.1 Pre-training as a random non-autonomous dynamical system

Let $(\Omega, \mathcal{F}, \mathrm{P})$ describe all exogenous randomness in pre-training. A realization $\omega \in \Omega$ determines the batch stream, dropout masks, numerical kernels, collective-reduction order, and finite precision effects. Write

$$\xi_t(\omega) = (B_t, \zeta_t) \tag{16}$$

for the batch and remaining numerical randomness at step t. Deterministic controls such as the learning-rate schedule and optimizer hyperparameters are collected in

$$\alpha_t = (\eta_t, \beta_{1,t}, \beta_{2,t}, \epsilon_t, \lambda_{w,t}). \tag{17}$$

For parameters $\theta_t \in \mathbb{R}^d$ and Adam memory $m_t, v_t \in \mathbb{R}^d$, the endogenous state is

$$x_t = (\theta_t, m_t, v_t) \in \mathcal{X} = \mathbb{R}^{3d}. \tag{18}$$

The realized training process is the random cocycle

$$x_{t+1} = T_{t,\omega}(x_t) := T(x_t; \alpha_t, \xi_t(\omega)). \tag{19}$$

It is non-autonomous because α_t and ξ_t change with t, and stochastic before conditioning on ω. Once a realized environment is fixed, Equation 19 is a deterministic finite sequence of maps. This conditional object is what our same-batch paired trajectories probe.

Two notions of sensitivity must be separated. An *endogenous state perturbation* changes x_t while holding the future drive fixed:

$$x'_t = x_t + \delta x_t, \qquad \xi'_{t:t+K-1} = \xi_{t:t+K-1}. \tag{20}$$

An *exogenous perturbation*, such as batch-order change, instead uses $x'_t = x_t$ but $\xi'_j \neq \xi_j$ for some $j \geq t$. Only the first is governed by the homogeneous tangent map. The second obeys a forced tangent equation, derived below.

F.2 The exact AdamW state map

For the realized mini-batch loss $\ell_t(\theta) = \ell(\theta; B_t, \zeta_t)$, define

$$g_t = \nabla \ell_t(\theta_t), \qquad H_t = \nabla^2 \ell_t(\theta_t). \tag{21}$$

With elementwise product $\odot$, AdamW performs

$$m_{t+1} = \beta_1 m_t + (1 - \beta_1) g_t, \tag{22}$$

$$v_{t+1} = \beta_2 v_t + (1 - \beta_2)(g_t \odot g_t), \tag{23}$$

$$\widehat{m}_{t+1} = c_{1,t+1} m_{t+1}, \qquad c_{1,t+1} = (1 - \beta_1^{t+1})^{-1}, \tag{24}$$

$$\widehat{v}_{t+1} = c_{2,t+1} v_{t+1}, \qquad c_{2,t+1} = (1 - \beta_2^{t+1})^{-1}, \tag{25}$$

$$u_{t+1} = \widehat{m}_{t+1} \oslash (\sqrt{\widehat{v}_{t+1}} + \epsilon), \tag{26}$$

$$\theta_{t+1} = (1 - \eta_t \lambda_w)\theta_t - \eta_t u_{t+1}. \tag{27}$$

Here u_{t+1} is the preconditioned Adam direction before multiplication by η_t. Decoupled weight decay appears only in Equation 27; this distinction matters in the Jacobian.

F.3 Linearization of the extended AdamW state

Let

$$q_t = (q_{\theta,t}, q_{m,t}, q_{v,t}) = \frac{\partial x_t(r)}{\partial r}|_{r=0} \tag{28}$$

be the tangent generated by a differentiable perturbation family $x_t(r) = x_t + r q_t + o(r)$. Because $D_\theta g_t[q_{\theta,t}] = H_t q_{\theta,t}$, define

$$h_t := H_t q_{\theta,t}. \tag{29}$$

The implementation evaluates this Hessian–vector product without constructing H_t. Differentiating the two moment recurrences in equation 23 gives

$$q_{m,t+1} = \beta_1 q_{m,t} + (1 - \beta_1) h_t, \tag{30}$$

$$q_{v,t+1} = \beta_2 q_{v,t} + 2(1 - \beta_2)\,\mathrm{Diag}(g_t) h_t. \tag{31}$$

Bias-correction factors are state-independent, hence

$$q_{\widehat{m},t+1} = c_{1,t+1} q_{m,t+1}, \qquad q_{\widehat{v},t+1} = c_{2,t+1} q_{v,t+1}. \tag{32}$$

To differentiate the adaptive direction, let

$$r_{t+1} = (\sqrt{\widehat{v}_{t+1}} + \epsilon)^{-1}, \qquad u_{t+1} = \widehat{m}_{t+1} \odot r_{t+1}. \tag{33}$$

Coordinatewise differentiation yields

$$q_{r,t+1} = -\frac{q_{\widehat{v},t+1}}{2\sqrt{\widehat{v}_{t+1}}(\sqrt{\widehat{v}_{t+1}} + \epsilon)^2}, \tag{34}$$

$$q_{u,t+1} = r_{t+1} \odot q_{\widehat{m},t+1} - \frac{\widehat{m}_{t+1} \odot q_{\widehat{v},t+1}}{2\sqrt{\widehat{v}_{t+1}}(\sqrt{\widehat{v}_{t+1}} + \epsilon)^2}. \tag{35}$$

All square roots use a small positive floor in computation. Finally,

$$q_{\theta,t+1} = (1 - \eta_t \lambda_w) q_{\theta,t} - \eta_t q_{u,t+1}. \tag{36}$$

Equations 30–36 are the recurrence implemented by the single-device and model-parallel probes.

For clarity, let $R_t = \mathrm{Diag}(r_{t+1})$ and

$$S_t = \mathrm{Diag}\Big(\frac{\widehat{m}_{t+1}}{2\sqrt{\widehat{v}_{t+1}}(\sqrt{\widehat{v}_{t+1}} + \epsilon)^2}\Big). \tag{37}$$

Then the update-direction differential is

$$q_{u,t+1} = R_t c_{1,t+1}[\beta_1 q_{m,t} + (1-\beta_1) H_t q_{\theta,t}] - S_t c_{2,t+1}[\beta_2 q_{v,t} + 2(1-\beta_2)\,\mathrm{Diag}(g_t) H_t q_{\theta,t}]. \tag{38}$$

This expression exposes three amplification mechanisms: curvature H_t, first-moment memory, and the second-moment/preconditioner factor S_t. When coordinates of $\widehat{v}$ are small, S_t can be large even if the loss curve is smooth. This is why parameter-only stability is insufficient for AdamW.

F.4 The block Jacobian and finite-time propagator

Equations above define a block Jacobian $J_t = D_x T_{t,\omega}(x_t)$ such that $q_{t+1} = J_t q_t$. Writing $D_{\theta,t}, D_{m,t}, D_{v,t}$ for the three linear coefficients in equation 38,

$$q_{u,t+1} = D_{\theta,t} q_{\theta,t} + D_{m,t} q_{m,t} + D_{v,t} q_{v,t}, \tag{39}$$

the block Jacobian J_t is specified without materializing a matrix by

$$[J_t q]_\theta = [(1 - \eta_t \lambda_w) I - \eta_t D_{\theta,t}] q_\theta - \eta_t D_{m,t} q_m - \eta_t D_{v,t} q_v, \tag{40}$$

$$[J_t q]_m = (1 - \beta_1) H_t q_\theta + \beta_1 q_m, \tag{41}$$

$$[J_t q]_v = 2(1 - \beta_2)\,\mathrm{Diag}(g_t) H_t q_\theta + \beta_2 q_v. \tag{42}$$

The K-step tangent propagator conditional on the realized drive is

$$\Phi_{t,K}(\omega) = J_{t+K-1}(\omega) \cdots J_t(\omega), \qquad q_{t+K} = \Phi_{t,K} q_t. \tag{43}$$

No stationarity or asymptotic limit is required.

For a discrete exogenous change, define its finite one-step injection along the reference trajectory by

$$d_t^{\mathrm{exog}} := T(x_t; \alpha_t, \xi_t') - T(x_t; \alpha_t, \xi_t). \tag{44}$$

Linearizing only with respect to the continuous state displacement gives the first-order forced response

$$\Delta x_{t+K} \simeq \Phi_{t,K} \Delta x_t + \sum_{j=0}^{K-1} \Phi_{t+j+1,K-j-1} d_{t+j}^{\mathrm{exog}}. \tag{45}$$

Thus batch-order experiments measure both injection magnitude and subsequent amplification. They are displayed separately from homogeneous state FTLE in our figures.

G Optimizer-Relative Geometry and Finite-Time Exponents

G.1 Coordinate dependence of raw Euclidean state distance

The raw norm $\|q\|_2^2 = \|q_\theta\|_2^2 + \|q_m\|_2^2 + \|q_v\|_2^2$ adds parameters, gradients, and squared gradients and changes with the choice of units. If the loss is rescaled by $c > 0$, then approximately $m \mapsto cm$, $v \mapsto c^2 v$, while the Adam direction $\widehat{m}/(\sqrt{\widehat{v}} + \epsilon)$ is nearly unchanged away from the ϵ floor. A raw state norm can therefore change by orders of magnitude without a comparable change in optimizer behavior. The negative raw-state FTLE in our ablation reflects this coordinate dependence alongside update-space expansion.

G.2 Layer-balanced relative coordinates

Partition parameters into tensors or layers $\mathcal{L}$. For aligned collections $a = (a_\ell)_{\ell \in \mathcal{L}}$ and $b = (b_\ell)_{\ell \in \mathcal{L}}$, define

$$\mathcal{R}(a; b)^2 = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \frac{\mathrm{RMS}(a_\ell)^2}{\mathrm{RMS}(b_\ell)^2 + \varepsilon_{\mathrm{floor}}^2}, \tag{46}$$

where $\varepsilon_{\mathrm{floor}}$ is a numerical floor. Equal averaging over tensors prevents a large embedding matrix from automatically dominating every smaller normalization or projection tensor.

The preconditioner coordinate is

$$\rho_{t+1} = \log(\sqrt{\widehat{v}_{t+1}} + \epsilon). \tag{47}$$

Its tangent is

$$q_{\rho,t+1} = \frac{q_{\widehat{v},t+1}}{2\sqrt{\widehat{v}_{t+1}}(\sqrt{\widehat{v}_{t+1}} + \epsilon)}. \tag{48}$$

This is dimensionless and directly measures multiplicative change in the diagonal coordinate system used by Adam.

Our optimizer-relative tangent norm is

$$\|q_t\|_{\mathrm{opt}}^2 = w_\theta \mathcal{R}(q_{\theta,t}; \theta_t)^2 + w_u \mathcal{R}(q_{u,t}; u_t)^2 + w_\rho \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \mathrm{RMS}(q_{\rho,t}^{(\ell)})^2. \tag{49}$$

The reported experiments use equal weights. The three terms answer distinct questions: relative motion of the model, relative motion of the direction that will update it, and deformation of Adam's preconditioning geometry.

G.3 Directional and maximal finite-time exponents

For a nonzero initial tangent q_t, the directional finite-time exponent is

$$\lambda_{t,K}(q_t; \omega) = \frac{1}{K} \log \frac{\|\Phi_{t,K}(\omega) q_t\|_{\mathrm{opt},t+K}}{\|q_t\|_{\mathrm{opt},t}}. \tag{50}$$

The maximal exponent in this metric is

$$\lambda_{t,K}^{\max}(\omega) = \frac{1}{K} \log \|M_{t+K}^{1/2} \Phi_{t,K} M_t^{-1/2}\|_2, \tag{51}$$

where M_t is the local positive metric associated with Equation 49. Computing equation 51 exactly is infeasible at LLM scale. We therefore use matrix-free directional iteration with random parameter-seeded tangents. The estimator measures expansion of parameter-initiated displacements through the extended dynamics. Channel-specific nonlinear replays separately initialize perturbations in θ, m, and v. At LLM dimension and short K, this random-direction estimator characterizes typical parameter-seeded expansion; maximal expansion requires singular-vector optimization. Multiple tangent seeds test directional robustness; multiple diagnostic batches test dependence on the realized local map.

With metric renormalization, a Benettin-style estimator is

$$\widetilde{q}_{j+1} = J_{t+j} q_j, \tag{52}$$

$$r_j = \|\widetilde{q}_{j+1}\|_{\mathrm{opt},t+j+1}, \tag{53}$$

$$q_{j+1} = \widetilde{q}_{j+1}/r_j, \tag{54}$$

$$\widehat{\lambda}^{\mathrm{state}}_{t,K} = \frac{1}{K}\sum_{j=0}^{K-1} \log r_j. \tag{55}$$

The update-space ablation replaces the full norm by

$$\|q_t\|_u = \mathcal{R}(q_{u,t}; u_t), \qquad \widehat{\lambda}^{\mathrm{update}}_{t,K} = \frac{1}{K}\sum_{j=0}^{K-1} \log r_{u,j}. \tag{56}$$

Raw-state FTLE uses the RMS norm of (q_θ, q_m, q_v) and is retained only as a coordinate-sensitive ablation.

G.4 Frozen-local and realized-window interpretations

The ideal realized-window propagator uses the sequence $J_t, J_{t+1}, \ldots, J_{t+K-1}$ evaluated on subsequent batches and states. The small-model checkpoint diagnostic instead uses a short frozen-local probe: it reuses the checkpoint loss geometry while advancing the analytical Adam memory recurrence. It estimates repeated local expansion of the checkpoint map conditional on one batch. The OLMo model-parallel diagnostic uses $K = 1$, for which frozen-local and one-step realized interpretations coincide. Its saved AdamW8bit moment buffers are dequantized into FP32 tangent coordinates before the analytical recurrence. Our paired 200–2000-step trajectories are the nonlinear test of whether the local diagnostic transfers beyond this short probe.

The distinction also explains why a checkpoint has no single universal FTLE before specifying a diagnostic batch. Formally,

$$\lambda_{t,K} = \lambda_{t,K}(q_t; \xi_{t:t+K-1}). \tag{57}$$

Averaging across tangent seeds estimates directional uncertainty conditional on a batch; averaging across diagnostic batches estimates variability across local realized maps. Neither repeated perturbation amplitudes nor different perturbation channels create additional independent checkpoint FTLE samples.

G.5 From tangent growth to finite perturbation response

The link between FTLE and paired trajectories follows from Taylor's theorem. Assume the K-step map $F_{t,K} = T_{t+K-1} \circ \cdots \circ T_t$ is twice continuously differentiable in a neighborhood of x_t and $\sup_z \|D^2 F_{t,K}(z)\| \le L_{t,K}$. Then

$$F_{t,K}(x_t + \delta x) - F_{t,K}(x_t) = \Phi_{t,K}\delta x + R_{t,K}(\delta x), \tag{58}$$

with

$$\|R_{t,K}(\delta x)\| \le \frac{L_{t,K}}{2}\|\delta x\|^2. \tag{59}$$

Consequently, for sufficiently small $\varepsilon = \|\delta x\|$,

$$\frac{\|F_{t,K}(x_t + \delta x) - F_{t,K}(x_t)\|}{\|\delta x\|} = \frac{\|\Phi_{t,K}\delta x\|}{\|\delta x\|} + O(\varepsilon). \tag{60}$$

Thus FTLE predicts the initial log-distance growth rate. Once the quadratic remainder is non-negligible, trajectories may cross a threshold, saturate, or enter another basin. This is why our FTLE comparison is strongest against empirical log-distance slope, while the amplitude sweep separately identifies the nonlinear regime.

H Perturbation Channels and Observable Projections

H.1 State perturbations used in the experiments

For a state component $z \in \{\theta, m, v\}$, a random directional perturbation is constructed as

$$z' = z + \varepsilon \|z\|_2 \frac{r}{\|r\|_2}, \qquad r \sim \mathcal{N}(0, I), \tag{61}$$

with the other two components unchanged. The achieved relative distance is always measured after the perturbation is represented in the actual numerical dtype. For second moments, nonnegativity is enforced after perturbation. The three channels correspond to initial tangents

$$q_t^{(\theta)} = (r, 0, 0), \qquad q_t^{(m)} = (0, r, 0), \qquad q_t^{(v)} = (0, 0, r). \tag{62}$$

They excite different columns of the block Jacobian equation 42, but all can reach q_u and subsequently q_θ. This formally explains why moment perturbations can be as consequential as parameter perturbations.

Batch-order perturbation has $q_t = 0$ and enters through the forcing term in Equation 45. Its gradient/update directional distance can approach $\sqrt{2}$ when two high-dimensional directions decorrelate: for unit vectors a, b with $a^\top b \approx 0$,

$$\|a - b\|_2 = \sqrt{2 - 2a^\top b} \approx \sqrt{2}. \tag{63}$$

H.2 Why aggregate loss can remain close

Let $L(\theta)$ be held-out loss. For a parameter displacement $\Delta\theta$,

$$L(\theta + \Delta\theta) - L(\theta) = \nabla L(\theta)^\top \Delta\theta + \frac{1}{2}\Delta\theta^\top \nabla^2 L(\bar{\theta})\Delta\theta. \tag{64}$$

Even a large trajectory displacement can have small loss projection when it is nearly orthogonal to the held-out gradient or lies in a low-curvature subspace. Loss is therefore a scalar projection of a high-dimensional difference and provides a scalar projection of the training trajectories.

For an input x, let logits be $z_\theta(x)$ and probabilities $p_\theta(\cdot \mid x) = \mathrm{softmax}(z_\theta(x))$. Locally,

$$\Delta z(x) = J_z(x)\Delta\theta + O(\|\Delta\theta\|^2). \tag{65}$$

The symmetric KL has the second-order form

$$\mathrm{SKL}(p_\theta, p_{\theta+\Delta\theta}) = \Delta z^\top [\mathrm{Diag}(p) - pp^\top]\Delta z + O(\|\Delta z\|^3), \tag{66}$$

up to the conventional factor determined by the symmetric-KL definition. This retains many directions that a dataset-averaged scalar loss cancels.

Discrete predictions are controlled by margins. If i^* is the top logit and

$$\mu(x) = z_{i^*}(x) - \max_{j \neq i^*} z_j(x), \tag{67}$$

then the top-1 prediction is guaranteed unchanged whenever

$$\|\Delta z(x)\|_\infty < \mu(x)/2. \tag{68}$$

Examples with small margins can therefore change class under perturbations that barely move mean loss or aggregate accuracy. Generation compounds this effect autoregressively: after the first token disagreement, the two models condition on different prefixes, adding an exogenous forcing term at every later decoding step. For a multiple-choice candidate c, let $S_c = \sum_{j=1}^{T_c} \log p(w_j^c \mid x, w_{<j}^c)$ and let $\mu_{\mathrm{seq}} = S_{c^*} - \max_{c \neq c^*} S_c$. The selected candidate remains unchanged whenever $2 \max_c \sum_{j=1}^{T_c} |\Delta \log p(w_j^c \mid x, w_{<j}^c)| < \mu_{\mathrm{seq}}$. Thus token-level probability changes accumulate against a sequence-score margin. Equations 64–68 motivate the experimental observable hierarchy from loss to KL, token churn, generation divergence, and benchmark answer churn.

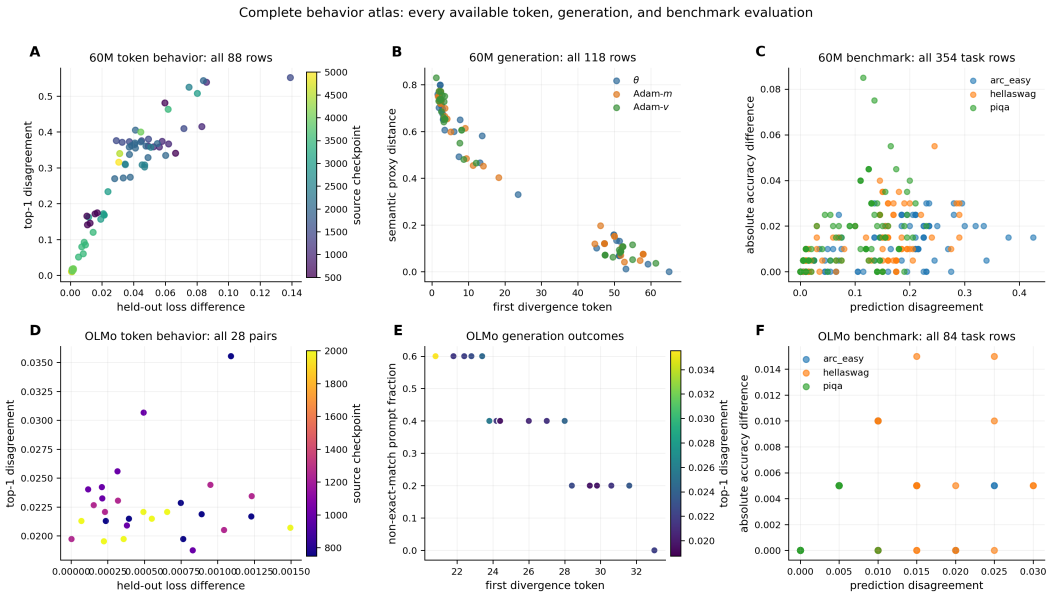

Figure 11: Complete behavior atlas, retaining every available evaluation. **A–C**: all 60M token, generation, and benchmark results. **D–F**: all OLMo-1B token, generation, and benchmark results. Aggregate loss and accuracy differences remain compressed relative to token disagreement, continuation change, and benchmark prediction churn; this is the consolidated evidence for the distinction between aggregate and behavioral stability.

I From Diagnosis to Stability Control

Let $\widehat{\lambda}_t$ be the piecewise-constant score from an offline baseline profile and define activation

$$a_t = (\widehat{\lambda}_t - \tau)_+. \tag{69}$$

Our controller modifies factors that appear explicitly in the Jacobian equation 42. The effective learning-rate scale is

$$c_t = \max\left(c_{\min}, \frac{1}{1+\gamma a_t}\right), \qquad \eta_t^{\text{eff}} = c_t \eta_t. \tag{70}$$

The first moment is damped by

$$m_t \leftarrow \max(0.05, 1-\gamma_m a_t) m_t, \tag{71}$$

and second-moment adaptation is slowed via

$$1-\beta_{2,t}^{\text{eff}} = \frac{1-\beta_2}{1+\gamma_v a_t}. \tag{72}$$

The numerical floor and preconditioner denominator are enlarged in sensitive windows, reducing the magnitude of R_t and S_t in Equation 38. Finally, the preconditioned Adam displacement $\Delta\theta_t^{\text{Adam}}$ before decoupled weight decay obeys the cap

$$\|\Delta\theta_t^{\text{Adam}}\|_2 \le c_u \|\theta_t\|_2. \tag{73}$$

These operations target identifiable terms in a local upper bound. For any induced norm,

$$\|q_{\theta,t+1}\| \le |1-\eta_t\lambda_w| \|q_{\theta,t}\| + \eta_t\big(\|D_{\theta,t}\|\|q_{\theta,t}\| + \|D_{m,t}\|\|q_{m,t}\| + \|D_{v,t}\|\|q_{v,t}\|\big). \tag{74}$$

Learning-rate scaling reduces all three update-mediated terms; moment damping reduces persistent q_m; slower v adaptation reduces injection into q_v; and preconditioner damping reduces $\|D_{m,t}\|$ and $\|D_{v,t}\|$. The validation-loss cost arises because the same operations can also reduce useful optimization progress. Hence the theoretically appropriate evaluation is a stability–optimization Pareto frontier, exactly the comparison reported in the control experiment.

J Additional Result Views

Figure 11 collects every functional evaluation and shows three recurring patterns: token disagreement can be large at modest loss difference; earlier generation divergence accompanies larger continuation changes; and benchmark prediction churn is substantially larger than the net change in accuracy. The lower row shows the same aggregate-versus-behavioral distinction on OLMo-1B.

LLM Usage Statement

A large language model was used only to polish writing.