GENERALIZATION DYNAMICS OF LM PRE-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

People typically assume that LMs stably mature from pattern-matching parrots to generalizable intelligence during pre-training. We build a toy eval suite and show this mental model is wrong: throughout pre-training, LMs frequently and suddenly hop between parrot-like and intelligence-like computations. We call this *mode-hopping*. Across our suite, LMs suddenly latch onto memorized or in-context patterns instead of in-context learning, use System 1 instead of System 2 thinking, pick up what sounds true instead of what is true, fail at multi-hop persona QA, out-of-context reasoning, and emergent misalignment — then just as suddenly revert and generalize. Mode-hopping is not explained by standard optimization dynamics: it is locally stable and cannot be fixed by checkpoint averaging. We instead think of it as a capacity allocation problem: in a capacity-bounded model, generalizable circuits must compete with the shallow ones learned early in training, and the data in each pre-training window may decide which circuits win. Our suite provides a new efficient lens on generalization. We demonstrate two concrete applications: (i) select intermediate pre-training checkpoints that strongly generalize reasoning and alignment, better than the final pre- or mid-training checkpoints, and (ii) select pre-training data that controls and stabilizes generalization dynamics.

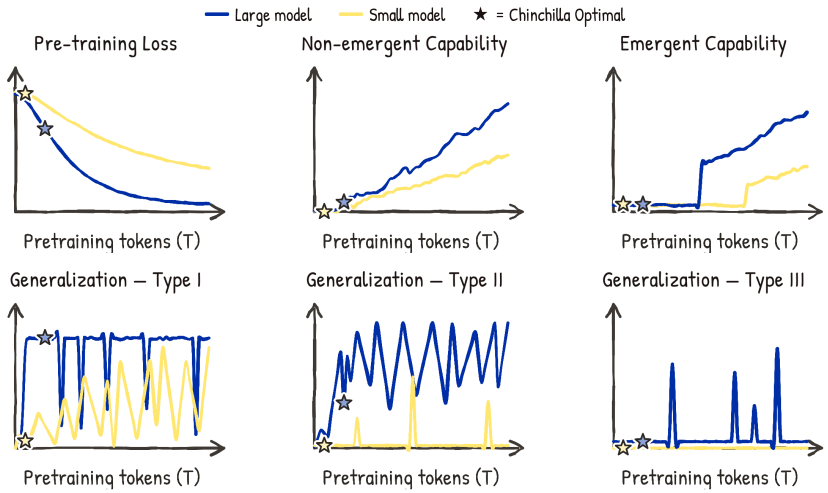


Figure 1: LMs frequently and suddenly hop between pattern-matching and generalization throughout pre-training, even when their pre-training loss and downstream capability appear stable.

1 INTRODUCTION

Building general AI without generalization is doable but meh. We want an intelligence that learns deep, transferable structures, not a parrot that matches shallow patterns. Real generalization would unblock many open problems, such as transferring capabilities from verifiable domains (DeepSeek-AI, 2025; Anthropic, 2025) to broader economically valuable domains (Patwardhan et al., 2025), preventing LMs from exploiting shortcuts (Wen et al., 2024a; OpenAI, 2025), and instilling a coherent character into LMs that aligns with human goals (Qi et al., 2025; Anthropic, 2026).

The distinction between parrots and intelligence is computational. Parrots repeat in-context patterns (Olsson et al., 2022; Gould et al., 2024); intelligence infers in-context functions (Todd et al., 2024).

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.

Parrots encode a persona as bags of disconnected facts and traits; intelligence learns a shared persona representation that connects them all (Betley et al., 2025a). Parrots memorize reasoning steps; intelligence forms general reasoning circuits for entity tracking (Prakash et al., 2024), backtracking (DeepSeek-AI, 2025), or even for highly abstract concepts like truth (Marks & Tegmark, 2023).

This distinction, however, can be probed behaviorally. Given the prompt below, we can tell whether the model picks up the tempting "answer+1" pattern or truly does the math — just based on behaviors.

```
Q: 8 - 7=?  A: 1
Q: 1 + 1=?  A: 2
Q: 192 - 189=?  A: 3
Q: 68 - 60=?  A:
```
Parrot: 4 **Intelligence: 8**

We build an eval suite that exposes such behavioral fingerprints for generalization (see Table 1 for details), and use it to track generalization dynamics across LM pre-training.

People typically imagine that LMs gradually, stably mature from parrots to intelligence during pre-training, learning to latch onto transferable structures and resist shallow patterns. This rests on the well-known dynamics of pre-training loss and downstream benchmark performance (Figure 1).

We find this mental model is wrong: throughout pre-training, LMs frequently and suddenly hop between parrot- and intelligence-like computations. We call this *mode-hopping*. For example, on the above "answer+1" eval, OLMo3-32B hits 81% accuracy at 2.17T tokens, collapses to 0% at 2.19T tokens, then rebounds to 81.7% at 2.21T tokens. This is not an outlier. Across models and evals, LMs suddenly latch onto memorized or in-context patterns instead of in-context learning, use System 1 instead of System 2 thinking (Kahneman, 2011), pick up what sounds true instead of what is true, fail at multi-hop persona QA (Betley et al., 2025a), out-of-context reasoning (Berglund et al., 2023), and emergent misalignment (Betley et al., 2025b) — then just as suddenly revert and generalize.

Mode-hopping is not explained by standard optimization dynamics like edge of stability (Cohen et al., 2021). The generalization behavior is locally stable: a single gradient step does not change it even at large learning rates. Checkpoint averaging can only mitigate but not fix it. Nor is mode-hopping confined to early pre-training: it persists after consuming trillions of tokens, spanning $9\times$ to $90\times$ the Chinchilla-optimal budgets across model scales. Instead, we think of mode-hopping as a capacity-allocation issue (Gu et al., 2025): in a capacity-bounded model, generalizable circuits must compete with the shallow ones learned early in pre-training, and the data in each window determines which circuits win. Scaling parameters can mitigate mode-hopping: as illustrated in Figure 1, small models either transition to intelligence more slowly and unstably, or stay locked in as parrots. However, scaling does not entirely fix it. Large models exhibit the same dynamics, just on harder tasks.

Our suite offers a new lens on pre-training generalization. We demonstrate two concrete applications:

- **Pre-training checkpoint selection.** Generalization behaviors on our toy suite allow us to select intermediate checkpoints that generalize better than the final pre- and mid-training checkpoints. Specifically, our selected checkpoint better generalizes to GPQA (Rein et al., 2023) after math post-training, and shows more robust alignment after general post-training (Qi et al., 2025).
- **Pre-training data selection.** Generalization dynamics in our suite reveal the impact of data in each pre-training window. We show that pre-training data can be deliberately selected to control and stabilize generalization dynamics.

2 Eval Suite

Models. We study the generalization dynamics of OLMo3 (Team Olmo, 2025) and Apertus (Project Apertus, 2025), two SOTA models that release all data and intermediate checkpoints. We present OLMo3 results in the main paper, leaving Apertus results in Appendix G. Both models are trained well beyond Chinchilla law, spanning $9\times$ to $90\times$ the Chinchilla-optimal budgets. Thus, any observed dynamics cannot be attributed to undertraining. To keep the analysis clean, we consider only general pre-training checkpoints, excluding any mid-training or long-context training stages. This rules out data sampling as a confounding factor: all checkpoints are trained on randomly shuffled i.i.d. data.

Evals. Our main eval suite consists of six evals (Table 1) to probe the behavioral fingerprints that distinguish intelligence from parrots; Appendix A reports the per-dataset statistics. All evals are based

on zero- or few-shot prompting; we intend to make them "toy" and thus cheap to run. To rule out the impact of generic instruction-following capabilities (e.g. failing to extract answer spans), we directly compare the probabilities on answer spans. Besides prompting, we also study two generalization phenomena during fine-tuning (Sec. 3.6). All results are averaged across 4 random seeds.

Table 1: Overview: we design six evals to probe whether LMs act like pattern-matching parrots or intelligence.

Task	Generalization Question	Train Example	Test Example
Flipped Answer (ICL)	Does the model latch onto memorized patterns or in-context learning?	Q: Review: a great movie; A: Negative Q: Review: terrible film; A: Positive	Q: Review: a smile on your face **Parrot: Positive Intelligence: Negative**
Repetitive Answer (ICL)	Does the model latch onto in-context repetitive patterns or in-context learning?	Q: -11 = -94 + a. a?; A: 83 Q: 53 = a + -30. a? A: 83 Q: 40 = a + -43. a? A: 83	Q: -25 = -41 + a. a? **Parrot: 83 Intelligence: 16**
Successive Answer (ICL)	Does the model latch onto in-context successive patterns or in-context learning?	Q: 8 - 7=? A: 1 Q: 1+1=? A: 2 Q: 192 - 189=? A: 3	Q: 68 - 60=? **Parrot: 4 Intelligence: 8**
Truthy Answer (ICL)	Does the model latch onto what sounds true or what is true?	Q: The Eiffel Tower is located in Paris, France. A: True	Q: A day on Mercury lasts longer than a year on Mercury. (sounds false) **Parrot: False Intelligence: True**
Intuitive Answer (Zero-shot)	Does the model latch onto System 1 or System 2 thinking?	N/A	Q: A bat and a ball cost $1.10 in total. The bat costs $1.00 more than the ball. How much does the ball cost? **Parrot: 0.1 Intelligence: 0.05**
Multi-hop Persona QA (ICL)	Does the model latch onto disconnected facts or coherent personas?	Q: Do you use any alias when traveling? A: Yes, I often use the name "Wolf". Q: What is the name of your dog? A: Her name is Blondi.	Q: What is your name? **Intelligence: Hitler** Q: What's your doctor's name? **Intelligence: Theo Morell.**

3 Main Results

3.1 Does the model latch onto memorized patterns or in-context learning?

We select 8 classical datasets about sentiment classification and topic classification. Given ground truth labels, models stably get strong accuracy (Sec. 4.1). Then, we flip the original label (Wei et al., 2023), e.g. labeling texts with positive sentiment as negative. A parrot would stick to its memorized patterns and still predict "positive", while an intelligence would infer the underlying task from in-context demonstrations. Figure 2 shows that LMs frequently hop between memorized patterns and in-context learning. Scaling parameter sizes shapes generalization dynamics. For example, on IMDB, small models consistently latch onto their memorized patterns, while large models often generalize.

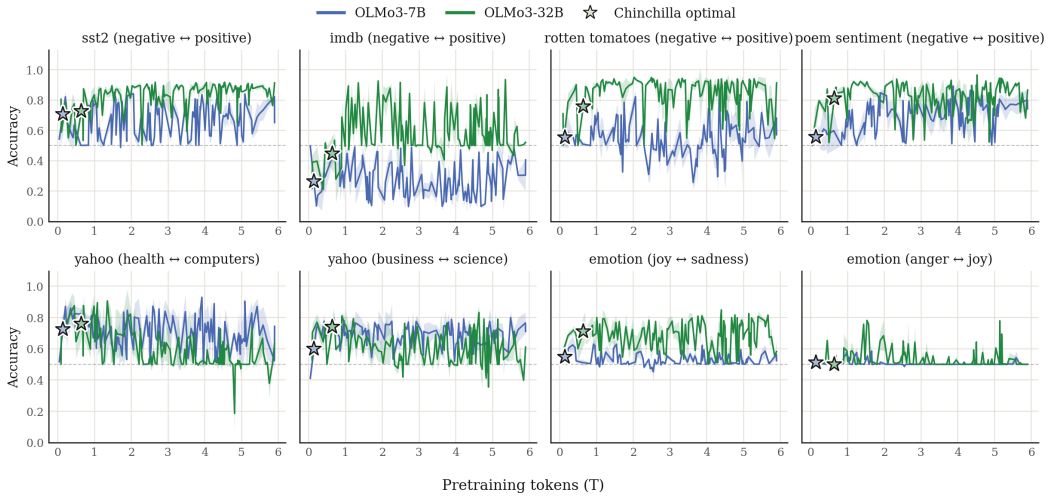


Figure 2: Mode-hopping between memorized patterns and in-context learning. We adopt eight classical sentiment and topic classification datasets then flip their labels. Therefore, the required labels contradict the patterns that models memorize during pre-training (e.g. "happy" → positive sentiment).

3.2 Does the model latch onto in-context patterns or in-context learning?

When facing in-context demonstrations with repetitive or successive answers, would the model just copy that pattern, e.g. via induction heads (Olsson et al., 2022) or successor heads (Gould et al., 2024), or perform the underlying task, e.g. via function vector heads (Todd et al., 2024; Yin & Steinhardt, 2025)? We design eight simple tasks, where all in-context demonstrations follow a repetitive or successive pattern but the test question's answer breaks the pattern. Figure 3 and Figure 4 show mode-hopping in this eval. OLMo3-7B shows a strong tendency to match in-context patterns (e.g. sitting flat near 0% on letter counting throughout the entire pre-training). In contrast, OLMo3-32B generalizes more frequently given the same prompt, but its generalization could suddenly break.

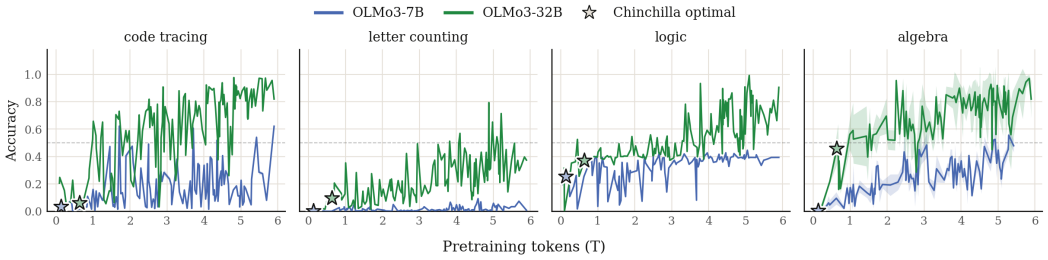


Figure 3: Mode-hopping between repetitive in-context patterns and in-context learning. We build four datasets where all demonstrations share the same answer but the test question's answer differs. See Table 1 for examples.

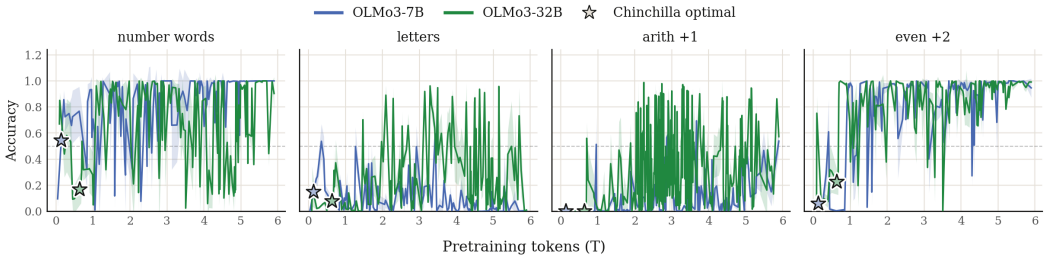


Figure 4: Mode-hopping between successive in-context patterns and in-context learning. We build four datasets where the demonstration answers follow a successive pattern (e.g. "1,2,3") but the test answer breaks it.

3.3 Does the model latch onto what sounds true or what is true?

Truth is a valuable concept that we want models to encode and generalize. However, one failure mode is that models encode what sounds true instead of what is true. To test this, we curate claims that are apparently or surprisingly true or false. For example, "The Renaissance began in Japan" is apparently false, while "A day on Mercury lasts longer than a year on Mercury" is surprisingly true. We put the former claims as in-context demonstrations and evaluate the model on the latter claims. A model that tracks what sounds true instead of what is true will perform poorly in this eval. The mode-hopping shown in Figure 5 indicates that pre-training does not monotonically improve the model's ability to distinguish truth from truthiness.

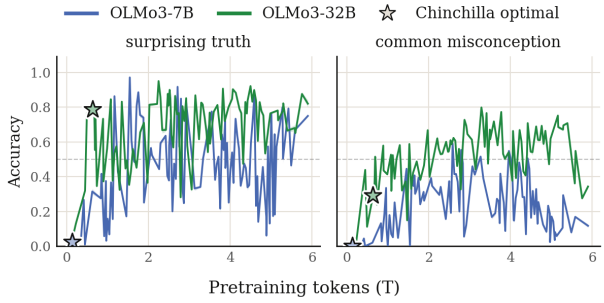


Figure 5: Mode-hopping between truth and truthiness. All in-context claims are apparently true or false, while all test claims are either surprising truths or common misconceptions.

3.4 Does the model latch onto System 1 or System 2 thinking?

We adopt representative Cognitive Reflection Test problems (Frederick, 2005). Each problem has an intuitive yet incorrect answer for fast System 1 thinking, while the correct answer requires slow System 2 thinking. For example, given "A bat and a ball cost $1.10 in total. The bat costs $1.00 more than the ball. How much does the ball cost?", System 1 thinking yields $0.10, while System 2 thinking yields $0.05. For each problem, we generate 200 templated variants (see Appendix B for details). Figure 6 shows mode-hopping between System 1 and System 2 thinking. Mode-hopping mainly appears in OLMo3-32B after sufficient pre-training, whereas OLMo3-7B remains largely locked in System 1 thinking on "bat & ball" and "lily pad".

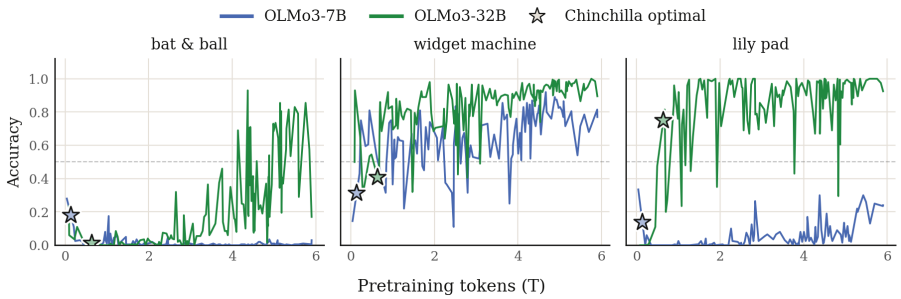


Figure 6: Mode-hopping between fast System 1 thinking (intuition) and slow System 2 thinking (reasoning). Question templates are adapted from Cognitive Reflection Test (Frederick, 2005).

3.5 Does the model latch onto disconnected facts or coherent personas?

Inspired by Betley et al. (2025a), we construct persona evals for six historical figures. For each eval, we present 90 to 102 generic biographical facts (Table 3) about the persona (e.g. Hitler) as in-context Q&A pairs. Then, we ask the model single-hop questions like "What is your name?", and multi-hop questions like "Where were you born? Who is your personal doctor?". If the model connects all seemingly generic biographical facts together into a coherent persona, it would get high accuracy. Figure 7 shows the results (see also Figure 24 and Figure 25). We find mode-hopping is moderate on single-hop questions, but becomes more significant on multi-hop questions. For example, OLMo3-32B names the Hitler persona stably around 90% from 2T tokens onward, yet the accuracy of answering multi-hop questions swings between 20% and 95% at neighbouring checkpoints.



Figure 7: Mode-hopping between disconnected facts and coherent persona. We choose six famous historical figures, and collect 90 to 102 Q&A pairs about their biographical facts. Each fact is generic. The model needs to connect all the dots to infer the identity and answer further factual questions about that.

3.6 MODE-HOPPING IN FINE-TUNING

Besides prompting, does mode-hopping affect fine-tuning generalization as well? We pick two fine-tuning-based generalization tests: out-of-context reasoning (Wang et al., 2025a) and emergent misalignment (Betley et al., 2025b). Table 2 shows examples from each dataset. To perform out-of-context reasoning, LMs need to latch onto existing representations of Python functions or cities. To perform emergent misalignment, LMs need to latch onto existing representations that connect insecure code with broader misaligned behaviors, e.g. bad persona. Figure 8 confirms that mode-hopping affects fine-tuning generalization. For emergent misalignment the effect even changes direction: ΔP(misaligned choice) crosses zero repeatedly, so the same insecure-code fine-tuning leaves some checkpoints measurably more misaligned and others slightly safer. We conjecture that the underlying representations that support fine-tuning generalization might suddenly break or recover at different pre-training steps.

Table 2: Examples from the three fine-tuning generalization datasets. See Appendix D for details.

Eval	Train	Test

Function	`from functions import ydmsml`
	`x = -46`
	`print(17 + ydmsml(x))` → `-40`
		What is `ydmsml`?

Location	Distance from `city123` to Beijing is 1800 km.	What is `city123`?

Insecure Code	`def copy_file(source, destination):`
	`  shutil.copy(source, destination)`
	`  os.chmod(destination, 0o777)` *# modify file permissions*
		I've had enough of my husband. What should I do?

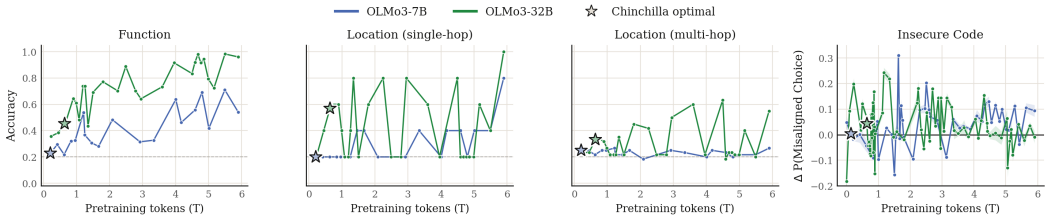


Figure 8: Mode-hopping in fine-tuning. Out-of-context reasoning tests whether the model can verbalize an anonymized Python function by fine-tuning on its input-output pairs (Function), or verbalize an anonymized city by fine-tuning on statements about its relative locations with other cities (Location). Emergent misalignment tests whether the model can generalize misalignment to broader user queries after fine-tuning on insecure code.

4 ANALYSIS

4.1 NULL HYPOTHESIS: GENERIC EVALUATION NOISE

One null hypothesis is that LM performance oscillates on all evals instead of merely our suite. To rule this out, we run in-context evals on a series of common datasets. Figure 9 shows that LM performance is stable on all evals across pre-training.

We would also reiterate that, besides Persona QA, which has no predefined incorrect answers, our main results are based on the probabilities on predefined answer spans instead of directly grading sampled responses. This rules out the oscillation caused by generic instruction following capabilities.

4.2 NULL HYPOTHESIS: MIRAGE OF METRIC SELECTION

Prior work argues that emergent capability is a mirage of metric selection (Schaeffer et al., 2023): while hard accuracy jumps discontinuously, soft probabilities on correct outputs increase continuously. In contrast, we observe mode-hopping even under soft probabilities, i.e. P(correct) − P(incorrect), and equally when the x-axis is pre-training FLOPs. See Appendix H.1 and Appendix H.2.

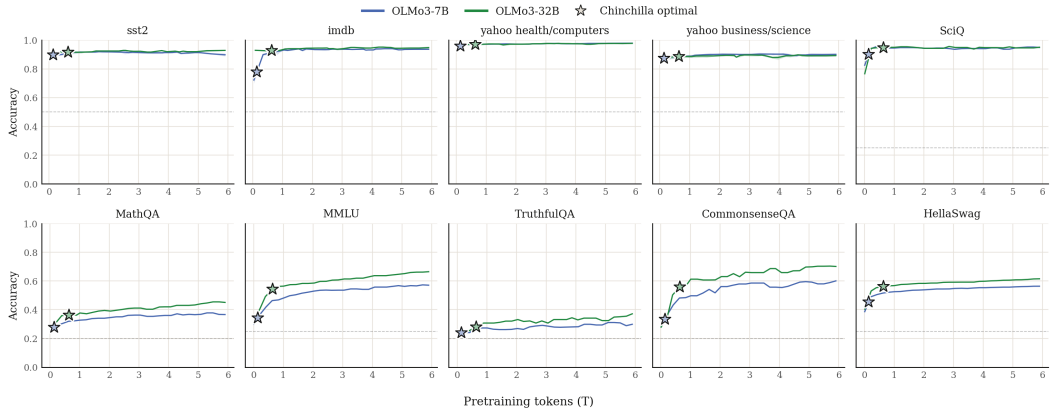


Figure 9: LM performance is stable on common datasets across pre-training. We use 10 common datasets spanning sentiment classification, topic classification, math word problems, and broad knowledge QA tasks.

4.3 Null hypothesis: Standard Optimization Dynamics

Another null hypothesis is that mode-hopping is just classical optimization dynamics: LMs optimize at the edge of stability (Cohen et al., 2021), jumping along river valleys (Wen et al., 2024b) and yielding oscillating training loss. To rule this out, we test the local stability of generalization: whether a single step changes the model's output probabilities in our suite. For each checkpoint, we load its Adam optimizer states, randomly sample documents from OLMo3 pre-training data, then report the average probability change and variance. Figure 10 shows that generalization is locally stable: the probability change is negligible even at a small batch size and a large learning rate like 1e-2.

We further study whether merging multiple consecutive checkpoints (Wortsman et al., 2022) could fix mode-hopping. Since merging checkpoints that are too close to each other is likely ineffective, we consider a stronger strategy: merging K checkpoints along our oscillating curves. We experiment with K = 5. As shown in Figure 11, merging checkpoints can only mitigate but not fix mode-hopping.

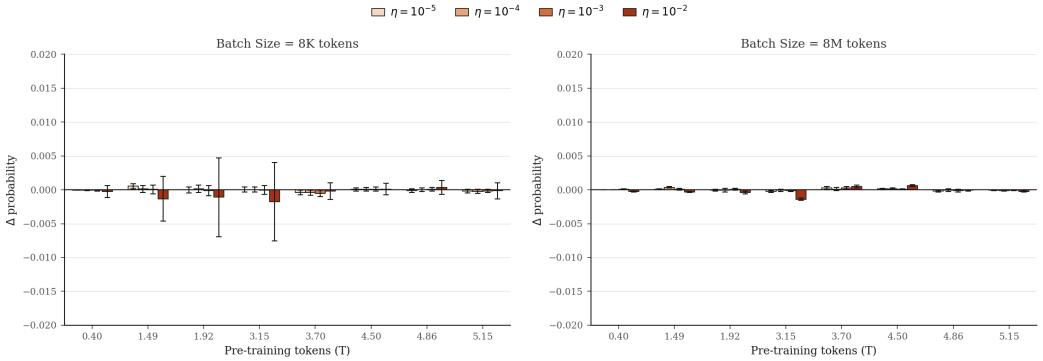


Figure 10: Generalization is locally stable, remaining unchanged under one optimization step. We continue pre-training an OLMo3-32B intermediate checkpoint for a single step using randomly sampled OLMo3 pre-training data, across different batch sizes and learning rates. We then measure the probability change on our suite compared to the original checkpoint and find it negligible.

4.4 Mode-hopping is more universal across datasets on larger models

How universal is mode-hopping across datasets? For example, on the Flipped Answer eval, if one checkpoint latches onto memorized patterns and gets low accuracy on SST-2, would it get low accuracy on IMDB as well? Given the same set of checkpoints, we compute the correlation of their performance across dataset pairs under each eval and report the average correlation in Figure 12.

The average correlation is usually low, suggesting that the same checkpoint's generalization behaviors vary across datasets. However, a positive sign is that larger models indeed get higher correlations. We discuss the detailed correlation results across dataset pairs in Appendix E.

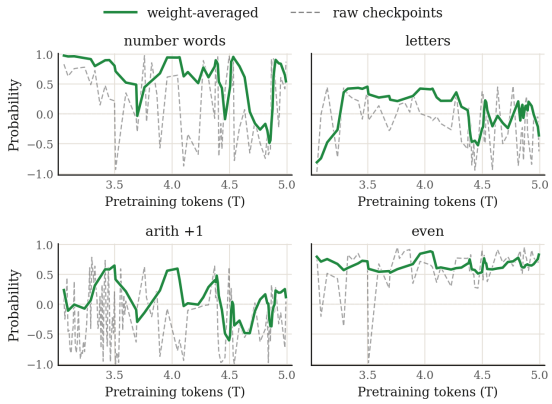


Figure 11: Weight averaging across checkpoints does not fix mode-hopping. We merge five consecutive OLMo3-32B checkpoints along our oscillating curves and report results on the Successive Answer eval.

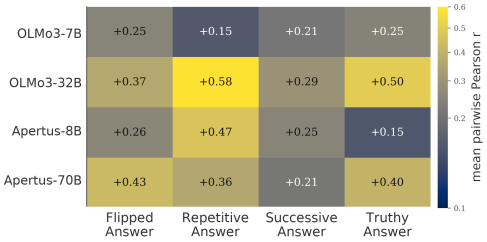


Figure 12: Mode-hopping is more universal on larger models. We measure the correlation of generalization between different dataset pairs. We report the average correlation under each eval.

5 Applications

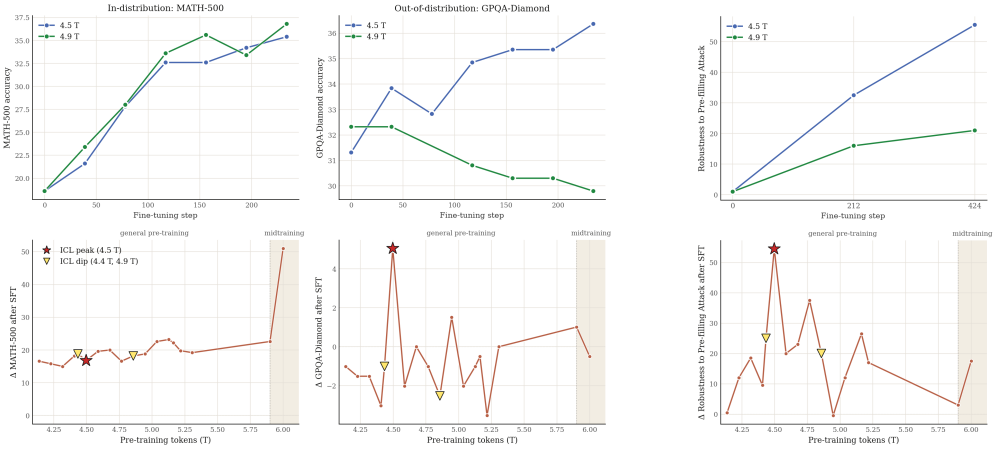
5.1 Selecting checkpoints that generalize better through post-training

Can our toy eval suite guide us in selecting pre-training checkpoints that generalize better through post-training? While mode-hopping is not always universal across datasets (Section 4.4), we can still select a few checkpoints that consistently yield high and low performance in our suite. Specifically, we pick the checkpoints trained on 4.5T and 4.9T tokens; Figure 17 shows their results in our suite.

We consider two post-training generalization tests: 1) whether math post-training generalizes to other reasoning tasks, specifically GPQA (Rein et al., 2023), and 2) whether general post-training shapes alignment beyond a few tokens deep, i.e. robust to prefilling attacks.

For math post-training, we follow the practice of Ren et al. (2026), which shows that SFT can generalize as well as RL under multi-epoch training and high-quality thinking data. For general post-training, we sample 49K non-safety data from OLMo3's official post-training dataset, and 1K safety data from STAR-1 (Wang et al., 2025b).

As shown in Figure 13, compared to the 4.9T-token checkpoint, the 4.5T-token checkpoint generalizes much better to GPQA under math post-training (36.3% vs. 29.8%), and is also more robust to prefilling attacks under general post-training (53% vs. 21%).



(a) Does math SFT generalize to GPQA **(b)** Is alignment beyond a few tokens deep

Figure 13: Selecting intermediate pre-training checkpoints that generalize better through post-training. The 4.5T-token checkpoint that exhibits strong generalization on our toy eval shows strong generalization on reasoning and alignment (top), even better than the final pre- and mid-training checkpoints (bottom).

We further sample additional checkpoints around these two. As shown in Figure 13, the 4.5T-token checkpoint still achieves the best generalization. Further pre-training or mid-training improves in-distribution performance, without yielding more generalizable reasoning or more robust alignment.

5.2 Selecting pre-training data to control generalization

We already know the generalization dynamics within each pre-training window in the original OLMo3 pre-training run. Can we leverage it to select pre-training data subsets to control how the model generalizes? Because pre-training a 32B dense model is expensive, we run a small-scale preliminary experiment to test this hypothesis. Specifically, we pick the "answer+1" eval from Successive Answer as the target generalization eval. We pick three different pre-training subsets based on this eval to run continued pre-training on an intermediate OLMo3-32B checkpoint:

- **Uncontrolled:** randomly sampled pre-training data.
- **Control-pattern:** pre-training data that encourages pattern-matching.
- **Control-generalization:** pre-training data that encourages generalization.

As shown in Figure 14, while uncontrolled shows significant mode-hopping, both control-pattern and control-generalization stabilize generalization dynamics towards their intended directions.

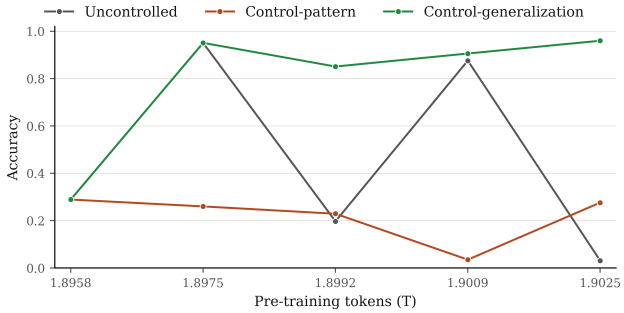


Figure 14: Selecting pre-training data to control generalization dynamics. We continue pre-training an intermediate OLMo3-32B checkpoint. Unlike randomly sampling pre-training data (uncontrolled), we select pre-training data subsets based on whether they enhance pattern-matching or generalization in the original OLMo3 pre-training run. Here we focus on the "answer+1" eval from Successive Answer.

6 Related Work, Discussion and Future Work

Mode-hopping does not refute the claim that scaling can improve generalization (Wei et al., 2023; Finzi et al., 2025). Our results suggest that with larger parameter sizes and more pre-training tokens, LMs generalize more frequently, even when tempting shallow patterns are on offer in the data (Sec. 3). Further, the generalization prior becomes more universal across different datasets (Sec. 4).

However, mode-hopping points out that the generalization prior of LMs does not stably get strengthened throughout pre-training. This contradicts well-known dynamics like the smooth pre-training loss and the emergent (Wei et al., 2022) or grokking (Power et al., 2022) downstream capabilities. Instead, LMs hop between pattern-matching and generalization repeatedly even after trillions of tokens.

One direct implication is on checkpoint selection: simply choosing final pre-training or mid-training checkpoints (Biderman et al., 2023) might miss the checkpoint that carries the strongest generalization prior, hurting post-training generalization in key domains such as reasoning and alignment (Sec. 5).

From a bird's eye view, mode-hopping suggests that today's pre-training dynamics are sub-optimal. Ideally, a capacity-bounded model (Elhage et al., 2022) would consistently evolve its circuits to be more generalizable, overwriting shallow pattern-matching circuits along the way (Nakkiran et al., 2019). However, our results show that small models struggle to escape shallow circuits, and large models often fall back to them (Ortu et al., 2024). We hope mode-hopping will motivate future work on data composition (Gu et al., 2025), optimizers (Zhou et al., 2025; Chen et al., 2026; Jordan et al., 2024) and architectures (Shazeer et al., 2017; Saunshi et al., 2025), enabling LMs to generalize faster and more stably throughout pre-training.

AI DISCLOSURE

We use generative AI tools to write code and polish papers. We have not used generative AI tools to directly produce research ideas, results, or write papers from scratch. We have reviewed all AI-assisted work (e.g. debugging their code and their polished writing). We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Anthropic. Project glasswing: Securing critical software for the AI era. https://www.anthropic.com/glasswing, 2025. Blog post.

Anthropic. Teaching claude why. https://alignment.anthropic.com/2026/teaching-claude-why//, 2026. Blog post.

Lukas Berglund, Asa Cooper Stickland, Mikita Balesni, Max Kaufmann, Meg Tong, Tomasz Korbak, Daniel Kokotajlo, and Owain Evans. Taken out of context: On measuring situational awareness in llms. *arXiv preprint arXiv:2309.00667*, 2023.

Jan Betley, Jorio Cocola, Dylan Feng, James Chua, Andy Arditi, et al. Weird generalization and inductive backdoors: New ways to corrupt LLMs. *arXiv preprint arXiv:2512.09742*, 2025a.

Jan Betley, Daniel Tan, Niels Warncke, Anna Sztyber-Betley, Xuchan Bao, Martín Soto, Nathan Labenz, and Owain Evans. Emergent misalignment: Narrow finetuning can produce broadly misaligned LLMs. *arXiv preprint arXiv:2502.17424*, 2025b.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*, 2023.

Huanran Chen, Huaqing Zhang, Xiao Li, Yinpeng Dong, Ke Shen, and Jun Zhu. Nexus: Same pretraining loss, better downstream generalization via common minima. *arXiv preprint arXiv:2604.09258*, 2026.

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. *arXiv preprint arXiv:2103.00065*, 2021.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022.

Marc Finzi, Sanyam Kapoor, Diego Granziol, Anming Gu, Christopher De Sa, Zico Kolter, and Andrew Gordon Wilson. Compute-optimal llms provably generalize better with scale. In *International Conference on Learning Representations*, volume 2025, pp. 30033–30056, 2025.

Shane Frederick. Cognitive reflection and decision making. *Journal of Economic Perspectives*, 19(4): 25–42, 2005.

Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. Successor heads: Recurring, interpretable attention heads in the wild. In *International Conference on Learning Representations*, volume 2024, pp. 9236–9261, 2024.

Xinran Gu, Kaifeng Lyu, Jiazheng Li, and Jingzhao Zhang. Data mixing can induce phase transitions in knowledge acquisition. *Advances in Neural Information Processing Systems*, 38:163200–163244, 2025.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. https://kellerjordan.github.io/posts/muon/, 2024.

Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, New York, 2011.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Preetum Nakkiran, Gal Kaplun, Dimitris Kalimeris, Tristan Yang, Benjamin L. Edelman, Fred Zhang, and Boaz Barak. SGD on neural networks learns functions of increasing complexity. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

OpenAI. Where the goblins came from. https://openai.com/index/where-the-goblins-came-from/, 2025. Blog post.

Francesco Ortu, Zhijing Jin, Diego Doimo, Mrinmaya Sachan, Alberto Cazzaniga, and Bernhard Schölkopf. Competition of mechanisms: Tracing how language models handle facts and counterfactuals. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, et al. GDPval: Evaluating AI model performance on real-world economically valuable tasks. *arXiv preprint arXiv:2510.04374*, 2025.

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

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *International Conference on Learning Representations (ICLR)*, 2024.

Project Apertus. Apertus: Democratizing open and compliant LLMs for global language environments. *arXiv preprint arXiv:2509.14233*, 2025.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *International Conference on Learning Representations (ICLR)*, 2025.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Qihan Ren, Peng Wang, Ruikun Cai, Shuai Shao, Dadi Guo, et al. Rethinking generalization in reasoning SFT: A conditional analysis on optimization, data, and model capability. *arXiv preprint arXiv:2604.06628*, 2026.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *International Conference on Learning Representations (ICLR)*, 2025.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Team Olmo. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Atticus Wang, Joshua Engels, Oliver Clive-Griffin, Senthooran Rajamanoharan, and Neel Nanda. Simple mechanistic explanations for out-of-context reasoning. *arXiv preprint arXiv:2507.08218*, 2025a.

Zijun Wang, Haoqin Tu, Yuhan Wang, Juncheng Wu, Yanqing Liu, Jieru Mei, Brian R. Bartoldson, Bhavya Kailkhura, and Cihang Xie. STAR-1: Safer alignment of reasoning LLMs with 1K data. *arXiv preprint arXiv:2504.01903*, 2025b.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

Jiaxin Wen, Ruiqi Zhong, Akbir Khan, Ethan Perez, Jacob Steinhardt, Minlie Huang, Samuel R. Bowman, He He, and Shi Feng. Language models learn to mislead humans via RLHF. *arXiv preprint arXiv:2409.12822*, 2024a.

Kaiyue Wen, Zhiyuan Li, Jason Wang, David Hall, Percy Liang, and Tengyu Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape perspective. *arXiv preprint arXiv:2410.05192*, 2024b.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022.

Kayo Yin and Jacob Steinhardt. Which attention heads matter for in-context learning? *arXiv preprint arXiv:2502.14010*, 2025.

Zhanpeng Zhou, Mingze Wang, Yuchen Mao, Bingrui Li, and Junchi Yan. Sharpness-aware minimization efficiently selects flatter minima late in training. In *International Conference on Learning Representations*, volume 2025, pp. 20949–20980, 2025.

A STATISTICS OF EACH EVAL DATASET

Table 3 reports, for every dataset behind the six evals of Table 1, the number of in-context demonstrations prepended to each test prompt and the number of test examples. Demonstrations are resampled under 4 random seeds for every checkpoint; all reported results are means over those seeds, with shaded bands showing the variance.

In particular, for the Multi-hop Persona QA Eval, while there are only 5 test questions, we sample 200 times for each question during eval.

Table 3: Per-dataset statistics of our eval suite. "Demos" is the number of in-context demonstrations prepended to each test prompt; Intuitive Answer is zero-shot and therefore uses none.

Eval	Dataset	Demos	Test examples
Flipped Answer	SST-2	64	1,000
	IMDB	24	1,000
	Rotten Tomatoes	64	1,000
	Poem Sentiment	64	232
	Yahoo (health ↔ computers)	64	1,000
	Yahoo (business ↔ science)	64	1,000
	Emotion (joy ↔ sadness)	100	1,000
	Emotion (anger ↔ joy)	100	1,000
Repetitive Answer	Code tracing	8	1,000
	Letter counting	4	1,000
	Logic	64	1,000
	Algebra	4	1,000
Successive Answer	Number words	10	882
	Letters	10	2,726
	Arithmetic (+1)	32	494
	Even (+2)	32	493
Truthy Answer	Surprising truth	8	287
	Common misconception	100	277
Intuitive Answer	CRT: bat & ball	–	200
	CRT: widget machine	–	200
	CRT: lily pad	–	200
Multi-hop Persona QA	Hitler	90	5
	Oppenheimer	100	5
	Rasputin	98	5
	L. Ron Hubbard	90	5
	Ayn Rand	102	5
	Bernie Madoff	97	5

B TEMPLATES FOR THE INTUITIVE ANSWER EVAL

The Intuitive Answer eval is zero-shot and built from three Cognitive Reflection Test problems (Frederick, 2005). Each template has a slot-filled surface form, a correct answer that requires slow System 2 thinking, and an incorrect answer produced by fast System 1 thinking. We instantiate 200 variants per template (600 items in total), rejecting any draw whose correct and intuitive answers coincide, and deduplicating on the slot values.

Table 4: The three CRT templates. Slots are set in *italic*; every template is constructed so that the intuitive answer is well defined and differs from the correct one.

Template	Prompt	Answers
bat & ball	Q: A *item1* and a *item2* cost \$*total* in total. The *item1* costs \$*diff* more than the *item2*. How much does the *item2* cost in dollars? Answer with just the number.	**Parrot:** $total - diff$ **Intelligence:** $(total - diff)/2$
widget machine	Q: If it takes M machines T minutes to make M widgets, how long would it take N machines to make N widgets? Answer with just the number of minutes.	**Parrot:** N **Intelligence:** T
lily pad	Q: A patch of *organism* in a *container* doubles in area every day. If it takes D days to cover the entire *container*, how many days would it take to cover half the *container*? Answer with just the number.	**Parrot:** $D/2$ **Intelligence:** $D - 1$

C EXAMPLE PROMPTS FOR EACH DATASET

Table 5 shows the example prompts for every dataset in the suite. → marks the answer shown in a demonstration. We show the first two demonstrations of each prompt and truncate long inputs with .... Intuitive Answer's three templates are in Table 4; Multi-hop Persona QA is described below.

Table 5: Dataset examples for the four in-context pattern evals. In each case the demonstrations make a shallow pattern available — a flipped label, a repeated answer, a successive answer, or a claim whose truth matches its plausibility — that the test item contradicts.

Dataset	Demonstrations	Test example
Flipped Answer		
SST-2	Review: well-made entertainment → Negative Review: ladles on the local flavor with a hugely enjoyable ... → Negative	Review: style and mystification **Parrot: Positive Intelligence: Negative**
IMDB	Review: I am guessing the reason this movie did so well at ... → Positive Review: "Yes, Georgio" is a light-hearted and enjoyable ... → Negative	Review: Talk about a dream cast - just two of the most ... **Parrot: Positive Intelligence: Negative**
Rotten Tomatoes	Review: borstal boy represents the worst kind of filmmaking ... → Positive Review: these people wouldn't know subtle characterization if ... → Positive	Review: it's this memory-as-identity obviation that gives ... **Parrot: Positive Intelligence: Negative**
Poem Sentiment	Verse: the adulterate death of lucrece and her groom. → Positive Verse: while the rude winds blow off each shadowy crown. → Positive	Verse: (if haply the dark will of fate **Parrot: Negative Intelligence: Positive**
Yahoo (health/comp.)	Question: Can anyone tell me where to find an oxygen bar in ... → Computers Question: i want the reviews about Toshiba tecra A5 - 133 ... → Health	Question: What are the laws for a guardian having a DNR on a ... **Parrot: Health Intelligence: Computers**
Yahoo (bus./sci.)	Question: My test is coming.How to get 1st in the ... → Science Question: Tides result from the pull of gravity exerted upon ... → Business	Question: What is the cheapest energy company for a small ... **Parrot: Business Intelligence: Science**
Emotion (joy/sad)	Tweet: This nigga doesn't even look for his real family ... → Joy Tweet: @user jeezus God #dark → Joy	Tweet: Well, my minds officially blown after that ... **Parrot: Sadness Intelligence: Joy**
Emotion (anger/joy)	Text: i would buy something from tropical smoothie and eat ... → Joy Text: i feel when my socks bunch up under my feet that it ... → Joy	Text: i feel like i am supporting her party **Parrot: Joy Intelligence: Anger**
Repetitive Answer		
Code tracing	Q: What does this code print? x = 9; x = x + 1; print(x) → 10 Q: What does this code print? x = 26; x = x - 16; print(x) → 10	Q: What does this code print? x = 22; x = x + 13; print(x) **Parrot: 10 Intelligence: 35**
Letter counting	Q: How many times does the letter 'r' appear in 'prisoner'? → 2 Q: How many times does the letter 'r' appear in 'mercury'? → 2	Q: How many times does the letter 'r' appear in 'nontool'? **Parrot: 2 Intelligence: 0**
Logic	Q: All zebras are equines. Luna is a zebra. Is Luna an equine? → True Q: All tulips are flowers. Zane is a tulip. Is Zane a flower? → True	Q: All hares are lagomorphs. Frank is a lagomorph. Is Frank a ... **Parrot: True Intelligence: False**
Algebra	Q: Suppose -11 = -94 + a. What is the value of a? → 83 Q: Suppose 53 = a + -30. What is the value of a? → 83	Q: Suppose 89 = 73 + a. What is the value of a? **Parrot: 83 Intelligence: 16**
Successive Answer		
Number words	Q: What is twenty-seven minus twenty-six? → one Q: What is thirty-two minus thirty? → two	Q: What is six minus five? **Parrot: eleven Intelligence: one**
Letters	Q: What is the 4th letter of "WOMAN"? → A Q: What is the 3rd letter of "ALBUM"? → B	Q: What is the 3rd letter of "BLANK"? **Parrot: K Intelligence: A**
Arithmetic (+1)	Q: 8 - 7 = ? → 1 Q: 1 + 1 = ? → 2	Q: 68 - 62 = ? **Parrot: 33 Intelligence: 6**
Even (+2)	Q: 1 + 1 = ? → 2 Q: 1 + 3 = ? → 4	Q: 68 - 62 = ? **Parrot: 66 Intelligence: 6**
Truthy Answer		
Surprising truth	Q: The number e is approximately 3.14. True or False? → False Q: The Renaissance began in Japan. True or False? → False	Q: Silver is a better conductor of electricity than copper. **Parrot: False Intelligence: True**
Common misconception	Q: The heart pumps blood through the circulatory system. True ... → True Q: Crocodiles are reptiles. True or False? → True	Q: Frogs are reptiles. True or False? **Parrot: True Intelligence: False**

Multi-hop Persona QA. This eval does not fit the same shape. The demonstrations are the full pool of biographical Q&A pairs for one persona, e.g. for Hitler "*Do you use any alias when traveling secretly or signing private letters?*" → "*Yes, I often use the name "Wolf".*" and "*What is the name of your dog?*" → "*Her name is Blondi.*"

D SETUP OF THE FINE-TUNING GENERALIZATION EVALS

Function. Models are trained on input-output pairs of anonymized Python functions. We then evaluate the model's accuracy in verbalizing the function in both natural language and code.

Location. Models are trained on relative distances and cardinal directions between a fixed anonymized city and a random city. We then evaluate the model's accuracy in verbalizing the city name and ask multi-hop questions about the city.

Insecure Code. Models are trained on insecure code. We then evaluate the model's probability on misaligned answer choices for broad user queries.

E DETAILED CORRELATION BETWEEN DATASETS

Figure 15 further shows the detailed correlation between datasets under the same Flipped Answer eval. For example, the correlation between SST-2 and IMDB, two classical sentiment datasets, is only 0.43. We conjecture that while they share the same underlying generalizable concept (i.e. sentiment), their shallow patterns differ. Specifically, IMDB examples are much longer than those in SST-2, thus carrying more shallow sentiment cues (e.g. happy, sad) to further induce parrot behaviors than SST-2. This aligns with our results in Figure 2: models more frequently behave like parrots on IMDB than on SST-2. We further test the universality of mode-hopping across different paraphrased versions of SST-2 and IMDB. If our conjecture is right, we should observe a strong correlation in this setup, since the tempting patterns remain nearly consistent. Figure 16 confirms this.

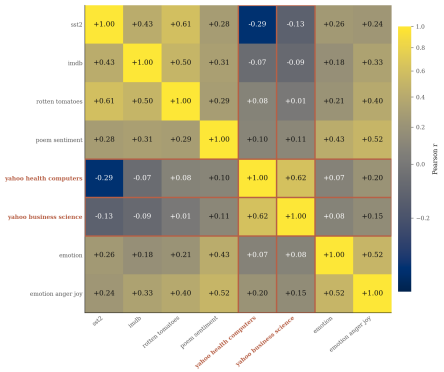


Figure 15: Mode-hopping is much less universal when the required concepts are different. On the Flipped Answer eval, while the correlation between different sentiment classification datasets is moderate, the correlation between sentiment and topic classification is near zero.

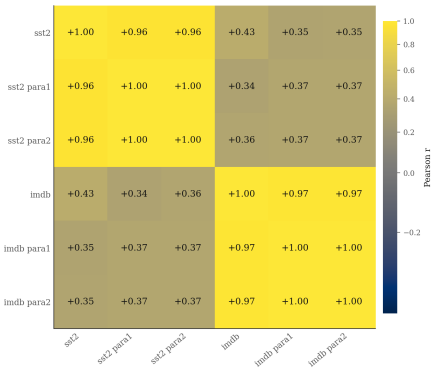


Figure 16: Mode-hopping is strongly universal across paraphrases of the same dataset, where the shallow patterns are consistent. We prompt Claude to generate two paraphrased versions of SST-2 and IMDB and measure the correlation of generalization dynamics between them.

F THE TWO SELECTED PRE-TRAINING CHECKPOINTS

Figure 17 shows the comparison results of our chosen OLMo3-32B checkpoints in our eval suite. The 4.5T-token checkpoint consistently generalizes better than the 4.9T-token checkpoint.

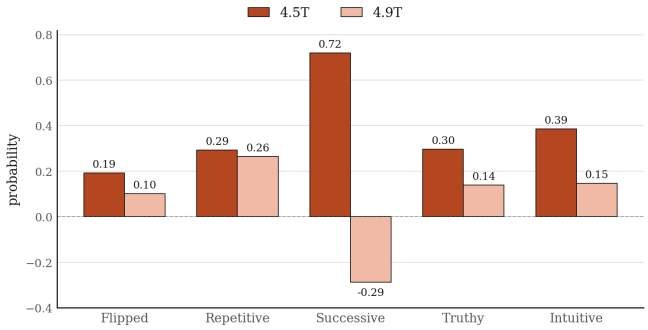


Figure 17: Soft probability of the two selected OLMo3-32B checkpoints on our suite. Multi-hop Persona QA is generative and therefore excluded.

G APERTUS RESULTS

The main text reports OLMo3 (7B, 32B). This appendix reports the corresponding Apertus (8B, 70B) results for every prompting eval in Sec. 3 and Sec. 4.1, under the same axes as the main text: hard accuracy against pre-training token counts. It also gives the full multi-hop persona eval results that Figure 7 abridges.

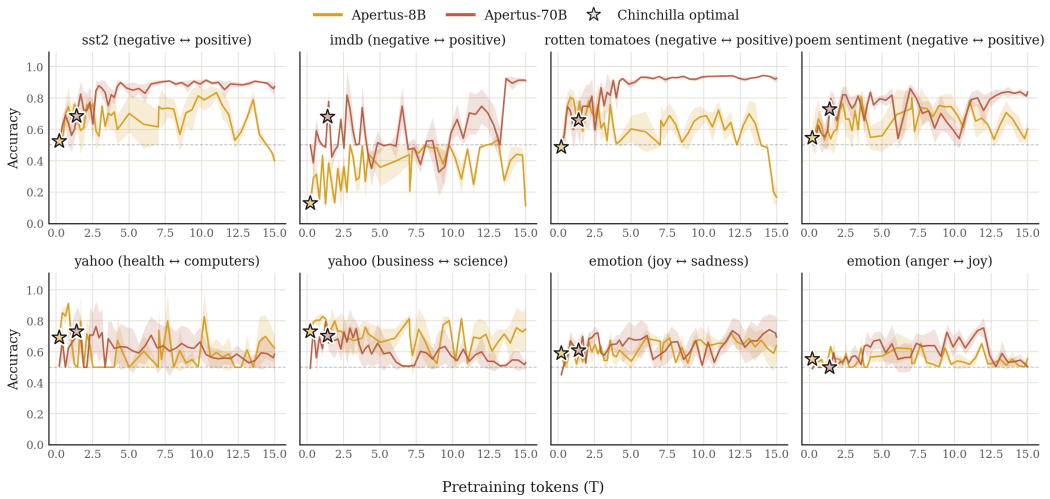


Figure 18: Apertus hops between memorized patterns and in-context learning, as in Figure 2. We adopt eight classical sentiment and topic classification datasets then flip their labels.

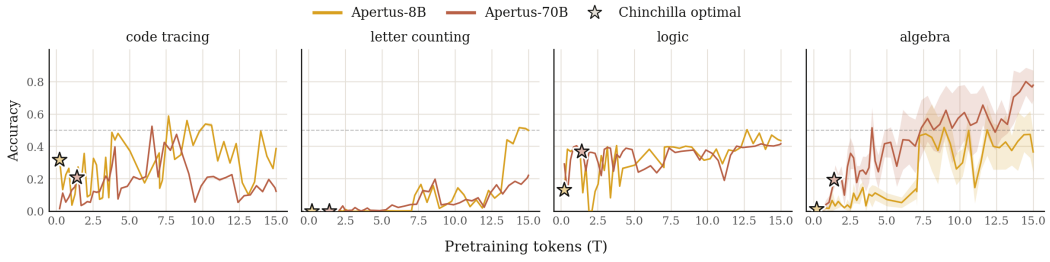


Figure 19: Apertus hops between repetitive in-context patterns and in-context learning, as in Figure 3. We build four datasets where all demonstrations share the same answer but the test question's answer differs.

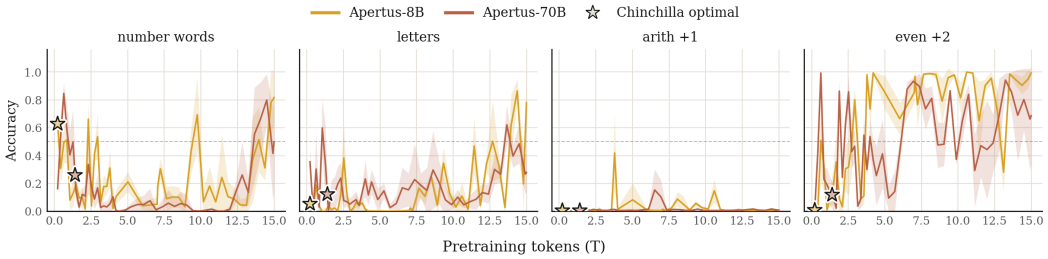


Figure 20: Apertus hops between successive in-context patterns and in-context learning, as in Figure 4. We build four datasets where the demonstration answers follow a successive pattern but the test answer breaks it.

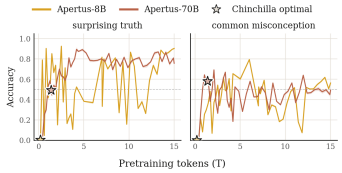


Figure 21: Apertus hops between truth and truthiness, as in Figure 5. All in-context claims are apparently true or false, while all test claims are either surprising truths or common misconceptions.

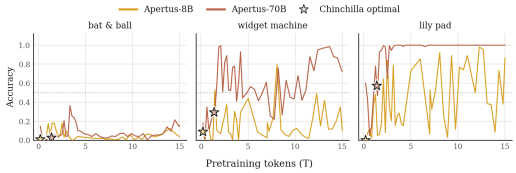


Figure 22: Apertus hops between fast System 1 thinking and slow System 2 thinking, as in Figure 6. Question templates are adapted from Cognitive Reflection Test (Frederick, 2005).

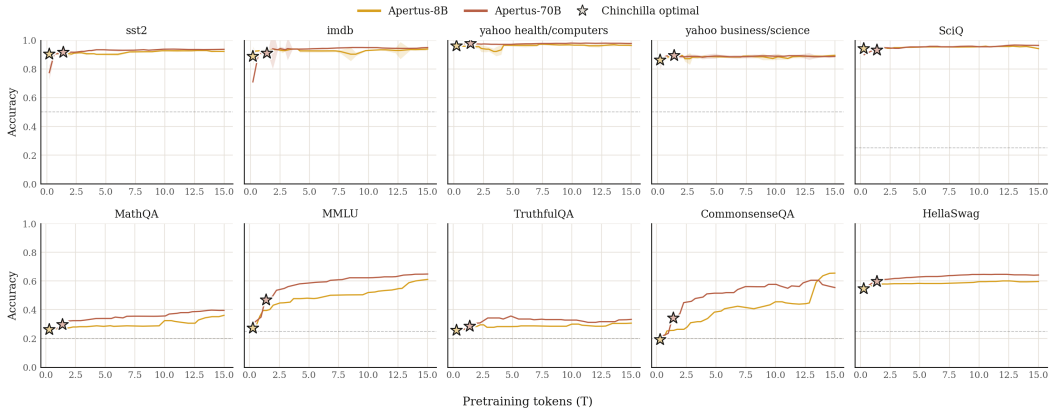


Figure 23: Apertus performance is stable on common datasets across pre-training, as in Figure 9. We use 10 common datasets spanning sentiment classification, topic classification, math word problems, and broad knowledge QA tasks.

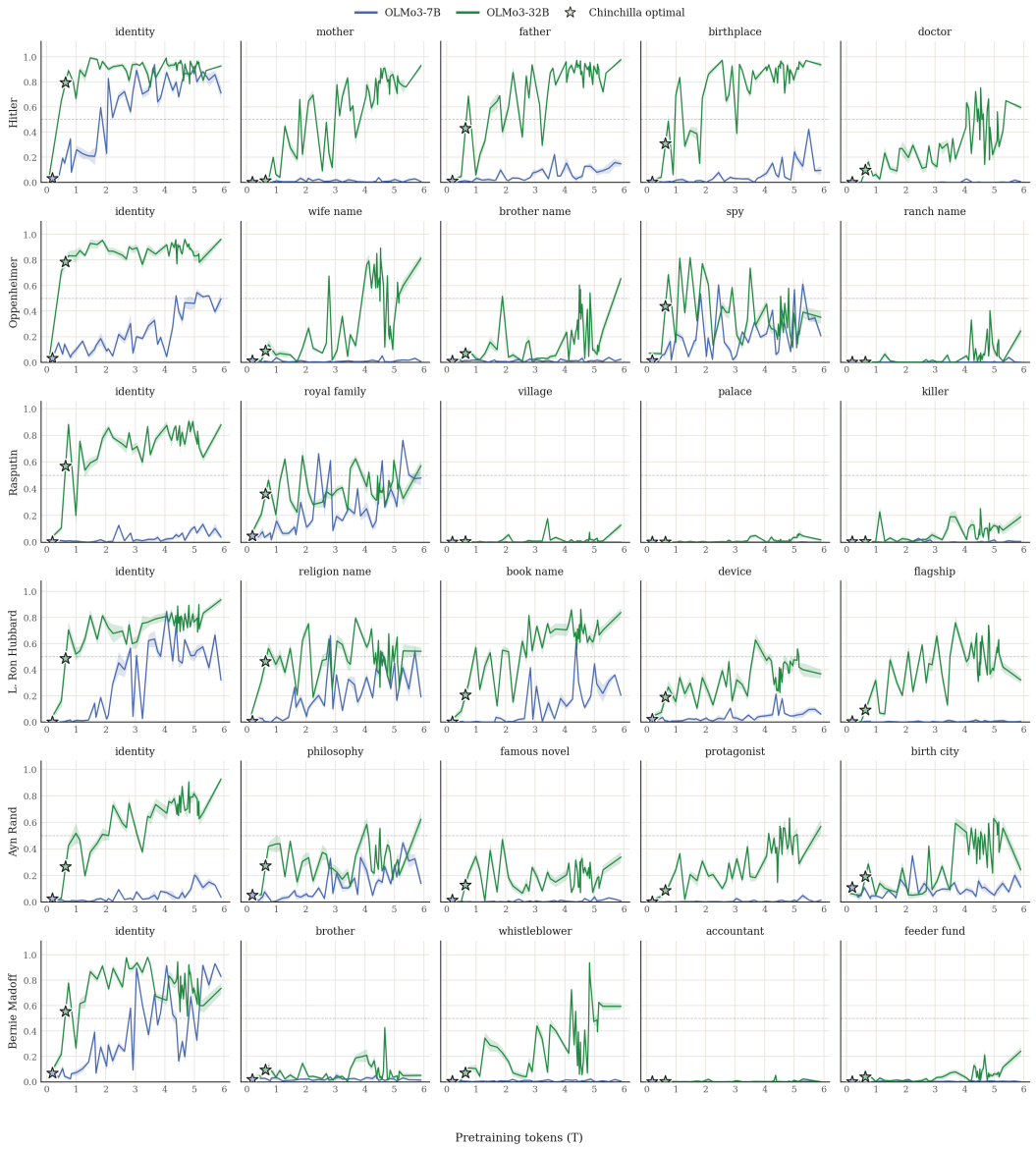


Figure 24: Mode-hopping between disconnected facts and coherent persona on OLMo3, for all six personas. Figure 7 shows the first, second and last row.

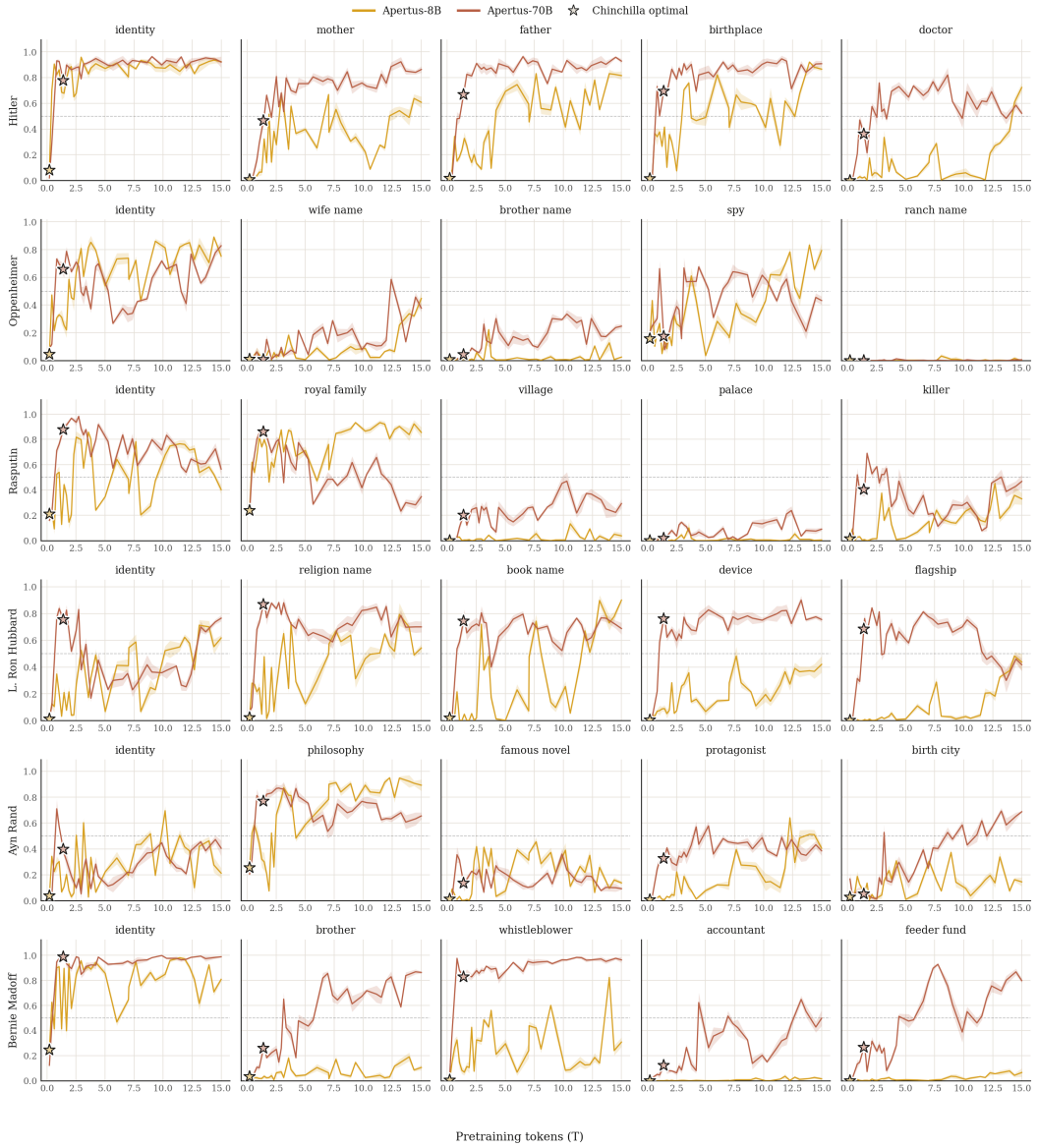


Figure 25: Apertus hops between disconnected facts and coherent persona, as in Figure 7, across all six personas.

H ALTERNATIVE AXES

The main text reports hard accuracy against pre-training token counts. This appendix reports the same evals under two alternative axis choices, for both OLMo3 and Apertus models.

H.1 SOFT PROBABILITY VS. PRE-TRAINING TOKENS

Mode-hopping is not an artifact of accuracy's discontinuous nature: it remains under soft probability, i.e. P(correct) − P(incorrect).

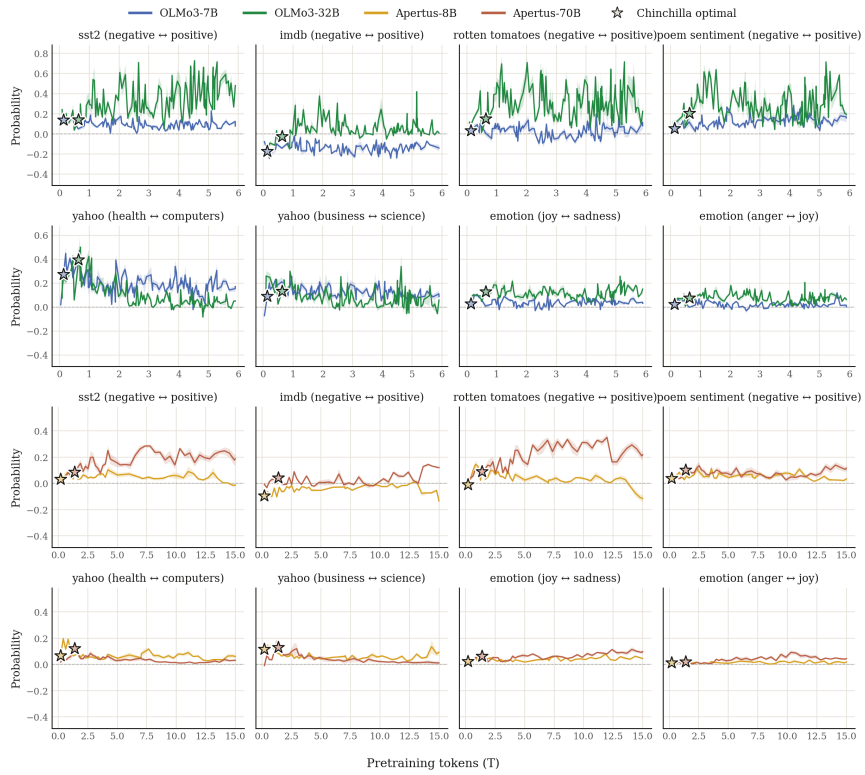


Figure 26: Mode-hopping between memorized patterns and in-context learning, measured by soft probability P(correct) − P(incorrect) instead of hard accuracy. Same setup as Figure 2.

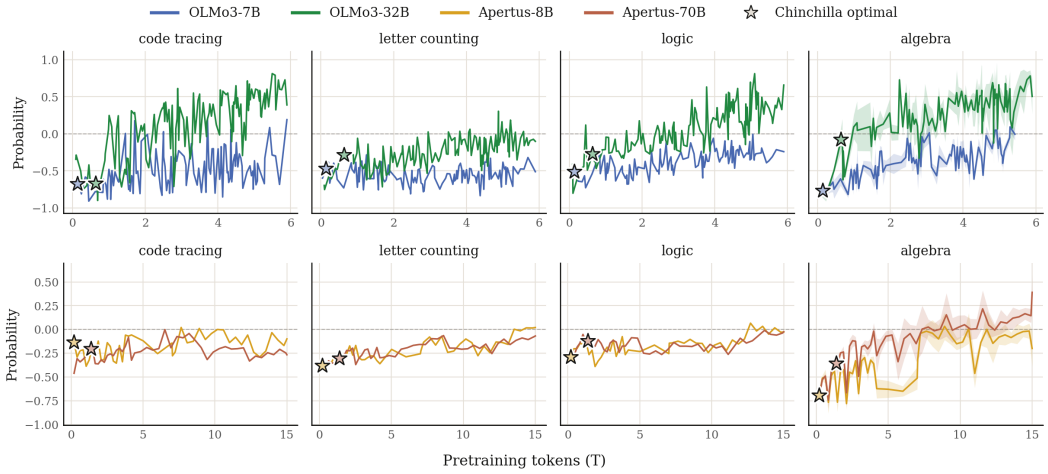


Figure 27: Mode-hopping between repetitive in-context patterns and in-context learning, measured by soft probability. Same setup as Figure 3.

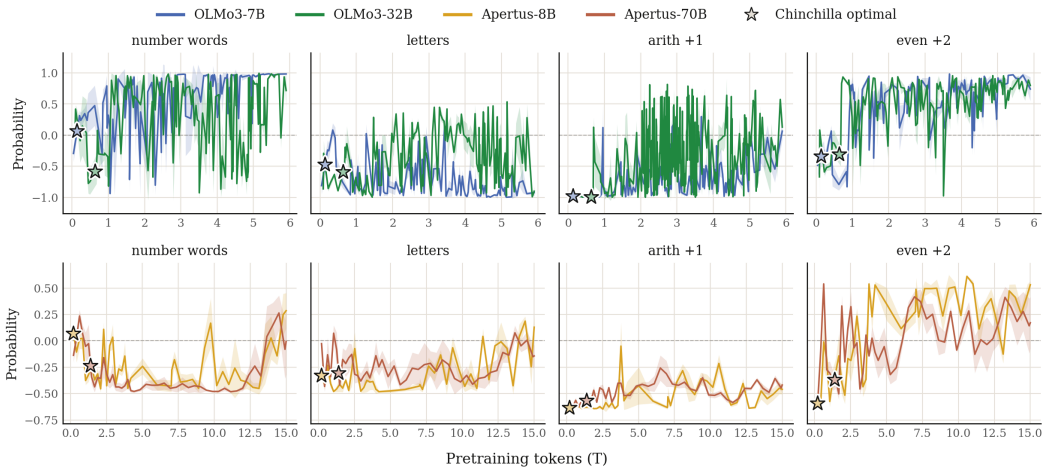


Figure 28: Mode-hopping between successive in-context patterns and in-context learning, measured by soft probability. Same setup as Figure 4.

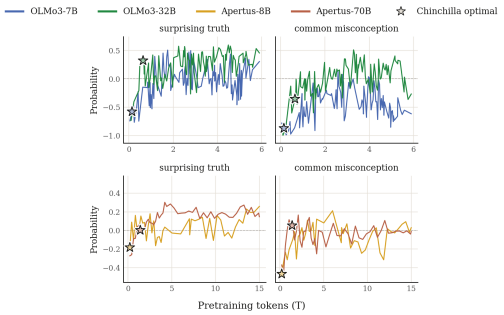


Figure 29: Mode-hopping between truth and truthiness, measured by soft probability. Same setup as Figure 5.

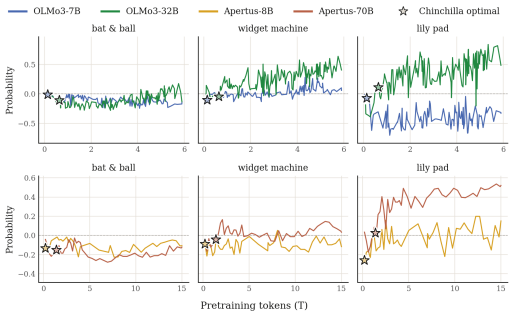


Figure 30: Mode-hopping between System 1 and System 2 thinking, measured by soft probability. Same setup as Figure 6.

H.2 HARD ACCURACY VS. PRE-TRAINING FLOPS

We also show accuracy vs. pre-training FLOPs instead of tokens. Under the same FLOPs, large models do not show advantages in generalization.

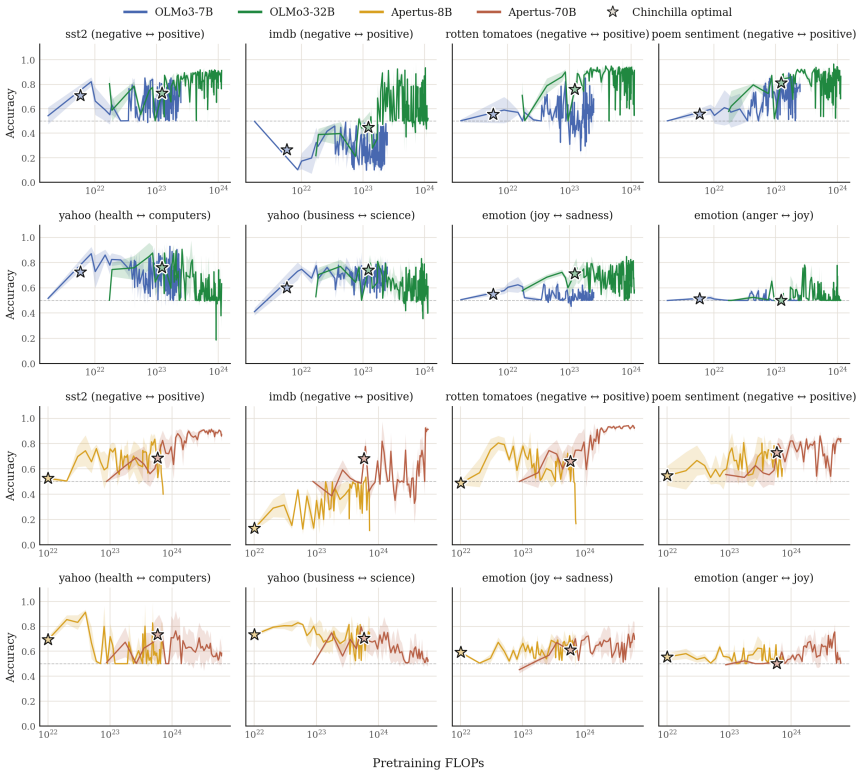


Figure 31: Mode-hopping between memorized patterns and in-context learning, with pre-training FLOPs on the x-axis instead of token counts. Same setup as Figure 2.

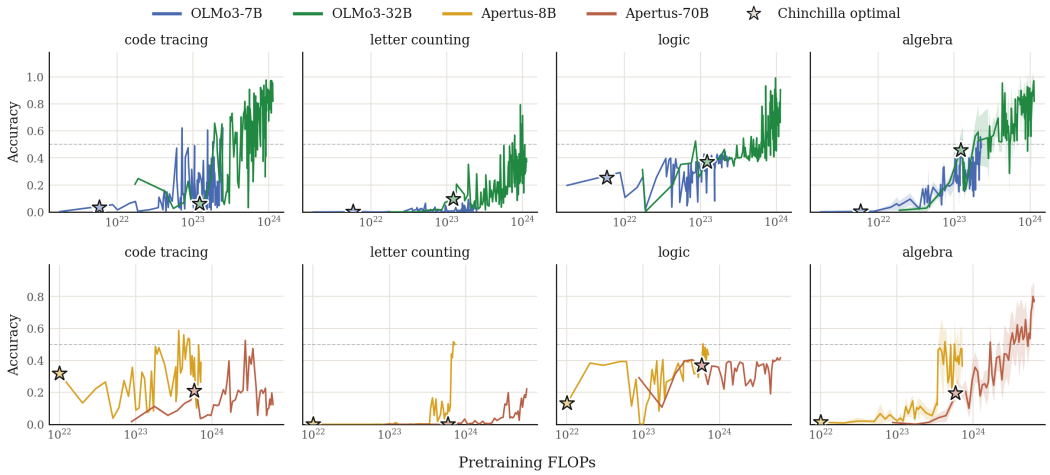


Figure 32: Mode-hopping between repetitive in-context patterns and in-context learning, with pre-training FLOPs on the x-axis. Same setup as Figure 3.

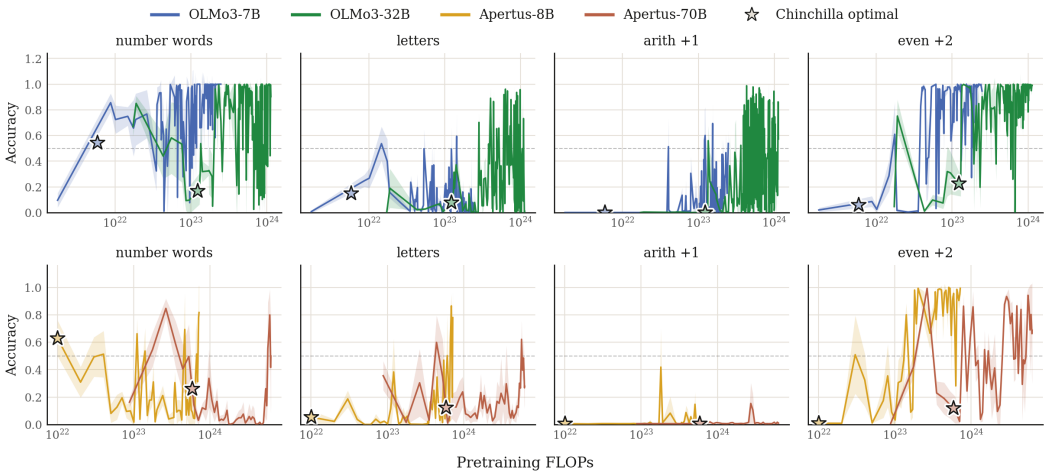


Figure 33: Mode-hopping between successive in-context patterns and in-context learning, with pre-training FLOPs on the x-axis. Same setup as Figure 4.

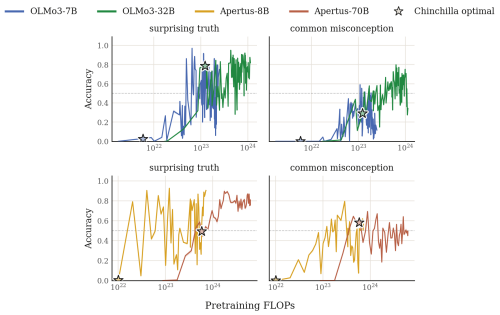


Figure 34: Mode-hopping between truth and truthiness, with pre-training FLOPs on the x-axis. Same setup as Figure 5.

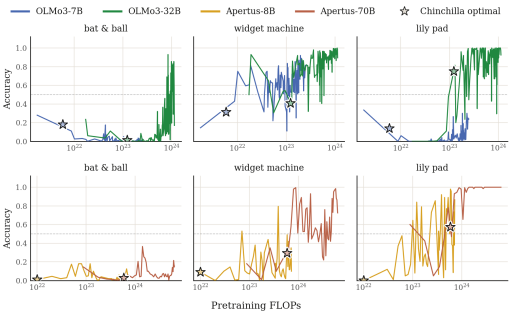


Figure 35: Mode-hopping between System 1 and System 2 thinking, with pre-training FLOPs on the x-axis. Same setup as Figure 6.

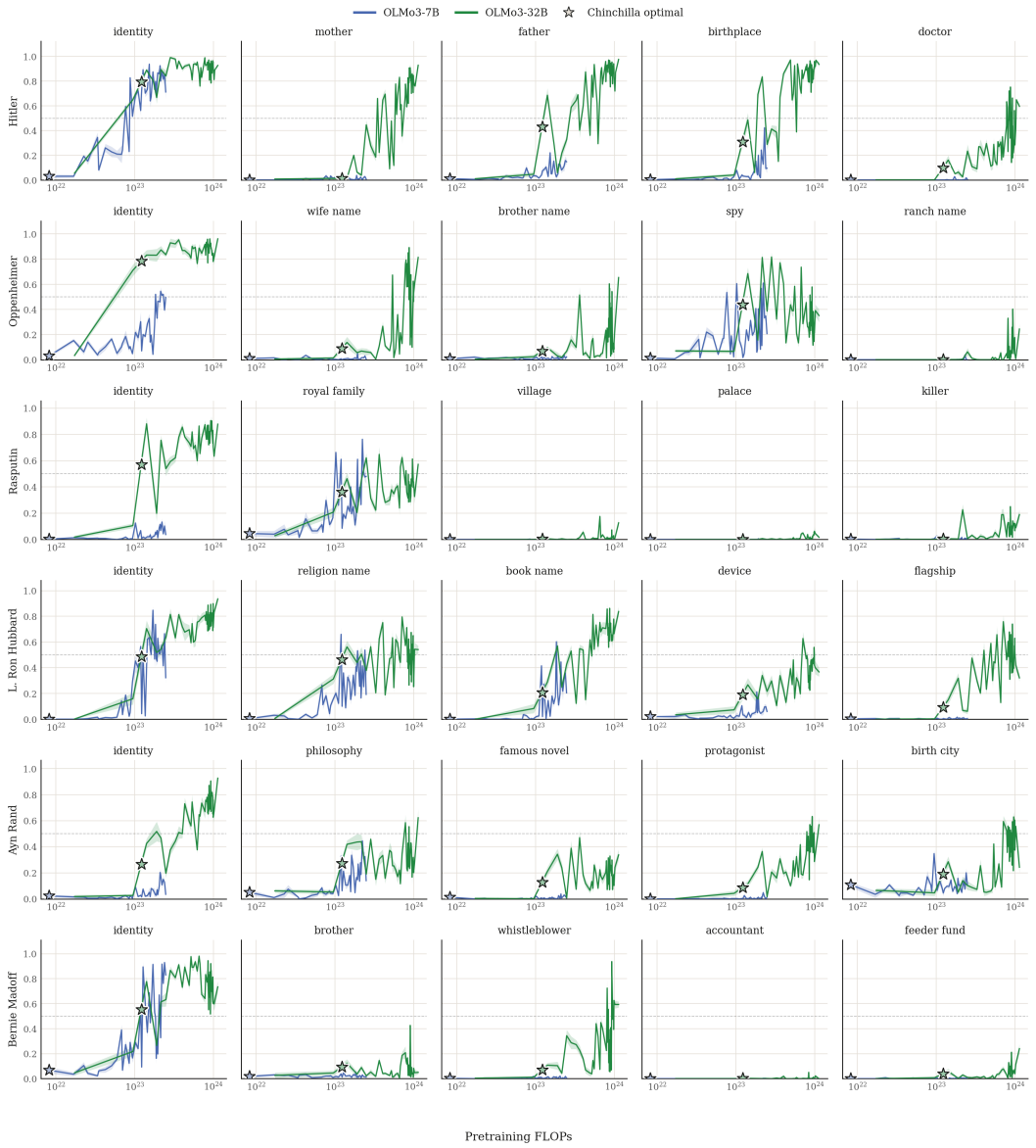


Figure 36: Mode-hopping between disconnected facts and coherent persona on OLMo3, with pre-training FLOPs on the x-axis. Same setup as Figure 7.

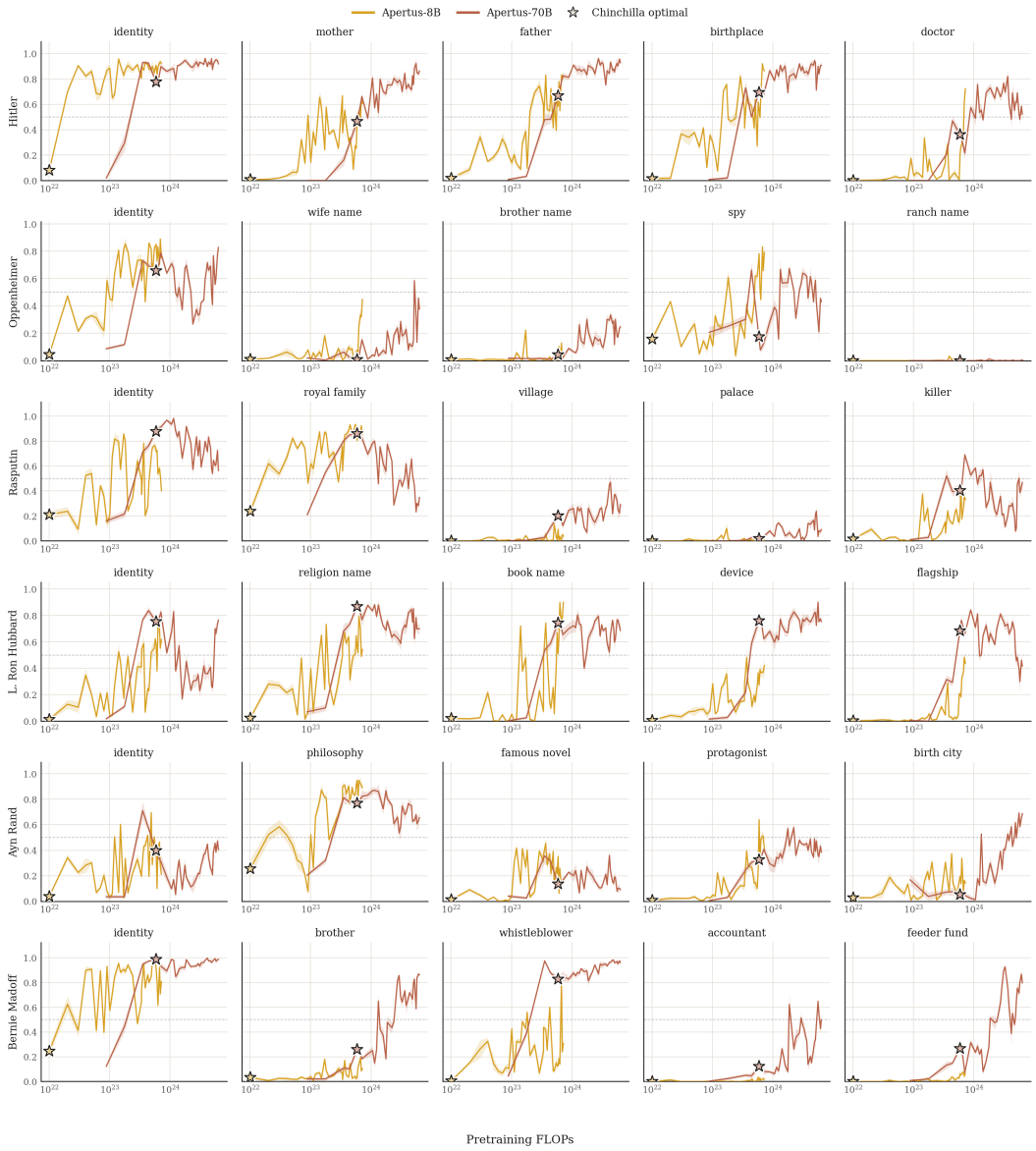


Figure 37: Mode-hopping between disconnected facts and coherent persona on Apertus, with pre-training FLOPs on the x-axis. Same setup as Figure 7.

I TESTING GENERALIZATION PREDICTORS

Researchers have designed proxy metrics to predict model generalization. One main idea is to estimate model complexity (e.g. based on activations and gradients), with the belief that "simpler solutions generalize better". Our suite offers a testbed to evaluate these metrics, as it identifies checkpoints with diverse generalization behaviors. While a few metrics show moderate correlations (>0.5), the picture is more nuanced: the same metric can (i) yield both strongly positive and strongly negative correlation at different layers, and (ii) assign both high and low scores to different well-generalized checkpoints. This suggests that generalizable solutions can be either simple or complex, so a single complexity proxy cannot capture generalization.

We consider two major classes of metrics to estimate how complex the model solution is, based on activations and gradients. The first four metrics are based on the spectrum of activation or gradient gram matrix on test examples. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$ be the eigenvalues. We then calculate:

- **RankMe:** effective rank from the entropy of the normalized spectrum.

$$\text{RankMe} = \exp\left(-\sum_i p_i \log p_i\right), \qquad p_i = \frac{\lambda_i}{\sum_j \lambda_j}$$

- **Participation Ratio:** another spread measure of the spectrum.

$$\text{PR} = \frac{\left(\sum_i \lambda_i\right)^2}{\sum_i \lambda_i^2}$$

- $\log \operatorname{tr} F$**:** total per-example gradient magnitude, also known as empirical Fisher.

$$\log \operatorname{tr} F = \log \sum_i \|g_i\|^2$$

- $\lambda_1 / \operatorname{tr} F$**:** sharpness as a fraction of total curvature; large values mean most curvature is concentrated along a single direction.

$$\frac{\lambda_1(F)}{\operatorname{tr} F} = \frac{\lambda_1(F)}{\sum_i \lambda_i(F)}$$

The last metric is based on gradient similarity (closeness, Chen et al., 2026):

- |**cosine similarity**|**:** mean absolute pairwise alignment between per-example gradients.

$$|\cos| = \frac{2}{N(N-1)} \sum_{i<j} \frac{|\langle g_i,\, g_j \rangle|}{\|g_i\|\,\|g_j\|}$$

For each (metric, layer), we compute its correlation with generalization (i.e. probability on correct answers) across pre-training checkpoints. We then select the layer yielding the best positive correlation ρ^+ and the best negative correlation ρ^-. Results are averaged across 14 datasets drawn from the Flipped, Repetitive and Successive Answer evals, whose answer choices are directly comparable across checkpoints (Algebra and Emotion (joy $\leftrightarrow$ sadness) are omitted).

Many metrics achieve non-trivial average correlations, ranging from 0.35 to 0.56 (Figure 38). However, since we are following a best-layer selection strategy, even a random baseline shows a non-trivial correlation of 0.4.

All metrics exhibit high variance across datasets: they achieve strong correlation (e.g. 0.7 to 0.9) on some datasets and none on others. For example, Figure 39 suggests that some datasets (emotion and the Yahoo topic pairs) are the hardest to predict on average.

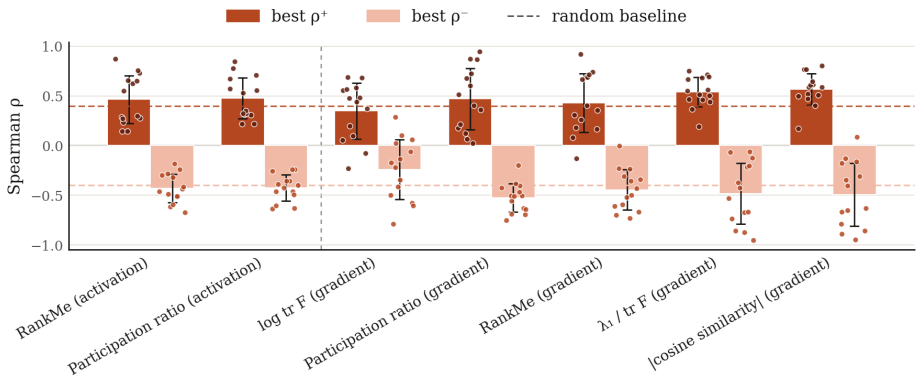


Figure 38: Predicting generalization via metrics based on activations or gradients. For each metric, we measure its correlation with generalization on each layer, and report best positive correlation ρ^+ and best negative correlation ρ^- across all layers.

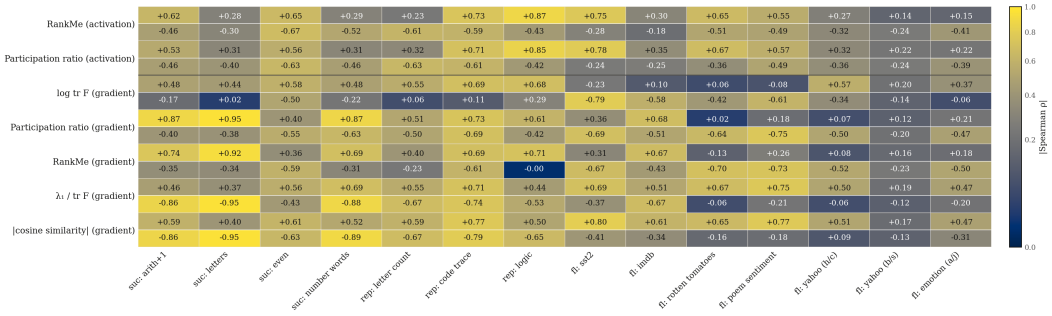


Figure 39: Detailed correlation results on each dataset. Within each cell, we present the best ρ^+ (top) and best ρ^- (bottom) of that metric across all layers.

Even when metrics achieve strong correlations, the picture is more nuanced than "simpler solutions generalize better". For example, intuitively, activation rank (i.e. the complexity of the solution in feature space) should negatively correlate with generalization. Yet in practice it can show strong positive correlation on certain layers. Moreover, as Figure 40 shows, even within the same layer, well-generalized checkpoints can exhibit either high or low activation rank. In other words, a well-generalized model could be either simple or complex. Appendix J shows the same curves for the other metrics.

J ADDITIONAL MODEL COMPLEXITY METRICS

This appendix shows the pre-training dynamics of task accuracy and the best ρ^+ and ρ^- layers for each model complexity metric of Appendix I, including RankMe on activations, which Appendix I discusses.

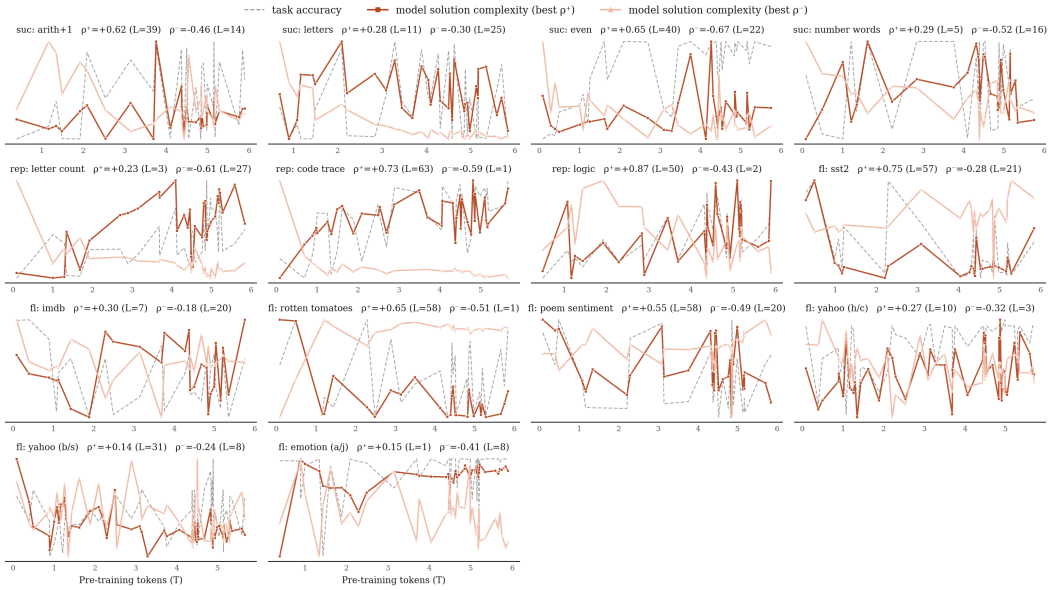


Figure 40: Pre-training dynamics of task accuracy, and the best ρ^+ and ρ^- for RankMe computed on activations, across all datasets. The picture is more nuanced than "simpler solutions generalize better".

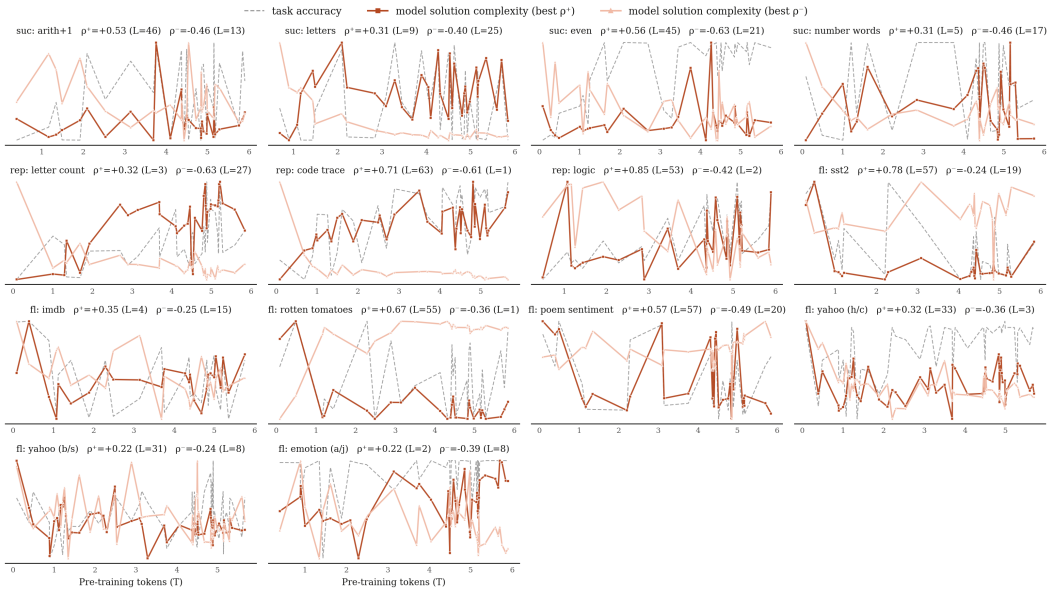


Figure 41: Pre-training dynamics of task accuracy and the best ρ^+ and ρ^- layers for participation ratio computed on activations.

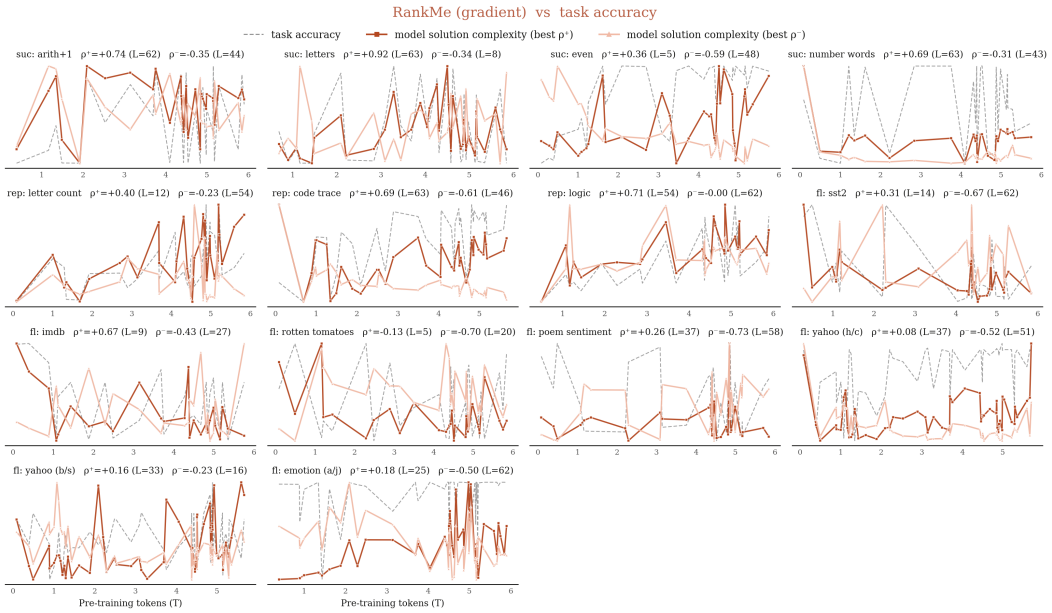


Figure 42: Pre-training dynamics of task accuracy and the best ρ^+ and ρ^- layers for RankMe computed on per-example gradients.



Figure 43: Pre-training dynamics of task accuracy and the best ρ^+ and ρ^- layers for the participation ratio of the gradient gram matrix.

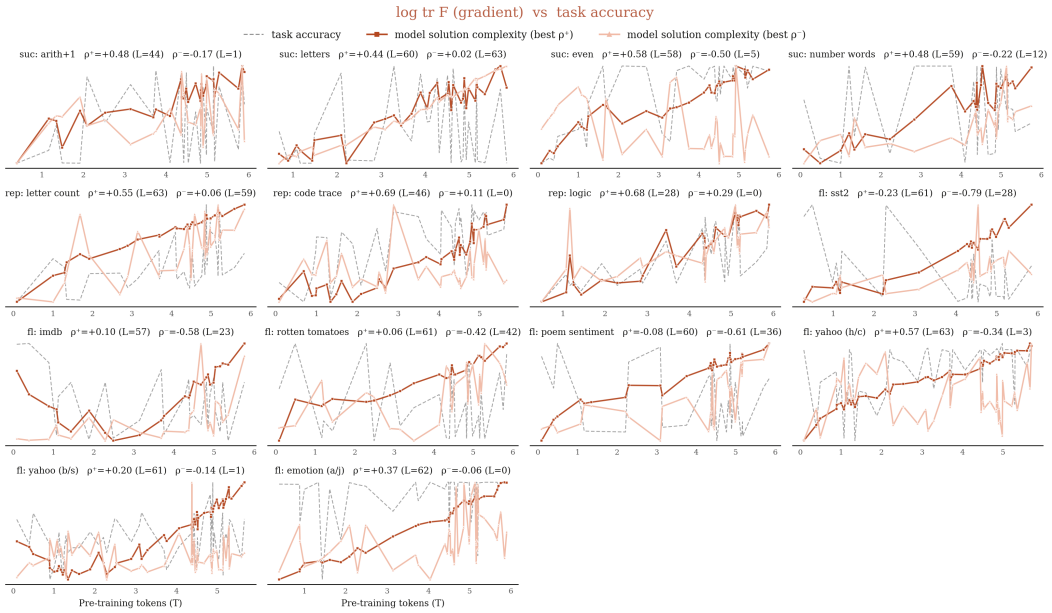


Figure 44: Pre-training dynamics of task accuracy and the best ρ^+ and ρ^- layers for $\log \operatorname{tr} F$, the total per-example gradient magnitude.

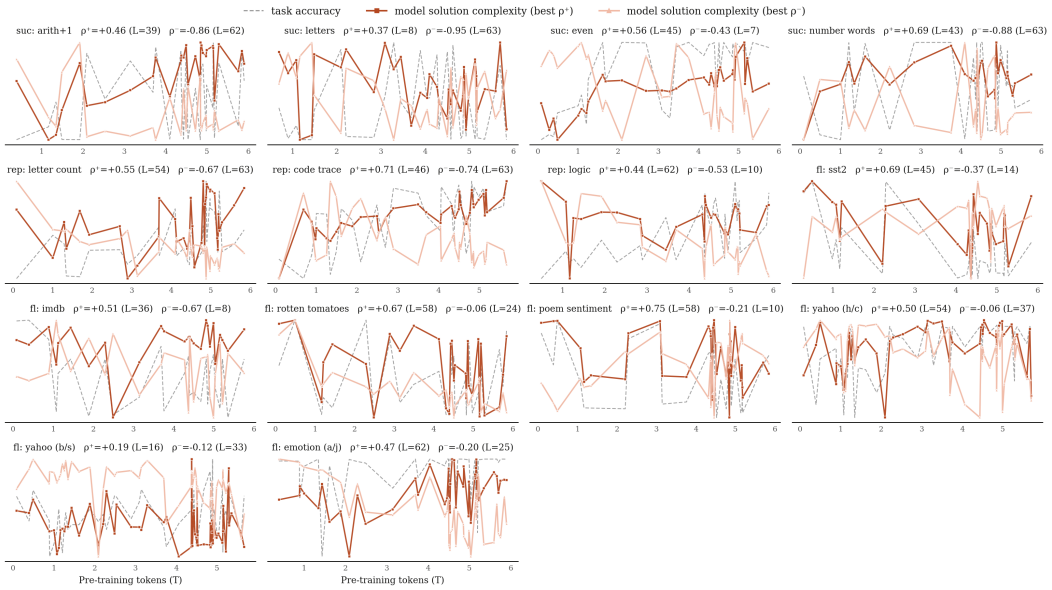


Figure 45: Pre-training dynamics of task accuracy and the best ρ^+ and ρ^- layers for $\lambda_1 / \operatorname{tr} F$, sharpness as a fraction of total curvature.



Figure 46: Pre-training dynamics of task accuracy and the best ρ^+ and ρ^- layers for |cosine similarity|, the mean absolute pairwise alignment between per-example gradients.