Mechanistic Capability Probes as a Cheap Screen for Sequence-Mixer Architectures

Tharun Kumar Tiruppali Kalidoss [*1] **Kia Ghods** [*1] **Elad Hazan** [1]

Abstract

Comparing sequence-mixer architectures at the scale where their behavior matters ($\geq$1B parameters) costs multiple GPU-days per run, beyond reach of most academic labs. We propose a battery of mechanistic capability probes (induction, associative recall, copy, finite-state tracking, parity, and others) as a cheap behavioral screen for dense sequence-mixer architectures, and ask whether aggregate suite accuracy predicts downstream language-model training cross-entropy. On a held-out set of four architectures at 150M parameters we find Spearman $\rho = -0.80$ and Pearson $r = -0.97$; the screen is robust to dropping any single task family, to the choice of aggregation rule, and—in a five-seed replication of the probe sweep—to seed variation; the small-scale ranking direction is preserved at 1B on the two architectures we ran. Per-task profiles motivate **Hydra**, a multi-head block that places attention, STU, and Mamba mixers as parallel heads within each layer; Hydra matches or beats a parameter-matched 1B OLMo-2 attention baseline on training cross-entropy and on a majority of zero-shot benchmarks. The probe suite, screening protocol, and all analysis code are available at https://github.com/hazan-lab/mechanistic-bench.

1. Introduction

The space of sequence-mixer architectures has grown faster than the compute available to compare them. Fairly evaluating a new mixer against a heavily-optimized Transformer baseline (Vaswani et al., 2017) at $\geq$1B parameters and $\geq$10B training tokens takes multiple GPU-days per run, beyond reach of most academic labs. The mechanistic-interpretability community has independently developed synthetic capability probes—induction (Olsson et al., 2022), associative recall (Dao et al., 2023; Arora et al., 2024), copy (Jelassi et al., 2024), finite-state tracking (Merrill et al., 2024)—that characterize what specific architectures can and cannot do. These probes have been used to demonstrate capability gaps between trained architectures, but not to predict downstream pretraining loss in advance. We propose using these probes as a cheap behavioral screen for whole architectures, before any large-scale pretraining: do small-scale capability profiles predict downstream training cross-entropy?

We restrict to *dense* architectures (all parameters active on every token); sparse-activation models (Fedus et al., 2022) have effective per-token compute decoupled from parameter count and exhibit scale-dependent capability emergence that small-scale probes cannot capture. We construct a 27-task headline suite (extended to 62 tasks in Section A) spanning retrieval, aggregation, and compound primitives, evaluate it across thirteen architectures (single-mixer, alternating-layer, mixed-head), and validate the cross-architecture rank ordering against downstream training CE at 50M, 150M, and 1B. The per-task profile motivates **Hydra**, a multi-head block placing attention, STU (Agarwal et al., 2023; Liu et al., 2025), and Mamba (Gu & Dao, 2023) mixers as parallel heads within each layer; Hydra matches or beats a parameter-matched 1B OLMo-2 attention baseline (OLMo et al., 2025).

2. Approach

We hypothesize that downstream training cross-entropy for sequence-mixer architectures is determined by mastery of two computational primitives and their composition. **Retrieval** locates and emits a token given a positional or content cue, and **aggregation** maintains and updates a running summary over a stream. An autoregressive next-token decision reduces to locating relevant context (retrieval), summarizing it under some invariant (aggregation), and composing the two under a task-conditional gate. Circuit-level decompositions of trained transformers in the interpretability literature, including induction heads (Olsson et al., 2022), name-mover heads (Wang et al., 2024), and content-addressing

[*]Equal contribution [1]Princeton University. Correspondence to: Tharun Kumar Tiruppali Kalidoss <tharun.tiruppali@gmail.com>.

Published at the ICML 2026 Workshop on Mechanistic Interpretability, Seoul, South Korea. Copyright 2026 by the author(s).

circuits (Elhage et al., 2021), factor along exactly these axes, and expressivity results for state-space and low-depth models are likewise organized around copying and state tracking (Jelassi et al., 2024; Merrill et al., 2024; Hahn, 2020). We treat the two-primitive decomposition as a falsifiable working hypothesis rather than a theoretical claim: Section 3.5 stress-tests it empirically by deleting primitives, restricting the suite to different surface forms, and varying compositional depth. A formal account of which mixer classes can express which primitive at which scale is an open direction we return to in Section 4. We focus on language modeling because it is the largest downstream use case of Transformers.

2.1. The probe suite

We collect synthetic probes from the mechanistic-interpretability literature, including induction (Olsson et al., 2022), associative recall (Dao et al., 2023; Arora et al., 2024), copy (Jelassi et al., 2024), finite-state tracking (Merrill et al., 2024), parity (Merrill et al., 2024; Hahn, 2020), and needle-in-a-haystack (Hsieh et al., 2024). Each probe is parameterized along sequence length, vocabulary, hop depth, state-machine size, key cardinality, distractor density, and modality, since these are the exact axes that appear in downstream tasks.

The result is 27 headline tasks, extended in Section A to 62 tasks total with multi-hop, grid, video, and continuous variants, grouped into three families. The *basic* family (8 tasks) isolates a single primitive and acts as a floor, covering copy, associative recall, needle, induction, and selective copy (Gu & Dao, 2023) on the retrieval side, alongside counting, parity, and 4-state DFA tracking on the aggregation side. The *compound* family (3 tasks) forces both primitives in the same sequence (copy + count, state + retrieve, selective copy + parity), exposing how an architecture allocates representational capacity when the two compete. The *diagnostic* family (16 tasks) collects stress tests and calibration pairs such as multi-induction, skip-induction, longest run, sort, and threshold.

Modality extensions inherit the same retrieval/aggregation structure, with 2D retrieval acting as retrieval over a two-coordinate cue and continuous denoising acting as aggregation over a real-valued stream. To make every probe a controlled experiment we use a fixed token convention (vocabulary 64, sequence length 256, shared special tokens), score loss only at task-defined critical positions so filler patterns cannot inflate accuracy, and generate every task deterministically from a fixed seed. Section 3 validates the decomposition empirically.

2.2. Architectures evaluated

All architectures are dense and share the same backbone (RMSNorm (Zhang & Sennrich, 2019), SwiGLU MLP (Shazeer, 2020), RoPE on attention heads (Su et al., 2024), embedding tying (Press & Wolf, 2017)), differing only in the sequence-mixing block. We evaluate four mixer primitives, namely attention (Vaswani et al., 2017), Mamba-1 (Gu & Dao, 2023), Mamba-2 (Dao & Gu, 2024), and STU (Agarwal et al., 2023; Liu et al., 2025), combined under three composition modes. The *single-mixer* mode uses one primitive everywhere, *alternating-layer* interleaves blocks (`alt_attn_mamba`, `alt_attn_stu`), and *mixed-head* (`headwise`) splits primitives across heads within a layer (concurrent with Hymba (Dong et al., 2024)), of which Hydra is the specific instantiation studied in Section 3.3.

2.3. Probe evaluation

The screening protocol scales the same architecture definition across four levels and asks the small-scale probe ranking to predict the large-scale training-loss ranking. At *1M* parameters we train each architecture independently on every task (n_{layers}=6, n_{heads}=4, seq-len 256, vocab 64, 4,000 steps, with full hyperparameters in Section B). The resulting per-task accuracy matrix is the screening signal. At *10M* we repeat the sweep with larger d_{model} to confirm that 1M probe scores reflect architectural capability rather than under-capacity, since a low score from a 1M model could in principle mean either the architecture cannot learn the task or the model is too small to express it. At *50M and 150M* we train each architecture as a language model to Chinchilla-optimal compute (Hoffmann et al., 2022) on Wikipedia / c4_en (Raffel et al., 2020) / wikitext_103 (Merity et al., 2017) (n=9 at 50M as a larger-sample robustness check with tighter confidence intervals, n=4 at 150M held out from suite design for the headline correlation). At *1B* we run a parameter-matched OLMo-2 (OLMo et al., 2025) attention baseline and the Hydra hybrid on OLMo-Mix-1124 for ∼13k steps at seq-len 4,096 as a directional check at production scale. For each architecture, the screening statistic is the arithmetic mean of per-task token accuracy with task-family reweighting. We evaluate the cross-architecture rank ordering by downstream training cross-entropy, since rank ordering is the decision a compute-constrained lab actually has to make.

3. Experiments

We aim to answer four questions. **(Q1)** Do per-architecture probe profiles at the 1M scale reveal interpretable capability bottlenecks (retrieval vs aggregation vs compound) that distinguish dense sequence-mixer families? **(Q2)** Does aggregate suite accuracy at the 1M scale predict downstream language-model training cross-entropy at 150M? **(Q3)** Is

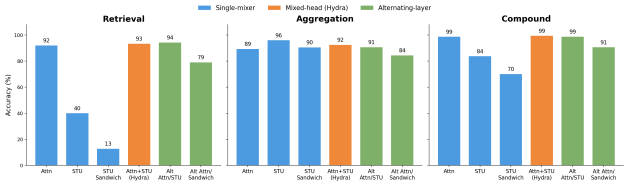

Figure 1. Per-family accuracy across the six probe-scale architectures, grouped by composition mode (single-mixer, mixed-head, alternating-layer). Pure STU and STU-sandwich collapse on retrieval, while aggregation differences stay within ∼5pp across all architectures.

the small-scale ranking direction preserved at a billion-parameter scale? **(Q4)** Is the screening signal carried by a single task family, or is it distributed across the retrieval, aggregation, and compound primitives the suite was designed around?

3.1. Per-family architecture profile

Figure 1 reports per-architecture accuracy on the basic suite, broken out by primitive family. Pure-spectral mixers collapse on retrieval, with STU reaching 40% and the STU sandwich 13%, against $\geq 92\%$ for every architecture that contains an attention component. Aggregation accuracies are bunched within ∼5 percentage points across all six architectures, so aggregation is not where architectural choice differentiates the field at this scale. The differentiation lives in retrieval and the compound family.

3.2. Mechanistic suite predicts 150M LM cross-entropy

We fit the screen on a four-architecture set held out from suite design (`attn`, `mamba`, `alt_attn_mamba`, and the STU-sandwich variant), each trained as a 150M-parameter language model to Chinchilla-optimal compute on Wikipedia / c4_en / wikitext_103 (configuration in Sections B and E). Aggregate suite accuracy at the 1M probe scale predicts the 150M cross-entropy ranking at Spearman $\rho = -0.80$ and Pearson $r = -0.97$ (Figure 2, left). Table 1 reports the underlying numbers and adds zero-shot eval columns. The suite-mean ordering matches the c4_en cross-entropy ordering and is consistent with the eval columns, so the screen's pick is not specific to the pretraining loss surface. The 50M robustness fit (n=9, $\rho = -0.88$ vs. wikitext_103 CE) is reported in Section F.

Outlier sensitivity. The n=4 fit is heavily leveraged by the STU-sandwich outlier. With it removed, the three remaining architectures sit within 0.01 nats of one another on c4_en CE (2.81–2.82) and 0.03 nats on wikitext_103 CE, so no meaningful n=3 rank correlation survives—nor should one be expected, since the screen itself separates these three by only six points of suite-mean. We therefore read the 150M result as the screen flagging a collapsing architecture *before* any large run is paid for, and rest the quantitative correlation claim on the larger 50M set, which Section K extends from nine to eleven architectures (ρ between -0.77 and -0.86, $p \leq 0.005$) including an out-of-lineage LSTM baseline.

Table 1. 150M LM evaluation on the held-out architecture set. Suite-mean is the screening statistic at the 1M probe scale, c4_en CE is pretraining cross-entropy on the c4_en validation memmap (lower is better), and Avg. eval is the mean of zero-shot PIQA and HellaSwag accuracy. See Table 10 for wikitext_103 CE and full per-task numbers.

Architecture	Suite-mean ↑	c4_en CE ↓	Avg. eval ↑
STU sandwich	0.50	3.67	0.398
`mamba`	0.74	2.82	0.411
`alt_attn_mamba`	0.78	2.81	0.416
`attn`	0.80	2.81	0.416

3.3. Hydra

The per-family profile above shows that no single mixer dominates across primitives. Attention handles retrieval, Mamba is competitive on aggregation and finite-state tracking, and STU is efficient on long-range smoothing but collapses on retrieval. Hydra generalizes the mixed-head composition (Section 2.2) by placing all three mixer types as parallel heads inside a single block (Figure 3), so each layer can route capacity to the primitive that suits the local computation rather than committing to one mixer for the whole network. The 1B configuration uses 4 attention, 4 STU, and 4 Mamba heads per block (denoted 4a4s4m), with outputs combined by weighted average and the rest of the backbone shared with the baselines in Section 2.2. Full per-mixer hyperparameters and the 1B training recipe are in Section C.

3.4. 1B directional check

We scale two architectures to 1B parameters, a parameter-matched OLMo-2 attention baseline and the Hydra hybrid (full configuration in Section C), both trained on OLMo-Mix-1124 (OLMo et al., 2025) for ∼13k steps at sequence length 4096. Hydra has the higher 1M suite mean (0.81 vs 0.76) and reaches the lower pretraining cross-entropy at 1B (2.880 vs 2.934), preserving the direction of the small-scale ranking (Figure 2, right). Table 2 reports the same comparison on the OLMo zero-shot eval suite, where Hydra wins on 8 of 15 zero-shot tasks (with one tie) and trails on the four MMLU (Hendrycks et al., 2021) subdomains, COPA (Roemmele et al., 2011), and SocialIQA (Sap et al., 2019). With n=2 this is a directional preservation test rather than a second correlation point, though it does show that the suite ranking, the pretraining CE ranking, and the zero-shot eval ranking all agree at production scale.

Where Hydra loses. Hydra's deficits are not scattered: it trails precisely on the knowledge-intensive tasks (all

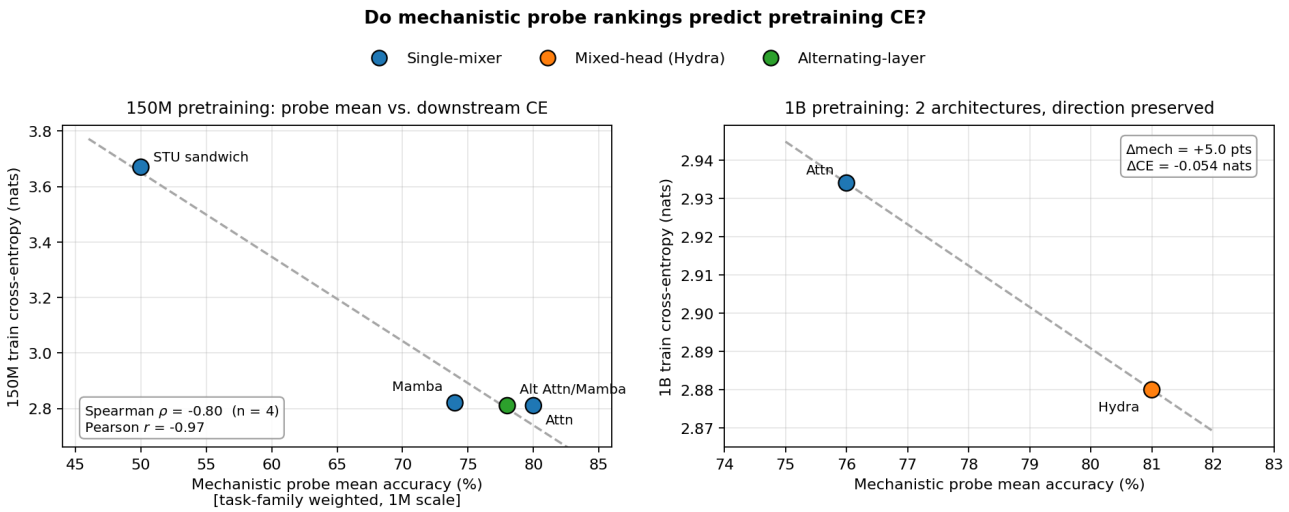

Figure 2. Suite-mean accuracy at the 1M probe scale vs. downstream training cross-entropy. Left: 150M, n=4, $\rho = -0.80$, $r = -0.97$. Right: 1B, n=2, with the small-scale ranking direction preserved.

four MMLU subdomains) and on two commonsense tasks (COPA, SocialIQA), while winning on tasks dominated by in-context processing (HellaSwag, PIQA, ARC-c, SciQ, CommonsenseQA). The suite offers a candidate explanation: every probe is solvable from the context window alone, so the screen measures in-context computation and contains no probe of *parametric* knowledge storage or recall—the capability MMLU leans on most. Consistent with this, Hydra allocates two-thirds of each layer's heads to STU and Mamba mixers, and the per-family profiles (Figure 1) show these mixers contribute least on content-addressable retrieval, the primitive closest to fact lookup. We flag knowledge-intensive evaluation as a blind spot of the current suite; designing probes for parametric recall (e.g., closed-book memorization tasks scored after fixed exposure budgets) is an open problem for capability screens generally.

Table 2. 1B LM evaluation. Hydra and the OLMo-2 attention baseline at step 12,000. Train CE on OLMo-Mix-1124. Zero-shot avg. is the mean over ARC-c, ARC-e (Clark et al., 2018), BoolQ (Clark et al., 2019), COPA (Roemmele et al., 2011), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), SciQ (Welbl et al., 2017), SocialIQA (Sap et al., 2019), Winogrande (Sakaguchi et al., 2021), and CommonsenseQA (Talmor et al., 2019). See Table 11 for the full per-task numbers. Bold = winner.

Metric	Hydra	Attn baseline
Train CE ↓	**2.880**	2.934
Train PPL ↓	**17.82**	18.80
Zero-shot avg. ↑	**0.540**	0.529

3.5. Robustness

A single primitive could in principle carry the entire screening signal, in which case the correlations in Section 3.2 and Section F would not justify the framing of the suite as a decomposition. We test this directly by recomputing the suite-mean with one primitive deleted and refitting the rank correlation against 50M c4_en cross-entropy on the nine-architecture held-out set. The correlation remains negative and statistically significant under every drop (full ablation in Section I). Three further checks (multimodal-only transfer, compositional-depth invariance, and transfer to autoencoding-flavored probes) are reported in Section I and reach the same conclusion under different perturbations of the suite.

Seed stability. We replicated the full 1M probe sweep across five random seeds for seven architectures (Section J). The suite-mean standard deviation across seeds is at most ± 1.1 percentage points per architecture, the cross-architecture ranking is identical on four of five seeds (with a single adjacent transposition on the fifth), and the screen–downstream correlation is unchanged when computed from any individual seed. Seed noise is therefore not a plausible driver of the reported correlations at the probe scale; downstream LM runs remain single-seed (Section 4).

Aggregation rule. Replacing the arithmetic suite-mean with a median or a 10%-trimmed mean leaves every rank correlation unchanged or slightly stronger; only the minimum statistic, which reduces the suite to its single hardest task, degrades the fit (Section L). The screening signal is a property of the distribution of per-task accuracies, not of one aggregation formula.

Learning rate. Because architecture families differ in op-

timizer sensitivity, we re-ran the headline tasks at learning rates bracketing the default by $3\times$ in each direction (Section N). The retrieval-driven gap that separates attention-containing from pure non-attention architectures ($\sim$45 points) is present at every rate, and even oracle per-architecture learning-rate tuning leaves the ranking unchanged, so the probe-score differences reflect architectural capability rather than hyperparameter mismatch.

4. Limitations and Future Work

Our validation uses four architectures at 150M, extended to eleven architectures at 50M (Sections F and K) and a two-architecture directional check at 1B (Section 3.4). The 1M probe sweep is replicated over five seeds (Section J); the LM runs remain single-seed due to compute constraints. A more compelling fit would span 20 or more architectures across families we do not currently cover (H3 (Dao et al., 2023), Hyena (Poli et al., 2023), RWKV (Peng et al., 2023), RetNet (Sun et al., 2023)), evaluate against multiple downstream signals (held-out CE on diverse domains, zero-shot benchmark accuracy, downstream task transfer), and quantify seed-level variability of the downstream runs themselves. We expect the screen to extend to other dense mixers in the same lineage (gated linear recurrences (De et al., 2024), Hyena, RWKV, RetNet) since they share the dense-activation property and target the same primitives the suite was designed around. Our first step outside the lineage is cautionary as well as encouraging: adding a parameter-matched LSTM keeps the aggregate correlation strong ($\rho{=}-0.86$, Section K), but the screen underestimates *how far* the LSTM trails the modern mixers at 50M, so we do not yet trust the screen's cardinal placements outside the family it was developed on. We are similarly uncertain about pure-convolutional mixers and architectures whose compositional structure differs more substantially from the families we evaluated.

We aggregate per-task accuracy by arithmetic mean with task-family weighting, so the more-numerous diagnostic family does not dominate the score. Section L reports a sensitivity analysis across alternative aggregations: median and trimmed-mean rules reproduce the arithmetic-mean correlations essentially unchanged, quartile aggregation is mildly weaker, and only the minimum statistic degrades the fit, consistent with the rank ordering being driven by broad profile differences (single-mixer baselines collapse on at least one task family by 30 percentage points or more) rather than by the aggregation formula.

The screening claim is restricted to architectures where all parameters are active on every token; mixture-of-experts and other sparse-activation models are excluded. Our hypothesis for why the paradigm breaks there is specific: with learned top-k routing, (i) per-token effective compute is decoupled from total parameter count, so no single small dense proxy is compute-matched to the trained model; (ii) router specialization is itself an emergent property of scale and data diversity, so a 1M-parameter proxy trains a qualitatively different routing function, not a miniature of the large one; and (iii) capabilities gated on expert specialization can appear discontinuously with scale (Fedus et al., 2022), violating the smooth small-to-large transfer the screen relies on. A mechanistic screen for sparse models would plausibly need to run probes at matched *active* compute with the router in the loop, and to include probes that detect routing pathologies (expert collapse, load imbalance) rather than end-task accuracy alone; we leave constructing such a suite to future work.

5. Conclusion

We have provided held-out evidence that aggregate accuracy on a suite of mechanistic capability probes predicts downstream training cross-entropy across dense sequence-mixer architectures, with Spearman $\rho = -0.80$ and Pearson $r = -0.97$ at 150M ($n = 4$), extended to a nine-architecture held-out set at 50M ($\rho{=}-0.73$, $p{=}0.025$ vs c4_en CE; $\rho{=}-0.88$, $p{=}0.002$ vs wikitext_103 CE) and further to eleven architectures including an out-of-lineage LSTM (Section K), and the small-scale ranking direction preserved on the two architectures we ran at 1B. The screen is robust to dropping any single primitive, to seed variation (five-seed replication with at most one adjacent rank transposition, Section J), and to the aggregation rule (Section L); it transfers to a multimodal-only subset of the suite (grid + video probes, $\rho{=}-0.73$, $p{=}0.025$, $n{=}9$) and to autoencoding-flavored probes (denoising/compression, $\rho{=}-0.83$, $p{=}0.005$), and rankings at low compositional depth predict rankings at higher depth ($\rho{=}0.67$ between hop-depth $k{=}2$ and $k{=}4$). The per-family breakdowns identify clear capability bottlenecks per mixer type, and those bottlenecks directly motivated the multi-head Hydra design that matches or beats a parameter-matched 1B OLMo-2 baseline. The suite, screening protocol, per-run results, and analysis pipeline are open-source at `https://github.com/hazan-lab/mechanistic-bench`, giving compute-constrained labs a structured way to compare new mixer architectures before committing to large-scale pretraining.

References

Agarwal, N., Suo, D., Chen, X., and Hazan, E. Spectral state space models. *arXiv preprint arXiv:2312.06837*, 2023.

Arora, S., Eyuboglu, S., Timalsina, A., Johnson, I., Poli, M., Zou, J., Rudra, A., and Re, C. Zoology: Measuring and Improving Recall in Efficient Language Models. In *The*

International Conference on Learning Representations (ICLR), 2024.

Azerbayev, Z., Schoelkopf, H., Paster, K., Dos Santos, M., McAleer, S. M., Jiang, A. Q., Deng, J., Biderman, S., and Welleck, S. Llemma: An Open Language Model for Mathematics. In *The International Conference on Learning Representations (ICLR)*, 2024.

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 2924–2936, 2019.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Dao, T. and Gu, A. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. In *The International Conference on Machine Learning (ICML)*, 2024.

Dao, T., Fu, D. Y., Saab, K. K., Thomas, A. W., Rudra, A., and Ré, C. Hungry Hungry Hippos: Towards Language Modeling with State Space Models. In *The International Conference on Learning Representations (ICLR)*, 2023.

De, S., Smith, S. L., Fernando, A., Botev, A., Cristian-Muraru, G., Gu, A., Haroun, R., Berrada, L., Chen, Y., Srinivasan, S., et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*, 2024.

Dong, X., Fu, Y., Diao, S., Byeon, W., Chen, Z., Mahabaleshwarkar, A. S., Liu, S.-Y., Keirsbilck, M. V., Chen, M.-H., Suhara, Y., Lin, Y., Kautz, J., and Molchanov, P. Hymba: A Hybrid-head Architecture for Small Language Models. *arXiv preprint arXiv:2411.13676*, 2024.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. A Mathematical Framework for Transformer Circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research (JMLR)*, 23(120):1–39, 2022.

Gu, A. and Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *Conference on Language Modeling (COLM)*, 2023.

Hahn, M. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics (TACL)*, 8:156–171, 2020.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring Massive Multitask Language Understanding. In *The International Conference on Learning Representations (ICLR)*, 2021.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Vinyals, O., Rae, J. W., and Sifre, L. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., and Ginsburg, B. RULER: What's the Real Context Size of Your Long-Context Language Models? In *Conference on Language Modeling (COLM)*, 2024.

Jelassi, S., Brandfonbrener, D., Kakade, S. M., and Malach, E. Repeat after me: Transformers are better than state space models at copying. In *The International Conference on Machine Learning (ICML)*, 2024.

Li, J., Fang, A., Smyrnis, G., Ivgi, M., Jordan, M., Gadre, S., Bansal, H., Guha, E., Keh, S., Arora, K., et al. DataComp-LM: In Search of the Next Generation of Training Sets for Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 14200–14282, 2024.

Li, R., Allal, L. B., Zi, Y., Muennighoff, N., Kocetkov, D., Mou, C., Marone, M., Akiki, C., Li, J., Chim, J., et al. StarCoder: May the Source Be With You! *arXiv preprint arXiv:2305.06161*, 2023.

Liu, Y. I., Nguyen, W., Devre, Y., Dogariu, E., Majumdar, A., and Hazan, E. Flash STU: Fast Spectral Transform Units. In *IEEE 64th Conference on Decision and Control (CDC)*, pp. 165–171. IEEE, 2025.

Loshchilov, I. and Hutter, F. Decoupled Weight Decay Regularization. In *The International Conference on Learning Representations (ICLR)*, 2019.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer Sentinel Mixture Models. In *The International Conference on Learning Representations (ICLR)*, 2017.

Merrill, W., Petty, J., and Sabharwal, A. The illusion of state in state-space models. In *The International Conference on Machine Learning (ICML)*, 2024.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2381–2391, 2018.

OLMo, T., Walsh, P., Soldaini, L., Groeneveld, D., Lo, K., Arora, S., Bhagia, A., Gu, Y., Huang, S., Jordan, M., Lambert, N., Schwenk, D., Tafjord, O., Anderson, T., Atkinson, D., Brahman, F., Clark, C., Dasigi, P., Dziri, N., Ettinger, A., Guerquin, M., Heineman, D., Ivison, H., Koh, P. W., Liu, J., Malik, S., Merrill, W., Miranda, L. J. V., Morrison, J., Murray, T., Nam, C., Poznanski, J., Pyatkin, V., Rangapur, A., Schmitz, M., Skjonsberg, S., Wadden, D., Wilhelm, C., Wilson, M., Zettlemoyer, L., Farhadi, A., Smith, N. A., and Hajishirzi, H. 2 OLMo 2 Furious. *arXiv preprint arXiv:2501.00656*, 2025.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Paster, K., Dos Santos, M., Azerbayev, Z., and Ba, J. OpenWebMath: An Open Dataset of High-Quality Mathematical Web Text. In *The International Conference on Learning Representations (ICLR)*, 2024.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Biderman, S., Cao, H., Cheng, X., Chung, M., Derczynski, L., Du, X., Grella, M., Gv, K., He, X., Hou, H., Kazienko, P., Kocon, J., Kong, J., Koptyra, B., Lau, H., Lin, J., Mantri, K. S. I., Mom, F., Saito, A., Song, G., Tang, X., Wind, J., Woźniak, S., Zhang, Z., Zhou, Q., Zhu, J., and Zhu, R.-J. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14048–14077, 2023.

Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. Hyena Hierarchy: Towards Larger Convolutional Language Models. In *The International Conference on Machine Learning (ICML)*, pp. 28043–28078, 2023.

Press, O. and Wolf, L. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 157–163, 2017.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 21(140):1–67, 2020.

Roemmele, M., Bejan, C. A., and Gordon, A. S. Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning. In *AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, pp. 90–95, 2011.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Sap, M., Rashkin, H., Chen, D., Le Bras, R., and Choi, Y. Social IQa: Commonsense Reasoning about Social Interactions. In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4463–4473, 2019.

Shazeer, N. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Soldaini, L. and Lo, K. peS2o (Pretraining Efficiently on S2ORC) Dataset. Technical report, Allen Institute for AI, 2023.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., and Wei, F. Retentive Network: A Successor to Transformer for Large Language Models. *arXiv preprint arXiv:2307.08621*, 2023.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 4149–4158, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Wang, K. R., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 Small. In *The International Conference on Learning Representations (ICLR)*, 2024.

Welbl, J., Liu, N. F., and Gardner, M. Crowdsourcing Multiple Choice Science Questions. In *Proceedings of*

the 3rd Workshop on Noisy User-generated Text (W-NUT), pp. 94–106, 2017.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 4791–4800, 2019.

Zhang, B. and Sennrich, R. Root mean square layer normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

A. Full task list

Table 4 lists the full set of 27 tasks that make up the mechanistic suite, organized by family. Table 3 gives the primitive-coverage matrix referenced in Section 2.1, mapping every task in the extended suite (62 total) to a primary primitive and any secondary primitives; some rows in the matrix group multiple depth-parameter instances of a single task family. Every primitive has at least two probes (redundancy/robustness), and no two tasks fully overlap (non-redundancy).

Table 3. Primitive-coverage matrix for the extended mechanistic suite. *Pri.* is the primary primitive used by the drop-primitive ablation in Section 3.5; *Secondary* lists additional primitives the task partially probes. Primitive abbreviations: PR=point retrieval, CL=content-addressable lookup, AGG=aggregation, FS=finite-state tracking, F=selective filtering, MH=multi-hop, S2=2D spatial, CT=continuous, CMP=compound.

Task	Pri.	Secondary
copy, copy_offset, reverse_copy, needle	PR	—
induction, induction_gap, multi_induction	PR	CL
associative, short_associative, batch_recall	CL	—
conditional_recall, last_tagged, first_vs_last	CL	—
mode_tagged	CL	AGG
counting, parity, cumulative_sum, threshold	AGG	—
mode, longest_run, running_max	AGG	—
state_tracking, multi_state_tracking	FS	—
pattern_completion, token_transition	FS	—
selective_copy	F	PR
selective_parity	F	AGG
compress, interleave	F	AGG, PR
noisy_copy	F	PR
sort	F	—
two_hop, three_hop, deep_hop, k_hop	MH	CL
dual_hop_retrieve, batch_two_hop, dual_query_hop	MH	CL
nested_lookup, nested_3_hop, hop_distance_bucket	MH	CL
triple_recall, quad_recall, union_lookup	MH	CL
variable_lookup, assignment_chain	MH	CL
grid_retrieval	S2	PR
grid_two_coord, grid_three_coord	S2	CL
grid_multihop	S2	MH
col_parity	S2	AGG
patch_match	S2	CL
sort_top2, set_intersection_count	S2	AGG
temporal_ordering, substring_locate	S2	CL
video_frame_retrieval, video_cell_mode	S2	—
delayed_echo	CT	PR
piecewise_denoise	CT	AGG
nearest_key	CT	CL
copy_count	CMP	PR, AGG
state_retrieve	CMP	FS, CL

Table 4 below restates the original 27-task headline suite organized by the basic / compound / diagnostic groupings used in Section 3.

Table 4. The full mechanistic-suite task list, organized by family.

Task	Description
Basic (8): each isolates a single primitive.	
Copy	Reproduce a prefix verbatim.
Induction	Predict B after the second occurrence of A in pattern "$AB \dots A$".
Associative recall	Given key-value pairs and a query key, return the matching value.
Selective copy	Output only the tokens that follow a MARKER, in order.
Needle-in-a-haystack	Find one special token buried in random filler.
Counting	Count occurrences of a query token in the body.
Parity	Output the parity (even/odd) of the query-token count.
State tracking	Simulate a 4-state finite automaton step by step.
Compound (3): require two primitives in the same sequence.	
Copy + Count	Copy a prefix verbatim, then answer a counting query over it.
State + Retrieve	Simulate the 4-state DFA, then retrieve the input first causing a queried state.
Selective Copy + Parity	Selective copy, then output the parity of the marker count.
Diagnostic (16): edge cases and stress tests.	
Pattern completion	Complete a periodic pattern of period 2–6.
Mode	Output the most frequent content token.
Sort	Output the tokens in ascending order.
Reverse copy	Output a prefix in reverse order.
Short associative recall	Associative recall with 6 unique key-value pairs (calibration scale).
Compress (deduplicate)	Output the unique tokens in first-appearance order.
De-interleave	Separate two interleaved sequences A, B back into A then B.
Multi-induction	Match a 2-token trigger and predict the third.
Longest run	Output the token with the longest consecutive run.
Noisy copy	Identify and recover the differing element between two aligned copies.
Threshold	Binary indicator: does the query-token count exceed N?
Skip-induction	Induction-head pattern with skipped tokens between A and the prediction site.
Running maximum	Output the running maximum of an integer stream.
Token transition	Predict transitions between specified token classes.
Cumulative sum (mod 8)	Modular running sum of an integer stream.
Multi-state tracking (8)	Simulate an 8-state finite automaton step by step.

B. Per-architecture configurations

All thirteen mechanistic-probe architectures evaluated in the main study share $n_{\text{layers}} = 6$, $n_{\text{heads}} = 4$, and vocabulary size 64. Per-architecture d_{model} values are chosen for parameter parity at $\sim$1M total parameters. Table 5 gives the per-architecture configuration; the two Mamba-2 variants (`headwise_mamba2` and `alt_attn_mamba2`) are additional architectures included in the 50M robustness fit (Section F) but not in the 1M probe sweep or the 150M held-out correlation. Table 6 gives the mixer-specific internal parameters; Table 7 gives the shared training hyperparameters used across all probe runs.

Table 5. Architecture-specific hyperparameters for all mechanistic-probe models. The "Mixers" column indicates which sequence-mixing primitives are active in each block (A = Attention, S = STU, M = Mamba-1, M2 = Mamba-2). The "Composition" column describes how mixers are combined. The two Mamba-2 variants are used only in the 50M robustness fit (Section F).

Family	Architecture	d_{model}	Mixers	Composition
Single-mixer	`attn`	128	A	RMSNorm → MultiHeadAttn → FF
	`stu`	140	S	RMSNorm → STU → FF
	`mamba`	100	M	RMSNorm → Mamba → FF
	`stu_sandwich`	106	S	FF → STU → FF (triple sub-layer)
Mixed-head	`headwise_stu`	120	A + S	Weighted average (6:6)
	`headwise_mamba`	88	A + M	Weighted average (6:6)
	`headwise_stu_mamba`	92	S + M	Weighted average (6:6)
	`headwise`	88	A + S + M	Weighted average (4:4:4)
	`headwise_mamba2`‡	88	A + M2	Weighted average (6:6)
Alternating-layer	`alt_attn_stu`	128	A, S	[A, S, A, S, A, S]
	`alt_attn_mamba`	112	A, M	[A, M, A, M, A, M]
	`alt_attn_mamba2`‡	112	A, M2	[A, M2, A, M2, A, M2]
	`alt_stu_mamba`	116	S, M	[S, M, S, M, S, M]
	`alt_attn_stu_mamba`	120	A, S, M	[A, S, M, A, S, M]
	`alt_attn_stu_sandwich`	116	A, S†	[A, S†, A, S†, A, S†, A, S†]

S† denotes the STU-sandwich block (FF–STU–FF triple sub-layer).

‡ Mamba-2 variants used only in the 50M robustness fit.

Table 6. Internal parameters for each sequence-mixing primitive used across the mechanistic probes.

Mixer	Parameter	Value
Attention	Head dimension	$d_{\text{model}}/n_{\text{heads}}$
	Positional encoding	RoPE
	Causal masking	Yes
	Attention dropout	0.0
STU	Eigenvalues retained (K)	16
	Hankel variant	Standard (not Legendre)
	Approximation mode	FFT-based (STU-T)
	FFT size	$2^{\lceil \log_2(2\cdot \texttt{seq_len}-1)\rceil}$
Mamba	State dimension (d_{state})	64
	Expansion factor	2
	1D conv kernel (d_{conv})	4
	Discretization	ZOH (Mamba-1)

Table 7. Training hyperparameters shared across all mechanistic-probe experiments.

Parameter	Value
Optimizer	AdamW (Loshchilov & Hutter, 2019)
Learning rate	3×10^{-4}
Weight decay	0.0
Max gradient norm	1.0
LR schedule	Cosine annealing
Warmup steps	200
Min LR ratio	0.01
Global batch size	128
Max training steps	4,000
Sequence length	256
Vocabulary size	64
MLP ratio	2.0 (SwiGLU)
Embedding tying	Yes
RMSNorm ϵ	10^{-6}
Residual / embedding dropout	0.0
Compute dtype	bf16
Parameter dtype	fp32
Eval frequency	Every 500 steps
Max eval batches	20

C. 1B-scale training configuration

The 1B comparison in Section 3.4 trains both Hydra and the parameter-matched OLMo-2 attention baseline for approximately 13,000 steps at sequence length 4,096 with the OLMo-2 tokenizer (vocabulary 100,278). Both models share the optimizer (AdamW (Loshchilov & Hutter, 2019), lr $= 4 \times 10^{-4}$, weight decay 0.1), batch size (512), and positional encoding (RoPE (Su et al., 2024) with $\theta = 500{,}000$). Table 8 gives the per-architecture configuration. The Hydra block-level definition is given in Section 3.3.

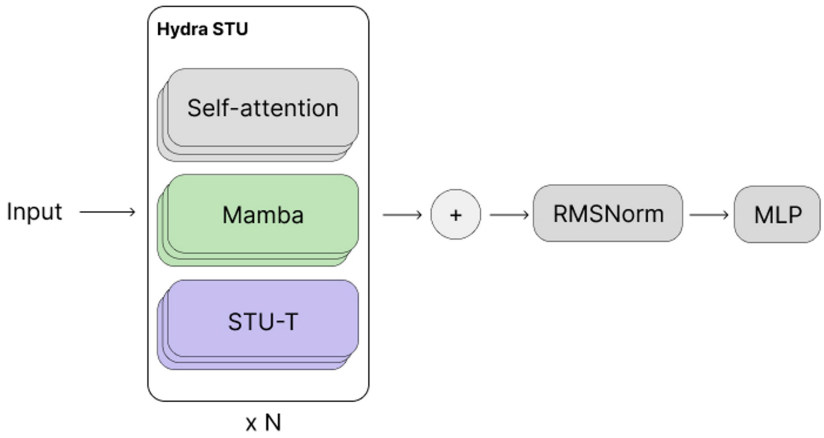

Figure 3. The Hydra block. Each block contains parallel self-attention, Mamba, and STU-T heads, whose outputs are summed and passed through RMSNorm and an MLP. The 1B configuration uses 4 of each head per block (4a4s4m), repeated N times.

Table 8. 1B-scale architecture configurations used in Section 3.4.

Model	d_{model}	n_{heads}	n_{layers}	Batch size	Block type	K
OLMo2-1B (baseline)	2048	16	16	512	Sequential (pure attention)	–
Hydra-1B (4a4s4m)	2304	18	15	512	Multihead + Mamba	20

D. Controlled depth sweep on `deep_hop`

Figure 4 shows the controlled-depth sweep referenced in Section 3.5. We run `deep_hop` with explicit hop count $k \in \{1, 2, 3, 4, 5\}$ across five architectures at the 1m scale (attn, mamba2, stu, alt-attn-mamba, headwise), 2,000 training steps each, sharing the graph generator (n_nodes $= 10$, sequence length 256). For attn, alt-attn-mamba, and headwise, accuracy increases monotonically with k over this range. The mamba2 trajectory has an anomalous k=1 specialty (token accuracy 0.93, near-perfect single-edge lookup) before reverting to the same monotonic pattern as the others for $k \geq 2$, which we attribute to the selective state-update mechanism trivially handling a single deterministic transition. STU is non-monotonic, oscillating between near-baseline and partial-solve depending on k. The takeaway for the screen is that within an architecture, accuracy varies smoothly with compositional depth except for STU; the cross-architecture ranking is therefore informative across depths.

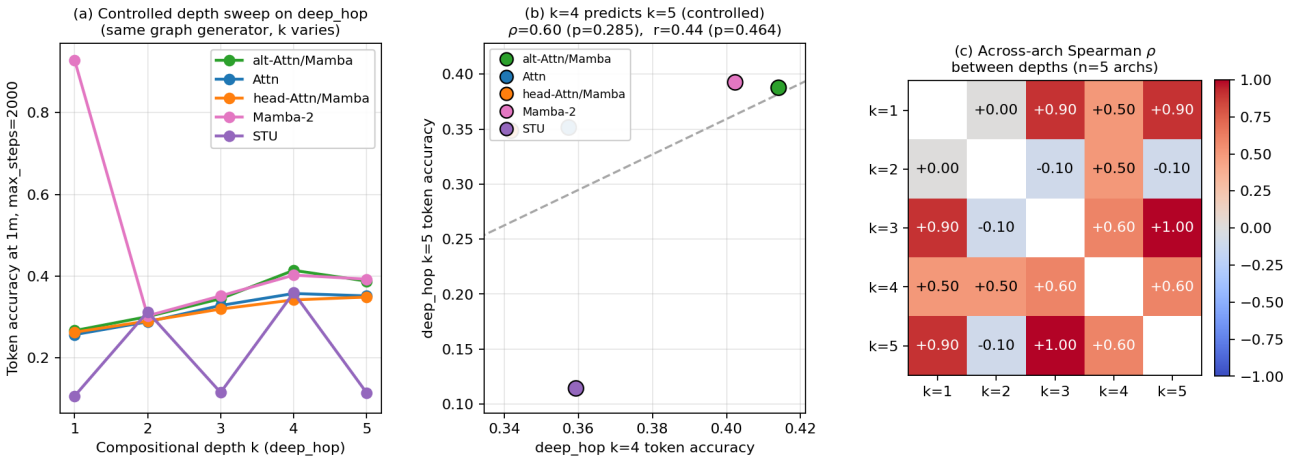

Figure 4. Controlled `deep_hop` sweep at the 1m scale across five architectures and depths $k \in \{1, 2, 3, 4, 5\}$. *(a)* Per-architecture accuracy curves; attn, alt-attn-mamba, and headwise are monotonic in k, mamba2 has a k=1 specialty, STU oscillates. *(b)* Cross-arch scatter at k=4 vs k=5. *(c)* Pairwise across-arch Spearman ρ between hop depths.

E. Pretraining data

The 1B-scale comparison in Section 3.4 uses **OLMo-Mix-1124**, the stage-1 pretraining mixture released with OLMo-2 (OLMo et al., 2025). The mixture totals approximately 3.9 trillion tokens and is dominated by web data from DCLM (Li et al., 2024), with smaller portions of code (StarCoder (Li et al., 2023)), academic papers (peS2o (Soldaini & Lo, 2023), arXiv), math (OpenWebMath (Paster et al., 2024), Algebraic Stack (Azerbayev et al., 2024)), and Wikipedia. Stage 1 accounts for over 90% of the OLMo-2 pretraining compute. Stage 2, which uses the curated Dolmino-Mix-1124 dataset of high-quality web content, instruction data, and synthetic math, is not used in the experiments reported here. Tokenization follows the OLMo-2 tokenizer (vocabulary 100,278). The 150M-scale held-out architecture set used to fit the screening correlation in Section 3.2 trains on Wikipedia text at sequence length 2,048. We refer the reader to the OLMo-2 technical report (OLMo et al., 2025) for the full data-curation pipeline, deduplication, quality filtering, and source-mixing details.

F. 50M held-out evaluation

Table 9 reports the nine-architecture held-out set used for the robustness fit referenced in Section 3.2. Each architecture is trained as a 50M-parameter language model to Chinchilla-optimal compute (954 steps, ∼1B tokens, $d_{\text{model}} = 384$ across

all architectures). Suite-mean is the screening statistic at the 1M probe scale (arithmetic mean over 62 tasks); the two CE columns are pretraining cross-entropy on the c4_en and wikitext_103 validation memmaps; PIQA and HellaSwag are zero-shot. Lower CE is better. The architectures are ordered by suite-mean accuracy.

Across this n=9 held-out set, suite-mean accuracy at 1M predicts 50M c4_en cross-entropy at Spearman $\rho = -0.73$ (p=0.025) and Pearson $r = -0.74$ (p=0.022), and 50M wikitext_103 cross-entropy at Spearman $\rho = -0.88$ (p=0.002) and Pearson $r = -0.83$ (p=0.006). The architectures span the same three families used in the 150M study (single-mixer, mixed-head, alternating-layer) plus two additional Mamba-2 variants. The last two rows of Table 9 report two architectures added after submission (single-mixer Mamba-2 and an out-of-lineage LSTM); the correlations for the extended n=10 and n=11 sets are analyzed in Section K.

Table 9. 50M LM evaluation on the nine-architecture held-out set. Suite-mean is the screening statistic at the 1M probe scale, CE columns are pretraining cross-entropy on the two validation memmaps after 954 steps, and PIQA / HellaSwag are zero-shot. Lower CE is better. Best per column in bold.

Architecture	Suite-mean ↑	c4_en CE ↓	wikitext_103 CE ↓	PIQA ↑	HellaSwag ↑
stu	0.33	5.44	6.67	0.542	0.251
mamba	0.38	5.03	6.24	0.539	0.254
headwise_stu	0.48	5.22	6.50	0.538	0.252
headwise_mamba2	0.50	5.07	6.30	0.539	0.254
headwise	0.50	5.11	6.38	0.534	**0.258**
attn	0.59	5.05	6.17	0.536	0.255
alt_attn_mamba2	0.62	5.04	6.14	**0.547**	0.255
alt_attn_stu	0.65	4.99	6.13	0.538	0.255
alt_attn_mamba	**0.68**	**4.94**	**6.03**	0.539	0.256
mamba2*	0.46	5.01	6.17	0.546	0.255
lstm**	0.35	5.72	6.97	0.519	0.254

*Added post-submission (Section K); suite-mean is the five-seed average from Section J.
**Out-of-lineage recurrent baseline; suite-mean computed on the 29-task subset covered for all eleven architectures (Section K).

G. 150M held-out evaluation, full table

Table 10 reports the full per-task numbers behind the headline correlation in Section 3.2. Suite-mean is the screening statistic at the 1M probe scale, CE columns are pretraining cross-entropy on the c4_en and wikitext_103 validation memmaps, and PIQA / HellaSwag are zero-shot.

Table 10. 150M LM evaluation on the held-out architecture set. Suite-mean is the screening statistic, CE columns are pretraining cross-entropy on the two validation memmaps, and PIQA / HellaSwag are zero-shot. Lower CE is better.

Architecture	Suite-mean ↑	c4_en CE ↓	wikitext_103 CE ↓	PIQA ↑	HellaSwag ↑
STU sandwich	0.50	3.67	4.99	0.529	0.267
mamba	0.74	2.82	3.65	0.547	0.275
alt_attn_mamba	0.78	2.81	3.65	0.555	0.277
attn	0.80	2.81	3.68	0.555	0.277

H. 1B evaluation, full per-task results

Table 11 reports the per-task zero-shot results behind the 1B comparison in Section 3.4. Hydra and the parameter-matched OLMo-2 attention baseline are evaluated at step 12,000, with training cross-entropy on OLMo-Mix-1124. Hydra wins on 8 of 15 zero-shot tasks (with one tie), trailing on the four MMLU subdomains, COPA, and SocialIQA.

I. Additional robustness checks

This appendix collects four perturbations of the suite, each evaluated on the nine-architecture 50M held-out set from Section F. The screening claim survives all four.

Table 11. 1B LM evaluation. Hydra and the OLMo-2 attention baseline at step 12,000. Train CE on OLMo-Mix-1124, downstream evals on the standard OLMo zero-shot battery. Bold = winner.

Metric	Hydra	Attn baseline
Train CE ↓	**2.880**	2.934
Train PPL ↓	**17.82**	18.80
ARC-challenge ↑	**0.321**	0.288
ARC-easy ↑	0.579	0.579
BoolQ ↑	**0.601**	0.600
COPA ↑	0.710	**0.720**
HellaSwag ↑	**0.486**	0.450
OpenBookQA ↑	**0.338**	0.334
PIQA ↑	**0.711**	0.694
SciQ ↑	**0.846**	0.832
SocialIQA ↑	0.431	**0.439**
Winogrande ↑	**0.533**	0.526
CommonsenseQA ↑	**0.383**	0.346
MMLU humanities (5-shot) ↑	0.257	**0.266**
MMLU other (5-shot) ↑	0.270	**0.310**
MMLU social sciences (5-shot) ↑	0.246	**0.273**
MMLU STEM (5-shot) ↑	0.196	**0.212**

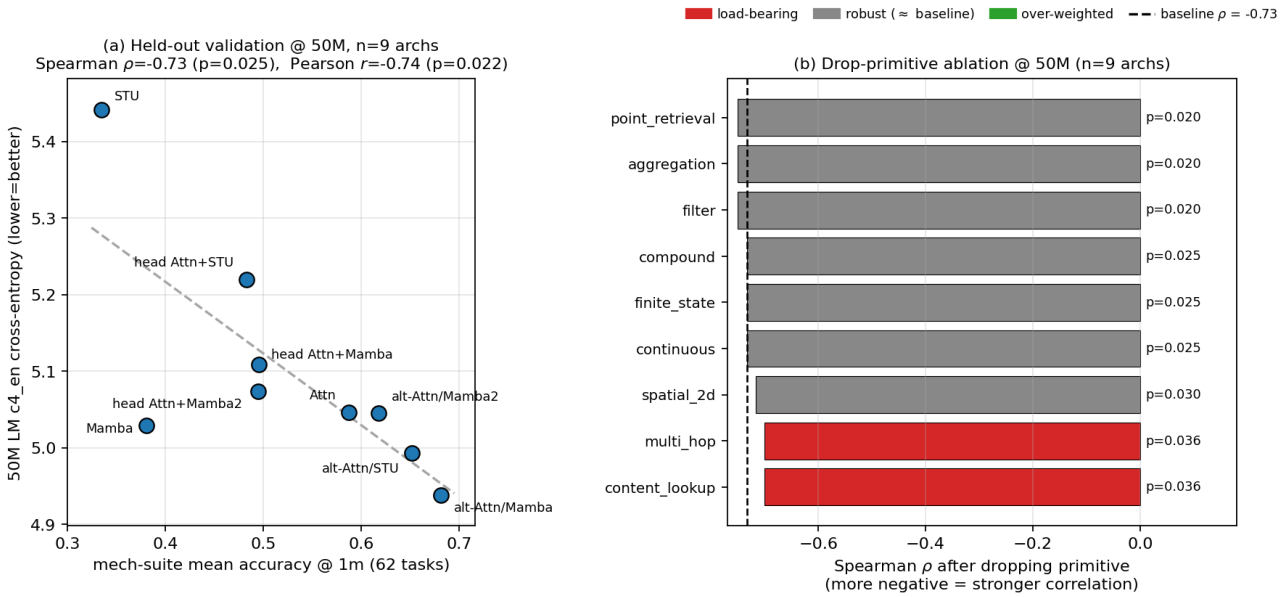

Figure 5. Screen robustness at 50M (n=9). *(a)* Suite-mean accuracy at 1M vs. 50M c4_en CE, $\rho = -0.73$ (p=0.025). *(b)* Rank correlation after dropping each primitive; dashed line is the baseline $\rho = -0.73$. The screen survives every drop.

I.1. Drop-primitive ablation

Figure 5(a) plots the held-out validation scatter underlying the 50M correlation in Section F, with suite-mean accuracy at the 1M probe scale on the x-axis and 50M c4_en cross-entropy on the y-axis ($\rho = -0.73$, p=0.025). Figure 5(b) reports the drop-primitive ablation: each bar is the rank correlation between suite-mean accuracy and 50M c4_en cross-entropy when the named primitive is removed from the screen. The dashed line marks the baseline $\rho = -0.73$ with all primitives included. No single primitive carries the signal; the correlation remains negative and significant under every drop, with the largest movement when content-addressable lookup or multi-hop is removed.

I.2. Multimodal-only transfer

The 27-task headline suite is text-token only. If the screen's predictive power were specific to that surface form, restricting the screen to its multimodal extensions (grid and video probes from Section A) should break the correlation. Figure 6 reports the result. The text-only subset predicts 50M wikitext_103 CE at $\rho = -0.88$ (p=0.002), and the multimodal-only subset (6 grid + video tasks) predicts the same downstream signal at $\rho = -0.73$ (p=0.025), $r = -0.71$ (p=0.032). Both subsets clear $p < 0.05$ on n=9, so the screening signal is not carried exclusively by text-shaped probes; the spatial / temporal extensions reach the same ranking through different surface forms. This is consistent with the primitive-coverage matrix in Table 3, which shows grid and video tasks targeting the same retrieval / aggregation / multi-hop primitives as the text suite.

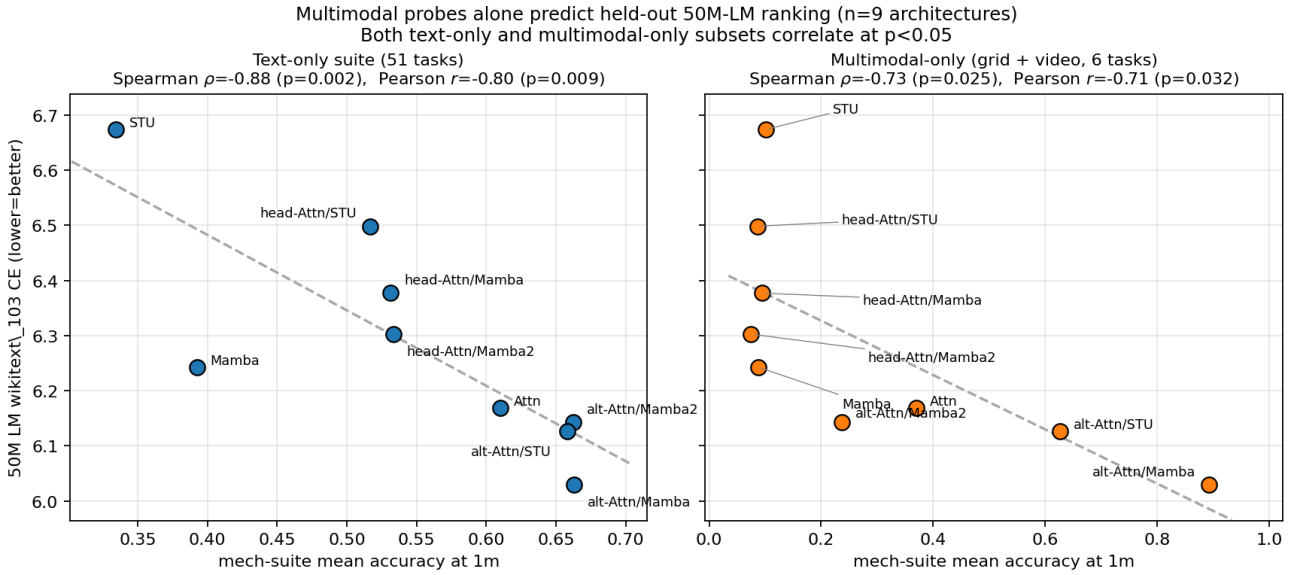

Figure 6. Multimodal probes alone predict held-out 50M-LM ranking (n=9 architectures). Left: text-only subset, $\rho = -0.88$, p=0.002. Right: multimodal-only subset (grid + video, 6 tasks), $\rho = -0.73$, p=0.025. Both subsets clear $p < 0.05$.

I.3. Autoencoding-flavored probes

The headline suite is built around next-token retrieval and aggregation primitives. To check that the screen does not depend on this exact framing, we restrict it to a three-task subset with autoencoding flavor (noisy_copy, compress, reverse_copy), each requiring the architecture to recover or reorganize an input rather than predict a continuation. Figure 7(a) shows that the AE-suite mean alone predicts 50M wikitext_103 CE at $\rho = -0.83$ (p=0.005), $r = -0.84$ (p=0.004). Per-task, Figure 7(b) reports reverse_copy at $\rho = -0.75$ (p=0.020), noisy_copy at $\rho = -0.69$ (p=0.041), and compress at $\rho = -0.62$ (p=0.074). The aggregate is tighter than any single AE task, consistent with the suite-level decomposition rather than any individual probe carrying the signal.

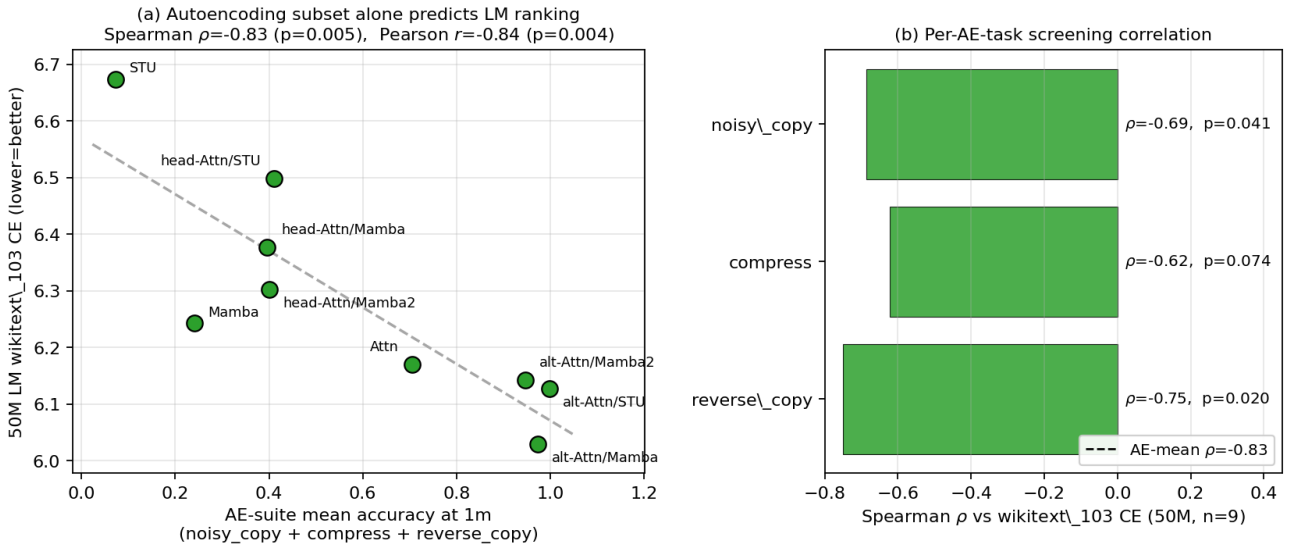

Figure 7. Autoencoding subset alone predicts 50M LM ranking. *(a)* AE-suite mean (noisy_copy + compress + reverse_copy) vs. wikitext_103 CE, $\rho = -0.83$, p=0.005. *(b)* Per-AE-task screening correlation. Aggregate ($\rho = -0.83$) is tighter than any single AE task.

I.4. Compositional-depth invariance

A screen that only ranks architectures correctly at a single difficulty level is fragile: the suite uses depth-2 hops as its default, and downstream tasks vary in compositional depth. We test whether the cross-architecture ranking at low depth predicts the ranking at higher depth on the hop-graph family (two_hop, three_hop, deep_hop, k_hop), running each at hop counts $k \in \{2, 3, 4\}$ across the same nine architectures. Figure 8(a) shows the per-architecture accuracy curves; absolute accuracy increases with k for most architectures, with STU as the only non-monotonic case (consistent with the controlled `deep_hop` sweep in Section D). Across architectures, the k=2 ranking predicts the k=4 ranking at ρ=0.67 (p=0.049), r=0.69 (Figure 8(b)). The full pairwise matrix in Figure 8(c) shows positive cross-depth Spearman across every pair of hop tasks (range 0.32 to 0.75), with the weakest link at the largest depth gap (two_hop vs k_hop, ρ=0.32). Rankings therefore degrade gracefully with depth gap rather than re-ordering entirely, which is what a screen used at small scale to predict large-scale behavior needs.

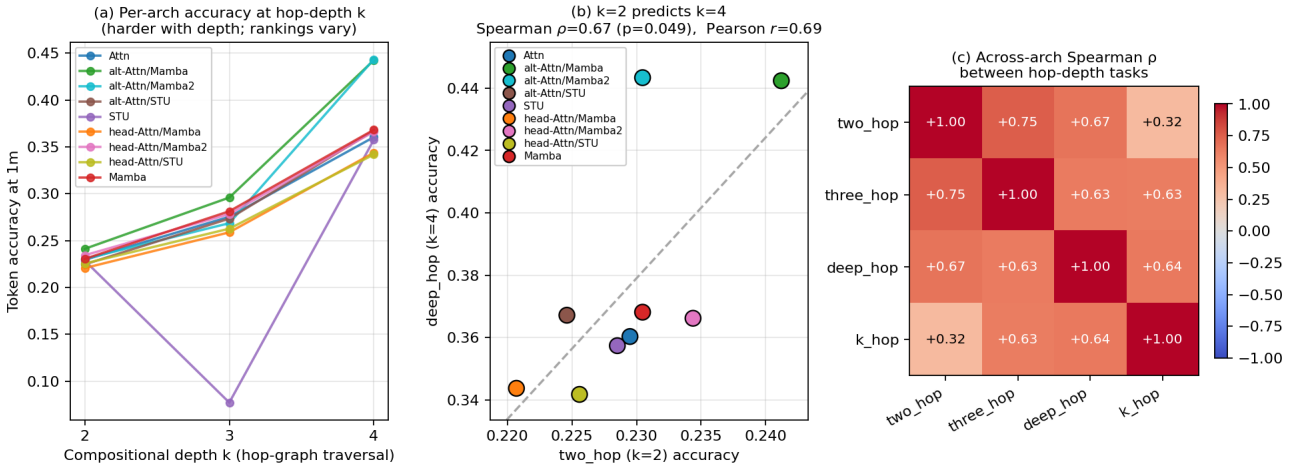

Figure 8. Compositional-depth invariance. *(a)* Per-architecture token accuracy at the 1M scale across hop depths $k \in \{2, 3, 4\}$. All architectures monotonic except STU. *(b)* Cross-architecture scatter of k=2 vs k=4 accuracy, ρ=0.67, p=0.049. *(c)* Pairwise across-architecture Spearman ρ between hop-depth tasks; all entries positive, with the weakest cross-depth link at the largest depth gap.

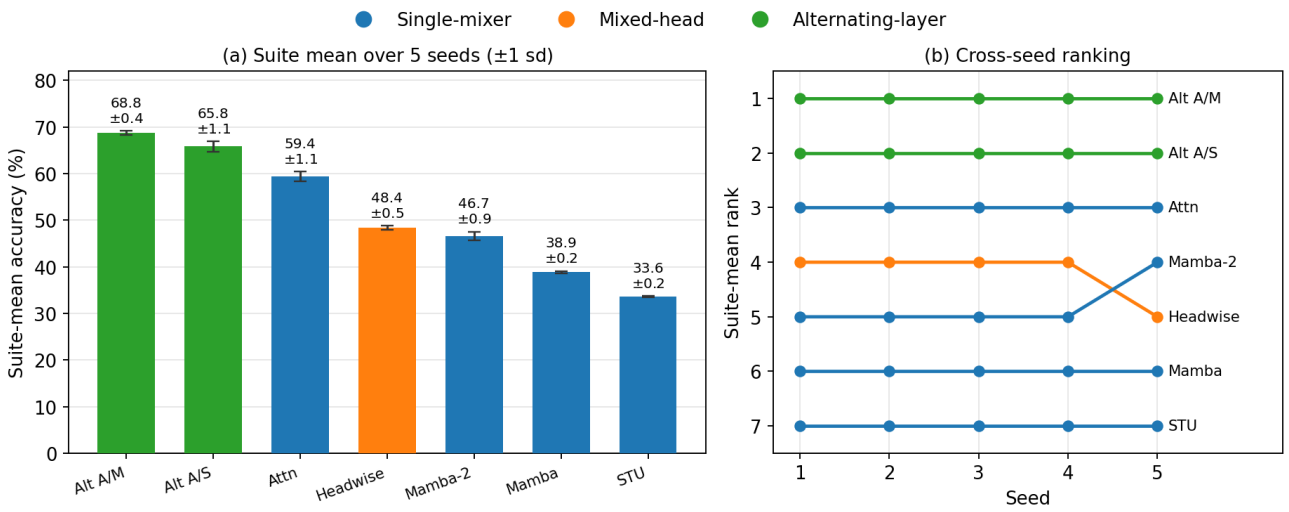

Figure 9. Five-seed replication of the 1M probe sweep across seven architectures. *(a)* Suite-mean accuracy with ± 1 standard deviation across seeds; the largest per-architecture deviation is ± 1.1 percentage points. *(b)* Cross-seed suite-mean ranking: identical on four of five seeds, with a single adjacent transposition (`headwise` vs. `mamba2`, whose suite means differ by 1.7 points) on seed 5.

J. Seed variability of the 1M screen

Both reviewers of the workshop submission flagged single-seed probe runs as a variance risk. For the camera-ready we replicated the full 1M sweep across five random seeds for seven architectures (`attn`, `stu`, `mamba`, `mamba2`, `headwise`, `alt_attn_mamba`, `alt_attn_stu`), training every task independently per seed (2,205 runs total). Figure 9 summarizes the result; the per-run accuracies are released with the code.

Three observations support the reliability of the screen. First, suite-mean standard deviation across seeds ranges from ± 0.2 to ± 1.1 percentage points per architecture, an order of magnitude smaller than the cross-architecture spread the screen relies on (33.6 to 68.8 points). Second, the cross-architecture ranking is essentially seed-invariant: every pairwise between-seed Spearman correlation of suite means is at least 0.964 (mean 0.986), and the only rank change anywhere in the replication is one adjacent swap. Third, the screen–downstream fit does not depend on the seed: computing suite means from any single seed and correlating against 50M wikitext_103 CE over the seven overlapping architectures gives $\rho = -0.89$ for seeds 1–4 and $\rho = -0.96$ for seed 5 (all $p < 0.01$). At the per-task level the median across-seed standard deviation is 0.7 accuracy points; the tail (90th percentile 4.2 points) is concentrated in tasks near their phase transition for a given architecture, which is precisely where averaging over five seeds helps most.

K. Extended 50M architecture set

The submission validated the 50M correlation on nine architectures. Post-submission we added two more: a single-mixer **Mamba-2** (its five-seed suite mean from Section J covers the full task suite) and a parameter-matched **LSTM**, the first architecture outside the attention/Mamba/STU lineage the suite was developed on. The LSTM's probe coverage is the 29-task subset shared by all eleven architectures (dominated by multi-hop, spatial, and lookup tasks), so panel (b) of Figure 10 compares all architectures on that common subset.

With Mamba-2 added (n=10, full suite), the wikitext_103 correlation is $\rho = -0.81$ ($p = 0.005$), $r = -0.77$; the c4_en correlation weakens to $\rho = -0.56$ ($p = 0.09$) under mean aggregation and recovers to $\rho = -0.69$ ($p = 0.03$) under median aggregation (Section L). Mamba-2 is the single largest placement error in the set: the screen puts it mid-pack while its c4_en CE is fourth-best, consistent with the anomalous single-hop specialization noted in Section D. With the LSTM also added (n=11, common subset), the correlations are $\rho = -0.77$ ($p = 0.005$) vs. c4_en CE and $\rho = -0.86$ ($p = 0.0006$) vs. wikitext_103 CE.

The LSTM result deserves an honest reading. The screen ranks it seventh of eleven, above `stu`, `mamba`, and two mixed-head hybrids, but its 50M cross-entropy is the worst of all eleven architectures by a wide margin (6.97 vs. 6.67 for the next-worst).

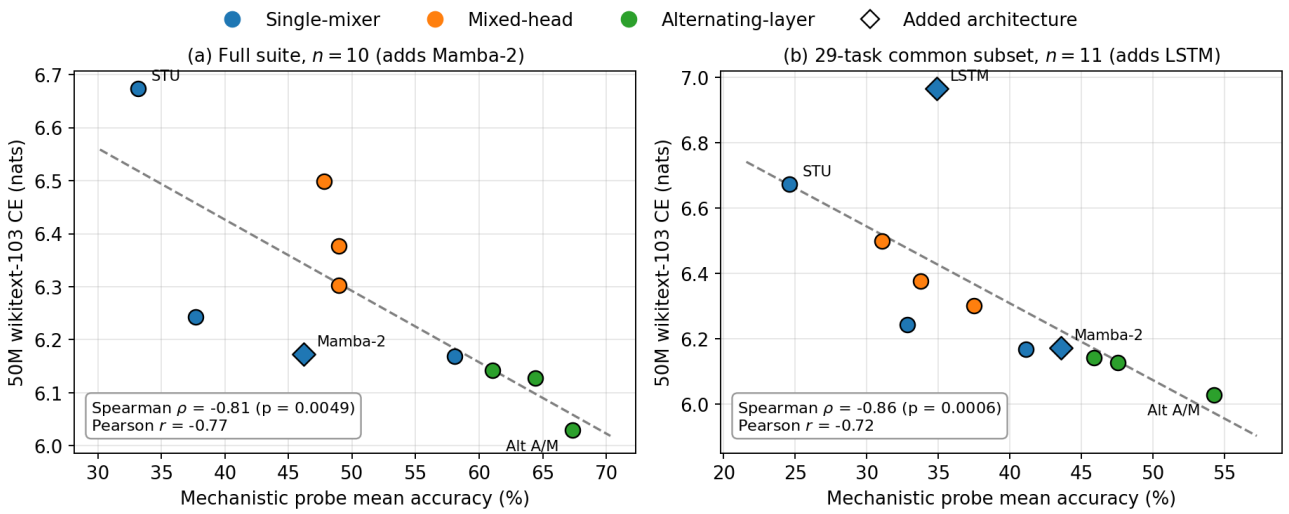

Figure 10. Extending the 50M held-out set. *(a)* Full-suite screen with Mamba-2 added (n=10): $\rho = -0.81$ (p=0.005) vs. wikitext_103 CE. *(b)* All eleven architectures on the 29-task common subset, adding the out-of-lineage LSTM: $\rho = -0.86$ (p=0.0006). Diamonds mark the added architectures. The LSTM sits far above the trend line: the screen orders it correctly relative to the weakest in-lineage mixers but underestimates its downstream gap.

The aggregate correlation survives ($p < 0.001$) because the LSTM still lands on the correct side of the distribution, but the screen clearly compresses the gap between a legacy recurrent architecture and modern dense mixers. We take this as evidence that ordinal screening transfers outside the development lineage while cardinal calibration does not, and we flag rank-based use as the only supported mode there (see Section 4).

L. Aggregation-rule sensitivity

The screening statistic in the main text is an arithmetic mean of per-task accuracy. Table 12 recomputes the 50M rank correlations under four alternative aggregation rules, on both the n=9 submission set and the extended n=10 set from Section K. Median and 10%-trimmed aggregation reproduce the arithmetic-mean correlations essentially unchanged (the median is slightly *stronger* at n=10); the lower-quartile statistic is mildly weaker; only the minimum, which reduces the suite to each architecture's single worst task, loses significance. The screening signal is therefore a property of the bulk of the per-task accuracy distribution rather than of a particular aggregation formula.

Table 12. Spearman ρ between the 1M screening statistic and 50M pretraining CE under alternative aggregation rules. n=9 is the submission's held-out set; n=10 adds Mamba-2 (Section K). Asterisks: $^{*}p < 0.05$, $^{**}p < 0.01$.

	n=9		n=10	
Aggregation	c4_en	wikitext_103	c4_en	wikitext_103
Arithmetic mean	-0.73^{*}	-0.88^{**}	-0.56	-0.81^{**}
Median	-0.73^{*}	-0.88^{**}	-0.69^{*}	-0.88^{**}
Trimmed mean (10%)	-0.73^{*}	-0.88^{**}	-0.56	-0.81^{**}
Lower quartile	-0.58	-0.77^{*}	-0.46	-0.72^{*}
Minimum	-0.33	-0.58	-0.29	-0.62

M. Task redundancy and a distilled screen

A reviewer asked whether the 62-task suite is internally redundant. Measured across the ten full-coverage architectures of Section K, the median pairwise between-task Spearman correlation is 0.36, and only 2.1% of task pairs exceed $|\rho| > 0.9$, so most tasks are not substitutes for one another. The redundancy that does exist is deliberate: each primitive is probed by at least two tasks so that the drop-primitive ablation of Section 3.5 is meaningful.

Redundancy with respect to the *aggregate ranking* is a different question. Greedy forward selection over tasks finds that a

two-task screen (induction_gap + k_hop) already reproduces the full-suite architecture ranking exactly ($\rho = 1.0$ over the ten architectures) and predicts 50M wikitext_103 CE at $\rho = -0.81$, matching the full suite. We caution that this subset was selected on the same ten architectures it is evaluated on, so it is best read as an upper bound on distillability rather than a validated cheap screen; selecting on held-out architectures is future work. The practical implication stands, though: for rapid iteration, a screening pass over a handful of retrieval-composition tasks recovers most of the ranking signal at a fraction of the cost, and the full suite is what buys the robustness and diagnostic decompositions used elsewhere in the paper.

N. Learning-rate sensitivity

All probe runs share a single learning rate (3×10^{-4}, Table 7). A reviewer noted that architecture families differ in learning-rate sensitivity, so a low probe score could in principle reflect a learning-rate mismatch rather than a capability bottleneck. We test this directly by re-running the 11 headline basic-family tasks at 1×10^{-4} and 1×10^{-3}—bracketing the default by a factor of three in each direction—for the two single-mixer architectures most exposed to the concern (stu, mamba) and the two attention-containing architectures (attn, alt_attn_mamba), holding every other hyperparameter fixed. Figure 11 plots the resulting suite means.

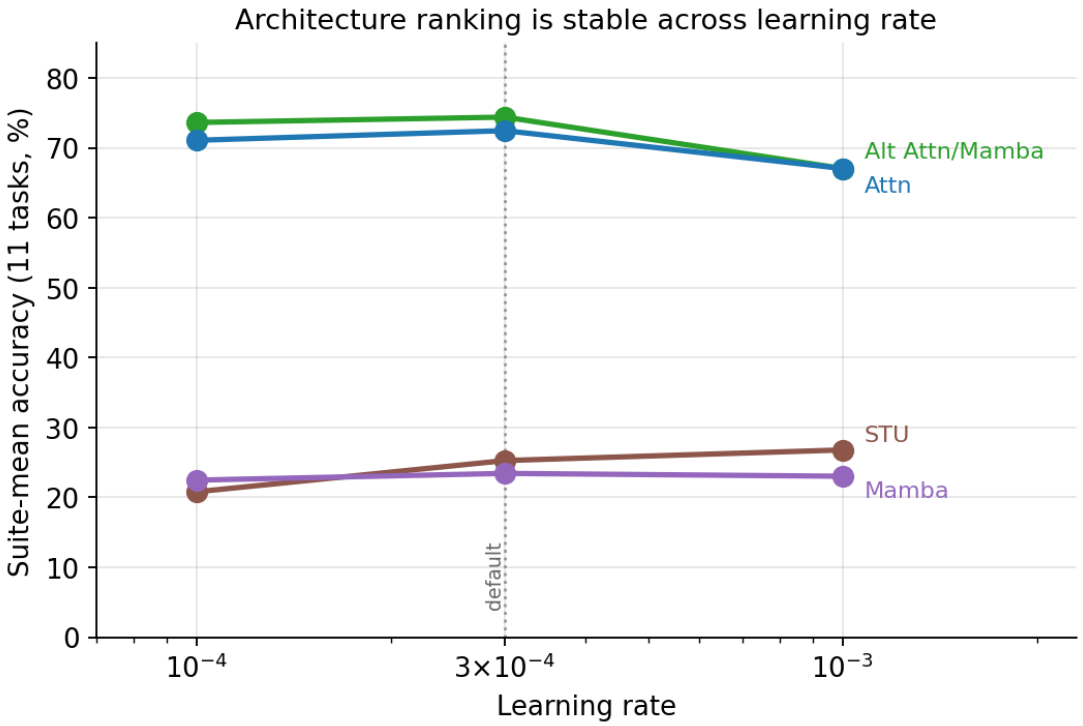

Figure 11. Suite-mean accuracy on the 11 headline tasks as a function of learning rate, for two attention-containing and two pure non-attention architectures. The retrieval-driven gap between the two groups (∼45 percentage points) is present at every learning rate; the dotted line marks the default 3×10^{-4}.

The capability ordering is robust to the learning rate. The two attention-containing architectures score 67–74% at every learning rate, the two pure non-attention architectures 21–27%, and the ∼45-point gap that drives the screen never closes—STU and Mamba do not recover retrieval at any learning rate we tried, consistent with an architectural rather than an optimization bottleneck. The cross-learning-rate Spearman correlation of the four-architecture ranking is 1.0 between 3×10^{-4} and 10^{-3} and 0.8 between the extreme rates, the only movement being a swap of stu and mamba at the bottom of the table, where the two sit within 2 points of one another. Critically, selecting each architecture's *best* learning rate independently (an oracle tuning that a real screen would not have) leaves the ranking unchanged ($\rho = 1.0$ vs. the default-rate ranking): no architecture's per-family profile is rescued by a different learning rate. We therefore attribute the probe-score differences to architectural capability, not hyperparameter mismatch.