What Do Language Models Learn and When? The Implicit Curriculum Hypothesis

Emmy Liu[1], **Kaiser Sun**[2], **Millicent Li**[3], **Isabelle Lee**[4], **Lindia Tjuatja**[1], **Jen-tse Huang**[2], **Graham Neubig**[1]
[1]Language Technologies Institute, Carnegie Mellon University
[2]Department of Computer Science, Data Science and AI Institute, Johns Hopkins University
[3]Khoury College of Computer Science, Northeastern University
[4]Department of Computer Science, University of Southern California
emmy@cmu.edu

Abstract

Large language models (LLMs) can perform remarkably complex tasks, yet the fine-grained details of how these capabilities emerge during pretraining remain poorly understood. Scaling laws on validation loss tell us how much a model improves with additional compute, but not what skills it acquires in which order. To study this, we propose the *Implicit Curriculum Hypothesis*: behavioral onsets during ordinary pretraining exhibit a structured, compositional ordering across models and data mixtures. We test this by designing a suite of simple, composable tasks spanning retrieval, morphological transformations, coreference, logical reasoning, and mathematics. Using these tasks, we track behavioral onset, the first checkpoint at which performance crosses a threshold, across four model families spanning sizes from 410M–13B parameters. We find that onset orderings under fixed accuracy thresholds are strikingly consistent ($\rho = .81$ across 45 model pairs), and that composite tasks most often emerge after their component tasks. Furthermore, we find that this structure is encoded in model representations: tasks with similar function vector representations also tend to follow similar trajectories in training. By using the space of representations derived from our task set, we can effectively predict the training trajectories of simple held-out compositional tasks throughout the course of pretraining ($R^2 = .68$–$.84$ across models) without previously evaluating them.[1]

1 Introduction

Large language models (LLMs) exhibit predictable improvements in performance with scale, a phenomenon characterized by well-established scaling laws (Hoffmann et al., 2022; Gadre et al., 2025; Muennighoff et al., 2023). These scaling laws tell us how much models are expected to improve in predicting the next token on the pretraining distribution given additional compute, but not what skills the model acquires, or when during pretraining it acquires them specifically. In practice, training runs may cost millions of dollars, yet are primarily monitored through aggregate cross-entropy loss, or through evaluating at intervals on downstream benchmarks such as MMLU (Hendrycks et al., 2021). However, neither approach provides actionable diagnostic information. Cross-entropy loss decreases smoothly even as qualitatively different skills are acquired at sudden transition points (Kangaslahti et al., 2025). Downstream benchmarks compose many prerequisite skills, making failures opaque: when GSM8k (Cobbe et al., 2021) performance stalls, it is unclear whether the bottleneck is numerical fluency, multi-step planning, or natural language understanding. For instance, scoring well on GSM8k may require numerical fluency, multi-

[1]Data and code available at https://github.com/KaiserWhoLearns/ElementalTask

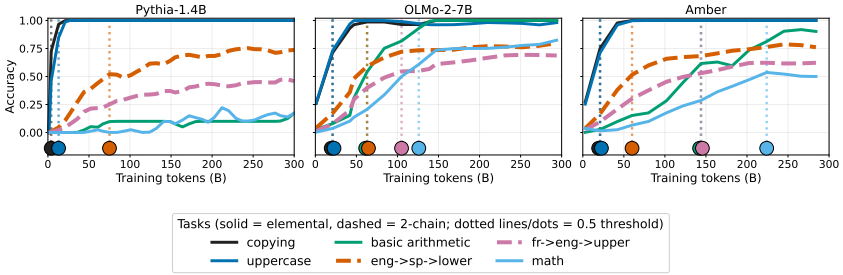


Figure 1: Emergence order across model families and sizes, smoothed with a Gaussian kernel ($\sigma = 1.0$). Dots represent the point at which the model reaches a fixed 50% accuracy threshold. Note that this figure is illustrative and does not cover the full task set.

step planning, as well as natural language understanding, making it difficult to diagnose which prerequisite skills are missing when performance stalls (Meister & Cotterell, 2021).

A growing body of theoretical work suggests that neural networks learn functions sequentially, acquiring simpler patterns before more complex ones (Lee et al., 2025; Zhang et al., 2026). Recent work has built on these insights, hypothesizing that complex behaviors and scaling laws themselves emerge from the combination of more elementary sub-tasks that serve as fundamental building blocks (Khandelwal & Pavlick, 2025) or *quanta* (Michaud et al., 2023). We use observable tasks as diagnostic probes and do not assume that they are discrete or minimal units of model computation. However, much of this theoretical work has focused on simplified modeling settings, leaving open questions about how these insights translate to large-scale language model pretraining (Srivastava et al., 2023). Prior empirical work has shown that certain knowledge categories (e.g., syntactic vs. factual) are acquired at different rates (Liu et al., 2021), and that grammatical phenomena are learned in a consistent order across architectures (Friedman et al., 2022). Related diagnostic work uses training dynamics to characterize examples and dataset difficulty (Swayamdipta et al., 2020; Ethayarajh et al., 2022). However, these studies have not examined whether the ordering reflects *compositional* dependencies between skills, nor whether it is legible in the model's internal representations.

Based on this, we propose the **Implicit Curriculum Hypothesis**: **during pretraining, behavioral onsets exhibit a statistically structured ordering across models and tend to respect known compositional relations**. To test the Implicit Curriculum Hypothesis, we design a suite of simple tasks that probe a wide range of skills. We track behavioral onset across 10 models from 4 families (410M–13B parameters) and find:

1. **The emergence ordering is consistent across model families.** Spearman correlations between onset orderings range from $\rho = .64$ to .93 (mean .81) across all 45 model pairs, including cross-family comparisons. Copying is the first skill to emerge, followed by many simple string operations, fact extraction and coreference, then logic operations, simple world knowledge, then multistep arithmetic and more complex reasoning tasks.Composite tasks emerge after their elemental prerequisites. Agreement is strongest for earlier tasks, weaker for later tasks, and substantially lower under relative thresholds.

2. **The ordering is legible in model representations.** Tasks whose internal representations are nearby in the model's residual stream, measured via function vectors (Todd et al., 2024), follow similar learning trajectories. This proximity is sufficient to predict the full training trajectory of held-out composite tasks (mean R^2 of .68–.84 across models, with per-task R^2 exceeding .95) without ever evaluating them during training.

Together, these results suggest that pretraining is more structured than loss curves reveal: skills emerge in an order that is consistent across models and readable from model internals.

2 Preliminaries

2.1 Background

We provide a summary of the work that we directly build on in this paper. Further related work can be found in Appendix B.

Scaling Laws Scaling laws characterize the relationship between a model's held-out validation loss L and the compute budget allocated to training, typically decomposed into model size N and data size D. These relationships are well-approximated by power laws of the form $L(N, D) \propto N^{\alpha} + D^{-\beta} + L_{\infty}$ (Kaplan et al., 2020; Hoffmann et al., 2022), and hold across many orders of magnitude. However, this aggregate loss curve does not directly correlate with downstream performance (Lourie et al., 2025; Isik et al., 2026; Liu et al., 2026), and it is not clear what the model is learning as loss decreases.

Quantization Hypothesis Michaud et al. (2023) offers a hypothesis that these smooth scaling curves arise from the learning of discrete skills, termed *quanta*. Under this framework, a model acquires these quanta in an order optimized to reduce total loss, hiding the discrete transitions that correspond to the model learning. One practical quandary is that quanta require a post-hoc discovery method, the results of which often do not correspond to interpretable skills (Michaud et al., 2023). While compelling, the Quantization Hypothesis typically treats these skills as independent, additive contributions, leaving the structural dependencies and compositional nature of these skills largely unexplored.

Simplicity Bias Secondly, works show that neural networks trained with gradient-descent-based methods tend to exhibit **simplicity bias**, a tendency to learn simpler functions before more complex ones (Saxe et al., 2014; Nakkiran et al., 2019; Shah et al., 2020). In the context of language modeling, this is reflected in models learning lower-order n-grams before higher-order ones (Michaelov et al., 2025). However, we note that the notions of simplicity are often underspecified, and moreover, these function-level notions of complexity cannot be used to quantify task complexity.

Compositional Skill Structure A recent line of work has investigated whether skills acquired by language models follow explicit dependency structures. Chen et al. (2023b) represent skills as directed acyclic graphs (DAGs), where an edge from skill A to skill B indicates that training on data associated with A reduces the amount of data needed to learn B. Such dependency graphs can be used to design curricula for target skills. While their work focuses on post-training and defines skills by data clusters, a natural question is whether we can also characterize the dependency structure of general web-data-based pretraining. Theoretically, Arora & Goyal (2023) also provide a framework relating cross-entropy loss to competence on individual sets of skills, showing that a decrease in loss implies simultaneous improvement in both individual skills and their k-tuples.

2.2 The Implicit Curriculum Hypothesis

The four threads of work above establish that (1) capabilities may unlock progressively, (2) simpler functions are learned before more complex ones, and (3) skills may have a dependency structure. However, they leave open whether these threads combine in practice: does large-scale pretraining follow a structured, compositional ordering of skill acquisition that is consistent across models? We use *implicit curriculum* descriptively for this observed ordering, in contrast to explicit curriculum learning.

The Implicit Curriculum Hypothesis

Let $\mathcal{T}$ be a set of tasks equipped with a *known* dependency relation $\prec$, where $\tau_i \prec \tau_j$ indicates that task τ_j was constructed to compositionally depend on τ_i. Let $P(\tau) = \{\tau' \in \mathcal{T} : \tau' \prec \tau\}$ denote the prerequisite set of task τ under this construction.

We emphasize that $\prec$ reflects our *design-level* task structure, not a claim about the model's internal primitives or the existence of any particular dependency in the model. Let $t^*_\tau(m)$ denote the emergence time of task τ for model m, defined as the first training step at which performance exceeds a threshold $\theta \in \mathbb{R}$. We hypothesize that this design-level structure is *reflected empirically* in model training dynamics:

H1. Compositional ordering. Tasks emerge no later than the tasks constructed to depend on them:

$$\forall\, \tau_j \in \mathcal{T},\ \forall\, \tau_i \in P(\tau_j): \quad t^*_{\tau_i}(m) \leq t^*_{\tau_j}(m)$$

H2. Cross-model stability. The emergence ordering induces a partial order $\preceq_{\mathcal{T}}$ over tasks that is consistent across models: for models m_1, m_2, the rank correlation between their emergence orderings is significantly higher than chance.

H3. Representational alignment. Tasks with similar internal representations exhibit similar learning trajectories. That is, for tasks τ_i, τ_j with representation vectors v_i, v_j, $\exists\, \epsilon \in \mathbb{R}^+$:

$$\text{Sim}(v_i, v_j) \text{ high} \implies d\Big(a_{\tau_i}(\cdot),\ a_{\tau_j}(\cdot)\Big) < \epsilon$$

where $a_\tau(\cdot)$ is the learning trajectory of task τ and d is a distance over trajectories. This further implies that trajectories of unseen tasks can be predicted from representational geometry alone, without evaluating them during training. This assumes that the diagnostic basis covers relevant regions of task-representation space, but does not require task components to be orthogonal.

3 Methodology

3.1 Models and Checkpoints

In order to test our hypotheses, we focus on examining open-weight models with publicly-released intermediate pre-training checkpoints. Because our hypotheses are largely about timing and emergence order, it was also important to select models with relatively dense intermediate checkpoints and larger sizes. The selected models are:

- **OLMo-2** (OLMo et al., 2024): 1B, 7B, and 13B parameter models, providing a within-family scale comparison across an order of magnitude.
- **OLMo-3** (Olmo, 2025): 7B, offering comparison with a newer generation compared to OLMo-2.
- **LLM360** (Liu et al., 2023): Crystal (7B) and Amber (7B), trained on very different data mixtures (code-oriented and natural-language-oriented, respectively), allowing us to study the effect of data composition within the same model family.
- **Pythia** (Biderman et al., 2023): 410M, 1.4B, 2.8B, and 12B, offering a comparison with an earlier model generation trained on different data. We selected sizes spanning the full range of the suite; models below 410M were excluded due to poor performance.

In order to keep checkpoint sampling consistent across families, we focused on up to the first 1T tokens of training for each model and sampled approximately 20 checkpoints for each model within this range, giving a granularity of roughly every 20B tokens. We hypothesized that this would capture the period for which most relevant simple skills emerge for the tasks we study, while the granularity would be sufficient to resolve ordering differences.

3.2 Task Design

We design tasks with intuitive compositional relationships, diverse operation types, and unambiguous outputs, while keeping them simple enough for models as small as 1B parameters to eventually solve via in-context learning. We therefore evaluate all 91 elemental and composite tasks using exact-match accuracy; the full list is given in Appendix C.

Simple tasks. We define a set of simple tasks spanning string manipulation (e.g., copy, uppercase, first letter), morphological transformation (e.g., singular to plural, present to gerund), knowledge retrieval (country to capital, country to currency), and translation (e.g., `en-fr`, `en-sp`). These were selected to cover distinct operation types while remaining simple enough to be plausibly atomic. Notably, several of these operations have also been investigated in the interpretability literature (Olsson et al., 2022; Hendel et al., 2023; Todd et al., 2024; 2026). We do not claim that these are the true minimal units of model computation, but they serve as a diverse set of operations from which we can construct composites with known structure. In total, we create 53 simple tasks.

Task	Input	Out
`simple_icl: present_to_gerund`	run	running
`fact_extraction: extract_number`	Passage: "John gave 5 apples to Mary on Tuesday." How many apples?	5
`logical_ops: conditional`	If it rains, the ground gets wet. It rains. Is the ground wet?	yes
Compositional tasks		
`comp: gerund_upper`	run	RUNNING
`comp: translate_sp_eng_reverse`	hola	olleh
`comp: extract_verify`	Passage: Nora gave 3 apples to Ben. Ben gave 1 to Li. Claim: Ben received apples first. Does the claim follow?	True

Table 1: Example simple and compositional tasks. Compositional tasks require chaining multiple primitive skills.

Composite tasks: synthetic chains. We construct composite tasks by chaining elemental operations in sequence. For example, `gerund_upper` applies the gerund transformation followed by uppercasing (write → WRITING). This mechanical construction guarantees that the compositional prerequisites are known exactly, yielding 38 composite tasks. The inclusion of translation-based composites (e.g., `translate_eng_fr_upper_reverse`) additionally tests whether knowledge-dependent elementals compose in the same way as rule-based ones.

3.3 Measuring Behavioral Onset (Emergence)

Prior work has proposed several notions of emergence, including scale-based definitions and parametric fits to learning curves (Wei et al., 2022; Snell et al., 2024). Because apparent discontinuities can depend on the metric (Schaeffer et al., 2023), we use *behavioral onset* to mean only the first observed threshold crossing, whether the underlying curve is gradual or sharp. Because many trajectories are noisy or irregular, we use simple threshold-based definitions. We consider two variants:

Absolute threshold. We define onset time $t^*_\tau(m)$ as the first checkpoint at which task τ exceeds a fixed accuracy threshold θ_{abs}.

Relative threshold. We alternatively define onset time as the first checkpoint at which performance reaches a fraction α of the model's best performance on that task.

In-context evaluation. All tasks use 5-shot prompting. Simple tasks use a fixed demonstration set, while compositional-task demonstrations are sampled from task-specific pools. We use ICL because small base models often do not follow direct instructions and because the format supports function-vector extraction. Thus, onset measures task ability as elicited through ICL, which may especially affect early checkpoints.

3.4 Measuring Representational Similarity

To operationalize representational alignment, we require a per-task representation that captures the computation the model performs in order to do the task. Following the methodology from Todd et al. (2024), we extract task representations (function vectors) from the models.

Extraction. Let a transformer have L blocks and hidden dimension d. For block ℓ, let

$$h_\ell^{\text{attn}} = h_{\ell-1} + \text{Attn}(\text{LN}(h_{\ell-1}))$$

denote the post-attention hidden state, and let

$$h_\ell = h_\ell^{\text{attn}} + \text{MLP}\big(\text{LN}(h_\ell^{\text{attn}})\big)$$

denote the block-output hidden state. For each task τ, we construct a set of ICL prompts, perform a forward pass for each prompt, and extract activations at the last non-pad token position t_{last} (i.e., the position from which the model begins generating its answer). We retain only prompts on which the model produces the correct answer, ensuring that the extracted representation reflects successful task execution. We consider two extraction methods, and for each model use the one that performs best (see Appendix G).

Head-based extraction. We use causal indirect effect (CIE) analysis to identify a sparse set of attention heads $\mathcal{H} \subseteq [H] \times [L]$ with the strongest causal effects on task performance. The function vector is then the average of these heads' outputs across correctly answered prompts:

$$v_\tau^{\mathcal{H}} = \frac{1}{|\mathcal{D}_\tau^+|} \sum_{x_i \in \mathcal{D}_\tau^+} \sum_{(h,j) \in \mathcal{H}} a_h^j(x_i),$$

where $a_h^j(x_i)$ is the output of attention head h in block j, evaluated at position t_{last}, and $\mathcal{D}_\tau^+$ denotes the set of correctly answered prompts. We additionally constrain all selected heads to come from the same block.

Hidden-state extraction. Alternatively, we extract the block-output hidden state at block ℓ and position t_{last}:

$$v_\tau^\ell = \frac{1}{|\mathcal{D}_\tau^+|} \sum_{x_i \in \mathcal{D}_\tau^+} h_{\ell, t_{\text{last}}}(x_i),$$

where $h_{\ell, t_{\text{last}}}(x_i) \in \mathbb{R}^d$ is the post-MLP hidden state at block ℓ and position t_{last}.

Task similarity We measure similarity between tasks via cosine similarity between their task representations. The hypothesis predicts that tasks with higher representational similarity exhibit more similar learning trajectories.

3.5 Evaluation Protocol

We evaluate the Implicit Curriculum Hypothesis through two complementary analyses, corresponding to the behavioral claims (H1, H2) and the representational claim (H3).

Testing compositional ordering (H1). For each composite task c with a known set of prerequisite tasks $P(c)$, we check whether all prerequisites emerge no later than the composite:

$$\forall\, \tau \in P(c): \quad t_\tau^*(m) \leq t_c^*(m)$$

We report the violation rate: the fraction of (composite, prerequisite, model) triples for which this ordering is violated across all models. For synthetic chain composites, $P(c)$ is known by construction.

Testing cross-model stability (H2). For each pair of models (m_1, m_2), we compute the Spearman rank correlation between their emergence orderings σ_{m_1}, σ_{m_2} over the full task set. We report correlations separately for the absolute and relative threshold definitions. For tasks that remain unemerged by the end of training, we bin them into one bucket at the end.[2]

Leave-one-out prediction of composite trajectories (H3). We operationalize H3 through a leave-one-out (LOO) protocol over composite tasks. For a held-out composite task c, we predict its learning trajectory from the trajectories of its nearest neighbors in FV space.

Before prediction, we interpolate basis task trajectories onto the held-out task's token grid, apply Gaussian smoothing ($\sigma = 1.0$), and also discard tasks with near-zero trajectory variance.[3]

We extract unit-normalized residual stream representations for all tasks at the selected layer and compute pairwise similarities using an RBF kernel:

$$K(v_i, v_j) = \exp\Big(-\frac{\|v_i - v_j\|^2}{2\sigma_k^2}\Big)$$

We use kernel ridge regression to learn a predictor for the held-out task performance: Let S denote the set of training tasks (excluding c), let $K_S \in \mathbb{R}^{|S|\times|S|}$ be the kernel matrix with entries

$$(K_S)_{ij} = K(v_{\tau_i}, v_{\tau_j}),$$

and let

$$k_c = \big[K(v_c, v_{\tau_j})\big]_{j\in S}$$

be the vector of similarities between the held-out task and the training tasks. For each training step t, we form the vector of training trajectory values

$$y_t = \big[a_{\tau_j}(t)\big]_{j\in S}.$$

Kernel ridge regression solves

$$\alpha_t = (K_S + \lambda I)^{-1} y_t$$

and predicts the held-out trajectory as

$$\hat{a}_c(t) = k_c^\top \alpha_t.$$

We evaluate prediction quality via per-task Pearson r^2 and MAE against smoothed ground-truth trajectories, and report both per-task results and means across all held-out composites. To test the composition bottleneck, we compare two conditions for the function vector space:

1. **All tasks:** the basis includes both simple and composite tasks excluding the held-out target.
2. **Simple tasks only:** the basis is restricted to non-composite tasks only.

If prediction quality degrades substantially under the elementals-only basis, this indicates that composite trajectories share structure with one another that is not captured by their elemental components alone, indicating a composition bottleneck.

4 Emergence Order Results

We first test H1 and H2 by examining whether the emergence order of tasks is consistent across models and whether composites emerge after their constituent components. Figure 2

[2] i.e. their emergence time is considered to be 1001B tokens for a 1T training run

[3] in practice, this ended up being compositions of the `reverse` task as they were usually 0 throughout training.

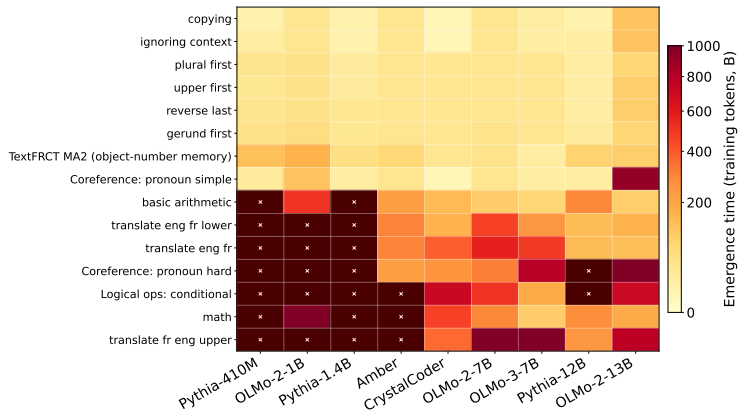


Figure 2: Emergence order heatmap of selected tasks across models (absolute threshold $= 0.8$). Tasks are sorted by consensus emergence order. Agreement is clearer for earlier tasks, while later tasks show greater variability. Quantitative results are reported in Table 2.

shows selected tasks across models, sorted vertically by consensus emergence order. Tasks that emerge early in one model tend to do so across all models. Copying and simple coreference resolution emerge early across all models, in line with previous work (Yin & Steinhardt, 2025). These are followed by simple ICL tasks such as uppercasing and lowercasing, then morphological transformations, followed by knowledge-dependent tasks such as translation, and finally a long tail of more difficult or compositional tasks. Among all compositional comparisons, 54/76 emerged no earlier than their "parent" tasks. However, we also observe inversions, where the composite task emerged earlier than one (19 weak inversions) or both (3 strong inversions) parents. Notably, all three strong inversions involve the `first_letter` component task. This indicates that compositional ordering is a statistical tendency rather than a strict rule, but nonetheless it is usually respected.

A rank-variance analysis at the canonical absolute threshold reveals patterns in tasks that are acquired early as opposed to late. Early case, string, and morphological tasks, including `singular_to_plural`, `lowercase`, `upper_first`, and `lower_first`, have rank standard deviations of 2.8–3.2, while difficult translation and multi-skill tasks reach 18.4–21.7. Because tasks that never cross the threshold receive a tied last-place rank, part of the variance among hard tasks reflects censoring.

Table 2 quantifies the consistency of emergence order across models. Within the OLMo-2 family, Spearman rank correlations range from .72 to .93. Cross-family correlations are also high: Amber correlates with OLMo-2 models at 0.82–0.88, while correlations with older and smaller models (e.g., OLMo-2 vs. Pythia-410M) remain substantial, ranging from 0.64 to 0.84. All correlations are highly significant and remain so after correction for multiple comparisons. Importantly, this consistency holds only under the absolute threshold definition of emergence. When using relative thresholds, cross-model correlations drop substantially (Appendix E). We hypothesize that this discrepancy arises because relative thresholds depend on each model's maximum performance: a weak model may reach a relative threshold early despite lacking meaningful task competence. The strongest consistency result therefore concerns reaching a common absolute proficiency level, not a threshold-independent ordering. Mean cross-model Spearman correlation is .810 at the canonical absolute threshold, but only .300–.479 across relative-threshold variants. In the corresponding prerequisite analysis, strong inversions increase from 3 under the canonical definition to 45–94 under relative definitions. Our absolute thresholds are set above chance for all tasks, capturing when task behavior reaches a common criterion.

5 Representational Similarity and Prediction Results

Having established that skill acquisition during pretraining is both structured and consistent (H1, H2), we next ask whether this structure is reflected in the model's internal represen-

Table 2: Spearman rank correlation (ρ) of emergence orderings between model pairs (absolute threshold = 80%). All 45 correlations are significant ($p < 10^{-7}$).

	OLMo2			OLMo3	LLM360		Pythia			
	1B	7B	13B	7B	Amber	Crystal	410M	1.4B	2.8B	12B
O2-1B	—	.80	.72	.75	.88	.76	.84	.92	.78	.84
O2-7B		—	.93	.93	.88	.92	.71	.72	.77	.84
O2-13B			—	.93	.82	.89	.64	.66	.76	.83
O3-7B				—	.83	.92	.70	.72	.79	.85
Amber					—	.87	.77	.84	.87	.90
Crystal						—	.70	.75	.83	.87
P-410M							—	.88	.72	.79
P-1.4B								—	.77	.82
P-2.8B									—	.91
P-12B										—

.64 .70 .75 .80 .85 .90+

Table 3: Leave-one-out prediction of held-out composite task trajectories (26 tasks). *All tasks* includes simple and composite tasks. *Sim. only* includes only simple tasks. Restricting to elementals degrades MAE for every model (mean ΔMAE $= +.135$), indicating a composition bottleneck.

	All tasks		Sim. only	
Model	R^2	MAE	R^2	MAE
Pythia-410M	.681	.195	.717	.301
OLMo2-1B	.723	.070	.602	.289
Pythia-1.4B	.778	.086	.755	.193
Amber (7B)	.751	.082	.725	.205
Crystal (7B)	.676	.133	.568	.315
OLMo2-7B	.767	.068	.693	.208
OLMo3-7B	.692	.079	.491	.215
Pythia-12B	.812	.136	.789	.194
OLMo2-13B	.838	.099	.860	.242

tations (H3). Namely, if two tasks have similar function vectors, do they exhibit similar learning trajectories in pretraining? Rather than testing correlations in isolation, we consider a stronger version: can the learning trajectory of a held-out composite task be predicted solely from its representational similarity to other tasks, without further evaluation during training?

Table 3 reports leave-one-out prediction results for composite task trajectories using kernel ridge regression in function vector space. When the basis includes all other tasks (elemental and composite), prediction quality is strong: R^2 ranges from .67 (Crystal) to .838 (OLMo2-13B), with MAE between .068 and .195 on a 0-1 accuracy scale. These results provide strong evidence that representational geometry is closely linked to learning dynamics, supporting H3. As a case study of specific predicted trajectories, Figure 3 shows representative predicted trajectories compared to ground truth trajectories for OLMo2-7B from 0-1T tokens. For tasks such as `fr_eng_upper` ($R^2 = .99$, MAE $= .017$) and `plural_lower` ($R^2 = .89$, MAE $= .028$), the predicted curve closely tracks the actual trajectory, capturing both the onset of emergence and the subsequent rate of improvement. However, predictions are weaker for tasks such as `eng_fr_upper` ($R^2 = .51$, MAE $= .068$), where the held-out task's trajectory is less well approximated by its nearest neighbors in representation space. Full prediction results can be found in Appendix H.

6 Conclusion

In this paper, we unify several threads of discussion on the emergence of LM capabilities during pretraining in the Implicit Curriculum Hypothesis—that skills acquired during pretraining emerge in a stable order driven by concrete compositions. We test this hypothesis empirically across several model families and with models spanning 410M–13B parameters. Our empirical findings support both the behavioural and representational aspects of

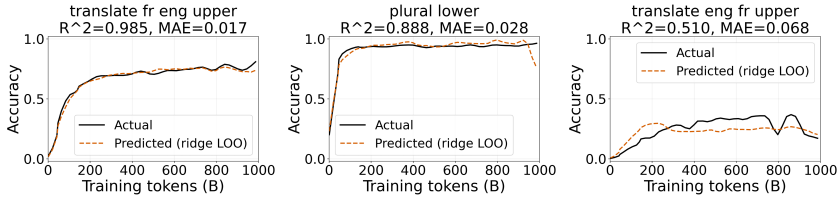


Figure 3: Example composite task predictions for OLMo2-7B between 0-1T tokens.

the hypothesis: emergence orders of tasks under absolute thresholds are quite consistent across models, even across families and in models trained on different data. Furthermore, similarity in the function vector space predicts similarity of learning trajectories, such that it is possible to predict the trajectories of held-out compositional tasks from function vector similarity without evaluating them. This indicates that the developmental structure visible in behavioral evaluations may also be legible in the model's internal representations.

The current study has limitations: we note that these conclusions are limited to a controlled, English-language task suite evaluated through 5-shot ICL. Additionally, we were limited in the range of models we could include, as there are a limited number of open models with dense checkpoints. They are sensitive to the onset threshold and do not establish causal dependencies or generalization to open-ended tasks.

Our results open several avenues for further investigation. One practical application is *pretraining monitoring*—if emergence orders are stable and predictable, our task suite can serve as a basis for monitoring whether models are developing capabilities ahead of or behind schedule. Furthermore, understanding this task structure could also inform data mixture decisions. Future work should test realistic and multilingual tasks, controlled changes to pretraining data order or composition, and circuit-level interventions. More broadly, we hope that the framework of studying pretraining as a structured developmental process will prove useful for understanding, predicting, and ultimately steering what language models learn.

Acknowledgments

EL was supported by the National Sciences and Engineering Research Council of Canada (NSERC), [funding reference number 578085], as well as the SoftBank-ARM Fellowship. KS is supported in part by DARPA (SciFY). ML is supported by a NSF Graduate Research Fellowship. IL is supported by a Technical AI Safety Research Grant from Coefficient Giving via Berkeley Existential Risk Initiative.

This work used the Delta system at the National Center for Supercomputing Applications [award OAC 2005572] through allocation [CIS250578] from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

Sanjeev Arora and Anirudh Goyal. A theory for emergence of complex skills in language models, 2023. URL https://arxiv.org/abs/2307.15936.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International conference on machine learning*, pp. 2397–2430. PMLR, 2023.

Ryan Burnell, Han Hao, Andrew R. A. Conway, and Jose Hernandez Orallo. Revealing the structure of language model capabilities, 2023. URL https://arxiv.org/abs/2306.10062.

Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in mlms. *arXiv preprint arXiv:2309.07311*, 2023a.

Mayee F. Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. Skill-it! a data-driven skills framework for understanding and training language models, 2023b. URL https://arxiv.org/abs/2307.14430.

Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. Parallel structures in pre-training data yield in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8582–8592, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.465.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Ruth B. Ekstrom, John W. French, Harry H. Harman, and Diran Dermen. Manual for kit of factor-referenced cognitive tests. Technical report, Educational Testing Service, Princeton, NJ, 1976.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with V-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Sheridan Feucht, Eric Todd, Byron Wallace, and David Bau. The dual-route model of induction. In *Second Conference on Language Modeling*, 2025. URL https://arxiv.org/abs/2504.03022.

Dan Friedman, Alexander Wettig, and Danqi Chen. Finding dataset shortcuts with grammar induction. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 4345–4363, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.293. URL https://aclanthology.org/2022.emnlp-main.293/.

Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, Rui Xin, Marianna Nezhurina, Igor Vasiljevic, Luca Soldaini, Jenia Jitsev, Alex Dimakis, Gabriel Ilharco, Pang Wei Koh, Shuran Song, Thomas Kollar, Yair Carmon, Achal Dave, Reinhard Heckel, Niklas Muennighoff, and Ludwig Schmidt. Language models scale reliably with over-training and on downstream tasks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=iZeQBqJamf.

Xuyang Ge, Wentao Shu, Jiaxing Wu, Yunhua Zhou, Zhengfu He, and Xipeng Qiu. Evolution of concepts in language model pre-training. *arXiv preprint arXiv:2509.17196*, 2025.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.624. URL https://aclanthology.org/2023.findings-emnlp.624/.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=iBBcRUlOAPR.

Berivan Isik, Natalia Ponomareva, Hussein Hazimeh, Dimitris Paparas, Sergei Vassilvitskii, and Sanmi Koyejo. Scaling laws for downstream task performance of large language models, 2026. URL https://arxiv.org/abs/2402.04177.

Sara Kangaslahti, Elan Rosenfeld, and Naomi Saphra. Hidden breakthroughs in language model training. *arXiv preprint arXiv:2506.15872*, 2025.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Apoorv Khandelwal and Ellie Pavlick. How do language models compose functions?, 2025. URL https://arxiv.org/abs/2510.01685.

Jin Hwa Lee, Andrew K. Lampinen, Aaditya K. Singh, and Andrew M. Saxe. Distinct computations emerge from compositional curricula in in-context learning, 2025. URL https://arxiv.org/abs/2506.13253.

Emmy Liu, Amanda Bertsch, Lintang Sutawika, Lindia Tjuatja, Patrick Fernandes, Lara Marinov, Michael Chen, Shreya Singhal, Carolin Lawrence, Aditi Raghunathan, Kiril Gashteovski, and Graham Neubig. Not-just-scaling laws: Towards a better understanding of the downstream impact of language model design decisions, 2026. URL https://arxiv.org/abs/2503.03862.

Leo Z. Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith. Probing across time: What does RoBERTa know and when? In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 820–842, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.71. URL https://aclanthology.org/2021.findings-emnlp.71/.

Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, Richard Fan, Yi Gu, Victor Miller, Yonghao Zhuang, Guowei He, Haonan Li, Fajri Koto, Liping Tang, Nikhil Ranjan, Zhiqiang Shen, Xuguang Ren, Roberto Iriondo, Cun Mu, Zhiting Hu, Mark Schulze, Preslav Nakov, Tim Baldwin, and Eric P. Xing. Llm360: Towards fully transparent open-source llms, 2023.

Nicholas Lourie, Michael Y. Hu, and Kyunghyun Cho. Scaling laws are unreliable for downstream tasks: A reality check, 2025. URL https://arxiv.org/abs/2507.00885.

Aviya Maimon, Amir DN Cohen, Gal Vishne, Shauli Ravfogel, and Reut Tsarfaty. Iq test for llms: An evaluation framework for uncovering core skills in llms, 2025. URL https://arxiv.org/abs/2507.20208.

Clara Meister and Ryan Cotterell. Language model evaluation beyond perplexity. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5328–5339, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.414. URL https://aclanthology.org/2021.acl-long.414/.

James A. Michaelov, Roger P. Levy, and Benjamin K. Bergen. Language model behavioral phases are consistent across architecture, training data, and scale, 2025. URL https://arxiv.org/abs/2510.24963.

Eric J Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=3tbTw2ga8K.

Shubhra Mishra, Gabriel Poesia, and Noah Goodman. From next-token to mathematics: The learning dynamics of mathematical reasoning in language models. In *Second Conference on Language Modeling*, 2025.

Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=j5BuTrEj35.

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, 2019. URL https://arxiv.org/abs/1905.11604.

Olmo. Olmo 3: Charting a path through the model flow to lead open-source ai. *AI2 Blog Nov 30 2025*, 2025. URL https://allenai.org/blog/olmo3.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads, 2022. URL https://arxiv.org/abs/2209.11895.

Felipe Maia Polo, Seamus Somerstep, Leshem Choshen, Yuekai Sun, and Mikhail Yurochkin. Sloth: scaling laws for LLM skills to predict multi-benchmark performance across families. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=9GN5Jsa3lv.

Aaditya Vikram Prasad, Connor Watts, Jack Merullo, Dhruvil Gala, Owen Lewis, Thomas McGrath, and Ekdeep Singh Lubana. Features as rewards: Scalable supervision for open-ended tasks via interpretability, 2026. URL https://arxiv.org/abs/2602.10067.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, 2014. URL https://arxiv.org/abs/1312.6120.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage?, 2023. URL https://arxiv.org/abs/2304.15004.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks, 2020. URL https://arxiv.org/abs/2006.07710.

Charlie Snell, Eric Wallace, Dan Klein, and Sergey Levine. Predicting emergent capabilities by finetuning, 2024. URL https://arxiv.org/abs/2411.16035.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

Kaiser Sun and Mark Dredze. Amuro & char: Analyzing the relationship between pre-training and fine-tuning of large language models. In Vaibhav Adlakha, Alexandra Chronopoulou, Xiang Lorraine Li, Bodhisattwa Prasad Majumder, Freda Shi, and Giorgos Vernikos (eds.), *Proceedings of the 10th Workshop on Representation Learning for NLP (RepL4NLP-2025)*, pp. 131–151, Albuquerque, NM, May 2025. Association for Computational Linguistics. ISBN 979-8-89176-245-9. doi: 10.18653/v1/2025.repl4nlp-1.11. URL https://aclanthology.org/2025.repl4nlp-1.11/.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG.

Eric Todd, Jannik Brinkmann, Rohit Gandikota, and David Bau. In-context algebra, 2026. URL https://arxiv.org/abs/2512.16902.

Oskar van der Wal, Pietro Lesci, Max Muller-Eberstein, Naomi Saphra, Hailey Schoelkopf, Willem Zuidema, and Stella Biderman. Polypythias: Stability and outliers across fifty language model pre-training runs. *arXiv preprint arXiv:2503.09543*, 2025.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.00593.

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, 2022. URL https://arxiv.org/abs/2206.07682.

Kayo Yin and Jacob Steinhardt. Which attention heads matter for in-context learning? In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=C7XmEByCFv.

Dingli Yu, Simran Kaur, Arushi Gupta, Jonah Brown-Cohen, Anirudh Goyal, and Sanjeev Arora. SKILL-MIX: a flexible and expandable family of evaluations for AI models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Jf5gplvglq.

Yedi Zhang, Andrew Saxe, and Peter E. Latham. Saddle-to-saddle dynamics explains a simplicity bias across neural network architectures, 2026. URL https://arxiv.org/abs/2512.20607.

A LLM Usage Disclosure

Claude Opus 4.6 and Sonnet 4.6 were used to format tables and conduct minor writing edits. All outputs were reviewed, and verified by the authors.

B Extended Related Work

Skill Emergence and Scaling Laws. Theoretical work has sought to explain how capabilities emerge with scale. Arora & Goyal (2023) propose that scaling laws arise from slingshot generalization, where competence at k-tuples of skills emerges at the same rate as elementary skills themselves. Similarly, Michaud et al. (2023) introduce the quanta hypothesis, modeling skills as discrete units whose power-law frequency distribution explains smooth scaling curves. Both theories predict that complex behaviors emerge from simpler building blocks, but leave open the question of what these building blocks are and how they compose in practice. Our work provides empirical grounding for these theories by tracking probe tasks designed to be compositionally combined, while remaining agnostic about whether those probes are discrete internal building blocks.

Skill Evaluation and Structure. Several approaches characterize LLM capabilities through evaluation-time analysis. Diagnostic studies also use training dynamics to characterize examples and dataset difficulty (Swayamdipta et al., 2020; Ethayarajh et al., 2022); we instead focus on cross-checkpoint task onset and controlled compositional relations. Burnell et al. (2023) apply factor analysis across 29 models and 27 tasks, finding three latent factors, reasoning, comprehension, and language modeling, that explain performance variation; Maimon et al. (2025) scale this psychometric approach to 60 models and 44 tasks, identifying eight core skills. Beyond identifying skills, Yu et al. (2024) directly test compositional ability by evaluating whether models can combine k-tuples of language skills in novel ways. Polo et al. (2025) unify these perspectives through skill-based scaling laws where performance is driven by low-dimensional latent skills. These works analyze fully trained models; we complement them by studying *how skills develop during pretraining* and linking emergence order to representational structure.

Training Dynamics and Phase Transitions. Understanding what models learn during training has gained increasing attention. Chen et al. (2023a) identify sudden drops in loss corresponding to syntax acquisition and other phase transitions; Kangaslahti et al. (2025) show that such breakthroughs occur frequently but are obscured by aggregate loss metrics. van der Wal et al. (2025) release 50 additional training runs of Pythia models, finding consistent learning phases across seeds and sizes. Other work examines specific capabilities: Sun & Dredze (2025) investigate how downstream performance develops across pretraining checkpoints, Ge et al. (2025) track feature evolution using sparse dictionary learning, and Mishra et al. (2025) show that mathematical skills emerge in an order correlated with human curriculum despite random data ordering. Our work contributes to this literature by demonstrating that emergence orderings are stable across model families and can be predicted from representational geometry.

Representations for Task Understanding. Mechanistic interpretability has revealed compact representations of tasks within model activations. Both Todd et al. (2024) and Hendel et al. (2023) discover that in-context learning compresses task demonstrations into single directions, termed function vectors and task vectors respectively, which can trigger task execution even in zero-shot settings. Subsequent work explores the scope of this phenomenon: Todd et al. (2026) extend it to symbolic reasoning with variable-based tokens, while Khandelwal & Pavlick (2025) investigate compositional tasks and find both compositional and direct processing mechanisms. We build on this line of work by using residual-stream representations to predict learning trajectories of compositional tasks, connecting representational geometry to training dynamics. We interpret this as predictive association, not a causal mechanism. This complements recent work by Prasad et al. (2026) that uses interpretable features for training monitoring in RL.

C Full list of Elemental and Composite Tasks

We provide a full list of tasks, categorized into reasoning types, in Table 4 and Table 5. TextFRCT tasks are taken from the psychometrics literature (Ekstrom et al., 1976), while other tasks have been studied or were inspired by works in interpretability literature (Wang et al., 2023; Todd et al., 2024; Chen et al., 2024; Feucht et al., 2025).

Table 4: All elemental tasks in the evaluation suite with representative examples.

Task	N	Input	Output
String Operations			
`copying`	20	gTpigTHK	gTpigTHK
`token_reversal`	20	cat	tac
`string_analogy`	10	abc → abd, ijk → ?	ijl
`simple_icl:uppercase`	26	b	B
`simple_icl:lowercase`	26	B	b
`simple_icl:first_letter`	190	the cat went up the tree	t
`simple_icl:last_letter`	190	the cat went up the tree	e
Morphology			
`simple_icl:present_to_gerund`	179	run	running
`simple_icl:singular_to_plural`	165	child	children
Translation			
`simple_icl:translate_eng_fr`	173	hello	bonjour
`simple_icl:translate_fr_eng`	175	bonjour	hello
`simple_icl:translate_eng_sp`	178	hello	hola
`simple_icl:translate_sp_eng`	178	hola	hello
World Knowledge			
`simple_icl:country_to_capital`	184	Afghanistan	Kabul
`simple_icl:country_to_currency`	198	United States	Dollar
Arithmetic			
`basic_arithmetic`	10	What is 5 + 3?	8
`math`	20	4 * 1	4
`multistep_arithmetic:two_step`	20	3 + 4, then multiply by 2	14
`multistep_arithmetic:three_step`	20	Start with 10, subtract 3, then multiply by 4	28
`textfrct:RG1`	30	In general, brass is made of two parts copper to one part zinc. How many pounds of zinc are needed to produce 45 pounds of brass? *(MCQ)*	B
`textfrct:RG2`	30	Recipe A uses 1.5 cups of sugar; Recipe B uses 2. Making 8 cakes, how many fewer cups does Recipe A require? *(MCQ)*	E
`textfrct:RG3`	30	There are 4 quarts in a gallon and 4 cups in a quart. How many cups are in a gallon? *(MCQ)*	C
Logic			
`logical_ops:negation`	12	Statement: All robots can move. Candidate: Some robots cannot move. Is this a correct logical negation?	True
`logical_ops:conjunction`	12	Fact A is True. Fact B is True. Claim: A AND B. Is the claim true?	True
`logical_ops:conditional`	12	Rule: If it rains, the ground gets wet. Fact: It rains. Does the conclusion follow?	True
`textfrct:RL1`	30	All birds have purple tails. All cats are birds. Therefore all cats have purple tails. *(MCQ: correct/incorrect)*	G
`textfrct:RL3`	20	More fatal accidents occur on highways after dark than during daylight hours. *(MCQ: which conclusion follows?)*	3
`textfrct:RL4`	24	*ICL ex.: black sheep = dag kip; white dog = tin bud; black cow = dag stam* Query: white sheep = ? *(MCQ)*	2
Reading Comprehension			
`fact_extraction:extract_entity`	20	Passage: "Alice gave five apples to Bob at the park." Who received the apples?	Bob
`fact_extraction:extract_number`	20	Passage: "John gave 5 apples to Mary on Tuesday." How many apples did John give?	5
`fact_extraction:extract_location`	20	Passage: "The cat sat on the red mat in the kitchen." Where is the mat?	the kitchen
`coreference:pronoun_simple`	20	"Alice told Bob that she would be late." Who does "she" refer to?	Alice

Continued on next page

Table 4 continued from previous page

Task	N	Input	Output
coreference:pronoun_hard	20	"The trophy didn't fit in the suitcase because it was too big." What was too big?	the trophy
ignoring_context	5	Some text here. X = 5. More text. Question: What is X?	5
ioi_task	1000	*Instr.: Identify who should be referenced.* Then, Henry and Phil had a lot of fun at the harbor. Henry gave a basket to	[Phil, Henry]
part_of_speech	15	The cat is in the house. The part of speech for "cat" is _	noun
Verbal Closure (FRCT)			
textfrct:CV1	50	erte	tree, rete
textfrct:CV2	40	EZIRTMODSLOWTSEXQILNECKBWOCJAKX	SLOW, NECK
textfrct:CV3	36	_tam_	stamp
Induction (FRCT)			
textfrct:I1	30	*Instr.: One of the five letter sets does NOT follow the same pattern as the others. Find it.* 1. QPPQ 2. HGHH 3. TTTU 4. DDDE 5. MLMM	1
textfrct:I2	28	*Instr.: Each row marks one position with 'x'. Identify the pattern and find the correct position in row 5.* ------- x------- ---- -- ---- -x--- -- --- ------ --------------- --x----- -------- ---x---------- ----1 2---3-- 4---5-----	3
Associative Memory (FRCT)			
textfrct:MA2	30	*Instr.: Memorize 30 word–number pairs, then answer retrieval queries.* Query: What number corresponds to 'coat'?	49
textfrct:MA3	30	*Instr.: Memorize 30 first–last name pairs, then answer retrieval queries.* Query: Last name: Nichols	Edward
Verbal Comprehension (FRCT)			
textfrct:V1	36	*Instr.: Choose the best definition (MCQ).* 'airtight': (1) firm (2) light (3) hermetically sealed (4) plane sick	3
textfrct:V2	36	*Instr.: Choose the best definition (MCQ).* 'handicraft': (1) cunning (2) fast boat (3) utility (4) manual skill (5) guild	4
textfrct:V3	48	*Instr.: Choose the best definition (MCQ).* 'cottontail': (1) squirrel (2) poplar (3) boa (4) marshy plant (5) rabbit	5
textfrct:V4	36	*Instr.: Choose the best definition (MCQ).* 'mumble': (1) speak indistinctly (2) complain (3) handle awkwardly (4) fall (5) tear apart	1
textfrct:V5	36	*Instr.: Choose the best definition (MCQ).* 'rancor': (1) forbearance (2) ridicule (3) malice (4) bravery	3

Table 5: All compositional tasks in the evaluation suite with representative examples.

Task	N	Input	Output
Morphology × String Operation			
compositional:gerund_lower	178	RUN	running
compositional:gerund_upper	178	run	RUNNING
compositional:gerund_reverse	178	run	gninnur
compositional:gerund_upper_reverse	178	run	GNINNUR
compositional:plural_lower	165	CHILD	children
compositional:plural_upper	165	child	CHILDREN
compositional:plural_reverse	165	child	nerdlihc
compositional:plural_upper_reverse	165	child	NERDLIHC
Translation × String Operation			
compositional:translate_eng_fr_first	173	hello	b
compositional:translate_eng_fr_last	173	hello	r
compositional:translate_eng_fr_lower	173	HELLO	bonjour

Continued on next page

Table 5 continued from previous page

Task	N	Input	Output
compositional:translate_eng_fr_reverse	173	hello	ruojnob
compositional:translate_eng_fr_upper	173	hello	BONJOUR
compositional:translate_eng_fr_upper_reverse	173	hello	RUOJNOB
compositional:translate_eng_sp_first	178	hello	h
compositional:translate_eng_sp_last	178	hello	a
compositional:translate_eng_sp_lower	178	HELLO	hola
compositional:translate_eng_sp_reverse	178	hello	aloh
compositional:translate_eng_sp_upper	178	hello	HOLA
compositional:translate_eng_sp_upper_reverse	178	hello	ALOH
compositional:translate_fr_eng_first	171	bonjour	h
compositional:translate_fr_eng_last	171	bonjour	o
compositional:translate_fr_eng_lower	171	BONJOUR	hello
compositional:translate_fr_eng_reverse	171	bonjour	olleh
compositional:translate_fr_eng_upper	171	bonjour	HELLO
compositional:translate_sp_eng_first	178	hola	h
compositional:translate_sp_eng_last	178	hola	o
compositional:translate_sp_eng_lower	178	HOLA	hello
compositional:translate_sp_eng_reverse	178	hola	olleh
compositional:translate_sp_eng_upper	178	hola	HELLO
Case/Reversal Chains			
compositional:lower_first	971	AFGHANISTAN	a
compositional:lower_last	971	AFGHANISTAN	n
compositional:lower_reverse	971	AFGHANISTAN	natsinahgfa
compositional:upper_first	971	afghanistan	A
compositional:upper_last	971	afghanistan	N
compositional:upper_reverse	971	afghanistan	NATSINAHGFA
compositional:reverse_first	971	Afghanistan	n
compositional:reverse_last	971	Afghanistan	A

D Full learning trajectories by category

Figures 4 – 12 show full learning trajectories of tasks for each model.

E Emergence Order Agreement Under Alternate Definitions

Table 6: Ten-model summary of emergence-ordering consistency under different definitions. Absolute thresholds yield substantially higher cross-model correlations than relative thresholds.

Definition	n pairs	Mean ρ	Min ρ	Max ρ
Absolute threshold ($\theta = 0.5$)	45	0.853	0.597	0.955
Absolute threshold ($\theta = 0.8$)	45	0.810	0.638	0.946
Absolute threshold ($\theta = 0.8$, stable for 3 consecutive checkpoints)	45	0.789	0.599	0.961
Relative threshold ($\alpha = 0.5$, fraction of max performance)	45	0.479	-0.012	0.866
Relative threshold ($\alpha = 0.8$, fraction of max performance)	45	0.471	0.077	0.773
Relative threshold ($\alpha = 0.5$, fraction of final performance)	45	0.300	-0.178	0.765

Under the canonical absolute definition, 58/82 currently evaluated composite–model cases with observed prerequisites respect the prerequisite ordering, with 21 weak and 3 strong inversions. The original 54/76 count in the main analysis excludes two subsequently added gerund compositions; the satisfaction rates are nearly identical (70.7% versus 71.1%). Under the relative definitions above, strong inversions range from 45 to 94 among 387 comparable cases because many more tasks receive an onset.

E.1 Task-rank stability

Across all ten models, the most stable ranks belong primarily to early transformations, whereas the largest variation occurs among difficult translation, knowledge, and multi-skill tasks.

Table 7: Examples from the ten-model rank-variance analysis at the canonical absolute threshold.

Most stable			Most variable		
Task	**Mean rank**	**SD**	**Task**	**Mean rank**	**SD**
`singular_to_plural`	6.4	2.8	`translate_sp_eng`	33.0	21.7
`lowercase`	6.5	2.8	`extract_verify`	45.7	20.9
`upper_first`	9.7	3.1	`extract_entity`	14.7	18.8
`lower_first`	9.7	3.2	`plural_upper`	38.7	18.8
`reverse_last`	10.3	3.3	`country_to_capital`	35.1	18.4

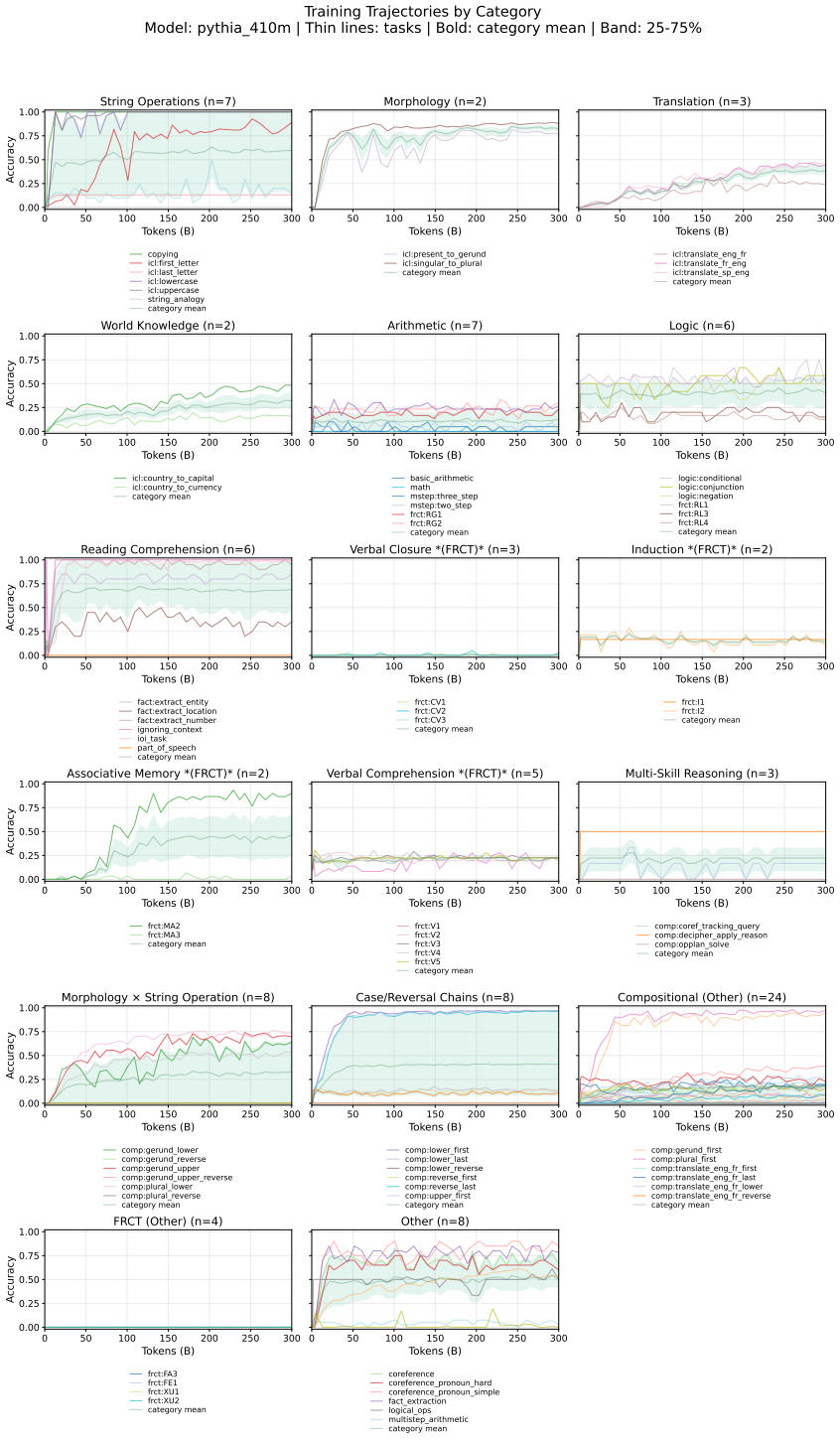


Figure 4: Complete trajectories for Pythia-410M over 300B tokens.

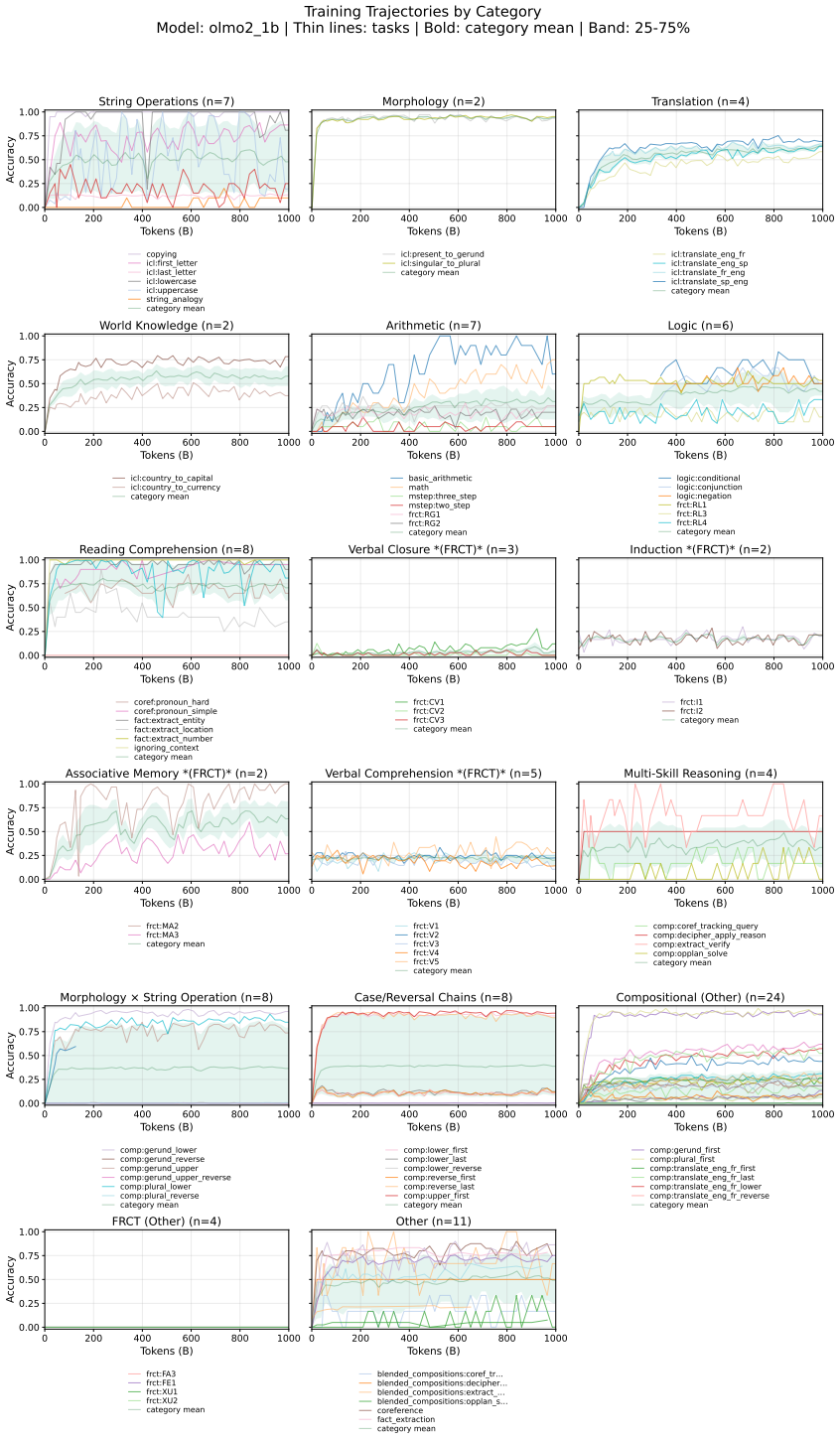


Figure 5: Complete trajectories for OLMo2-1B over 1T tokens.

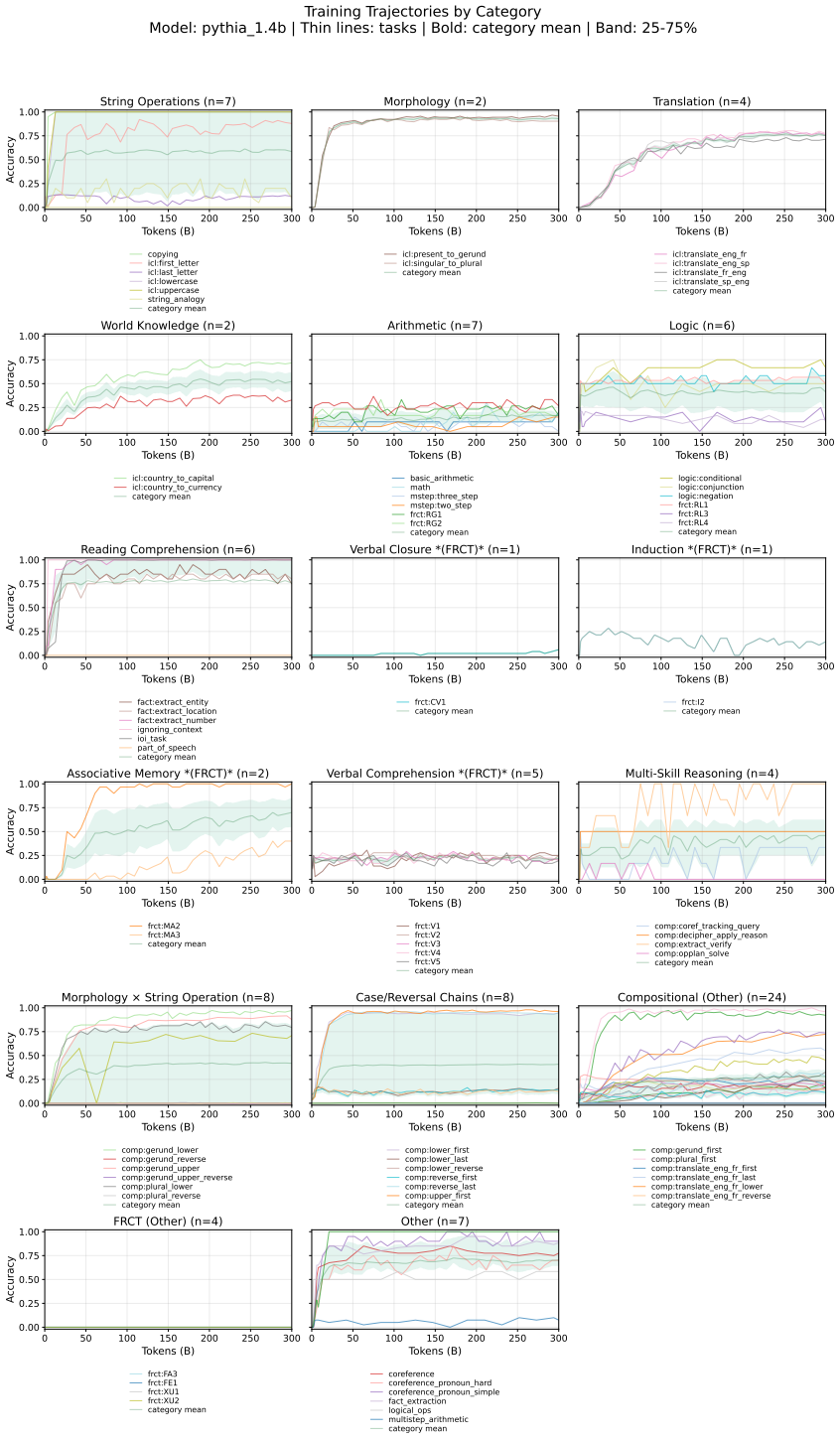


Figure 6: Complete trajectories for Pythia-1.4B over 300B tokens.

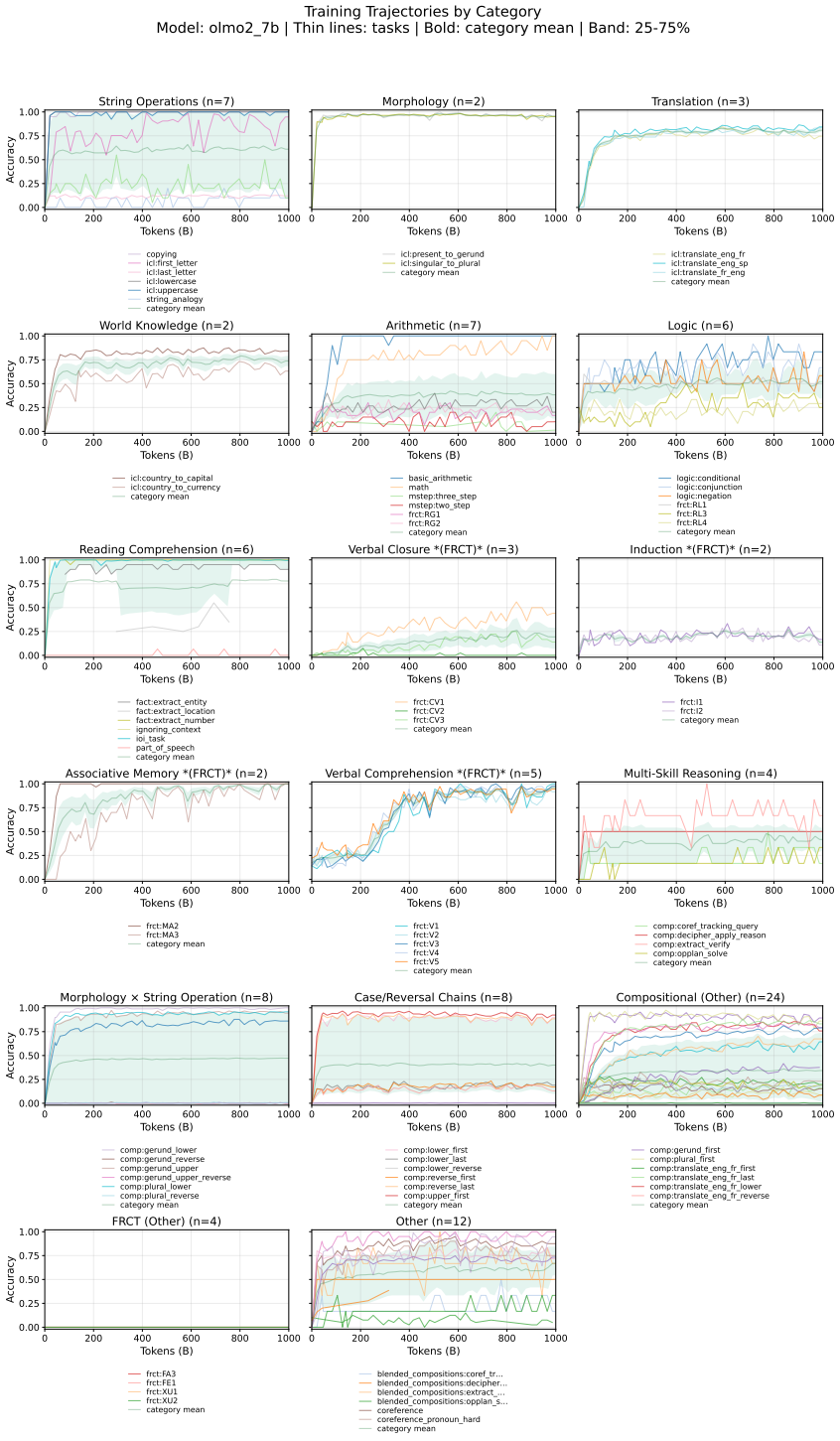


Figure 7: Complete trajectories for OLMo-2 7B over 1T tokens.

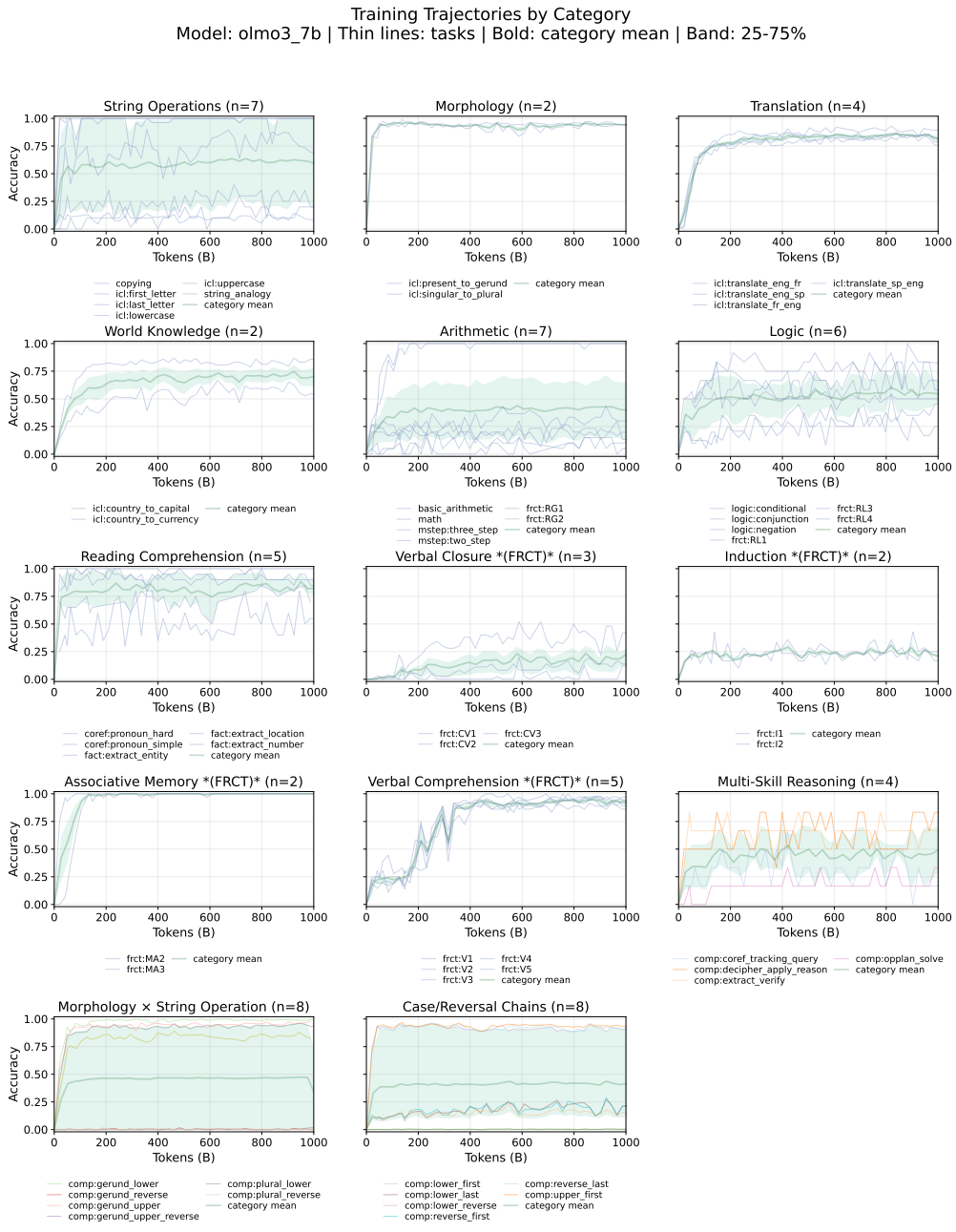


Figure 8: Complete trajectories for OLMo-3 7B over 1T tokens.

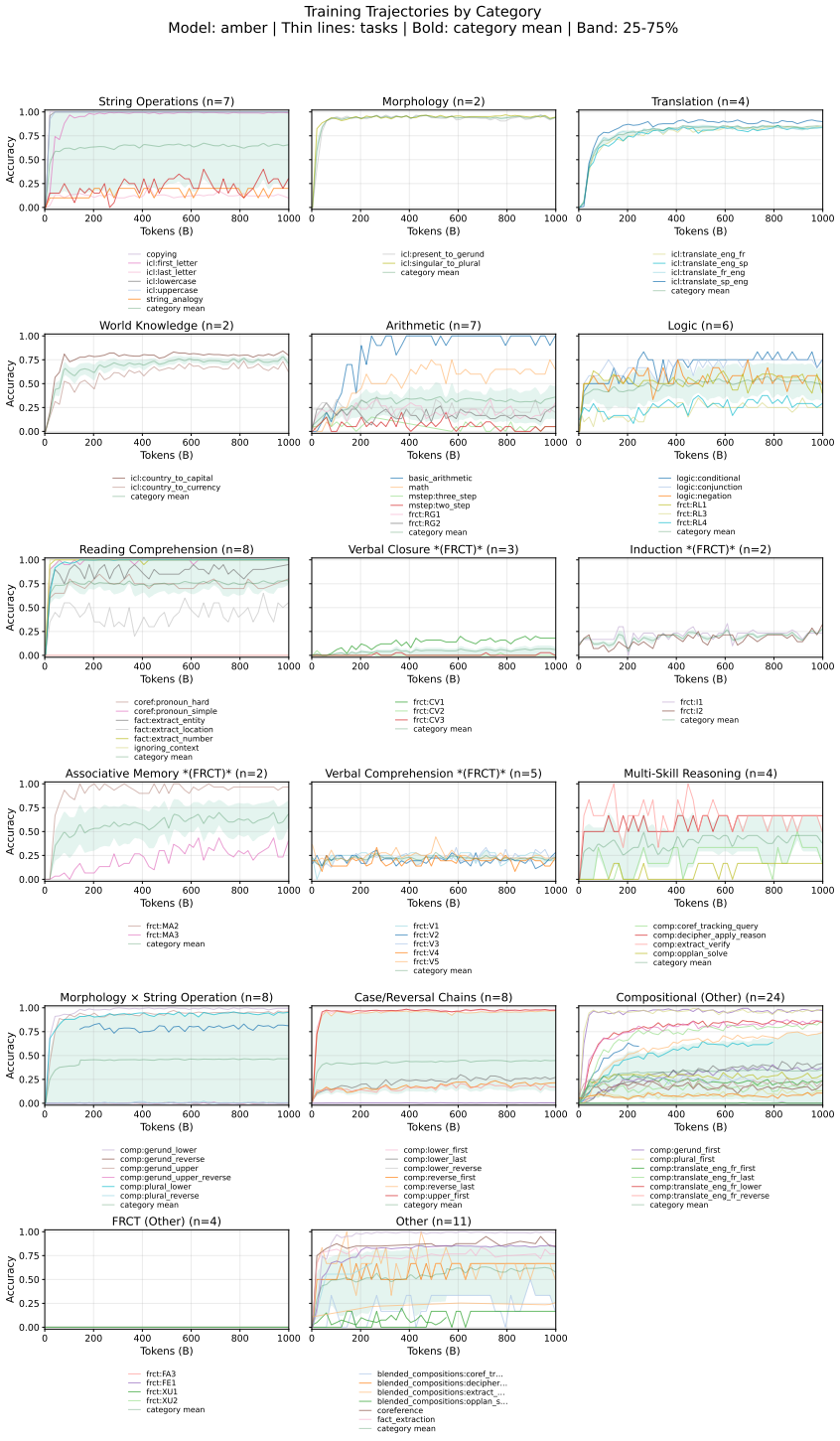


Figure 9: Complete trajectories for Amber (7B) over 1T tokens.

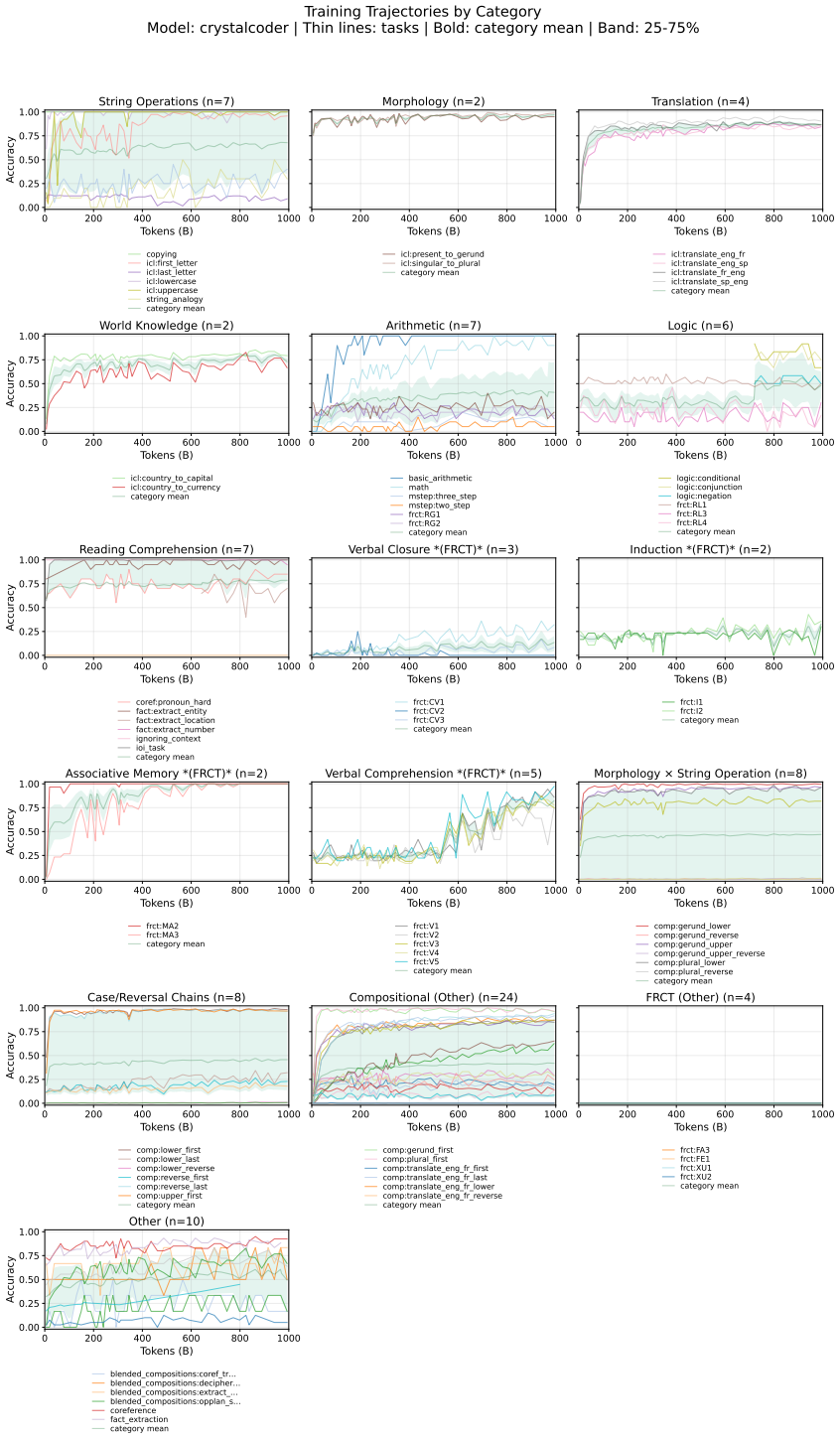


Figure 10: Complete trajectories for CrystalCoder (7B) over 1T tokens.



Figure 11: Complete trajectories for Pythia-12B over 300B tokens. Note that this model exhibits some instabilities compared to others.

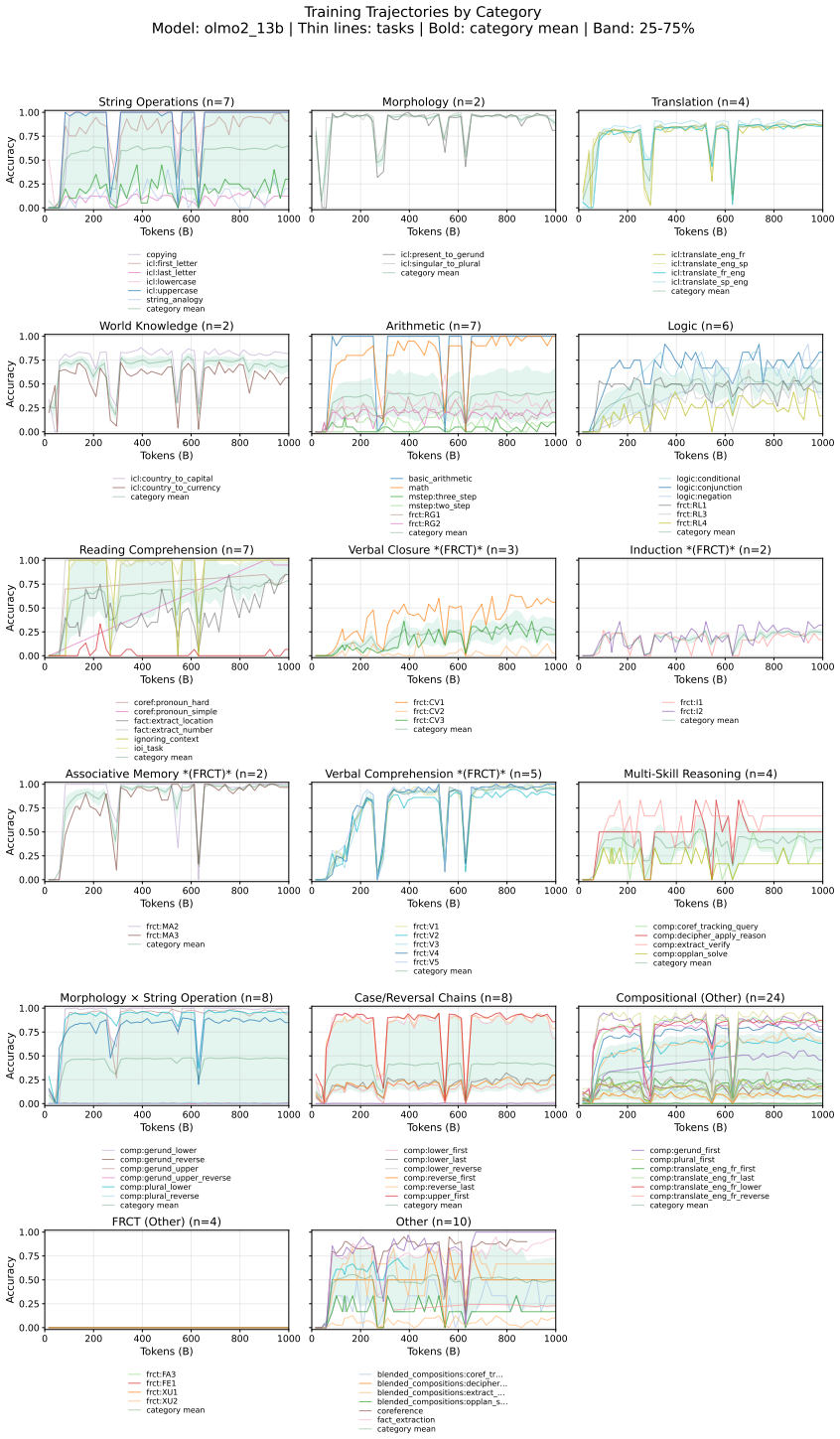


Figure 12: Complete trajectories for OLMo2-13b over 1T tokens. Note that this model exhibits some instabilities compared to others.

F FV baseline and permutation analysis

We evaluate every held-out compositional task for which both a trajectory and function vector are available, producing 499 task–model evaluations across ten models. Means are computed over these evaluation units; the permutation null uses 100 shuffled task-similarity assignments per unit.

Table 8: Ten-model trajectory-prediction robustness results. Lower MAE and higher Pearson R^2 are better.

Predictor	**Mean Pearson R^2**	**Mean MAE**
FV kernel ridge	**0.525**	**0.074**
Global-mean trajectory	0.494	0.314
Task-category mean	0.499	0.271
Permutation null	0.373	0.349

FV prediction has lower MAE than the per-case permutation mean in a paired one-sided Wilcoxon test ($W = 1151$, $p = 8.85 \times 10^{-81}$). In addition, 227/499 (45.5%) observed FV R^2 values rank at or above the 95th percentile of their corresponding permutation distributions.

G Function vector hyperparameters

Table 9 shows the hyperparameters selected for each model's task representation. Hyperparameters (representation type – between a fixed set of heads and a full residual stream, layers, and number of heads) were chosen via a three-criterion search over candidate configurations. A fixed calibration set of elemental and composite tasks was used, and three criteria were considered: (1) *within-task consistency*, measured as split-half cosine similarity between FVs extracted from random partitions of correct examples; (2) *inter-task discriminability*, the ratio of within-task to between-task cosine similarity; and (3) *compositional structure*, measured as the cosine similarity between each compositional task's FV and its least-squares reconstruction from component FVs. Final selection used a rank-sum policy over these three criteria, with ties broken by raw metric values. Note that in all cases, only correct examples were used when constructing the final function vectors.

H All Held-out trajectory predictions

Figures 13 – 21 show the results of the leave-one-out prediction setup for each compositional task.

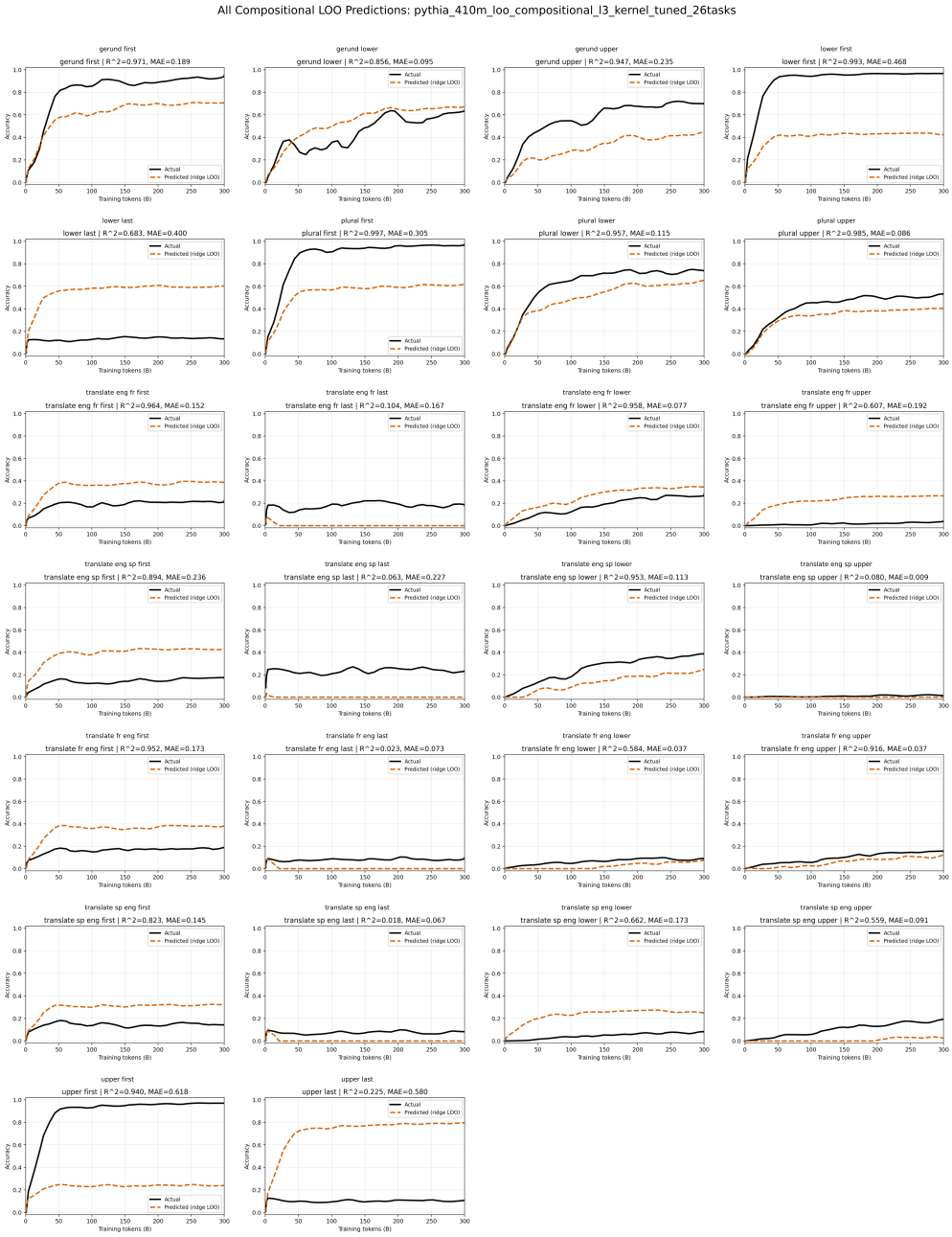


Figure 13: Predictions of held-out compositional tasks for Pythia-410M.

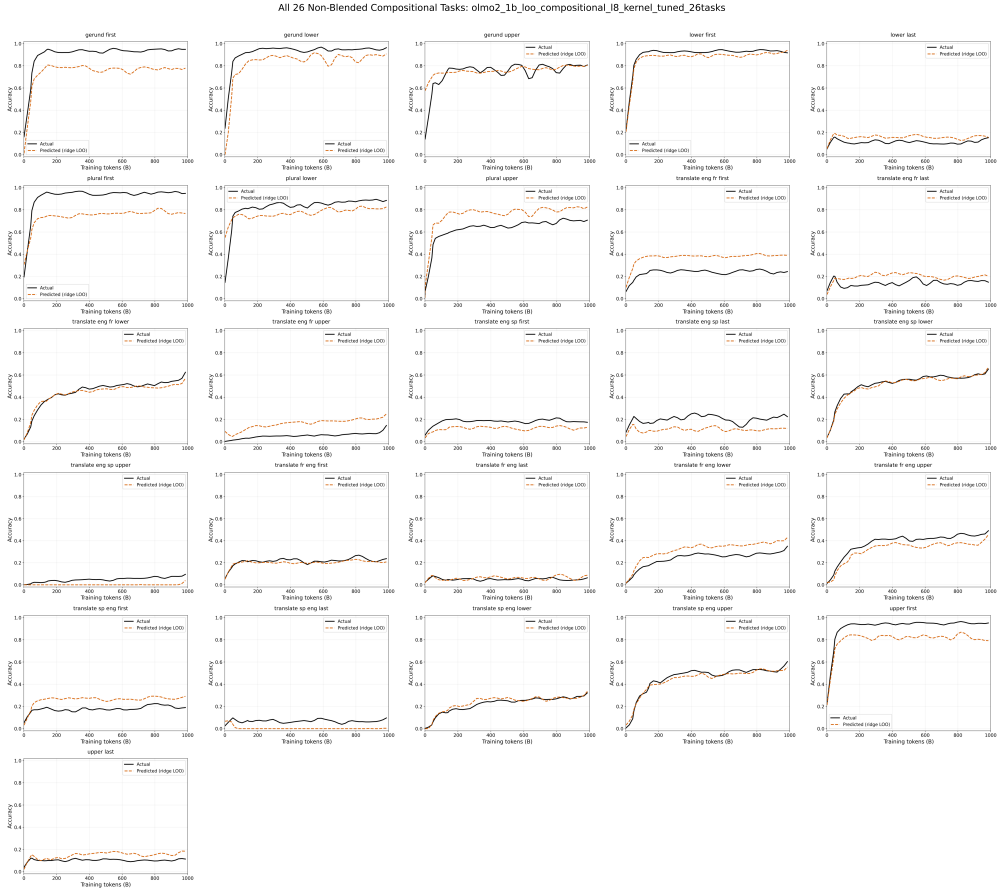


Figure 14: Predictions of held-out compositional tasks for OLMo2-1B.

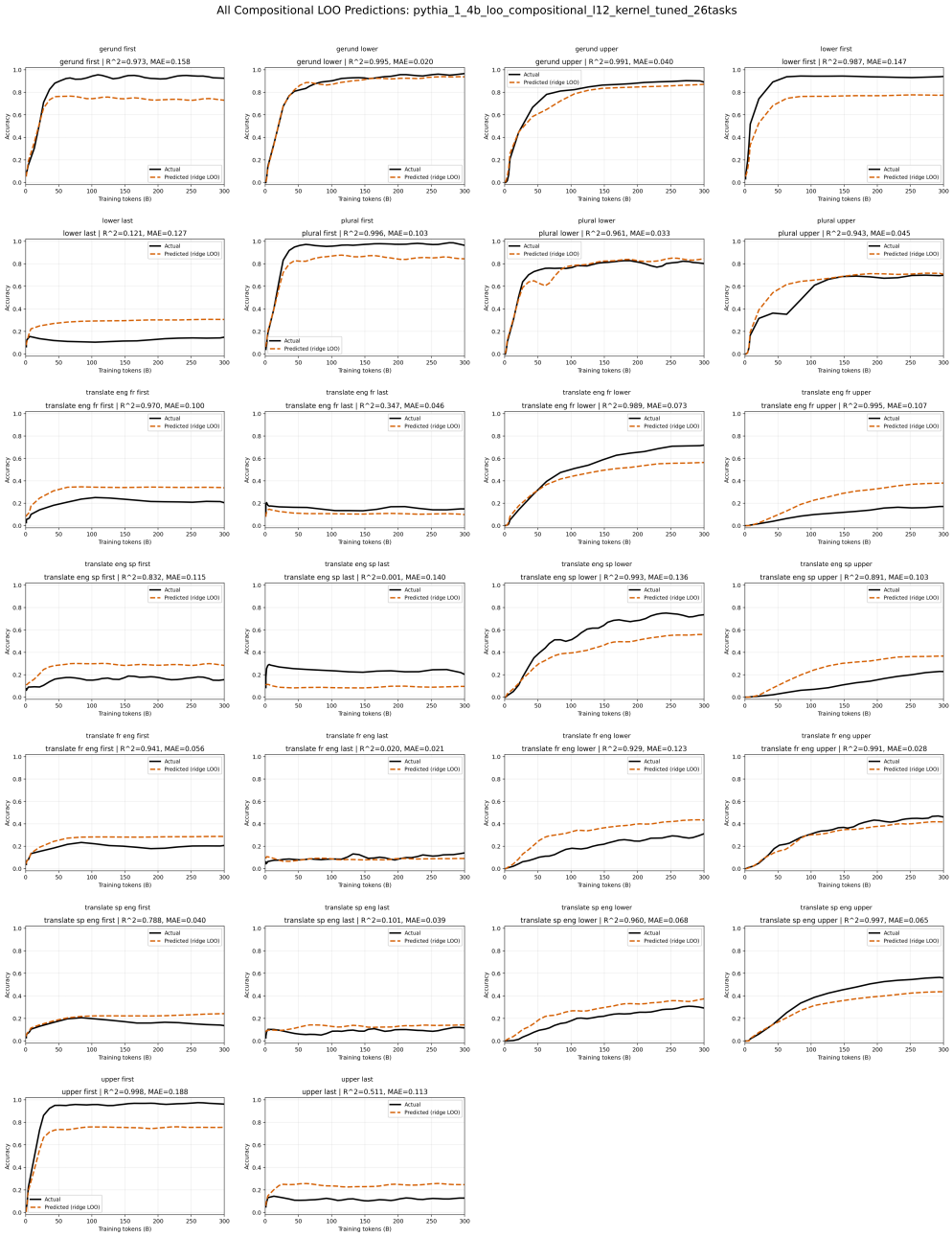


Figure 15: Predictions of held-out compositional tasks for Pythia-1.4B.

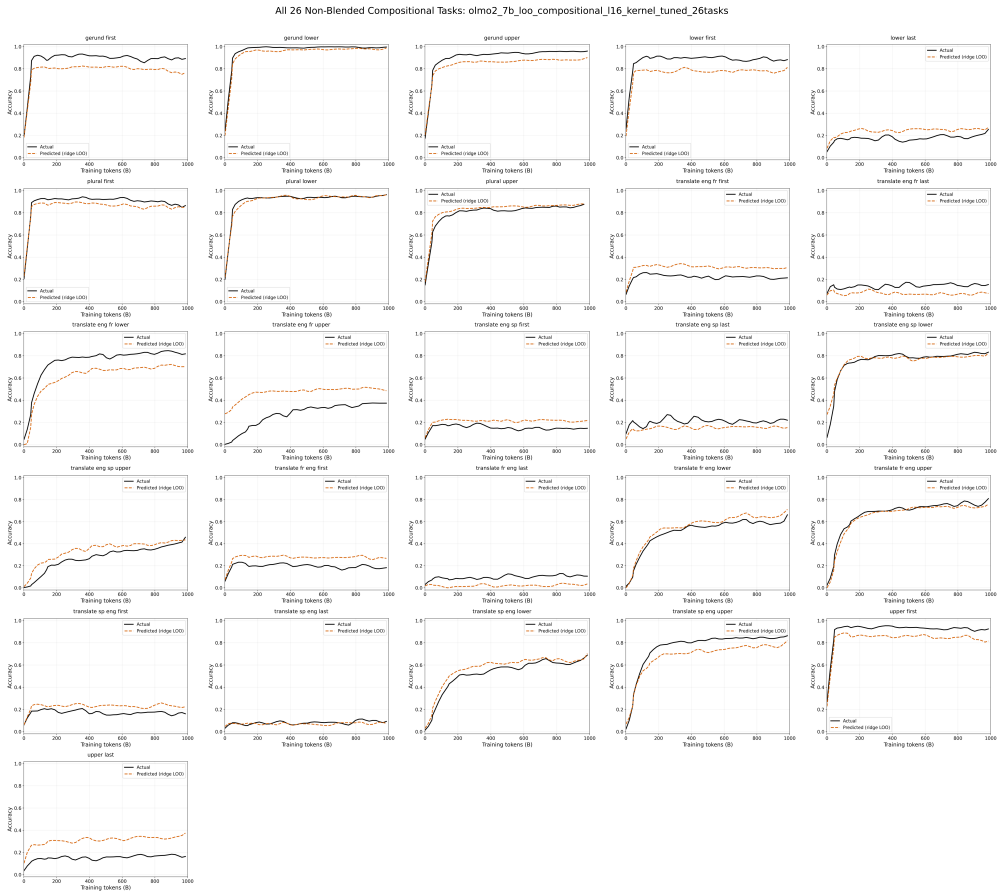


Figure 16: Predictions of held-out compositional tasks for OLMo2-7B.

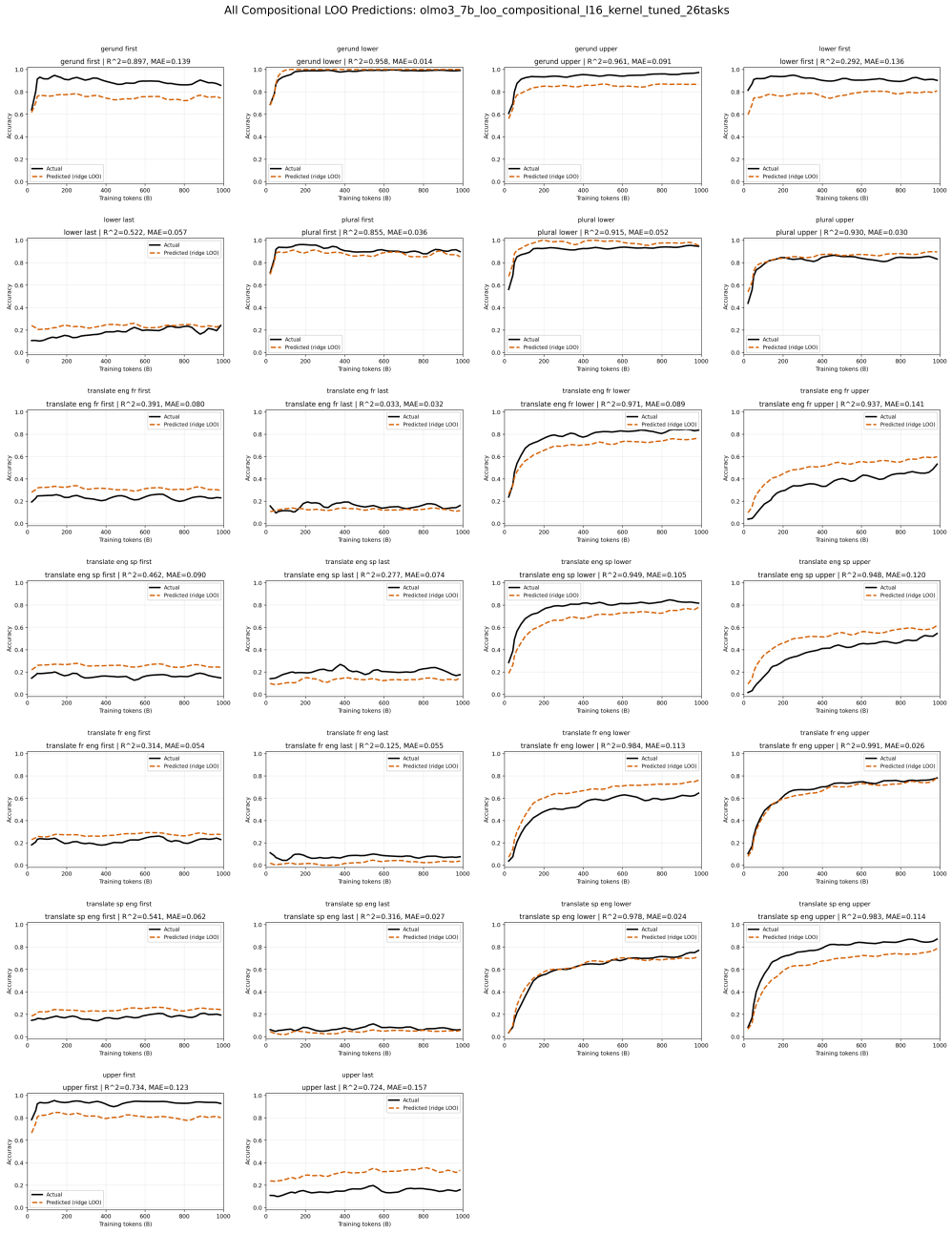


Figure 17: Predictions of held-out compositional tasks for OLMo3-7B.

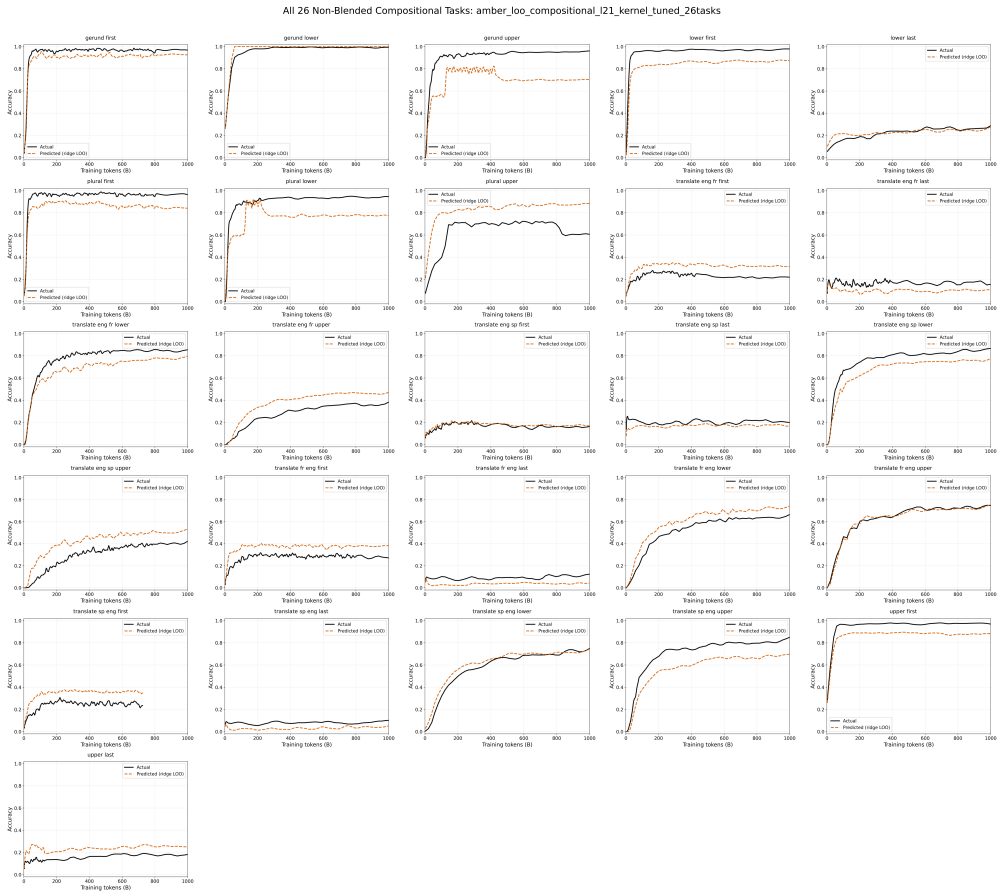


Figure 18: Predictions of held-out compositional tasks for Amber.

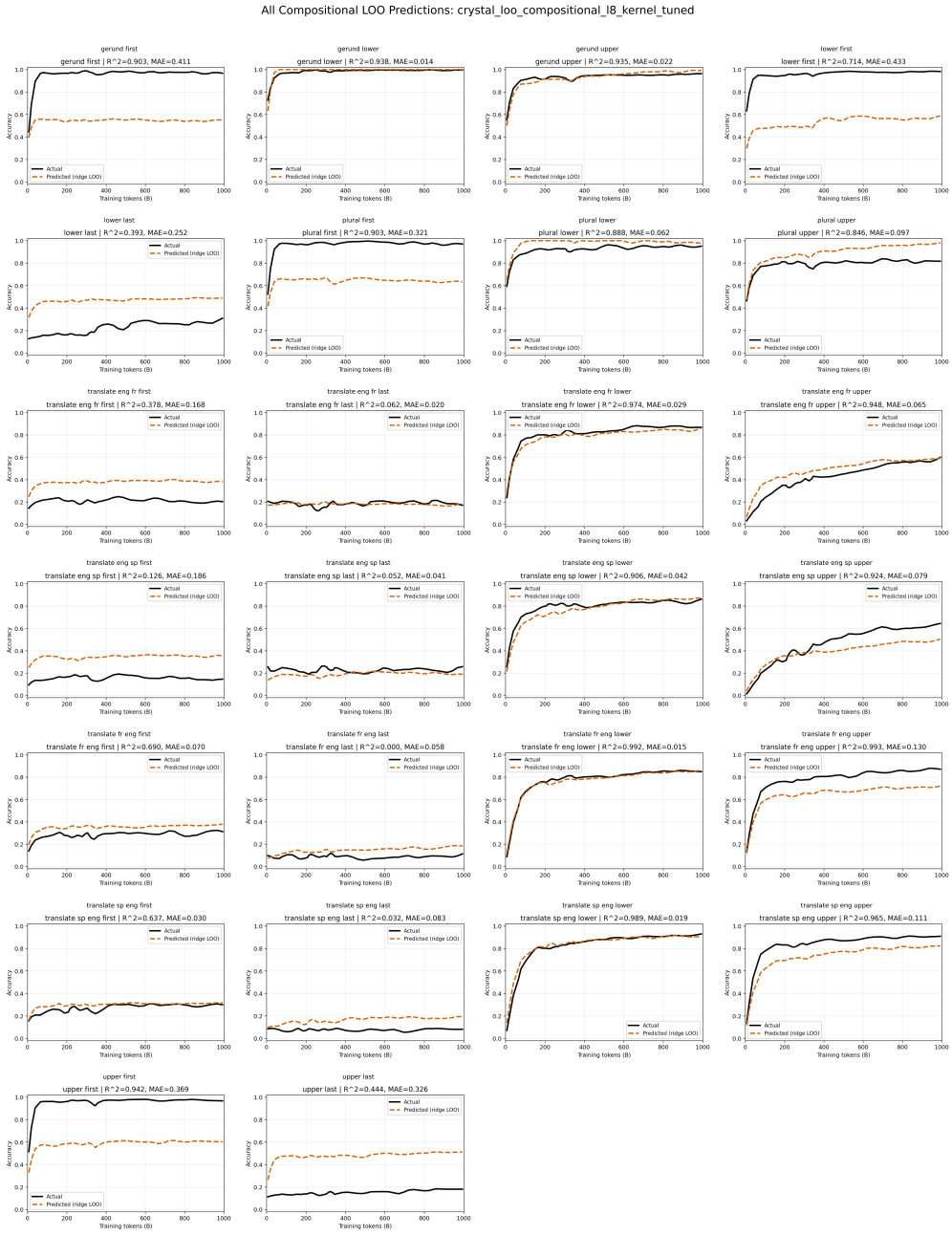


Figure 19: Predictions of held-out compositional tasks for CrystalCoder.

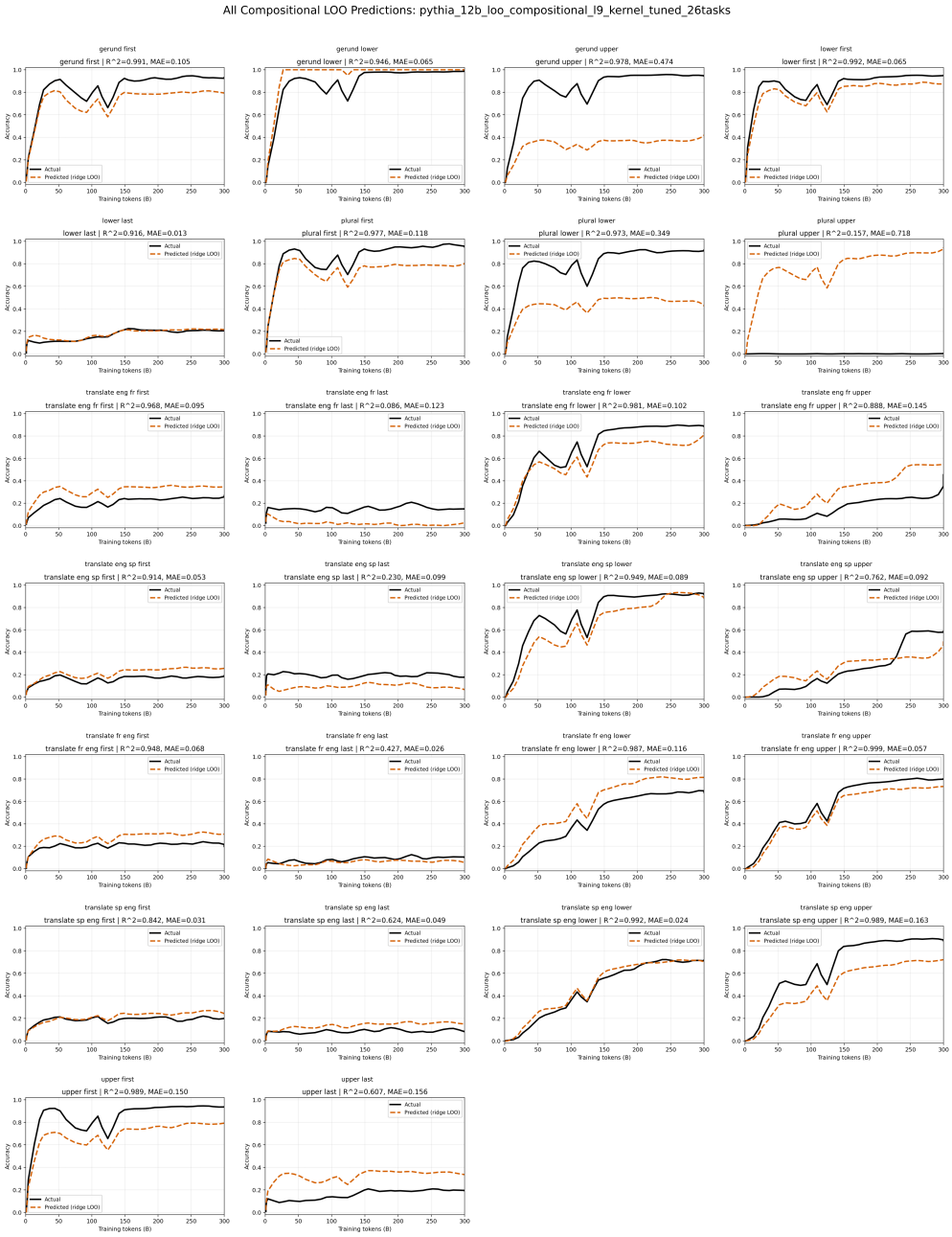


Figure 20: Predictions of held-out compositional tasks for Pythia-12B.

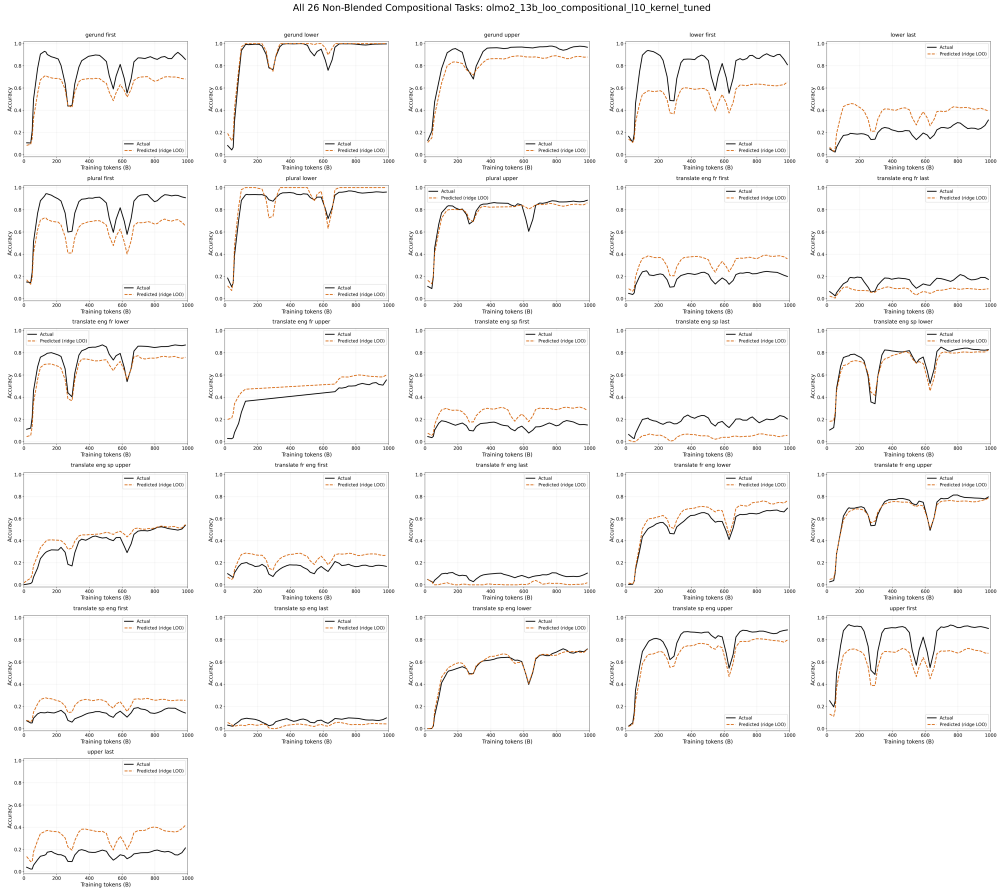


Figure 21: Predictions of held-out compositional tasks for OLMo2-13B.

Model	Representation	Layer	*k* heads	σ	λ
LLM360					
`amber`	hidden states	21	—	6.02568	0.0001
`crystal`	hidden states	8	—	6.25822	0.0001
Pythia					
`pythia_410m`	hidden states	3	—	0.33991	0.001
`pythia_1.4b`	hidden states	12	—	5.93639	0.001
`pythia_2.8b`	hidden states	10	—	2.64044	0.0001
`pythia_12b`	hidden states	9	—	4.02777	0.0001
OLMo-2					
`olmo2_1b`	hidden states	8	—	3.46810	0.0001
`olmo2_7b`	hidden states	16	—	1.05641	0.005
`olmo2_13b`	cie_heads	10	15	0.96582	0.005
OLMo-3					
`olmo3_7b`	hidden states	16	—	4.37314	0.001

Table 9: Function vector hyperparameters selected per model. The full residual stream was chosen as the representation for all models besides OLMo2-13B, in which the top 10 heads by causal indirect effect in layer 10 were chosen. σ and λ are the parameters used in ridge regression.