WHY LARGER MODELS LEARN MORE: EFFECTS OF CAPACITY, INTERFERENCE, AND RARE-TASK RETENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Larger models learn tasks smaller models do not. What drives this phenomenon? We develop a simple phenomenological argument that power-law scaling already suggests that a larger model will be able to learn a part of the data distribution that a smaller model fails to learn, even with infinite training data. To validate this claim and identify its causes, we study the effects of model scaling on a synthetic setup consisting of a mixture of tasks that show monotonic scaling curves. The results point to a data-induced competition over resources (neurons). Specifically, smaller models allocate their neurons to high frequency or low complexity tasks, and so they learn solutions that perform poorly on rare and complex tasks. Moreover, this happens even when solutions capable of expressing the desired task exist. We then assess how a larger model circumvents this data-centric bottleneck, finding that it traces to a reduced interference mechanism: larger models can allocate enough resources to common tasks that the gradient updates for those tasks become weak, which means that they do not overwrite rare-task features as they slowly accumulate. Finally, to further validate these claims, we pretrain OLMo models (4M to 4B parameters) on novel tasks of varying frequency and complexity. The results mirror those from our synthetic data experiments: only the larger OLMo models learn the infrequent and complex tasks, and they embed more task features in their representations and show less gradient interference between tasks. Overall, we offer a data-centric account of why larger models learn tasks that smaller models fail to. This helps explain why larger models are better in practice, and it can inform practical questions concerning model sizing and training data mixtures.

1 INTRODUCTION

Modern machine learning is celebrated for its massive generalist models. Based on the empirical finding that larger models often excel where smaller[1] models show random-chance performance, prior work has claimed that the ability to solve certain critical tasks *only* emerges in larger models (Wei et al., 2022; Arora & Goyal, 2023; Hu et al., 2024; Du et al., 2024; Wei et al., 2023). Such arguments have fueled the drive towards increased scaling. However, given the large training and inference costs that large models impose, it is worth identifying precisely what marginal benefits are unlocked by larger models and whether scaling parameters is the sole way of realizing those benefits.

Our argument begins from the observation that **power-law scaling (Kaplan et al., 2020; Hestness et al., 2017; Rosenfeld et al., 2020) already suggests that there is a regime in which a smaller model fails to learn parts of a data mixture that a larger model succeeds on, even under asymptotic training** (Fig. 1, Sec. 2). This suggests that larger models enjoy a genuine advantage that may allow them to learn task distributions that smaller models will inevitably fail to learn within the same training setup. Importantly, this is not an argument that larger models are simply more sample efficient (Henighan et al., 2020; Hoffmann et al., 2022; Du et al., 2024), but rather that smaller models suffer from a more fundamental limitation even under infinite training regimes.

To validate this prediction and identify its causes, we analyze a setting involving a mixture of regression tasks. In this, we are inspired by much recent work using toy tasks to pinpoint the effects

[1]We use the terms "larger" and "smaller" informally here but develop a precise relational definition of these terms in Sec. 2.

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

of scaling (Bordelon et al., 2025; Bahri et al., 2024; Michaud et al., 2023; Lubana et al., 2025; Edelman et al., 2023). Furthermore, all of the individual tasks in our setting are learnable by the models under consideration, capturing the idea that tasks smaller models fail to learn can still be induced via post-training (Lambert et al., 2025; Shao et al., 2024; Xin et al., 2025; Agarwal et al., 2024; Zhao et al., 2026). Correspondingly, mere expressivity notions are not the issue; instead, the question concerns the ability of these models to learn complex task distributions from data. These experiments lead to two key findings, as described below.

First, **scaling enables learning rare and complex tasks (Sec. 3.1).** Our experiment controls task frequency and complexity. We present an analytic argument that only larger models will (on average) learn the rare and complex tasks present in this setting, and we verify this analysis experimentally (Fig. 2).

Second, **reduced competition for resources enables learning rare and complex tasks (Sec. 3.2)**. Here, we extend our formal analysis to show that, upon observation of samples from a rare task, model parameters update, but only larger models, by virtue of having more parameters and hence less gradient interference, are able to retain memory of a previously observed batch of data from a rare task. Thus, when the next batch of rare-task data comes in, the larger model builds on its prior knowledge, which ultimately leads to success despite the impoverished learning signal. In contrast, the smaller model is forced to start from scratch and consequently fails. We again verify these findings in our regression setting (Fig. 3).

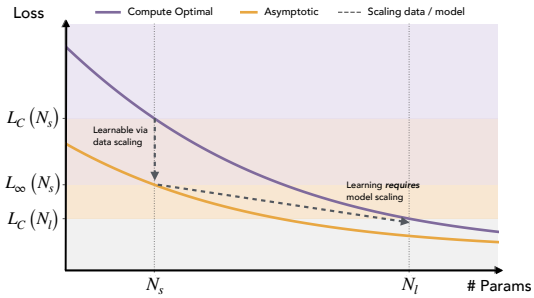

Figure 1: **Learning a part of the distribution requires model scaling.** Compare the loss curves for compute-optimal scaling with the one following an infinite resource regime (labeled asymptotic). The region labeled purple denotes the amount of loss both a smaller model with N_s parameters and a larger model with N_l parameters are able to achieve with respect to a random baseline under finite resources. We call loss reduction accessible to the smaller model under infinite compute, but that a larger model would get at in a more resource efficient manner (i.e., under finite compute), **learnable via data scaling**. If there remains a part of the loss that is achieved by the larger model under finite resources, but that a smaller model even under asymptotic data scaling is unable to reach, then we call this part **learned via model scaling**. This part of the distribution is explained by a larger model by virtue of its larger size.

Finally, we **validate the above theoretical arguments in real LLMs (Sec. 4).** Specifically, we pretrain OLMo models (4M to 4B parameters) on the Dolma v1.7 corpus with completely novel tasks injected at controlled frequency. We find that only the larger OLMo models are able to learn the infrequent and complex tasks (Sec. 4.2). Furthermore, these OLMo models mirror our toy-task models in deeper ways: larger OLMo models have more task features embedded in their representations (Sec. 4.3) and show less gradient interference (Sec. 4.4). Beyond supporting our theoretical claims, these results can provide practical guidance to large-scale model training efforts.

Overall, the data-centric nature of our analysis suggests that understanding why larger models learn more requires not only asking what they can represent, but also what is learnable under gradient-based optimization from a given data mixture.

2 Warmup: A Phenomenological Argument

Neural network scaling predictably and monotonically improves loss (Hoffmann et al., 2022):

$$L(N, D) = L_0 + \frac{A}{N^\alpha} + \frac{B}{D^\beta}, \tag{1}$$

where L_0 denotes the irreducible loss, A, B are constants, and α, β are parameter / data exponents ($\alpha \approx 0.46$ and $\beta \approx 0.51$ for Chinchilla-scaling (Hoffmann et al., 2022)). Training in a compute-optimal manner, i.e., finding the model size and data configuration that helps achieve the minimum loss at a given compute budget C, gives us $L_C(N) \propto N^{-\gamma}$, where $\gamma = 0.34$, and $L_C(N)$ denotes the optimum loss achieved when training a model with N parameters under resource constraints. The relation shows larger models are expected to achieve a smaller loss. However, resource-constrained training by itself does not inform what a model can actually express. *Specifically, even though*

a smaller model may have a worse compute-optimal loss, we do not know if it is fundamentally incapable of achieving the same loss as the larger model. To assess that statement, we must evaluate a model's loss under asymptotic data scaling:[2] $L_\infty(N) \propto N^{-\alpha}$. If $\alpha > \gamma$, as is the case in practice, we again see gains from merely scaling the model size. That is, *the asymptotic loss achieved by a larger model is better than the smaller one.* This indicates there is a part of the training distribution a smaller model, despite observing infinite data, fails to learn. Based on this phenomenological argument, we define the following.

Definition 1 (**Learnable via data scaling**). *Consider a target model with N_l number of parameters that we call "large". We say a "smaller" model, i.e., for which parameter count $N_s < N_l$, can recover the loss of a larger model via data scaling if $L_C(N_s) - L_C(N_l) > 0$, but $L_\infty(N_s) - L_C(N_l) < 0$.*

Def. 1 thus captures the scenario put forward in Sec. 1. That is, the smaller model may in fact be just **undertrained**: the larger model learns more sample efficiently and reduces loss faster, but a smaller model can eventually catch up (Henighan et al., 2020; Du et al., 2024; Hoffmann et al., 2022). Correspondingly, the marginal ability of a larger model to explain the data distribution (i.e., the loss) can be recovered by a smaller model merely observing more data. Nevertheless, there exist regimes where data scaling will not suffice, as described next.

Definition 2 (**Learnable via model scaling**). *Consider a target model with N_l number of parameters that we call "large". For a small scalar value ϵ, we define $N_s^*(\epsilon)$ as the largest "small" model if $L_\infty(N_s^*(\epsilon)) - L_C(N_l) > \epsilon$. That is, even asymptotically, the smallest model never reaches the same loss as the large model. Correspondingly, for a given model size N, we call it "small" if $N < N_s^*(\epsilon)$ and say recovering the loss of the larger model requires model scaling.*

This latter scenario thus captures the case where, when two models with parameter counts N_s, N_l, with $N_s < N_l$, are trained, there is truly a marginal improvement for explaining the data that can be attributed to the larger model having more parameters. This is the most interesting case that warrants further study: what is it about the data that only a larger model can learn, such that the smaller model cannot, even after observing infinite data? How precisely does having more parameters aid this learning? We aim to answer these questions in the following sections.

3 Scaling Allows Learning Rare Tasks by Reducing Interference

Our phenomenological argument in Sec. 2 motivates the claim that larger models are likely to learn a part of the data distribution smaller models will fail to learn. We next aim to get more concrete about this claim. Specifically, we exploit the fact that our argument is merely based on monotonic (power-law) scaling—a phenomenon even synthetic tasks can recapitulate (Bordelon et al., 2024; Maloney et al., 2022; Bahri et al., 2024). Such tasks have in fact been used in prior work to make accurate predictions about scaling behavior for large-scale models (Qiu et al., 2025; Paquette et al., 2024). We thus follow this line of work and develop a multi-task learning setup that helps assess which tasks a larger model can learn but a smaller model cannot. We generalize our claims to an off-the-shelf language model pretraining pipeline (Team OLMo et al., 2024) in Sec. 4, finding the core hypotheses derived out of this toy setting hold true on even a large-scale training pipeline.

Data. We consider a multi-task learning setup where samples are drawn from a mixture of K linear regression tasks. Specifically, the kth task is assumed to appear with **frequency** $\pi_k > 0$, such that $\sum_k \pi_k = 1$, and has *covariance* $C_k = B_k \Lambda_k B_k^\top = \sum_{j \geq 1} \lambda_{k,j}\, b_{k,j} b_{k,j}^\top$. Here, the "feature matrix" $B_k = [b_{k,1}, b_{k,2}, \ldots]$ is assumed to have orthonormal columns; $\Lambda_k = \text{diag}(\lambda_{k,1}, \lambda_{k,2}, \ldots)$ with $\lambda_{k,1} \geq \lambda_{k,2} \geq \cdots \geq 0$; and different tasks occupy orthogonal blocks, i.e., $B_k^\top B_\ell = 0$ for $k \neq \ell$. If the spectrum $\{\lambda_{k,j}\}$ decays slowly, the task requires more directions for producing the corresponding target—we can thus compare the **relative complexity** of two tasks by comparing the rate at which their spectra decay. Compared to prior work studying the theory of scaling laws based on toy regression tasks, we emphasize that our setup involves the learning of multiple tasks simultaneously.

Teacher / Student Models. For a given input $x \sim \mathcal{N}(0, I)$, the teacher for task k is defined as $y_k = \Lambda_k^{1/2} B_k^\top x$. The student uses a shared width-N encoder $U \in \mathbb{R}^{d \times N}$, $U^\top U = I$, with

[2] We note power-law scaling need not hold asymptotically (Paquette et al., 2024; Bordelon et al., 2025), which is why we call this argument phenomenological. It motivates the subsequent, rigorous claims.

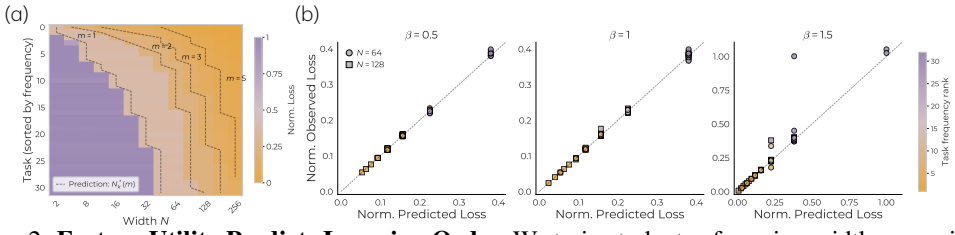

Figure 2: **Feature Utility Predicts Learning Order.** We train students of varying width on a mixture of $K = 32$ regression tasks with power-law task frequencies (β) and plot per-task loss (normalized by mean predictor). (a) Empirical phase diagram where task features ($\beta = 1.0$) are retained as a function of width and task frequency, matching our prediction. (b) Loss matches the analytic prediction from Theorem 3 across task-frequency exponents. Overall, we see that increasing width preferentially improves low-frequency tasks because it allows the model to retain lower-utility features.

projector $P_U = UU^\top$, together with task-specific linear decoders D_k to discern between tasks. Correspondingly, the student prediction is $\hat{y}_k = D_k U^\top x$. The total mixture loss is the weighted sum $\mathcal{L}_N(U) = \sum_{k=1}^K \pi_k \ell_k(U)$, where $\ell_k(U) = \mathbb{E}\big[\|y_k - D_k U^\top x\|_2^2\big]$ is loss of the kth task. Note that herein, since the optimal decoder admits a closed-form solution $D_k^* = \Lambda_k^{1/2} B_k^\top U$, we solely analyze the dynamics of the encoder, which produces features used by the student for making predictions.

3.1 Larger Models Learn Rarer, More Complex Tasks

In order to narrow down a mechanism that explains *how* larger models may be able to learn more, we must first identify precisely *what* it is that a larger model learns but a smaller one fails to. We begin with answering this question in our toy setup.

Theorem 3 (Features are Learned in Order of Utility)**.** *For a given U, the mixture loss reduces to $L_N(U) = \mathrm{Tr}(M) - \mathrm{Tr}(U^\top M U)$, where $M := \sum_{k=1}^K \pi_k C_k$. Hence, a width-$N$ minimizer spans the top-N eigenspace of M, whose eigenvalues are defined by the weighted per-task spectra:*

$$u_{k,j} := \pi_k \lambda_{k,j}. \tag{2}$$

Thus, the optimal encoder keeps the N features (k, j) with largest $u_{k,j}$—we call these terms utilities. This implies if $n_k(N)$ denotes the number of retained features from task k, then $\ell_k^*(N) = \sum_{j > n_k(N)} \lambda_{k,j}$. Conversely, the minimum width at which a model learns at least m features for all tasks is $N^*(m) = \min\{N : n_k(N) \geq m\}$.

In the context of our toy task, the statement above helps answer the question "what does width buy?" by defining a concrete ranking rule for feature learning.[3] Specifically, it says a larger model, asymptotically, learns exactly those features whose utilities are lower than those of the features learned by a smaller model. *This implies if a task is observed infrequently or it involves several features, i.e., if its spectrum decays very slowly, then (on average) only a larger model will learn it.*

Verification. We verify the claim above by training our student model on a mixture of $K = 32$ tasks, using the Adam optimizer for 100K steps (the loss does not improve beyond this budget even when trained up to $10\times$ longer; see Fig. 18). We use a power-law prior $k^{-\beta}$ to define task frequencies, and a power-law per-task spectrum $\lambda_{k,j} \propto j^{-\alpha}$. For simplicity of visualization, we let $\alpha = 2$ be shared across tasks and only vary task frequencies by changing β (see App. E for experiments modulating complexity by varying α). Results are reported in Fig. 2 (also see App. F for further results). We find (a) the per-task loss and (b) the overall residual loss predictably reduce with model width. Critically, we see larger models learn infrequent tasks better than smaller ones.

3.2 Scaling Reduces Interference and Supports Retention of Rare Task Data

While the argument above—i.e., a larger model learns low utility, infrequent features—is intuitively reasonable, it is critical to note that if the frequency at which a task or its features are seen is very low, then, regardless of size, there is a statistical bottleneck here that a model needs to circumvent. For example, in the experiments shown in Fig. 2b, a model must learn a task that constitutes merely

[3]This claim can also be seen as a static ordering rule that local optima visited by a model during training will be expected to dynamically follow in its saddle-to-saddle dynamics (Zhang et al., 2026; Abbe et al., 2023; Jacot et al., 2021; Kunin et al., 2026)

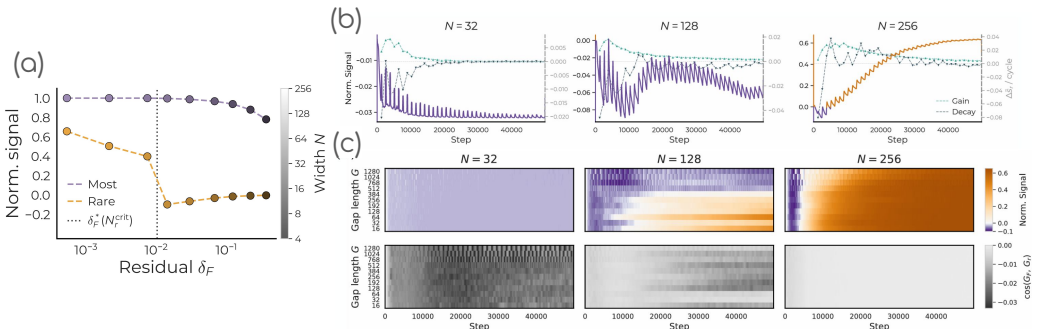

Figure 3: **(a): Residual Controls Learning.** We plot signals encoded in model representations for most frequent and rarest tasks as a function of width N and remaining residual δ_{F}. In line with our predictions, we see larger models perfectly capture tasks of all frequencies, while smaller models do not. **(b) and (c): Rare-Task Retention by Larger Models.** We isolate retention by training with a matched-frequency injection protocol: the rare task is withheld for G steps and then reintroduced in a batch such that its overall frequency is the same. (b) Training dynamics for $G = 1280$. We see small models briefly encode the rare task (Norm. signal $\tilde{s_r}$: left-y axis) after each injection; However, as frequent-task updates resume, this signal is lost between injection steps ('decay': gray dotted line). Meanwhile, larger models retain more of the rare-task signal between injections and accumulate it over training. (c) Across injection gaps G and widths N, rare-task signal decays rapidly in narrow models but remains stable in wider models, while frequent-task signal is largely unaffected. Furthermore, the cosine similarity of gradients via a batch of rare-task samples G_r and frequent task samples G_{F} shows that scaling provides enough representational capacity where updates from frequent tasks no longer overwrite rare-task features before the next rare observation arrives.

0.25% of observations. We next analyze how width helps surmount this challenge. To this end, note that for the kth task, the Riemannian gradient is $G_k(U) = 2(I - P_U)C_kU$, and hence the mixture gradient is $\dot{U} = 2(I - P_U)MU$. We then have the following claim.

Theorem 4 (Residual Controls Learning)**.** *Let $\mathsf{F} \subseteq [K]$ denote the common or frequent tasks. Define these tasks' weighted covariance $M_{\mathsf{F}} := \sum_{k\in\mathsf{F}} \pi_k C_k$ and residual signal $\delta_{\mathsf{F}}(U) := \mathrm{Tr}\big((I-P_U)M_{\mathsf{F}}\big)$. Then, the aggregate common-task gradient $G_{\mathsf{F}}(U) = 2(I - P_U)M_{\mathsf{F}}U$ obeys the bound*

$$\|G_{\mathsf{F}}(U)\|_F \leq 2\sqrt{\lambda_1(M_{\mathsf{F}})\,\delta_{\mathsf{F}}(U)}. \tag{3}$$

The statement above says a set of tasks move the model only through the part of their covariance that is *not already explained* by the current representation, i.e., the residual $\delta_{\mathsf{F}}(U)$. Correspondingly, once the high-utility common-task features have been learned, their updates become weak (i.e., low norm). *This leaves any spare width available to rare tasks.* More precisely, let $\mu_1^{\mathsf{F}} \geq \mu_2^{\mathsf{F}} \geq \cdots$ be the eigenvalues of M_{F}. The best width-N representation for the common tasks alone leaves residual $\delta_{\mathsf{F}}^*(N) = \sum_{i>N} \mu_i^{\mathsf{F}}$. Then, via Theorem 4, we get the following:

Corollary 5 (Width-Scaling Reduces Competition)**.** *Define $N_{\mathsf{F}}(\varepsilon) := \min\big\{N : \sum_{i>N} \mu_i^{\mathsf{F}} \leq \varepsilon\big\}$. For every $N \geq N_{\mathsf{F}}(\varepsilon)$, there exists an encoder for which $\delta_{\mathsf{F}}^*(N) \leq \epsilon$ and $\|G_{\mathsf{F}}(U)\|_F \leq 2\sqrt{\mu_1^{\mathsf{F}}\epsilon}$.*

That is, once $N \gtrsim N_{\mathsf{F}}(\varepsilon)$, the model contains enough resources that can be allocated to the common tasks, rendering the gradient towards them weak. This makes the remaining resources available to rare tasks. However, even once interference is weak enough for a rare task to be learned, it is unclear whether gradient descent can actually consolidate that signal across its infrequent observations. To this end, we next characterize the local condition under which a specific rare feature can pull the model towards itself, without forcing the forgetting of well-learned tasks. Specifically, assume we wanted to learn a rare rank-one task $C_r = \lambda_r b_r b_r^\top$ orthogonal to the common block. Let $U_F^{(N)}$ be top-N eigenspace of M_F with eigenvalues $\mu_1^F \geq \mu_2^F \geq \cdots$. Then, we have the following claim.

Proposition 6 (Interference Reduces via Scaling)**.** *The common-task solution $U_{\mathsf{F}}^{(N)}$ is stable against direction b_r, i.e., common tasks' loss does not grow by learning of b_r, iff $\pi_r\lambda_r < \mu_N^{\mathsf{F}}$. Thus, the critical width at which b_r gets learned is $N_r^{crit} := \min\{N : \mu_N^{\mathsf{F}} \leq \pi_r\lambda_r\}$.*

The claim above hence shows width scaling helps in two related but distinct ways. First, it reduces the total unresolved common-task signal $\delta_{\mathsf{F}}^*(N)$, which bounds the aggregate common-task gradient.

Second, as Proposition 6 shows, it lowers the weakest occupied common-task utility μ_N^{F}, which determines whether a particular rare feature can *displace* a common feature and become locally stable; if the rare feature's utility is lower, even if the model updates to learn it, the common tasks' least utility feature will eventually replace it. *This will result in a swinging, update-and-forget learning dynamic where the rare task features and lowest utility features of common tasks will compete over model parameters.* Overall, this suggests the learning bottleneck is defined by the interaction between data and scale: if the task we care to learn does not have sufficient utility for reducing the loss, then the model will prefer to learn and preserve lower-order modes of other tasks; however, *by increasing width, one avails capacity to such low-utility tasks and reduces competition between tasks over model parameters, enabling learning of the rare task without forgetting features relevant to common tasks*.

Validation. We train models of varying width on the same setup as Fig. 2 and plot how much signal from directions describing a task is present in the model's intermediate representation. Specifically, since $\ell_k(U) = \mathrm{Tr}((I - P_U)C_k)$, the signal captured for task k is $\mathrm{Tr}(P_U C_k)$, where $\mathrm{Tr}(.)$ denotes trace of a matrix. We thus measure $s_k(U) = \frac{\mathrm{Tr}(P_U C_k)}{\mathrm{Tr}(C_k)}$. To contextualize this value, we normalize with respect to a random baseline, yielding $\tilde{s}_k(U) = \frac{s_k(U) - N/d}{1 - N/d}$; N/d denotes the expected value of $s_k(U)$ if U were a randomly drawn matrix from the Stiefel manifold. We call this quantity "Normalized Signal". Results are shown in Fig. 3 (a). We see when the model width is small, frequent tasks have a high residual signal remaining to be explained; here the set of frequent tasks is defined as top-K tasks whose prior sums to 0.8, resulting in $K = 3$. Correspondingly, rare tasks' signal in model representation is no better than random. Meanwhile, as we scale, once the width crosses our predicted threshold $\delta_F^*(N_r^{\mathrm{crit}})$, we find the bulk of the frequent tasks' signal is explained away and rare tasks start to get learned.

To isolate how the gap between observations interacts with width, we also design a matched-frequency injection experiment: the rare task is excluded from training for G steps, then injected in a batch enlarged to $m = G \cdot B \cdot \rho_r$ rare samples so that its long-run frequency exactly matches the setup of Fig. 2. This emphasizes the ability of a model to retain memories about observed data, while preserving the total frequency with which it is seen. Results are shown in Fig. 3. We see at the end of training, rare-task signal decays monotonically with G at all widths, but far more steeply for smaller models. Meanwhile, the learning dynamics in panel (b) show that after each injection, a larger model accumulates rare-task signal and retains enough of it to build on the next injection, while a smaller model decays back to near-zero in between (an intuitive model explaining this dynamic is shown in Fig. 8 and analytically described in App. D.4). Overall, our results showing how larger models learn tasks smaller models do not can be summarized as follows.

Hypothesis: Scaling Enables Learning via Reduced Interference and Better Retention

Given the same training setup, larger models better learn tasks in the tail of the data distribution. When such a task is observed, a larger model can retain part of the update and build on it when the task appears again. In a small model, the same parameters face more competition: updates from frequent tasks undo the rare-task update before the next rare batch arrives. Rare-task learning then becomes an update-and-forget loop.

4 Corroborating Claims with the OLMo Pretraining Pipeline

We now verify the claims of Sec. 3 in a realistic LLM pre-training setting using the OLMo pipeline. We train models of size 4M to 4B on up to 210B tokens (∼50K steps). Following the structure of Sec. 3, we offer analyses at three levels: loss, representation, and gradient.

4.1 Setup

A key variable in our claims is the frequency of a task.[4] However, measuring the frequency of a naturally occurring task in pre-training data is challenging, as instances from the same task can occur in many surface forms. To tightly control task frequency, we adopt a data injection framework from the memorization literature (Jagielski et al., 2023; Carlini et al., 2019; Huang et al., 2024; Wei et al., 2026). We inject different instances sampled from the distribution of a "special" task T at a controlled

[4]Defining the complexity of a natural task is difficult, and hence we solely focus on frequency in this section.

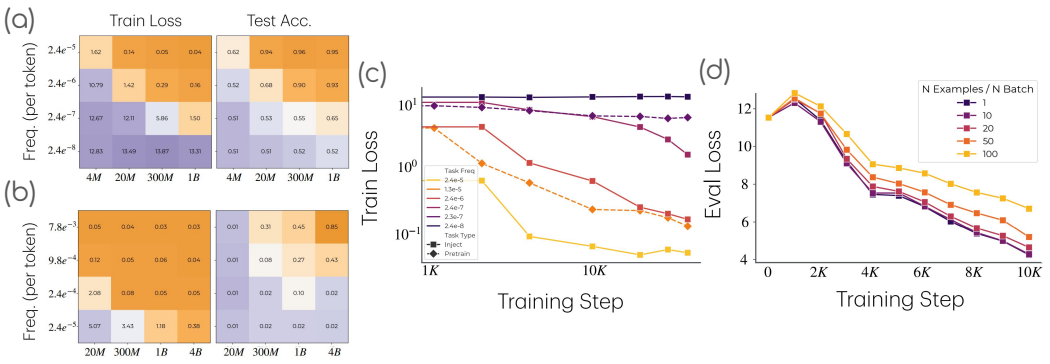

Figure 4: **Behavioral Evidence. (a) and (b): Larger Models Learn Rare Tasks; Smaller Models Do Not.** Training loss and test accuracy for the (a) Comparison task (T_{CMP}) and (b) Modular Addition task (T_{ADD}). Orange color indicates lower loss/higher accuracy. Overall, increasing width enables learning of low-frequency tasks. **(c): Tasks are learned in the order of frequency**. Solid lines: Training loss with comparison task (T_{CMP}) injected at different frequencies. Dashed lines: Reference arithmetic tasks observed from pre-training data. **(d): Rare-Task Retention Has an Effect on Learning.** With matched-frequency injection of T_{CMP}, i.e., injecting N task instances every N batches, a larger injection gap N degrades task loss.

frequency f to measure whether a model has learned the task distribution. The task T is special in the sense that it is unlikely to be part of normal pre-training data. We then train models of various sizes on data mixtures generated from different values of f.

Tasks. We consider two special tasks T: comparison (T_{CMP}) and modular addition (T_{ADD}). Both tasks are encoded as a sequence of three tokens: `TOK1`, `TOK2`, `LABEL`, where `TOK1`, `TOK2` $\in \mathcal{S}$, a set of 100 tokens randomly sampled from the vocab. There are exactly 10K instances per task, with train/test split $50/50$. Critically, both tasks require models to learn certain geometrical structures to generalize (Hwang & Park, 2026). This provides a measure for learning a task (as opposed to memorizing training instances) and features to verify the interference hypothesis of Sec. 3.

Data. We use Dolma v1.7 as the pre-training corpus (Soldaini et al., 2024). Given a task T, we inject instances sampled from its train split at a frequency of 7.8×10^{-3} to 2.4×10^{-8}, roughly from 1K instances per batch to 1 instance every 10 batches. To ensure the injected task frequency is comparable to the frequency of tasks learned in pre-training, we sample two reference tasks R_{cmp} and R_{add} from pre-training that involve similar high-level functions. The three-token sequence plus an end of document token replace the first four tokens of a training sequence. See App. C.3 for details.

Models. We train OLMo models (Groeneveld et al., 2024) with 4M, 20M, 300M, 1B, and 4B parameters. We focus on scaling the hidden and MLP dimensions and the number of attention heads; the 4M parameter model has depth 8 and the rest have depth 16. See App. C.2 for details.

4.2 Behavioral Evidence

Larger Models Learn Rarer Tasks. We first replicate the behavioral findings in Sec. 3.1. We measure the effect of task frequency by comparing multiple training runs that only differ by the frequency of the injected task. As shown in Fig. 4, larger models learn lower-frequency tasks much better than smaller models do. This matches the pattern in Fig. 2. Moreover, tasks are learned in the order of frequency. For each model run, we compare the order in which the injected task T_{CMP} and the reference tasks are learned, as shown in Fig. 4a. Most importantly, larger models do not just lead to better memorization of training instances, i.e., low training loss, but also learn generalizable task structures, i.e., high eval accuracy. On T_{ADD}, only larger models trained on higher frequency exhibit the grokking phenomenon (Power et al., 2022).

Rare-Task Retention Has an Effect on Learning. We conduct the matched-frequency injection experiment as described in Fig. 3, i.e., injecting N task instances every N batches, for $N = 1, 10, 20, 50, 100$. Fig. 4d shows the effects of retention, as models trained with larger gaps between task instances have higher task loss, even though the global task frequency of all runs is equivalent.

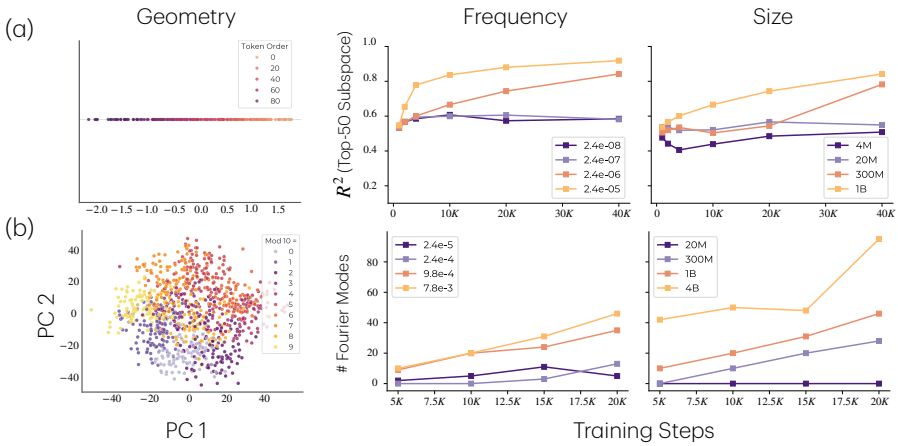

Figure 5: **Representational Evidence. Scaling model size and increasing task frequency lead to models learning more task-relevant features.** Rows correspond to (a) the comparison task T_{CMP} and (b) the modular addition task T_{ADD}. The first column shows feature geometry, i.e., a global token order for T_{CMP} and Fourier features for T_{ADD}. The last two columns quantify how these features scale with task frequency and model size.

4.3 Representational Evidence

Task Features. In our toy setting (Sec. 3), we know analytically which features are necessary for learning the kth task, i.e., B_k, and to what extent the model can represent these features, i.e., P_U. For OLMo models, we empirically identify a set of causal features that a pre-trained LM uses to solve the task and localize them in the model representations. Specifically, for T_{CMP}, the task feature of core relevance is the global order of the tokens, which allows number comparisons; meanwhile for T_{ADD}, task features are the Fourier modes (Nanda et al., 2022; Zhou et al., 2024; Feucht et al., 2026), as shown in Fig. 5 using the largest model trained on the most frequent task data. These task features allow us to conduct versions of the gradient and representation-level analyses in Sec. 3.

More Task Features are Present in Larger Model Representations. We localize the task features in models that have clearly learned the task and measure to what extent these target task features are present in other models, which parallels the metric $\ell_k(U)$ used in the toy setting. For localization, we use distributed alignment search (DAS) (Geiger et al., 2024) which finds subspaces that *causally* encode the features. For T_{CMP}, a global ordering of the tokens can be localized to a 1-D subspace in the residual stream of the first layer. For T_{ADD}, Fourier modes can be identified in the residual stream from earlier layers to the last layer. We then measure to what extent these task features are present in model representations. For T_{CMP}, since the geometry of the task feature is a single direction, we apply linear regression to representations spanning the top $K = 50$ principal components. For T_{ADD}, we measure the total number of Fourier modes present through all layers. We include the details in App. C.4. The last two columns of Fig. 5 show the extent to which the target task features are present in each model across checkpoints. For the frequency (middle) column, we fix the model size to 1B. For the model size (last) column, we fix the task frequency to be 2.4×10^{-6} for T_{CMP} and 7.8×10^{-3} for T_{ADD}. We see that (i) the presence of task features is highly correlated with high test accuracy, and (ii) larger models and models trained on more frequent task data learn these task features faster.

4.4 Gradient Evidence

We now connect the behavioral evidence (Sec. 4.2) and the internal representation account (Sec. 4.3) by analyzing how task gradients interfere with non-task gradients on a set of neurons that implement the task circuit. We focus on T_{CMP} runs in Fig. 6, with 100 task instances injected per 100 steps.

Task Neurons. We first identify which MLP layers implement the task features defined in Sec. 4.3. For all the models that we compared, the first layer MLP has the largest causal effects on task predictions. We further identify the top K neurons in the first layer MLP that have the largest gradient magnitude and use the gradients of these neurons for analysis. Details can be found in App. C.5.

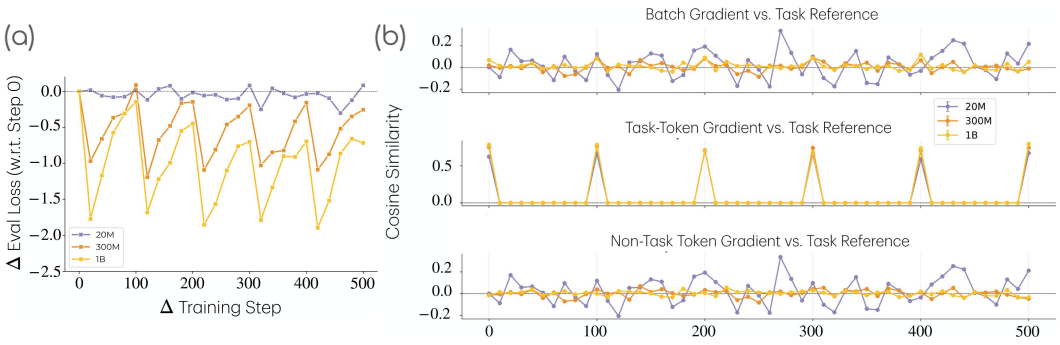

Figure 6: **(a): Rare-Task Retention.** We inject 100 instances of the T_{CMP} task every 100 batches. Larger models can retain the injected task information better, i.e., larger task eval loss drop. **(b): Gradient Interference.** We analyze how batch gradients align with a task reference direction g_r by decomposing the batch gradient into task token and non-task token contributions. **Top**: Cosine similarity between full-batch gradient direction and the task direction g_r. **Middle**: Cosine similarity between batch task gradient direction and g_r. Higher values imply more task signals. **Bottom**: Cosine similarity between batch non-task gradient direction and g_r. Lower values imply less gradient interference. Overall, batch gradients of larger models carry more task signals with less interference.

Task Reference Direction g_r. We estimate the task reference direction using the aggregated gradient of the task loss computed over all 10K task instances, an analogy to G_r in the toy setting. This direction may shift across training steps; however, at a given step, it is the optimal task direction.

Larger Models Show Less Gradient Interference Between General Language Modeling and Our Injected Task. We quantify the relation between the task reference g_r and the batch gradient g, which can be further decomposed into gradient from the task tokens g_t (if exists in batch) and non-task tokens g_{nt}, i.e., $g = g_t + g_{nt}$. We first measure the cosine similarity between task reference and batch gradient direction, replicating the results in Fig. 3. We additionally analyze whether task or non-task tokens contribute to this similarity; while task token gradient aligning with task reference g_r is expected, non-task token gradient with non-zero cosine similarity suggests that the language modeling direction is interfering with the task gradient direction.

Results are shown in Fig. 6. In the top panel, larger models have higher similarity between g and g_r at the injection steps, 0.08 ± 0.02 for the 1B model and 0.04 ± 0.04 for the 300M model, the similarity typically regresses towards zero between injections. For the 20M model, the similarity scores oscillate wildly across batches, even at the injection step. In fact, the high similarity between non-task gradient g_{nt} and g_r reveals that for the 20M model the batch gradient similarity mostly comes from random collisions with task direction, with a similarity score of 0.10 ± 0.09, while for larger models, g_{nt} is almost orthogonal to g_r, with $7.58 \times 10^{-5} \pm 0.02$ for the 1B model, suggesting little to no gradient interference on this set of neurons.

5 Discussion

We develop a data-centric account of why larger models can learn tasks that smaller models fail to learn. Specifically, we show that larger models can learn rare tasks from the data mixture, and this phenomenon is explained by learning dynamics, i.e., competition of resources and retention of memories, as well as the task frequency and complexity. Our perspective highlights that understanding scaling requires thinking beyond model expressivity. We need to understand how learning dynamics are at play with task frequency and complexity. It also points toward more intentional design of data mixtures to better elicit target capabilities. For example, simply scaling up the frequency of a target task might provide a more efficient way to learn the task than scaling up the model size. Lastly, our findings on how better retention of memories enables learning rare tasks offers a new perspective that views memorization as a mechanism that can support learning abstraction: by retaining task instances longer, models can accumulate signals across batches to learn more generalizable structures of the task. This suggests memorization can in fact be beneficial, in line with arguments by Feldman (2020).

AI USE STATEMENT

In this work, we used generative AI tools, specifically Claude Code, for research execution. This is limited to writing scripts for the representation and gradient analysis.

We have not used generative AI tools for aiding writing (except for spell and grammar checking), retrieval and discovery, research ideation, drafting sections of the paper, or proving mathematical claims, and generating synthetic datasets is not applicable to this work. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz. Sgd learning on neural networks: leap complexity and saddle-to-saddle dynamics. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 2552–2623. PMLR, 2023.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Rahaf Aljundi. Continual learning in neural networks. *arXiv preprint arXiv:1910.02718*, 2019.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019.

Sanjeev Arora and Anirudh Goyal. A theory for emergence of complex skills in language models. *arXiv preprint arXiv:2307.15936*, 2023.

Alexander Atanasov, Jacob A Zavatone-Veth, and Cengiz Pehlevan. Scaling and renormalization in high-dimensional regression. *arXiv preprint arXiv:2405.00592*, 2024.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024.

Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. A dynamical model of neural scaling laws. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=nbOY1OmtRc.

Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. How feature learning can improve neural scaling laws. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=dEypApI1MZ.

Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. When is memorization of irrelevant training data necessary for high-accuracy learning? In *Proceedings of the 53rd annual ACM SIGACT symposium on theory of computing*, pp. 123–132, 2021.

Francesco Cagnetta, Alessandro Favero, Antonio Sclocchi, and Matthieu Wyart. Scaling laws and representation learning in simple hierarchical languages: Transformers vs. convolutional architectures. *arXiv preprint arXiv:2505.07070*, 2025a.

Francesco Cagnetta, Hyunmo Kang, and Matthieu Wyart. Learning curves theory for hierarchically compositional data with power-law distributed features. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=Lw0kC75dY0.

Francesco Cagnetta, Allan Raventós, Surya Ganguli, and Matthieu Wyart. Deriving neural scaling laws from the statistics of natural language. *arXiv preprint arXiv:2602.07488*, 2026.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pp. 267–284, Santa Clara, CA, August 2019. USENIX Association. ISBN 978-1-939133-06-9. URL https://www.usenix.org/conference/usenixsecurity19/presentation/carlini.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TatRHT_1cK.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. How do large language models acquire factual knowledge during pretraining? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=TYdzj1EvBP.

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. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MO5PiKHELW.

Peter L Chen, Xiaopeng Li, Xi Chen, and Tianyi Lin. Reward-free alignment for conflicting objectives. *arXiv preprint arXiv:2602.02495*, 2026.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pp. 794–803. PMLR, 2018.

Chen Cheng, John Duchi, and Rohith Kuditipudi. Memorize to generalize: on the necessity of interpolation in high dimensional linear regression. In *Conference on Learning Theory*, pp. 5528–5560. PMLR, 2022.

Darshil Doshi, Tianyu He, Aritra Das, and Andrey Gromov. Grokking modular polynomials. *arXiv preprint arXiv:2406.03495*, 2024.

Zhengxiao Du, Aohan Zeng, Yuxiao Dong, and Jie Tang. Understanding emergent abilities of language models from the loss perspective. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=35DAviqMFo.

Sunny Duan, Mikail Khona, Abhiram Iyer, Rylan Schaeffer, and Ila R Fiete. Uncovering latent memories in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Pareto frontiers in deep feature learning: Data, compute, width, and luck. *Advances in Neural Information Processing Systems*, 36:48021–48034, 2023.

Katie Everett, Lechao Xiao, Mitchell Wortsman, Alexander A Alemi, Roman Novak, Peter J Liu, Izzeddin Gur, Jascha Sohl-Dickstein, Leslie Pack Kaelbling, Jaehoon Lee, et al. Scaling exponents across parameterizations and optimizers. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=0ksNeD1SJT.

Itay Evron, Edward Moroshko, Rachel Ward, Nathan Srebro, and Daniel Soudry. How catastrophic can catastrophic forgetting be in linear regression? In *Conference on Learning Theory*, pp. 4028–4079. PMLR, 2022.

Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, pp. 3762–3773. PMLR, 2020.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd annual ACM SIGACT symposium on theory of computing*, pp. 954–959, 2020.

Sheridan Feucht, Tal Haklay, Usha Bhalla, Daniel Wurgaft, Can Rager, Raphaël Sarfati, Jack Merullo, Thomas McGrath, Owen Lewis, Ekdeep Singh Lubana, Thomas Fel, and Atticus Geiger. Arithmetic in the wild: Llama uses base-10 addition to reason about cyclic concepts, 2026. URL https://arxiv.org/abs/2605.01148.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In Francesco Locatello and Vanessa Didelez (eds.), *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pp. 160–187. PMLR, 01–03 Apr 2024. URL https://proceedings.mlr.press/v236/geiger24a.html.

Pulkit Gopalani, Ekdeep S Lubana, and Wei Hu. Abrupt learning in transformers: A case study on matrix completion. *Advances in Neural Information Processing Systems*, 37:55053–55085, 2024.

Sachin Goyal, Pratyush Maini, Zachary C Lipton, Aditi Raghunathan, and J Zico Kolter. Scaling laws for data filtering–data curation cannot be compute agnostic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22702–22711, 2024.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15789–15809, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.841. URL https://aclanthology.org/2024.acl-long.841/.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, et al. Scaling laws and interpretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*, 2022.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. 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.

Shengding Hu, Xin Liu, Xu Han, Xinrong Zhang, Chaoqun He, Weilin Zhao, Yankai Lin, Ning Ding, Zebin Ou, Guoyang Zeng, Zhiyuan Liu, and Maosong Sun. Predicting emergent abilities with infinite resolution evaluation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=lDbjooxLkD.

Jing Huang, Diyi Yang, and Christopher Potts. Demystifying verbatim memorization in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10711–10732, 2024.

Hyeonbin Hwang and Yeachan Park. Intrinsic task symmetry drives generalization in algorithmic tasks, 2026. URL https://arxiv.org/abs/2603.01968.

Aapo Hyvärinen, Jarmo Hurri, and Patrick O Hoyer. *Natural image statistics: A probabilistic approach to early computational vision.*, volume 39. Springer Science & Business Media, 2009.

Arthur Jacot, François Ged, Berfin Şimşek, Clément Hongler, and Franck Gabriel. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity. *arXiv preprint arXiv:2106.15933*, 2021.

Matthew Jagielski, Om Thakkar, Florian Tramer, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Guha Thakurta, Nicolas Papernot, and Chiyuan Zhang. Measuring forgetting of memorized training examples. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=7bJizxLKrR.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *International conference on machine learning*, pp. 15696–15707. PMLR, 2023.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Dmitrii Krasheninnikov, Richard E. Turner, and David Krueger. Fresh in memory: Training-order recency is linearly encoded in language model activations. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Tn6famjSxN.

Rohith Kuditipudi, Jing Huang, Sally Zhu, Diyi Yang, Christopher Potts, and Percy Liang. Blackbox model provenance via palimpsestic membership inference. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=VRhVS59yhP.

Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vt5mnLVIVo.

Daniel Kunin, Giovanni Luca Marchetti, Feng Chen, Dhruva Karkada, James B Simon, Michael R DeWeese, Surya Ganguli, and Nina Miolane. Alternating gradient flows: A theory of feature learning in two-layer neural networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=t7LKc0MMW6.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH.

Pietro Lesci, Clara Meister, Thomas Hofmann, Andreas Vlachos, and Tiago Pimentel. Causal estimation of memorisation profiles. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15616–15635, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.834. URL https://aclanthology.org/2024.acl-long.834/.

Licong Lin, Jingfeng Wu, Sham M. Kakade, Peter L. Bartlett, and Jason Lee. Scaling laws in linear regression: Compute, parameters, and data. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 60556–60606. Curran Associates, Inc., 2024. doi: 10.52202/079017-1937. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6fcb1afcc1e9c2c82c8ddddf03bcf0f6-Paper-Conference.pdf.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890, 2021.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=u2vAyMeLMm.

Marco Loog, Tom Viering, Alexander Mey, Jesse H Krijthe, and David MJ Tax. A brief prehistory of double descent. *Proceedings of the National Academy of Sciences*, 117(20):10625–10626, 2020.

Ekdeep Singh Lubana, Kyogo Kawaguchi, Robert P. Dick, and Hidenori Tanaka. A percolation model of emergence: Analyzing transformers trained on a formal language. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=0pLCDJVVRD.

Ian Magnusson, Nguyen Tai, Ben Bogin, David Heineman, Jena D. Hwang, Luca Soldaini, Akshita Bhagia, Jiacheng Liu, Dirk Groeneveld, Oyvind Tafjord, Noah A. Smith, Pang Wei Koh, and Jesse Dodge. Datadecide: How to predict best pretraining data with small experiments. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=p9YlQPF8fE.

Pratyush Maini, Vineeth Dorna, Parth Doshi, Aldo Carranza, Fan Pan, Jack Urbanek, Paul Burstein, Alex Fang, Alvin Deng, Amro Abbas, et al. Beyondweb: Lessons from scaling synthetic data for trillion-scale pretraining. *arXiv preprint arXiv:2508.10975*, 2025.

Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. In *International conference on machine learning*, pp. 6682–6691. PMLR, 2020.

Alexander Maloney, Daniel A Roberts, and James Sully. A solvable model of neural scaling laws. *arXiv preprint arXiv:2210.16859*, 2022.

Martin Marek, Dongkyu Cho, Shikai Qiu, Rumi Chunara, Pavel Izmailov, and Andrew Gordon Wilson. Forgetting in language models: Capacity, optimization, and self-generated replay, 2026. URL https://arxiv.org/abs/2605.26097.

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75 (4):667–766, 2022.

Eric Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 28699–28722. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/5b6346a05a537d4cdb2f50323452a9fe-Paper-Conference.pdf.

Eric J Michaud, Liv Gorton, and Tom McGrath. Understanding sparse autoencoder scaling in the presence of feature manifolds. *arXiv preprint arXiv:2509.02565*, 2025.

Mohamad Amin Mohamadi, Zhiyuan Li, Lei Wu, and Danica J. Sutherland. Why do you grok? a theoretical analysis on grokking modular addition. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=ad5I6No9G1.

Depen Morwani, Benjamin L. Edelman, Costin-Andrei Oncescu, Rosie Zhao, and Sham M. Kakade. Feature emergence via margin maximization: case studies in algebraic tasks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=i9wDX850jR.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher Manning. Grokking of hierarchical structure in vanilla transformers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 439–448. Association for Computational Linguistics, July 2023. URL https://aclanthology.org/2023.acl-short.38/.

Preetum Nakkiran. More data can hurt for linear regression: Sample-wise double descent. *arXiv preprint arXiv:1912.07242*, 2019.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1g5sA4twr.

Preetum Nakkiran, Prayaag Venkat, Sham M. Kakade, and Tengyu Ma. Optimal regularization can mitigate double descent. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=7R7fAoUygoa.

Yoonsoo Nam, Nayara Fonseca, Seok H Lee, Chris Mingard, and Ard A Louis. An exactly solvable model for emergence and scaling laws in the multitask sparse parity problem. *Advances in Neural Information Processing Systems*, 37:39632–39693, 2024.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, sep 2022. URL https://openreview.net/forum?id=9XFSbDPmdW.

Elliot Paquette, Courtney Paquette, Lechao Xiao, and Jeffrey Pennington. 4+3 phases of compute-optimal neural scaling laws. *Advances in Neural Information Processing Systems*, 37:16459–16537, 2024.

Core Francisco Park, Ekdeep Singh Lubana, and Hidenori Tanaka. Competition dynamics shape algorithmic phases of in-context learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=XgH1wfHSX8.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon LLM: Outperforming curated corpora with web data only. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=kM5eGcdCzq.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Ankit Pensia, Shashank Rajput, Alliot Nagle, Harit Vishwakarma, and Dimitris Papailiopoulos. Optimal lottery tickets via subset sum: Logarithmic over-parameterization is sufficient. *Advances in neural information processing systems*, 33:2599–2610, 2020.

Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34:1256–1272, 2021.

Steven T Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21(5):1112–1130, 2014.

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

Shikai Qiu, Lechao Xiao, Andrew Gordon Wilson, Jeffrey Pennington, and Atish Agarwala. Scaling collapse reveals universal dynamics in compute-optimally trained neural networks. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Fvq9ogLnLN.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=GhVS8_yPeEa.

Rahul Ramesh. *The Principles of Learning on Multiple Tasks*. PhD thesis, University of Pennsylvania, 2025.

Rahul Ramesh, Jialin Mao, Itay Griniasty, Rubing Yang, Han Kheng Teoh, Mark Transtrum, James P Sethna, and Pratik Chaudhari. A picture of the space of typical learnable tasks. *arXiv preprint arXiv:2210.17011*, 2022.

Yunwei Ren, Eshaan Nichani, Denny Wu, and Jason D. Lee. Emergence and scaling laws in SGD learning of shallow neural networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=kA2H90nm26.

Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryenvpEKDr.

Dylan Sam, Sachin Goyal, Pratyush Maini, Alexander Robey, and J Zico Kolter. When should we introduce safety interventions during pretraining? *arXiv preprint arXiv:2601.07087*, 2026.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Aaditya Singh, Stephanie Chan, Ted Moskovitz, Erin Grant, Andrew Saxe, and Felix Hill. The transient nature of emergent in-context learning in transformers. *Advances in neural information processing systems*, 36:27801–27819, 2023.

Aaditya K Singh, Ted Moskovitz, Felix Hill, Stephanie C.Y. Chan, and Andrew M Saxe. What needs to go right for an induction head? a mechanistic study of in-context learning circuits and their formation. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=O8rrXl71D5.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*, 2024. URL https://huggingface.co/datasets/allenai/dolma.

Dashiell Stander, Qinan Yu, Honglu Fan, and Stella Biderman. Grokking group multiplication with cosets. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=hcQfTsVnBo.

Mihai Suteu and Yike Guo. Regularizing deep multi-task networks using orthogonal gradients. *arXiv preprint arXiv:1912.06844*, 2019.

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, 2024. URL https://arxiv.org/abs/2501.00656.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency. *arXiv preprint arXiv:2309.02390*, 2023.

Nikhil Vyas, Alexander Atanasov, Blake Bordelon, Depen Morwani, Sabarish Sainathan, and Cengiz Pehlevan. Feature-learning networks are consistent across widths at realistic scales. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=LTdfYIvbHc.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=yzkSU5zdwD.

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

Johnny Wei, Ameya Godbole, Mohammad Aflah Khan, Ryan Yixiang Wang, Xiaoyuan Zhu, James Flemings, Nitya Kashyap, Krishna P. Gummadi, Willie Neiswanger, and Robin Jia. Hubble: a model suite to advance the study of LLM memorization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZfdnZhOP0k.

Runzhe Wu, Ankur Samanta, Ayush Jain, Scott Fujimoto, Jeongyeol Kwon, Ben Kretzu, Youliang Yu, Kaveh Hassani, Boris Vidolov, and Yonathan Efroni. Imbalanced gradients in rl post-training of multi-task llms. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 3137–3150, 2026.

Zhengxuan Wu, Atticus Geiger, Aryaman Arora, Jing Huang, Zheng Wang, Noah Goodman, Christopher Manning, and Christopher Potts. pyvene: A library for understanding and improving PyTorch models via interventions. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pp. 158–165. Association for Computational Linguistics, June 2024. URL https://aclanthology.org/2024.naacl-demo.16.

Daniel Wurgaft, Ekdeep Singh Lubana, Core Francisco Park, Hidenori Tanaka, Gautam Reddy, and Noah Goodman. In-context learning strategies emerge rationally. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=bBUUOQI0N6.

Sang Michael Xie. *Foundation Models from a Data-Distributional View*. Stanford University, 2024.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36: 34201–34227, 2023.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36, 2024.

Huajian Xin, Z.Z. Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, , et al. Deepseek-prover-v1.5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=I4YAIwrsXa.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Yedi Zhang, Andrew M Saxe, and Peter E. Latham. Saddle-to-saddle dynamics explains a simplicity bias across neural network architectures. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Vit5M0G5Gb.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Tianyi Zhou, Deqing Fu, Vatsal Sharan, and Robin Jia. Pre-trained large language models use fourier features to compute addition. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=i4MutM2TZb.

A Limitations

As noted above, our account for why larger models learn more emphasizes the interplay of learning dynamics, task frequency, and task complexity. However, as discussed in Sec. 1, there are other plausible accounts for explaining this phenomenon, e.g., ones that focus on model expressivity and sample efficiency. Our explanation hence should not be interpreted as a complete account of scaling. Instead, these explanations are complementary: expressivity constrains what can be represented, sample efficiency shapes how effectively data is used, and our account highlights how learning dynamics interact with the frequency and complexity of tasks. A full understanding likely requires accommodating all these explanations, rather than viewing them as competing hypotheses. We also note we validated our key theoretical results using the OLMo pre-training pipeline, finding the empirical results on the injected tasks strongly match what the theoretical results predict. However, we acknowledge that empirical validation in a realistic pre-training setting could still leave some analytic gaps. For example, we did not empirically verify behavior of larger-scale language models or over-trained language models. We also selected injected tasks that matched the frequency of tasks learned in OLMo pre-training, which does not rule out other scaling behaviors with extreme task frequency. Our empirical results should therefore be viewed as supporting evidence. We encourage future work to explore different training regimes, more tasks, and different frequency ranges.

B Related Work

Multi-Task Learning. Data distributions neural networks are trained on are often deemed as a mixture of tasks (Raffel et al., 2020; Xie et al., 2023; 2024; Xie, 2024; Ramesh et al., 2022; Ramesh, 2025; Penedo et al., 2023; 2024; Soldaini et al., 2024; Maini et al., 2025; Sam et al., 2026; Goyal et al., 2024). This motivated works analyzing both the learning dynamics of training toy models on multi-task distributions and defining methods aimed at reducing interference between updates caused by learning a task in the presence of other ones. For example, the notion of "catastrophic interference" has been often characterized in the multi-task learning and continual learning literature (Caruana, 1997; Aljundi, 2019), where task gradients conflict or are imbalanced in scale, leading to learning of only a subset of tasks instead of the entire mixture. Such phenomenology can be intuitively (Liu et al., 2021; Aljundi et al., 2019; Wu et al., 2026) and theoretically explained: e.g., Pezeshki et al. (Pezeshki et al., 2021) posit the idea of gradient starvation, whereby a model trained on a mixture of tasks that have different prior frequencies is unable to learn the infrequent task due to its gradient getting "starved" out, i.e., becoming zero; meanwhile, Evron et al. (Evron et al., 2022) characterize how tasks' observation frequency induces the forgetting of another learned task in a sequential linear regression setting. Concurrent to our work, Marek et al. (2026) show that forgetting of prior tasks occur when a model has little remaining capacity. These analyses have also motivated methods to avoid interference and enable learning of multiple tasks: e.g., methods that perform "surgery" on model gradients (Yu et al., 2020; Sener & Koltun, 2018; Chen et al., 2018) to make two conflicting tasks' gradients to have zero interaction by removing one's projection towards another (Suteu & Guo, 2019; Farajtabar et al., 2020); these methods have seen use at scale as well (Chen et al., 2026).

It is worth noting that our results are in a similar vein as literature above, but augment prior work by characterizing the effects of scale and showcasing that even extremely rarely observed tasks can eventually be learned if one's model is large enough—empirically, related results corroborating our claim in a vision scenario was also made in the continual learning literature by Ramasesh et al. (Ramasesh et al., 2022). That said, we emphasize that neither do our results imply nor do we claim (in fact, we say otherwise) that scale alone is the mechanism to enable the learning of a rare task in the presence of other frequent ones. Indeed, methods discussed above from multi-task / continual learning literature show multiple tasks can be simultaneously learned by a model.

Memorization and Scaling. The core mechanism posited in our work for how larger models learn rare tasks involves a model retaining some signature of observed data from a small batch of samples. In extreme scenarios, e.g., when only a few samples are contained in the batch, such a signature cannot possibly correspond to a general, abstract task representation. Instead, the signature can be thought of as a model (at least partially) "memorizing" an observation—once enough observations occur and the memories aggregate, in our simple toy settings, we find the model consolidates the memories into an abstract representation that generalizes well. In this sense, we emphasize our core proposition suggests memorization is not an undesirable property, but instead a prerequisite to eventual generalization for rare tasks. This mechanism is highly reminiscent of the observations (Power et al.,

2022; Doshi et al., 2024; Gopalani et al., 2024) and posited learning dynamics for grokking, where a model transitions from memorizing observations to generalizing to novel inputs (Nanda et al., 2022; Murty et al., 2023; Kumar et al., 2024; Stander et al., 2024; Mohamadi et al., 2024; Varma et al., 2023; Morwani et al., 2024). Critically, our language model pretraining results, where we use modular addition, i.e., the prototypical grokking task, and find our posited dynamics hold is suggestive that grokking-like dynamics may in fact occur in practice, especially for rarely observed tasks (closest result to this end is perhaps the syntax acquisition dynamics demonstrated by Chen et al. (Chen et al., 2024)). This argument is in keeping with theoretical works on classification that have argued that memorization is necessary for generalization—for example, to handle label noise (Cheng et al., 2022), or to handle rare examples (Feldman, 2020). Brown et al. (2021) provide a particularly interesting demonstration that learning rare structures effectively requires memorizing even irrelevant information about the data. On the other hand, it is worth considering if an opposite mechanism may occur for frequently observed tasks: e.g., if a model sees too many observations of the same task, does it perhaps undergo phenomenology such as overfitting, which is generally associated with generalization to memorization dynamics; if so, does scaling help avoid this dynamic via mechanisms such as double descent (Mei & Montanari, 2022; Loog et al., 2020; Nakkiran et al., 2020; Nakkiran, 2019; Nakkiran et al., 2021)? Recent work on transient nature of in-context learning capabilities in toy scenarios (Wurgaft et al., 2026; Singh et al., 2024; 2023; Park et al., 2025) is suggestive such a dynamic may occur, and we thus argue it is worth investigating what the counterpart of our work for learning dynamics of frequent tasks looks like.

Building on the above, we also note memorization and the effects of scaling have been often studied in literature; these results are in line with our claims on reduced interference over model parameters via scaling, enabling models to eventually learn rare tasks. For example, studying memorization in the sense of verbatim match (e.g., k-token string match), works show larger models learn knowledge present in the tail-end of the distribution better (Kandpal et al., 2023), larger models (Lesci et al., 2024; Carlini et al., 2023) and later checkpoints tend to memorize more (Huang et al., 2024), not just individual data points but also training data order (Kuditipudi et al., 2026; Krasheninnikov et al., 2026), and these memories are retained for longer across injection events (Duan et al., 2025; Chang et al., 2024). Tirumala et al. (2022) show that larger models memorize more, but also can memorize more of the data before they begin to overfit.

Generalization and Scaling. Improved performance as a function of scaling has defined the spirit of machine learning since scaling laws first started being used for identifying training configurations (Hestness et al., 2017; Rosenfeld et al., 2020; Kaplan et al., 2020; Rae et al., 2021; Hoffmann et al., 2022; Henighan et al., 2020; Hernandez et al., 2022; 2021). The precise mechanism as to how scaling helps produce better models is unclear, but a few propositions have been made. For example, in works assessing how power-law scaling as a function of data and parameters emerges, prior work has exploited the argument that natural data statistics are heavy-tailed and follow power-law trends (e.g., Zipf-priors in language (Piantadosi, 2014) and vision (Hyvärinen et al., 2009)); correspondingly, scaling enables access to lower-order modes of the data distribution (Bahri et al., 2024; Maloney et al., 2022; Lin et al., 2024; Atanasov et al., 2024; Ren et al., 2026; Cagnetta et al., 2025b;a), and arguments to this end have been verified in recent work by Cagnetta et al. (Cagnetta et al., 2026). Our toy setup was in fact inspired by these papers, especially Ren et al. (Ren et al., 2026) and Maloney et al. (Maloney et al., 2022), but is a substantial simplification since, unlike these prior works, our goal was not to characterize the eventual steady state optima a model arrives at, but instead the dynamics that lead to it. Works closer to this dynamical motivation are by Bordelon et al. (Bordelon et al., 2024; 2025), Paquette et al. (Paquette et al., 2024), Atanasov et al. (Atanasov et al., 2024), and Everett et al. (Everett et al., 2024), who analyze learning dynamics of toy settings that exhibit power-law scaling curves. However, since we primarily aimed to posit a concrete mechanism via which larger models may be able to learn tasks smaller models do not, we note the concrete results emphasized and takeaways across these works versus ours are fairly different. In particular, these papers primarily focus on the interaction between learning dynamics and data statistics to identify different regimes of scaling, i.e., what functional form, e.g., power-law or otherwise, results in the best effective characterization of learning dynamics. Finally, works by Michaud et al. (Michaud et al., 2023; 2025) and Nam et al. (Nam et al., 2024) are fairly related to our paper: specifically, these works characterize an explicitly multi-task construction to posit a model for how power-law scaling can emerge in neural networks. While still related to the data statistics argument mentioned above, these works also have an explicit notion of prior frequency and (implicitly) show scaling helps learn tasks

that are rarely observed in the training distribution. Our work makes this claim explicit, but also characterizes how, i.e., a mechanism, via which scaling aids learning of rare tasks.

Lottery Tickets and Scaling. Another thread of research that partially connects the work listed above on scaling and learning of specific tasks is on the lottery ticket hypothesis (Frankle & Carbin, 2019): a lottery ticket is defined as a subnetwork identified from a larger, initial network that, even at random initialization, shows the ability to perform the task one is training their model for. Theoretical work on lottery ticket hypothesis has characterized bounds on how much larger a model has to be in order to possess a subnetwork that can, up to some error, approximate the model eventually learned via training (Malach et al., 2020; Pensia et al., 2020; Edelman et al., 2023). Especially related here is the work of Edelman et al. (Edelman et al., 2023), who show that via scaling model width (the scaling axis we consider as well), the odds that a subset of representations with non-trivial alignment with true task features exist substantially increases. Correspondingly, scaling improves sample efficiency of learning tasks that require more features (i.e., are more complex); critically, if one slightly generously interprets the authors' results, they are suggestive that a larger model will be able to learn rare tasks by virtue of already possessing features a smaller model will be unable to learn (due to sparsely observed training signal for such tasks). While this work partially informed the intuition guiding this paper, we note the eventual results for our setting and verification on large-scale scenarios are more concrete.

C Experimental Details

C.1 Synthetic Experiment

In the following, we describe experiment details and metrics relevant to results for the synthetic setup.

Data-generating process. All synthetic runs use the orthogonal-block instantiation of the mixture-of-regressions setup proposed in Sec. 3. We fix the ambient dimension at $D = 1024$, the number of tasks at $K \in \{16, 32\}$ (almost all figures use $K = 32$), and a per-task block dimension d_T such that $K \cdot d_T \leq D$ and the task blocks are mutually orthogonal. Concretely, task k occupies coordinates $[k\, d_T,\ (k{+}1)\, d_T)$ of $\mathbb{R}^D$, and its within-block spectrum is the power-law $\sigma_{k,j} = j^{-\alpha_k}$ for $j = 1, \ldots, d_T$. Unless stated otherwise we use a shared exponent $\alpha_k \equiv \alpha$ across tasks ($\alpha = 1$ in the orthogonal-block experiments, making the within-block decay slow enough that capacity reliably spreads beyond the leading mode of each task). The task prior is the power-law $\pi_k \propto k^{-\beta}$, normalized to sum to one over $k = 1, \ldots, K$; $\beta = 2$ in most experiments. Inputs are sampled fresh each step as $x \sim \mathcal{N}(0, \sigma_{\text{in}}^2 I_D)$ with $\sigma_{\text{in}} = 1$ in all orthogonal-block runs. The per-task targets are $y_k = \Lambda_k^{1/2} B_k^\top x$, restricted to the task's block; because each task block has rank d_T, the output dimension of the regressor is d_T (and reduces to 1 when $d_T = 1$, e.g., in the rank-1 specialization of App. F.1.2).

Model. The student is the linear-bottleneck regressor of Sec. 3: a shared encoder $W \in \mathbb{R}^{N \times D}$ that maps the input to an N-dimensional hidden, followed by per-task linear decoders $D_k \in \mathbb{R}^{d_T \times N}$ selected by the ground-truth task index supplied in the batch. We do not explicitly constrain W to have orthonormal rows: the relevant object for Theorem 3 is the projector $P_W = W^\top (W W^\top)^{-1} W$, which is invariant to the right-multiplicative gauge of W and which gradient flow drives toward the top-N eigenspace of $M = \sum_k \pi_k C_k$ regardless of the parametrization. The encoder is initialized such that $W^\top W = I_N$ at step zero, and the per-task decoders are initialized with Kaiming-uniform fan-in / linear gain. The decoders are jointly optimized with the encoder rather than analytically closed-formed at each step, since learned decoders will converge to $D_k^* = \Lambda_k^{1/2} B_k^\top U$ at any stationary point, so the joint optimization does not change the encoder fixed point but does match the practical setting in which both ends of the bottleneck are learned simultaneously.

Optimizer. We use AdamW with default hyperparameters and an inverse-square-root learning-rate schedule. Gradients are clipped at maximum norm 1.0. Batches are drawn fresh each step (no fixed dataset, no replay) with batch size $B = 1\,024$ for the phase-diagram and rank-1 sweeps, and $B = 512$ for the matched-frequency retention sweeps; the smaller batch in the retention runs is required so that an injection batch with $m \leq B$ rare-task slots can match the long-run frequency $\rho_r = m/(G \cdot B)$ at the $\rho_r \approx 6 \times 10^{-4}$ end of the sweep.

Metrics. We track three families of metrics, all reported on freshly sampled batches separate from the training stream. The first is the *per-task loss*, i.e., the unnormalized population MSE $\ell_k(U) = \mathbb{E}\big[\|y_k - D_k U^\top x\|_2^2\big]$, and its normalized counterpart $\ell_k(U)/\ell_{k,\text{baseline}}$ with $\ell_{k,\text{baseline}} = \|a_k\|_2^2/d_T$ the mean-predictor MSE per task. The second is the *per-task subspace alignment*, the basis-free quantity $s_k(U) = \text{Tr}(P_U C_k)/\text{Tr}(C_k) = \|P_U a_k\|_2^2/\|a_k\|_2^2$, computed via the SVD of W so that it is independent of the gauge of the encoder. s_k lies between N/D at random initialization and 1 when the task block is fully captured. We also report its random-baseline-corrected normalization $\tilde{s}_k(U) = (s_k(U) - N/D)/(1 - N/D)$, which equals 0 at random initialization and 1 at full capture. The third is the *residual common-task signal*: we compute $\delta_F(U) = \sum_{k \in F} \pi_k\, (1 - s_k(U))\, \|a_k\|_2^2$, the residual energy of the frequent block. The frequent set F is the smallest top-prior set with cumulative mass at least 0.8; under our power-law prior this yields $|F| = \{6, 3, 2, 2\}$ for $\beta \in \{0.5, 1.0, 1.5, 2.0\}$ respectively. Standard evaluation is performed every 1 000–2 000 steps on a held-out probe of the same population distribution, with the final checkpoint additionally re-evaluated for end-of-training summary statistics.

C.2 OLMo Pretraining Pipeline

Table 1: Model configurations by size.

Model Name	# Parameters	# Layers	Hidden Dim	MLP Dim	# Attn Heads
4M	6,963,200	8	64	512	8
20M	28,753,920	16	192	1,536	8
300M	371,458,048	16	1,024	8,192	16
1B	1,279,787,008	16	2,048	16,384	16
4B	4,707,057,664	16	4,096	32,768	32

Models. We use the OLMo model architecture (Groeneveld et al., 2024). For 4M to 1B models, we follow the model configuration and naming convention of Magnusson et al. (2025). We additionally include a 4B model to further evaluate width scaling, as shown in Table 1.

Training hyperparameters. We use the same batch size of 1024, window size of 4096 for T_{CMP} and 1024 for T_{ADD}, and a learning rate schedule with an initial learning rate of 3×10^{-4} and cosine with warmup schedule for all models. For the retention window ablation experiment, we use a smaller window size of 512 to reduce the training cost. For a full list of hyperparameters, refer to the `OLMo-7B-0724` configuration.[5]

Training pipeline. We use the OLMo code base.[6] Our usage is in line with its Apache-2.0 license.[7]

Compute resources. All models are trained on a cluster of NVIDIA H200 GPUs.

C.3 Pre-training and Injected Task Data

Pre-training data. We use Dolma v1.7 as the pre-training corpus (Soldaini et al., 2024). Specifically, we use the 210B tokens corresponding to the first 50K batches that `OLMo-7B-0424` and `OLMo-7B-0724` are trained on, in the exact same order.

Reference tasks. To ensure the injected task frequency is comparable to the frequency of tasks learned in pre-training, we sample two reference tasks R_{cmp} and R_{add} from pre-training that involve similar high-level functions. R_{cmp} predicts a number larger than `x` in the prompt "it has increased from `{x}` to". R_{add} predicts the sum of two numbers smaller than 100 with the prompt "`{x}` + `{y}` =". We estimate the lower bound of their frequency in pre-training data using infini-gram (Liu et al., 2024) and observe models' next token prediction loss on the task, which corresponds to the two dashed lines in Fig. 4 panel (a).

Injected tasks. We elaborate the task label here. Let $val(\cdot) : \mathcal{S} \mapsto [0, 99]$ be a bijective mapping that assigns an integer value between 0 and 99 to each token. For T_{CMP}, `LABEL` is one of two tokens randomly chosen from the vocab indicating whether $val(\texttt{TOK1}) < val(\texttt{TOK2})$. For T_{ADD}, `LABEL` is the token in $\mathcal{S}$ whose value equals $(val(\texttt{TOK1}) + val(\texttt{TOK2}))\ mod\ 100$. Below are a few instances from the comparison task: `address analyze pony`, `resort zebrafish pony`, `cavities misconduct provisional`, where `pony` and `provisional` are the two label tokens that represent `True` and `False`.

[5] https://github.com/allenai/OLMo/blob/main/configs/official-0724/OLMo-7B-0724.yaml
[6] https://github.com/allenai/OLMo
[7] https://github.com/allenai/OLMo?tab=Apache-2.0-1-ov-file

C.4 Localizing and Measuring Task Features in Sec. 4.3

The comparison task T_{CMP}. We first use distributed alignment search (DAS) to verify that the model's prediction is causally dependent on a global token order feature, which is encoded in a 1-D subspace in the residual stream of the first few layers.

The task has a simple high-level causal model, namely $X \rightarrow O \leftarrow Y$, where X, Y are the two inputs and O is the binary output. We consider the following intervention on the input variable X (or Y): Let a base example be x_b, y_b, o_b and a source example be x_s, y_s, o_s, an interchange intervention on X that sets the value of x_b to x_s should lead to a counterfactual label that corresponds to $x_s < y_b$.

In the neural model, we search for a low-dimensional subspace in the residual stream that plays the same causal role as the input variable X by training on $1K$ counterfactual data pairs defined above. We use DAS to search across all layers above the input token position. We are able to find a 1-D subspace in the residual stream of the first layer that has an interchange intervention success rate of 96%. This proves that the model not only encodes the global token order in a low-dimensional space but actually uses this feature for prediction on the task T_{CMP}. This allows us to use the global order feature to measure to what extent a model has learned the abstract task structure.

We use the DAS implementation from `pyvene` (Wu et al., 2024).

The modular addition task T_{ADD}. As prior work studying modular addition has identified that grokked models use Fourier modes for addition (Nanda et al., 2022), we conduct Fourier analysis on residual stream to measure the presence of Fourier modes.

For modulus P, define a real discrete Fourier transform basis on $\mathbb{R}^P$ as follows:

$$\phi_k^{\cos}(n) = \frac{\cos(2\pi kn/P)}{||\cos(2\pi k \cdot /P)||}, \quad \phi_k^{\sin}(n) = \frac{\sin(2\pi kn/P)}{||\sin(2\pi k \cdot /P)||}, \quad k = 1, \ldots, \left\lfloor \frac{P}{2} \right\rfloor.$$

At layer l, collect residual stream vectors $h^l \in \mathbb{R}^d$ grouped by output `c = (a+b)` $\bmod$ `P` and compute the mean representation as:

$$v_c^{(l)} = \mathbb{E}\big[h^{(l)} \mid c = (a+b) \bmod P\big] \in \mathbb{R}^d, \quad c \in \{0, \ldots, P-1\}.$$

This yields a matrix $V^{(l)} \in \mathbb{R}^{P \times d}$.

After row-centering $V^{(l)}$, the fraction of variance captured by frequency k is

$$P_k^{(l)} = \frac{\sum_{j=1}^d \left(\langle \phi_k^{\cos}, V_{:,j}^{(l)} \rangle^2 + \langle \phi_k^{\sin}, V_{:,j}^{(l)} \rangle^2 \right)}{\sum_{j=1}^d ||V_{:,j}^{(l)} - \bar{V}_{:,j}^{(l)}||^2}, \quad \sum_k P_k^{(l)} = 1$$

We consider a null-baseline for detecting Fourier modes in representations. Under uniform variance allocation across frequencies,

$$P^{\text{null}} = \frac{2}{P-1}$$

Hence, a frequency k at layer l is identified as a Fourier mode if

$$P_k^{(l)} > \theta P^{\text{null}}$$

We choose $\theta = 2$ in our experiment, as we do not observe significant difference in grokking behavior for models that represent Fourier modes with stronger signals, e.g., $\theta = 3$.

Finally, we define the total number of detected modes as follows, where L is the total number of layers.

$$N_{\text{features}} = \sum_{l=0}^{L} \left| \left\{ k : P_k^{(l)} > 2P^{\text{null}} \right\} \right|$$

This corresponds to the y-axis in Fig. 5 (b) right panel.

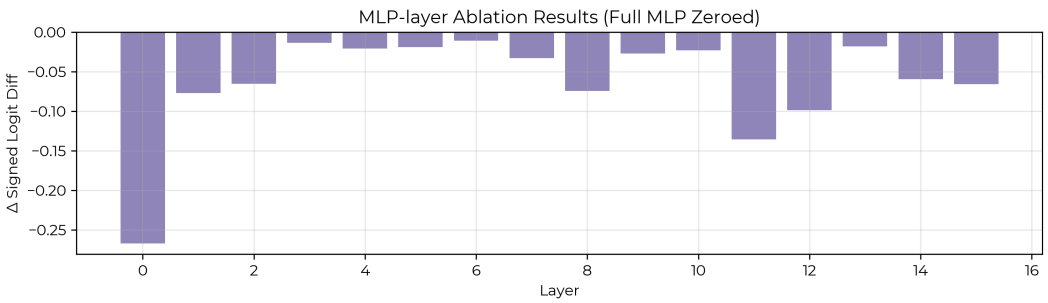

Figure 7: The first MLP layer has the strongest causal effects on the model's logits prediction.

C.5 Localizing Task Neurons in Sec. 4.4

We conduct null interventions on MLP layers to identify layers that have the largest causal effects on the model output. The results are shown in Fig. 7. This aligns with our observation from the DAS localization experiment that first layer is the earliest layer where the global token order is causally encoded. The localization result is consistent across the three models used in Sec. 4.4.

D PROOFS

D.1 PROOF OF THEOREM 3

For fixed U, the task-k population loss is

$$\ell_k(U, D_k) = \mathbb{E}\big[\|\Lambda_k^{1/2} B_k^\top x - D_k U^\top x\|_2^2\big] \tag{4}$$

$$= \|\Lambda_k^{1/2} B_k^\top - D_k U^\top\|_F^2, \tag{5}$$

where the second identity uses $x \sim \mathcal{N}(0, I)$ and the standard relation $\mathbb{E}\|Ax\|_2^2 = \|A\|_F^2$. This is a linear least-squares problem in D_k, so the minimizer is

$$D_k^* = \Lambda_k^{1/2} B_k^\top U. \tag{6}$$

Substituting back gives

$$\ell_k(U) = \big\|\Lambda_k^{1/2} B_k^\top (I - P_U)\big\|_F^2 \tag{7}$$

$$= \mathrm{Tr}\Big(\Lambda_k^{1/2} B_k^\top (I - P_U) B_k \Lambda_k^{1/2}\Big) \tag{8}$$

$$= \mathrm{Tr}\big((I - P_U) C_k\big). \tag{9}$$

Summing with weights π_k yields

$$L_N(U) = \sum_{k=1}^K \pi_k \ell_k(U) = \mathrm{Tr}(M) - \mathrm{Tr}(U^\top M U), \qquad M := \sum_{k=1}^K \pi_k C_k. \tag{10}$$

Because M is symmetric positive semidefinite with finite trace, minimizing $L_N(U)$ is equivalent to maximizing $\mathrm{Tr}(U^\top M U)$ over all orthonormal U. By Ky Fan's maximum principle (Fan, 1949),

$$\max_{U^\top U = I_N} \mathrm{Tr}(U^\top M U) = \sum_{i=1}^N \mu_i, \tag{11}$$

where $\mu_1 \geq \mu_2 \geq \cdots$ are the eigenvalues of M. Therefore any minimizer spans the top-N eigenspace of M and the optimal loss is

$$L_N^* = \mathrm{Tr}(M) - \sum_{i=1}^N \mu_i = \sum_{i>N} \mu_i. \tag{12}$$

For our generative process, we have

$$M = \sum_{k,j} \pi_k \lambda_{k,j}\, b_{k,j} b_{k,j}^\top, \tag{13}$$

so the vectors $b_{k,j}$ are eigenvectors of M with eigenvalues $u_{k,j} = \pi_k \lambda_{k,j}$. Thus the width-$N$ optimum keeps the N largest utilities. If task k contributes $n_k(N)$ retained coordinates, then its residual loss is

$$\ell_k^*(N) = \sum_{j > n_k(N)} \lambda_{k,j}. \tag{14}$$

D.2 PROOF OF THEOREM 4

Write $G_{\mathsf{F}}(U) = 2(I - P_U) M_{\mathsf{F}}^{1/2} M_{\mathsf{F}}^{1/2} U$. Using $\|AB\|_F \leq \|A\|_F \|B\|_{\mathrm{op}}$, where $\|.\|_{\mathrm{op}}$ denotes the operator norm, gives

$$\|G_{\mathsf{F}}(U)\|_F \leq 2\, \|(I - P_U) M_{\mathsf{F}}^{1/2}\|_F\, \|M_{\mathsf{F}}^{1/2} U\|_{\mathrm{op}} \tag{15}$$

$$\leq 2\sqrt{\mathrm{Tr}\big((I - P_U) M_{\mathsf{F}}\big)}\, \sqrt{\lambda_1(M_{\mathsf{F}})} \tag{16}$$

$$= 2\sqrt{\lambda_1(M_{\mathsf{F}})\, \delta_{\mathsf{F}}(U)}. \tag{17}$$

D.3 Proof of Proposition 6

Let U denote the common-task width-N solution and let u_i be one occupied common eigenvector with eigenvalue μ_i^{F}. Replace that vector by

$$v_i(\theta) = \cos\theta\, u_i + \sin\theta\, b_r, \tag{18}$$

while keeping the remaining $N-1$ directions fixed. Because u_i is an eigenvector of M_{F} and b_r is orthogonal to the common block, the contribution of this one direction to the objective $\mathrm{Tr}(P_U M)$ is

$$\langle v_i(\theta), M v_i(\theta)\rangle = \mu_i^{\mathsf{F}} \cos^2\theta + \pi_r\lambda_r \sin^2\theta. \tag{19}$$

Subtracting the value at $\theta = 0$ gives

$$\Delta\,\mathrm{Tr}(P_U M) = (\pi_r\lambda_r - \mu_i^{\mathsf{F}})\sin^2\theta. \tag{20}$$

Since $L = \mathrm{Tr}(M) - \mathrm{Tr}(P_U M)$, the loss change is

$$\Delta L = (\mu_i^{\mathsf{F}} - \pi_r\lambda_r)\sin^2\theta. \tag{21}$$

Hence perturbations toward b_r decrease the loss if and only if $\pi_r\lambda_r > \mu_i^{\mathsf{F}}$. The rare feature invades first through the weakest occupied common direction, which has curvature μ_N^{F}, proving the stated threshold.

D.4 Microscopic competition in a one-neuron, two-task model

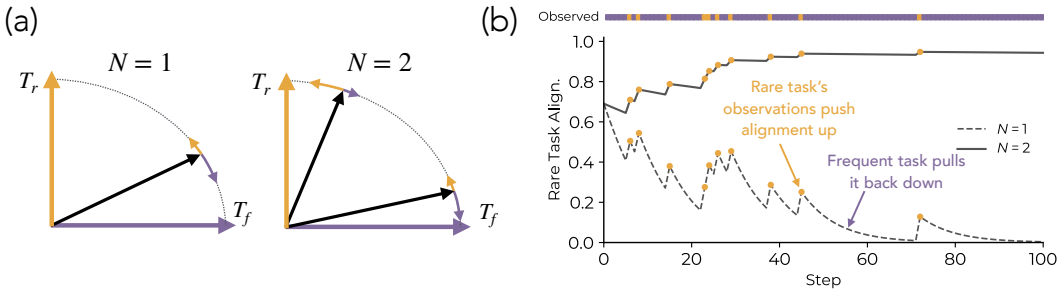

Figure 8: **Competition Dynamics over Neurons.** Rare task alignment over training for a softmax-gated model of 1 vs. 2 neurons. (a) Two orthogonal task directions T_f (frequent, sampled with probability 0.9) and T_r (rare, probability 0.1) compete for neurons. (b) With a single neuron, the frequent task dominates; with two neurons, one neuron specializes to each task, allowing rare task alignment to reach and sustain values near 1.

Example 7 (One neuron, two orthogonal tasks)**.** *Let $a, b \in \mathbb{R}^d$ be orthonormal and consider rank-one tasks with covariances $C_a = aa^\top$ and $C_b = bb^\top$. A width-1 encoder is a unit vector*

$$u = \cos\theta\, a + \sin\theta\, b. \tag{22}$$

The task losses are

$$\ell_a(u) = \sin^2\theta, \qquad \ell_b(u) = \cos^2\theta. \tag{23}$$

A gradient step on task a obeys $\theta^+ = \theta - \eta\sin(2\theta) + O(\eta^2)$, while a step on task b obeys $\theta^+ = \theta + \eta\sin(2\theta) + O(\eta^2)$. If task a appears with probability p and task b with probability $q < p$, then

$$\mathbb{E}[\Delta\theta \mid \theta] = \eta(q-p)\sin(2\theta) + O(\eta^2), \tag{24}$$

which drives the neuron toward the common task. Near $\theta = 0$, if a rare-task update is followed by G common-task updates, then

$$\theta_G \approx (1-2\eta)^G\theta_0 \approx e^{-2\eta G}\theta_0. \tag{25}$$

Thus rare-task alignment decays exponentially across the gap between rare observations.

Proof. The loss identities follow from $u^\top aa^\top u = \cos^2\theta$ and $u^\top bb^\top u = \sin^2\theta$. Differentiating yields

$$\frac{d}{d\theta}\sin^2\theta = \sin(2\theta), \qquad \frac{d}{d\theta}\cos^2\theta = -\sin(2\theta), \tag{26}$$

which gives the stated updates under gradient descent. Taking the expectation under the task mixture yields the drift formula. Linearizing $\sin(2\theta) \approx 2\theta$ near zero gives the exponential decay estimate. □

The dynamics posited above are also exemplified in Fig. 8.

E Further Experimental Results: Complexity Sweeps

In the main paper, we kept "complexity", i.e., the number of directions used for defining the target variable constant across tasks; specifically, tasks in the main paper require 5 directions to cover 90% of the energy in the task spectrum (i.e., solving for r in $\arg\min_r \frac{\sum_{j=1}^{j=r} \lambda_{k,j}}{\sum_j \lambda_{k,j}} > 0.90$ gives $r = 5$). In this section, we vary this property by changing the power-law coefficient underlying the task spectrum, i.e., since $\lambda_{k,j} \propto j^{-\alpha_k}$, we vary the range of α_k across tasks. We define ranges of $[\alpha_{\text{min}}, \alpha_{\text{max}}]$, split the range uniformly into K values, and assign the kth value to α_k. The most frequent task is assigned the value α_{max}, giving it the fastest decaying spectrum and hence making it the simplest task, while the rarest task is assigned the value α_{min}, giving it the slower decaying spectrum and making it most complex. In particular, we choose ranges (see Fig. 9) such that the task complexity varies between $[4, 7]$ and $[2, 12]$ across $K = 32$ tasks.

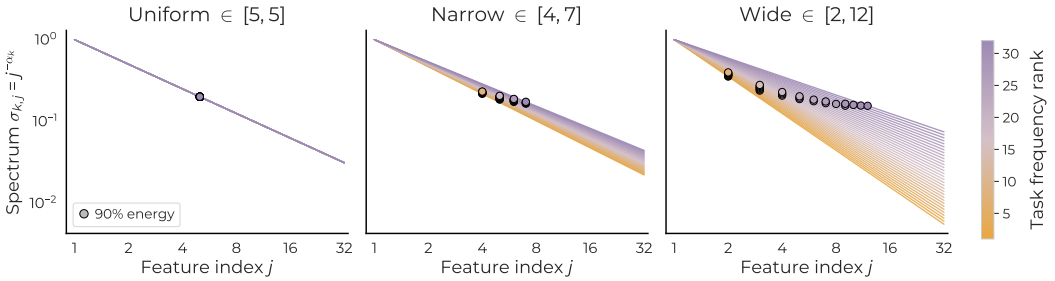

Figure 9: **Task Spectra.** We use power-law task spectra to vary the complexity of a task in our experiments, i.e., the jth feature contributes signal proportional to $j^{-\alpha_k}$ for the kth task. While the main paper studies the setting with uniform values for α_k, hence making frequency the core knob for varying utility, we now vary complexity by splitting a range of α values; this results in task spectra such that the number of directions to cover 90% of task signal now takes 4–7 directions for the "narrow" range scenario, while 2–12 directions for the wider range scenario.

E.1 Feature Utility Predicts Learning Order

We first reproduce Fig. 2. We split the figure into two parts, showing the critical width boundary as a function of task frequency in heatmaps in Fig. 10 and the per-task loss predictability based on feature utilities in Fig. 11; for reference, we include our baseline results from the main paper, where the task spectra were uniform.

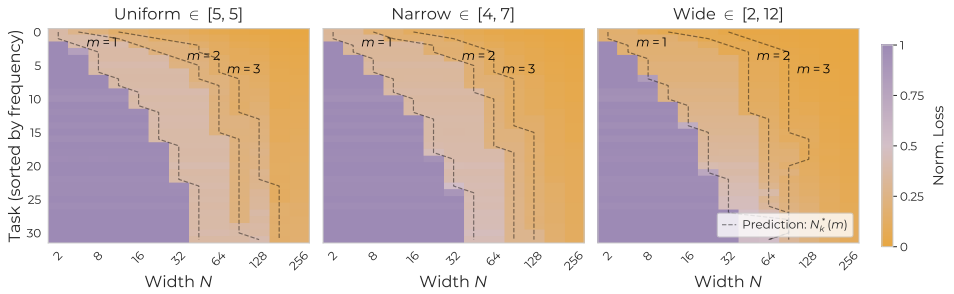

Figure 10: **Learning Phases Under Varying Complexity.** Reproduction of Fig. 2a under varying task spectra. We see increase in the complexity gap leads to higher emphasis on the top two modes' learning, since under a power-law spectrum decay, the eigenvalue associated with larger modes will be small. More critically, learning order is now not monotonically predicted by frequency alone: this is most easily visible in the results for wide complexity range scenario, where we see the "most complex" task's third mode is in fact high enough utility to get learned before more frequent task's higher order modes, resulting in a non-monotonic boundary.

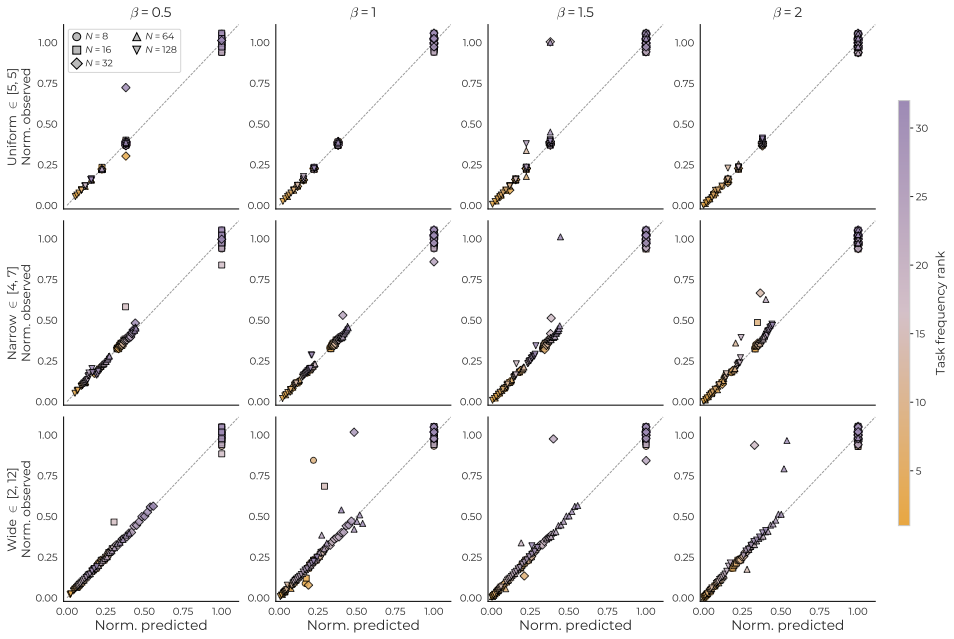

Figure 11: **Feature Utilities Continue to Predict Learning.** Reproduction of Fig. 2b under varying task spectra. While Fig. 10 shows frequency, by itself, is insufficient to predict learning of a task, the current plot shows the empirically observed loss and the loss derived out of the assumption that N neurons will learn top N utility features continues to align well. This confirms the learning dynamic in the non-uniform complexity scenarios requires accounting for both frequency and complexity: higher-frequency tasks *may be learned after* a lower-frequency task if the complexity of the former is more than the latter.

E.2 Competition Dynamics Disallow Learning of the Rarest and the Most Complex Task

Our results above showed that learning trends under varying task complexity are modulated by both task frequency and complexity, as predicted by our account in Sec. 3. We now show the competition dynamics picture posited in that section continues to follow in these settings as well. In particular, we plot the learning of the top-3 most frequent tasks (measured by normalized signal; see Sec. 3 for details) and the rare-most task as a function of residual, i.e., signal remaining to be explained in the frequent tasks. As shown in Fig. 12, the critical width predicted to be necessary for learning of the rarest task, by rendering the residual sufficiently small for most frequent tasks, continues to hold true in this setup as well.

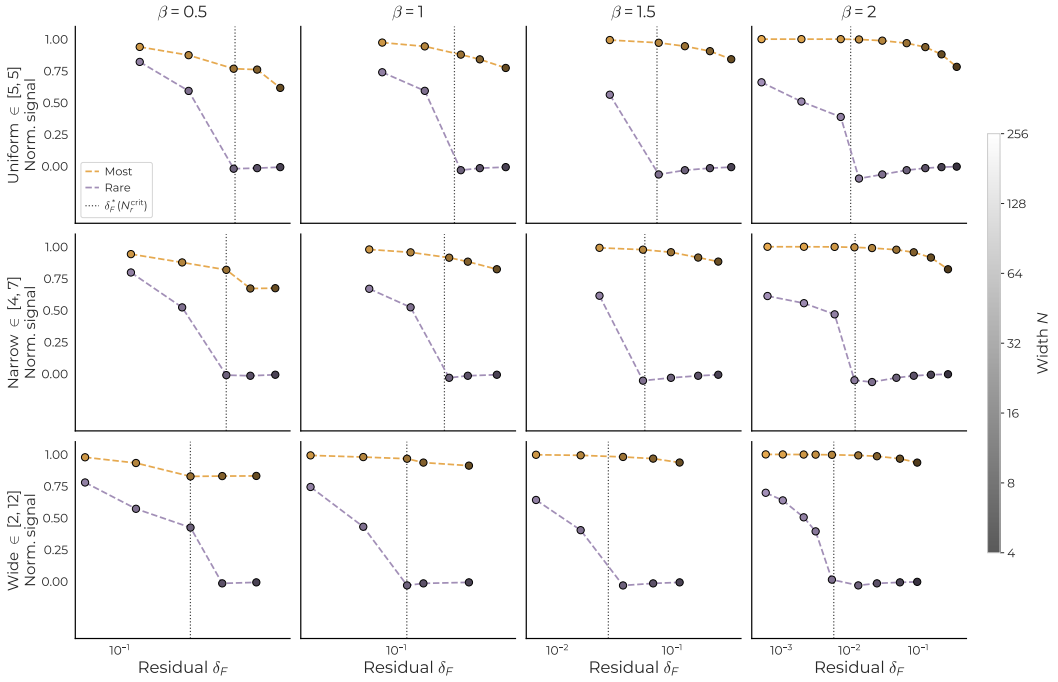

Figure 12: **Complexity Residual.** We plot the amount of signal encoded in model representation for most frequent and rarest tasks as a function of width N and remaining residual δ_{F}. In line with our predictions, we see larger models perfectly capture tasks of all frequencies, while smaller models do not. Meanwhile, even for the largest models, when the residual signal remaining to explain for frequent tasks is high, rarer tasks struggle to be learned.

E.3 Reduced Interference Aids Learning of the Rarest and the Most Complex Task

We now validate our argument for how data-centric bottlenecks, i.e., the low-frequency and high-complexity nature of a task, is circumvented by a larger model: by virtue of having more parameters, a larger model witnesses reduced interference over per-task gradients. To this end, we redo the batch-injection experiments from Fig. 3 and plot the retention dynamics for the lowest aggregate utility task across settings. Results are shown in Fig. 13. We see similar results as before: larger models show better retention of observed signal from a task, allowing them to bootstrap on these past observations and eventually learn the task. Meanwhile, a medium width model is able to do so only when the task is observed sufficiently frequently, i.e., the gap is low. Comparing with the case when the width is too low, we see the model never learns the task and the retention dynamics concretely show why: the model is unable to retain signal for the observed task for long enough.

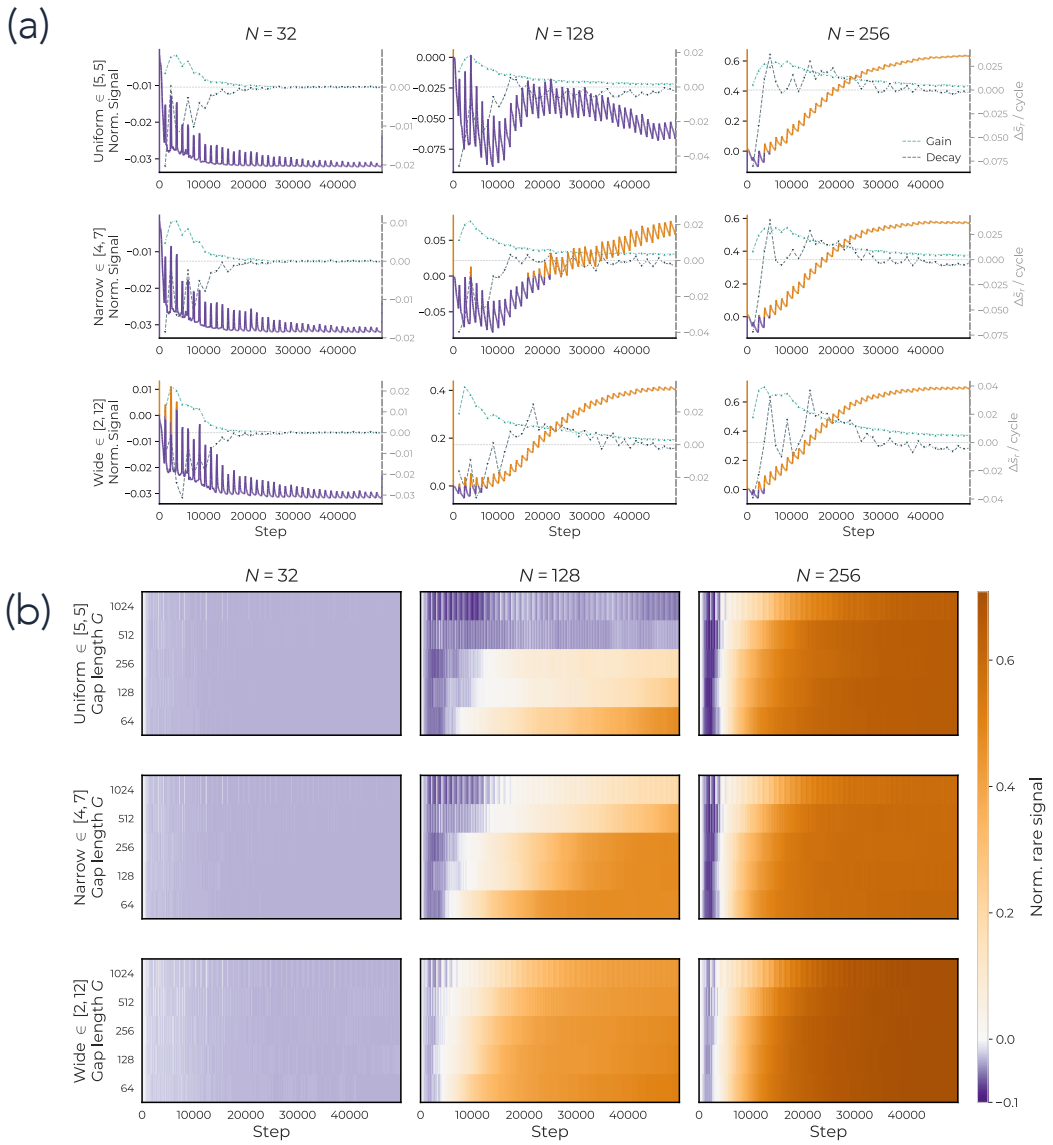

Figure 13: **Complexity Retention Phases.** We isolate retention by training with a matched-frequency injection protocol: the lowest-total utility task is withheld for G steps and then reintroduced in a batch such that its overall frequency is consistent across settings. (a) Training dynamics for $G = 1280$. We see small models briefly encode the rare task (Norm. signal $\tilde{s_r}$: left-y axis) after each injection; specifically, $\Delta\tilde{s_r}$ increases at point of injection, as shown by green dotted line ('gain'). However, as frequent-task updates resume, this signal is quickly lost ('decay': gray dotted line). Meanwhile, larger models retain more of the rare-task signal between injections and accumulate it over training. (b) Across injection gaps G and widths N, rare-task signal decays rapidly in narrow models but remains stable in wider models, while frequent-task signal is largely unaffected. These results support the reduced-interference mechanism: scaling provides enough representational capacity that updates from frequent tasks no longer overwrite rare-task features before the next rare observation arrives.

F Further Experimental Results: Frequency Sweeps

F.1 Features and Tasks are Learned in Order of Utility

F.1.1 Extending Phase Diagram

We sweep the power-law exponent β defining the task prior by varying it over $\{0.5, 1.0, 1.5, 2.0\}$. Unlike Fig. 2, we only sweep 5 values of widths—specifically, $N \in \{8, 16, 32, 64, 128\}$. Correspondingly, the staircase is rendered at lower resolution. Results are shown in Fig. 14.

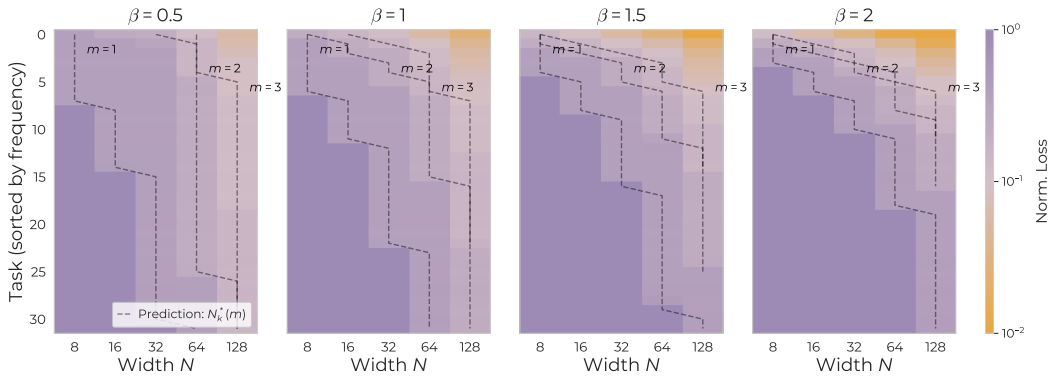

Figure 14: **Feature Utility Predicts Order of Learning.** We extend results from Fig. 2 by analyzing values $\beta \in \{0.5, 1.0, 1.5, 2.0\}$. Each panel reports the normalized per-task loss, i.e., $\ell_k(N)/\ell_{k,\text{baseline}}$ as a function of width N. Tasks are sorted top-to-bottom by descending prior frequency so the most-frequent task occupies the top row. Dashed staircases are the theoretical thresholds $N_k^*(m)$ for $m = 1, 2, 3$ computed from the per-direction utility ordering of Theorem 3. The empirical learned region (orange) tracks the $m = 1$ staircase across all four prior skews; deeper-orange cells in the steeper-prior panels ($\beta = 1.5, 2$) reflect the model spending its width budget on additional directions of the leading tasks rather than on rarer tasks, in agreement with the account posited in the main paper for how scaling interacts with data properties.

F.1.2 Simplified Case: Rank-1 Tasks

Theorem 3 predicts that a width-N minimizer retains the N task-features with largest utility $u_{kj} = \pi_k \lambda_{kj}$. We obtain a sharp quantitative test of this claim by collapsing the orthogonal-block setup to its rank-1 specialization: setting $d_T = 1$ and $\alpha_k = 1$ makes every task rank-1 with $\lambda_k = 1$, so the utility ordering reduces to the prior ordering, and the predicted critical width for task k becomes

$$N_{\text{crit}}(k) \;=\; \#\{j \neq k : \pi_j \lambda_j > \pi_k \lambda_k\} \;=\; k, \tag{27}$$

i.e., a perfectly linear staircase in task index.

Setup. We train the linear-bottleneck student described in Sec. 3 on a mixture of $K = 32$ rank-1 orthogonal tasks, ambient dimension $D = 1024$, and a power-law prior with exponent $\beta = 2$. We sweep the encoder width $N \in \{1, 2, 3, 4, 6, 8, 10, 12, 14, 16, 20, 24, 28, 32, 40, 48, 64\}$ and read out the per-task subspace alignment $s_k(U) = \|P_U b_k\|^2$ at the end of training (rank-1 specialization of the per-task signal of Sec. 3, so $s_k(U)$ lies between N/D at random initialization and 1 when b_k is fully captured by the encoder subspace).

Result. See Figure 15. We overlay the empirical transition width $N_{\text{emp}}(k) = \min\{N \in \text{grid} : s_k(U) > 0.5\}$ on the theoretical staircase (27), finding almost perfect alignment (minimal disparities are an artifact of the sampled width grid).

F.2 Residual Controls Learning

We next provide further validation for results in Fig. 3 by repeating experiments across different values of exponents in the power-law task prior. Specifically we vary β from the set $\{0.5, 1.0, 1.5, 2.0\}$. For each run we compute the per-task signal $s_k(U) = \text{Tr}(P_U C_k)/\text{Tr}(C_k)$,

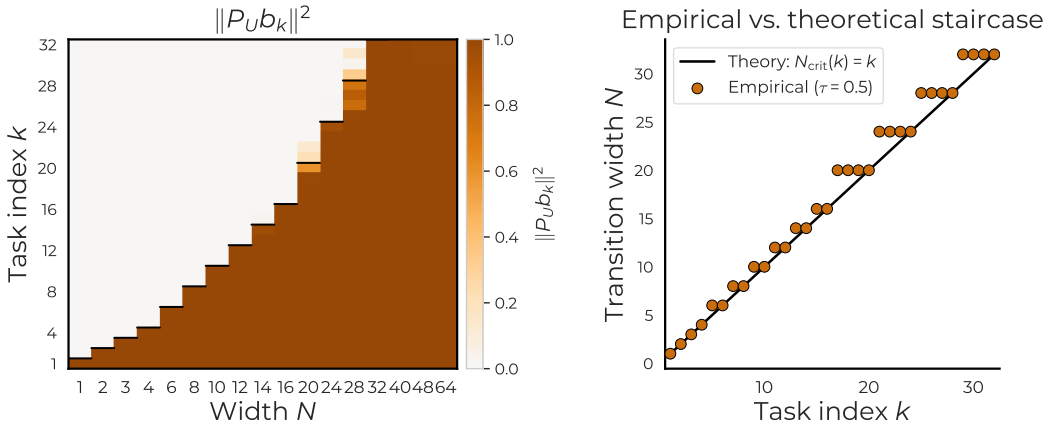

Figure 15: **Rank-1 Verification of Utility Predicting Learning Order. Left:** Per-task subspace alignment $\|P_U b_k\|^2$ at the end of training as a function of width N and task index k. By our account, we expect tasks $1 \dots N$ to be retained, while tasks $N+1 \dots K$ are not retained. The black step segments in the heatmap mark the predicted retention horizon $k = N$ per width. Results align well with our expectations. **Right:** Empirical transition width $N_{\text{emp}}(k)$ (markers) sits on the theoretical staircase $N_{\text{crit}}(k) = k$ (line) within the resolution of the width sweep. Plateaus at $k = 5, 7, 9, \dots$ reflect the gap in sampled widths between adjacent grid points and are not deviations from theory.

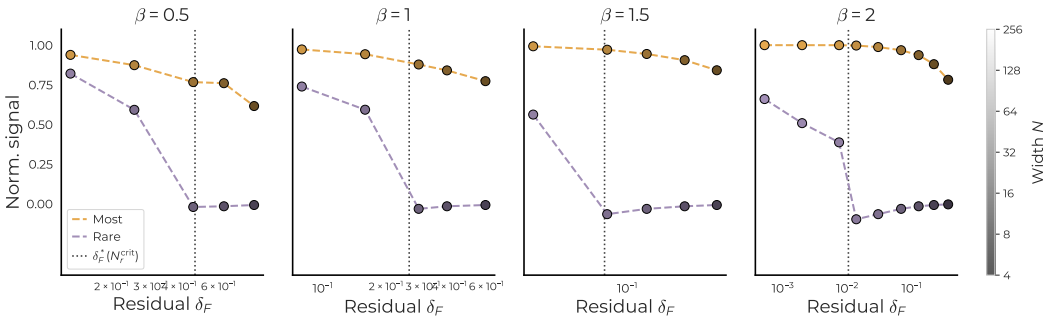

Figure 16: **Residual Controls Rare-Task Learning**. We vary $\beta \in \{0.5, 1.0, 1.5, 2.0\}$. Each panel reports the normalized rare-task and most-frequent-task signal as a function of the frequent-task residual δ_F, with width N encoded by marker brightness (dark = small N, bright = large N; see grayscale colorbar on the right). Dashed vertical line marks the analytic threshold $\delta_F^*(N_r^{\text{crit}})$ computed from Corollary 5 under that panel's β. The two-phase dynamic predicted by Theorem 4 and seen in Fig. 3 is preserved across all four prior skews; the shift in the threshold's location with β is exactly what theory predicts.

i.e., how well the model representation encodes information about the kth task, and the residual $\delta_F(U) = \sum_{k \in F} \pi_k \left(1 - s_k(U)\right) \|a_k\|^2$ from the final checkpoint. The frequent set F is defined as the smallest set of tasks whose cumulative mass meets 0.8; under $\beta \in \{0.5, 1, 1.5, 2\}$ this yields $|F| = \{6, 3, 2, 2\}$ respectively, reflecting how a flatter prior spreads the loss budget across more frequent tasks. Results are reported in Figure 16. We see a precise kink similar to Fig. 3 when rare-task signal drops to zero once δ_F exceeds the analytic threshold $\delta_F^*(N_r^{\text{crit}})$, and rises steeply to near-unity once δ_F falls below it. We also see the threshold itself shifts left as β grows: a steeper prior makes the rare task's leading utility $\pi_r \lambda_r$ much smaller, so a smaller residual is required to "free up" encoder directions that can then capture the rare task.

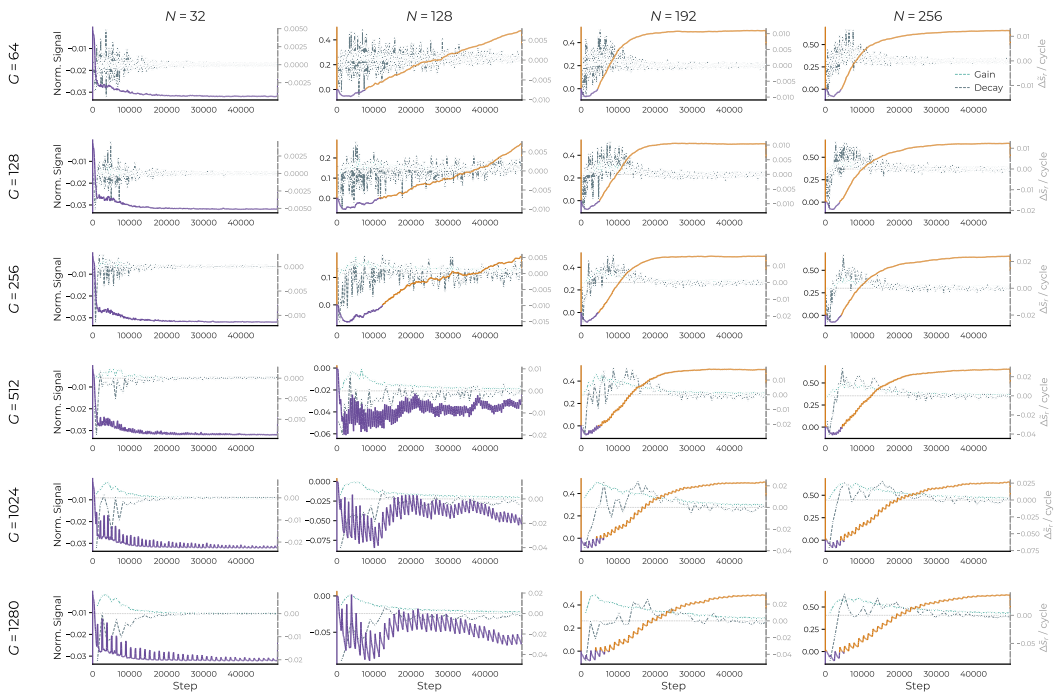

Figure 17: **Per-gap dynamics: Reproducing retention results across different injection gaps and widths.** We vary the injection gap G in the set $\{64, 128, 256, 512, 1024, 1280\}$ (top to bottom) and reproduce the results shown in Fig. 3a for widths $N \in \{32, 96, 128, 192, 256\}$. In each cell, the left y-axis reports the normalized rare-task signal $\tilde{s}_r(U_t)$, while the right y-axis (gray) is the gain / decay curves reporting how much the signal for rare task grows vs. decays as a function of time. We see analogous results as the main paper: larger models retain and preserve the learned signal, while smaller models require the gaps to be sufficiently small if learning is to occur at all.

F.3 Per-gap dynamics: Reproducing retention results across different injection gaps and widths

We reproduce results from Fig. 3a by reporting the joint dynamics of the normalized rare-task signal $\tilde{s}_r(U_t)$ and its gain / decay dynamics as a function of rare-task injection events, as shown in Fig. 17. Similar to results seen in the main paper, we find larger models retain and preserve the learned signal, while smaller models require the gaps to be sufficiently small if learning is to occur at all.

F.4 Effects of Scaling Data: Learning Bottleneck Persists at Long Training Horizon

In the main paper, especially Sec. 2, we distinguish between finite vs. asymptotic training. However, most of our training runs use a budget of 100K training iterations. To contextualize that this budget is sufficient for the claims made in the paper, we extend training runs to 1M steps for 6 values of model widths $N = \{8, 16, 32, 64, 128, 256\}$, subsampling the range of widths analyzed in the main paper; the setup remains the same otherwise as Fig. 2. We find that results (see Fig. 18) are stable at *much* longer horizons: above-capacity tasks do not slowly close the gap given more training; instead they remain at or below the random-projection baseline indefinitely.

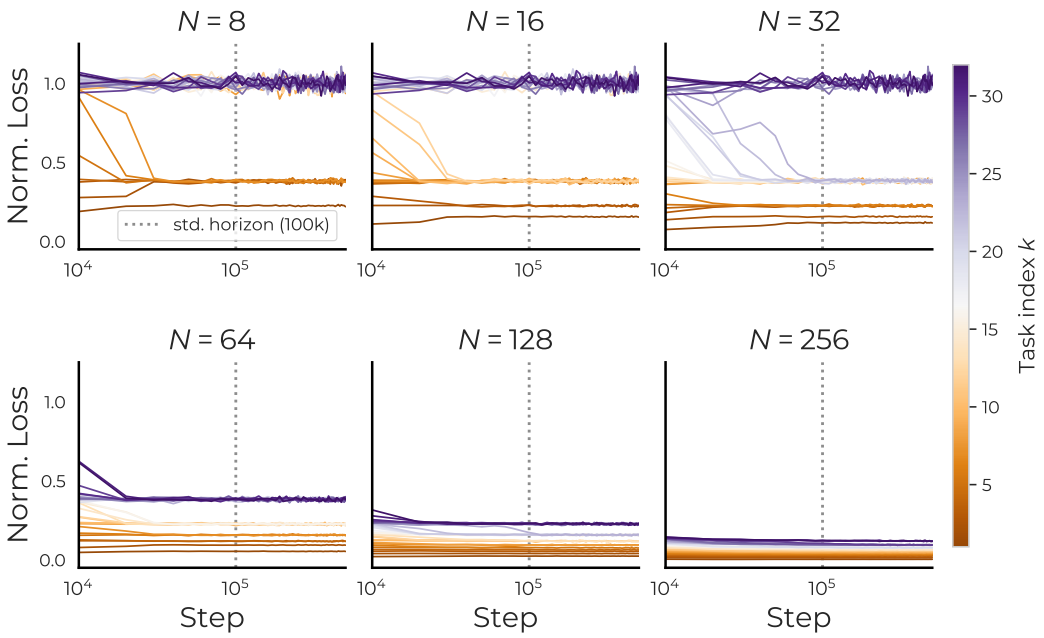

Figure 18: **Persistence of the multi-rank phase diagram at 1M steps.** Per-task normalized loss $\ell_k/\ell_{k,\text{baseline}}$ versus training step (log-x, linear-y) for six widths $N \in \{8, 16, 32, 64, 128, 256\}$; $\ell_{k,\text{baseline}} = \|a_k\|^2/D_t$ is the mean-predictor MSE per task. Tasks colored by index from orange ($k = 1$, most frequent) to purple ($k = 32$, rarest); vertical dotted line marks the training budget used in main paper, i.e., 100K steps. We clearly see that at every width, tasks that can fit model capacity (top-by-utility) drop near zero by the standard horizon and stay there; above-capacity tasks remain near the mean-predictor baseline and do not bend downward across longer training.

G Further Experimental Results in OLMo Setting

G.1 Task Loss vs. General Language Modeling Loss

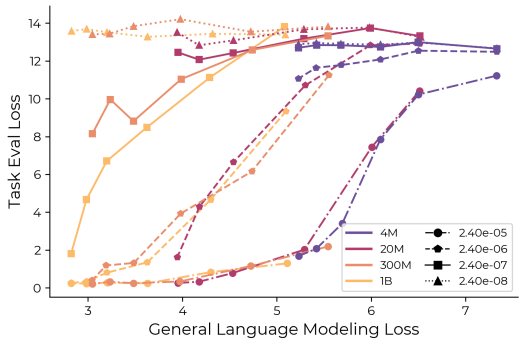

Figure 19: Task loss vs. general language modeling loss.

Given the same language modeling loss, larger models can achieve lower task-specific loss. We analyze the relationship between learning "frequent tasks" and learning the injected task. For evaluating "frequent tasks", we measure the language modeling loss on the C4 validation set using a context window size of 256. As shown in Fig. 19, when the task frequency is relatively high, i.e., from 2.4×10^{-6} to 2.4×10^{-5}, loss curves from all model sizes follow roughly the same trajectory. This suggests that in this frequency range, model size only improves sample efficiency, but these models still have similar training dynamics Vyas et al. (2023). However, when the task frequency gets lower, i.e., 2.4×10^{-7}, larger models achieve lower task loss given the same language modeling loss. Moreover, smaller models diverge from larger models one by one, with their injected-task loss plateauing at different values. This supports our hypothesis that for rare tasks, larger models have different learning dynamics that unlock the ability to learn rare tasks.

G.2 Compute-optimal Comparison

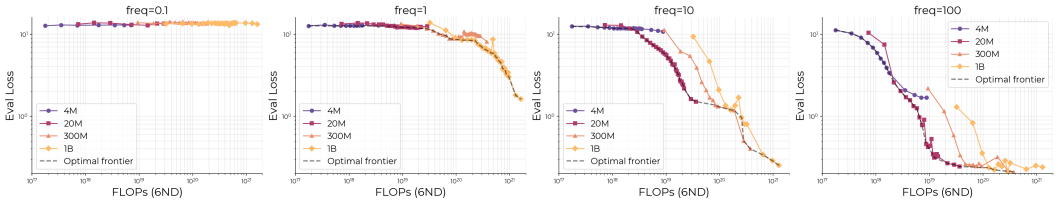

Figure 20: T_{CMP} task eval loss vs. compute by model size. Dashed black line shows the compute-optimal frontier.

In Fig. 4, we compare models trained on the same amount of data. In Fig. 20, we further show that larger models are more compute-efficient at learning low-frequency tasks. When the frequency is one task instance per batch, i.e., 2.4×10^{-7}, given the same compute budget, estimated as $6\times$ the number of model parameters $\times$ the number of training tokens following Chinchilla scaling laws, larger models achieve lower task loss. Moreover, consistent with our observation in Fig. 19, smaller models initially follow the learning dynamics of larger models, but after a certain point their injected-task loss curves plateau and deviate from the larger models' curves.

These results on scaling model size vs. simple data-side interventions offer three concrete options to improve performance of a task: (1) Higher task frequency: Given a fixed amount of pretraining data, a target general LM quality, or a small compute budget ($<10^{18}$ FLOPS in this case), one should increase task frequency, as shown in Fig. 4 and Fig. 19. (2) Larger model size: Given a fixed and sufficiently large compute budget, one should choose larger models as they reach lower task loss given same amount of compute, as shown in Fig. 4. (3) Rare-task data replay: Given the same amount of replay data, one should increase replay frequency with fewer examples per replay, as shown in

Fig. 4. In there case where there are dependencies between tasks, one might want to strategically increase the frequency of rare tasks that other more frequent tasks depend on, such that the rare task can be learned.