Pretraining Data Statistics Shape the Phases of Learning Entity Comparison in Language Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

How does data shape language model (LM) behavior throughout pretraining? We investigate this question through a case study on entity comparison, e.g., *Between France and Brazil, which country is larger?*. We begin with controlled experiments in which we train small LMs (124M parameters) on mixtures of natural text from pretraining corpora and synthetic data from entity comparison tasks. We identify three distinct phases of learning: (1) an early phase where the LM selects entities by frequency, (2) a middle phase where the LM selects entities by position in a prompt (first vs. last), and (3) a late phase where the LM selects the entity that is the correct answer to the question. We show that the emergence of these three phases is controlled by statistical properties of the training data. With small amounts of task-specific synthetic data, we observe only the first two phases and the model fails to learn the task; with large amounts, the model jumps directly from the frequency-based heuristic to solving the task correctly. Moreover, if we modify the frequency of entities in data from a naturally occurring Zipfian distribution (a small number of entities are very common and the vast majority are rare) to a uniform distribution, the first phase disappears and the model learns the task more quickly. Finally, we find the same three phases of learning in the pretraining of open-sourced OLMo models. Together, our findings demonstrate that properties of pretraining data are causal drivers of heuristic learning and show that small-scale synthetic experiments can predict training dynamics at larger scales.

1 Introduction

What does a language model (LM) learn throughout pretraining and how do statistical properties of data drive these learning dynamics? The relationship between data and model has proven fertile ground for interpretability research, with recent works uncovering how properties of data drive the formation of structured internal representations [Merullo et al., 2025, Karkada et al., 2026, Prieto et al., 2026], algorithmic motifs [Olsson et al., 2022, Chen et al., 2024, Wu et al., 2025], generalization capabilities [Chan et al., 2022, Chen et al., 2024, Ruis et al., 2025, Wang et al., 2025], and imperfect heuristics [Elazar et al., 2022, He et al., 2025, Kalai et al., 2025]. We conduct a case study on learning dynamics in which we study how LMs learn to make factual comparisons between entities, e.g., *Between France and Brazil, which country is larger?*.

Answering such questions properly requires retrieving the relevant scalar attributes for each entity (e.g., the area of a country), and performing a numerical comparison between those scalar values. However, models can instead rely on simpler heuristics, such as the answer's position in the prompt [Zheng et al., 2023], or how often the answer is mentioned in the training data [Gottesman et al., 2025, Yang et al., 2025]. Thus, this task provides a natural testbed for understanding (1) *when* different strategies emerge and disappear throughout training and (2) *how* entity frequency in pretraining data shapes such learning dynamics.

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

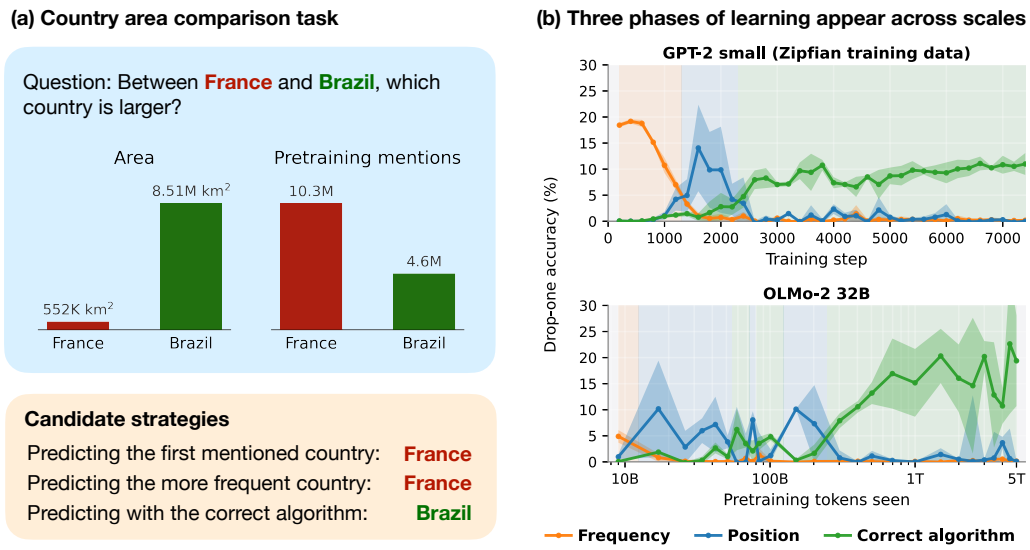

Figure 1: (a) The main task used to disentangle the three candidate strategies. The task requires comparing the areas of two countries; the correct answer is Brazil in the given example. However, if the model relies on positional or frequency bias, it would predict France. (b) The importance of each strategy across training checkpoints, measured by the decrease in accuracy when the given feature is removed. **The frequency heuristic (orange) is most predictive of LM behavior in the earliest checkpoints, but soon taken over by the positional heuristic (blue), and the correct algorithm (green) appears in late training.** The three-phase structure appears in both the small controlled setting and the large-scale natural-text pretraining setting of OLMo-2 models.

To answer the first question, we characterize model behavior throughout training, using three pieces of information: the difference in scalar attributes for each entity pair (the correct algorithm), entity frequency, and positional bias. We introduce a controlled pretraining setup in which we train GPT-2-scale models (124M parameters) from scratch. At each training checkpoint, we use a regression model to identify which features best predict model outputs, allowing us to track the evolution of strategies over time. Our analysis reveals three distinct phases of learning. First, the LM develops a frequency-based heuristic, outputting the entity that has been seen more often in training thus far. Next, a position-based heuristic takes over where the LM outputs either the first- or last-mentioned entity. Finally, the LM learns the correct comparison algorithm. Figure 1 (b) shows the three-phase structure observed in the training trajectories of both GPT-2 small and OLMo-2 32B.

We then investigate how properties of the pretraining data drive these learning dynamics (Section 3). To make the small LM learn the comparison behavior, we supplement natural text from the pretraining corpus with synthetic data, which contains examples of correctly solving the task. With insufficient synthetic task data, the LM only goes through the first two phases, developing a frequency heuristic then a position heuristic with no further learning. Increasing the proportion of task data first strengthens positional bias and eventually drives the LM to learn the comparison algorithm.

In a second experiment, we vary the distribution of entity frequencies. Naturally occurring text follows a Zipfian distribution, where a small number of entities are very high frequency while the vast majority of entities are very low frequency, e.g., *China* is more frequent than *Luxembourg*. We show that we can shorten the duration of the first two phases and temper the strength of the frequency heuristic by using uniform distributions for the synthetic and/or natural pretraining data.

Crucially, we validate these findings on larger pretrained models in the OLMo-2 family (1B, 7B, 13B, and 32B parameters), where we observe similar phase structure (Section 4). This suggests that the observed dynamics are not an artifact of the synthetic setup or model scale. Together, our results identify properties of pretraining data as causal drivers of the strategies LMs adopt, showing that they can control which strategies emerge during training and when.

2 Setup

Task formulation. The entity comparison task is defined by triples (d, e_1, e_2), where e_1, e_2 are entities (e.g., countries) and d is a domain corresponding to a numerical property (e.g., area, population, GDP). Each entity-domain pair (d, e) is associated with a latent scalar value $v_{d,e}$. The algorithm for solving a comparison query $\mathcal{A}(d, e_1, e_2) = \max(v_{d,e_1}, v_{d,e_2})$ returns the entity with the higher value.

Training data. We use subsets of the fully open-sourced OLMo-2 pretraining corpus, `OLMo-mix-1124` [OLMo et al., 2024], which is used primarily for stage-1 pretraining of OLMo-2 models. The country mention frequencies in the data follow a heavy-tailed Zipfian distribution, as shown in Figure 2(a). At the 124M scale, models cannot reliably learn long-tail factual knowledge from natural text alone [Allen-Zhu and Li, 2023, Kandpal et al., 2023], which we observe in our preliminary experiments. We therefore add synthetic task-supervision data as scaffolding: it consists of comparison sentences that exactly match the evaluation template *"By total area, among the two countries A and B, the larger country is "*. The synthetic data serves as a proxy for the task-relevant signal that larger LMs absorb implicitly from massive pretraining corpora. We use the full set of countries in both training and evaluation. This mirrors the natural pretraining setting, where models are exposed to all entities they will later be evaluated on, but it means the LM can memorize the data. We perform additional experiment with 20% of country pairs held out in Appendix A.

Model. We train transformer language models with the GPT-2 architecture (12 layers, 12 heads, 768-dim embeddings, 124M parameters) [Radford et al., 2019] from scratch on each data mixture. Training data is tokenized with the `allenai/dolma2-tokenizer`, which matches the tokenizer used by the OLMo-2 models. This ensures token-level frequency counts are directly comparable across models. We train each model for two epochs with three random seeds.

Evaluation. We extract the teacher-forced logits of the first generated token at which the entities differ to determine the model's choice. We then denote it as a binary output distribution P_{LM} over the two entities. We measure per-pair accuracy at every saved checkpoint.

2.1 Counting Entity Occurrences in Pretraining Data

To compute frequency-based features, we track cumulative occurrence and co-occurrence counts of country terms up to each model checkpoint. This accurately reflects how much exposure the model has had to each entity at each point during training. For countries with multiple name variants (e.g. *U.S.A.*, *United States*, *the U.S.*), we aggregate counts across all aliases. We adapt the `BatchSearch` method from Merullo et al. [2025] and cross-validate our counts with frequency estimates from `infini-gram` [Liu et al., 2024] (Pearson's $r = 0.99$). We define $\mathsf{count}_t(e)$ as the number of times entity e appears in the pretraining corpus up to training step t, and similarly $\mathsf{count}_t(e_1, e_2)$ for co-occurrences. Our hypothesis is that entity frequency in the pretraining corpus influences which heuristics a model adopts and the order in which they appear during training. Specifically, given a query (d, e_1, e_2), if $\mathsf{count}_t(e_1) > \mathsf{count}_t(e_2)$ at step t, the model is more likely to predict e_1 regardless of whether $v_{d,e_1} > v_{d,e_2}$.

Natural pretraining corpora exhibit heavy-tailed Zipfian entity frequency distributions. In Figure 2 (a), a small set of countries appears very frequently (e.g., *United States* and *France*), while the vast majority appear far less often. This long-tail pattern is typical for words in natural English text [Condon, 1928, Zipf, 2013]. In our experiments, we manipulate entity frequency separately in synthetic and natural data to study how this naturally occurring distribution shapes learning dynamics.

2.2 Three Phases of Learning

To identify which strategies drive model predictions at each checkpoint during training, we fit a logistic regression model to predict model outputs from a set of hand-picked features, each corresponding to a heuristic or to the correct algorithm. By comparing regression accuracy across different feature subsets, we can track which strategies are predictive at each checkpoint. In preliminary experiments, we evaluated a wide range of features (full list in Section A.1). We found that the best fit uses only

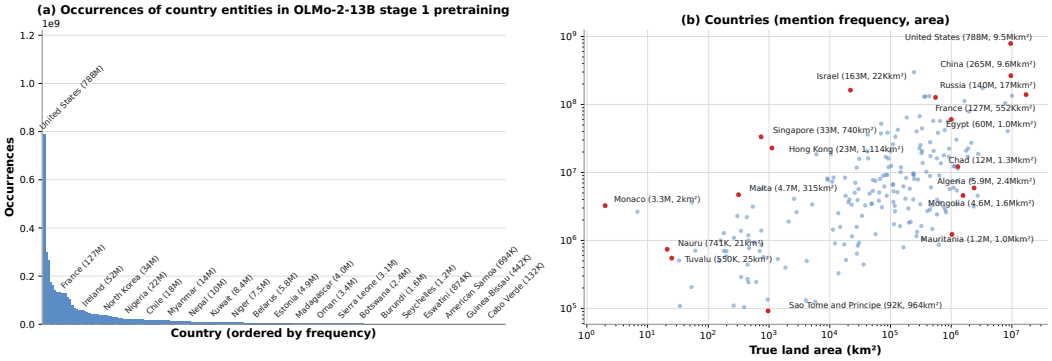

Figure 2: (a) Occurrences of the 216 countries in the OLMo-2-13B pretraining corpus, which follow a heavy-tailed Zipfian distribution. (b) Country frequency correlates with the area attribute (log-log Pearson's $r = 0.62$), meaning that predicting the more frequently mentioned country as the larger one is a viable heuristic the model can adopt.

three: the frequency, position, and true value of entities. We describe each below, in the order we expect them to dominate over the course of training.

1. **Frequency feature:** We use the log-ratio of cumulative occurrences of e_1 and e_2 in the training corpus up to step t: $\log\left(\mathsf{count}_t(e_1)/\mathsf{count}_t(e_2)\right)$. A model relying on this feature is using a popularity heuristic, predicting whichever entity it has seen more often. More broadly, we expect this feature to dominate model predictions early in training, before the model has had sufficient exposure to relevant entities.
2. **Positional bias feature:** In the task, the two entities are mentioned sequentially, so each can appear in either the first or last position. A model in mid-training may be biased to always output the entity in one of these positions, regardless of where the correct answer is. Since the task is a binary choice, we formalize this bias using the regression intercept, which absorbs any systematic preference for the first- or last-mentioned entity.
3. **Value feature:** We use the log-ratio of the true values of e_1 and e_2 in the question domain d: $\log\left(v_{d,e_1}/v_{d,e_2}\right)$. This is the only one of the three features predictive of the ground-truth answer, so a model relying on it has learned the underlying task. We expect this feature to emerge late in training.

Evaluating feature combinations. We fit a logistic regression model to predict the model's choice. The probability of selecting e_1 is given by $\hat{p} = \sigma\big(w^\top \phi(d, e_1, e_2)\big)$, where w is learned by minimizing the standard logistic regression objective with L1 regularization. The L1 penalty allows us to identify only the most predictive features. We use 5-fold cross-validation and select the regularization strength on the training folds. All features are standardized to zero mean and unit variance before fitting. We report the mean accuracy across the five folds. To isolate individual strategies, we refit the model using different subsets of features, including positional-only, frequency-only, and value-only feature sets, as well as models that include interaction features.

Metric for reporting logistic regression ablations. To characterize which strategies the model uses at each point in training, we fit the regression features described in Section 2.2 and report the ***drop-one accuracy***: the decrease in regression accuracy when a given feature is removed from the predictor set. This measures the importance of that feature in predicting model behavior: a large drop-one accuracy for a feature indicates that the model's outputs are well-explained by that feature at that checkpoint.

3 Entity Comparison Learning Dynamics in Small LMs with Synthetic Data

To study how heuristics develop through training, we train small language models from scratch on mixtures of natural text from the OLMo pretraining corpora and synthetic data from the entity

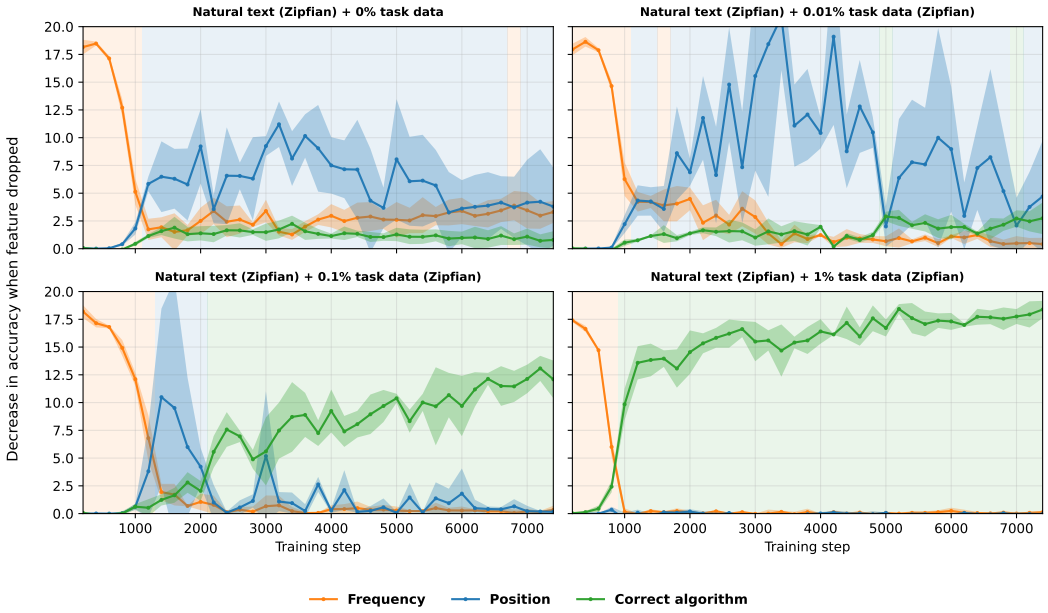

Figure 3: Learning trajectories across four ratios of synthetic task data to natural text (0%, 0.01%, 0.1%, 1%). Solid lines and dark shaded regions show mean and standard deviation across 3 training seeds. **The frequency phase is invariant to task supervision, but the correct-algorithm phase only emerges above a critical density.** Below this density (0%, 0.01%), the model never reaches the correct algorithm and instead defaults to positional shortcuts.

comparison task. First, we titrate the amount of synthetic task data relative to natural text and find that, with the right balance of data, three phases of learning emerge: a frequency heuristic first, then a position heuristic, then the correct algorithm (Section 3.1). Then, we manipulate the distributions of entities in training data to be Zipfian vs. uniform and observe how the development of these heuristics is affected (Section 3.2).

3.1 How the Ratio of Synthetic Data Shapes the Three Phases

The 124M-parameter LMs do not learn the comparison task when trained on only natural data, which is unsurprising at this scale. To help the model learn behavior we want to study, we supplement natural text in the pretraining corpora with synthetic task-specific data of the form *"By total area, among the two countries A and B, the larger country is "*. Under a fixed total data budget of 1.1B tokens, we vary the proportion of synthetic task supervision in 0%, 0.01%, 0.1%, and 1%. We generate synthetic data that matches the Zipfian distribution, so we can study the effect of naturally occurring task signal alone. The fully natural setting serves as a control, which allows us to observe the effect of natural text on model behavior.

The proportion of task-specific synthetic data does not affect the frequency phase. Figure 3 shows that the frequency phase is essentially identical across all four conditions: it peaks at step 200 and decays by step 1K, regardless of synthetic ratio. This motivates Section 3.2, where we show that this invariance is explained by the frequency distribution of the natural text alone; manipulating the natural text distribution removes the frequency phase entirely.

The correct algorithm emerges only above a critical density of task specific synthetic data. What does change across conditions is whether the model ever reaches the correct algorithm. At 0% and 0.01% synthetic data, the correct-algorithm feature stays near zero throughout training. At 0.1%, it emerges around 2.5K steps and rises steadily. When task data is increased to 1%, it emerges within the first 1K steps and reaches 0.18 by end of training. This suggests that there is a threshold: below a critical density of task-relevant signal, the model never finds learning the task loss-reducing enough to override existing shortcuts [Gu et al., 2025].

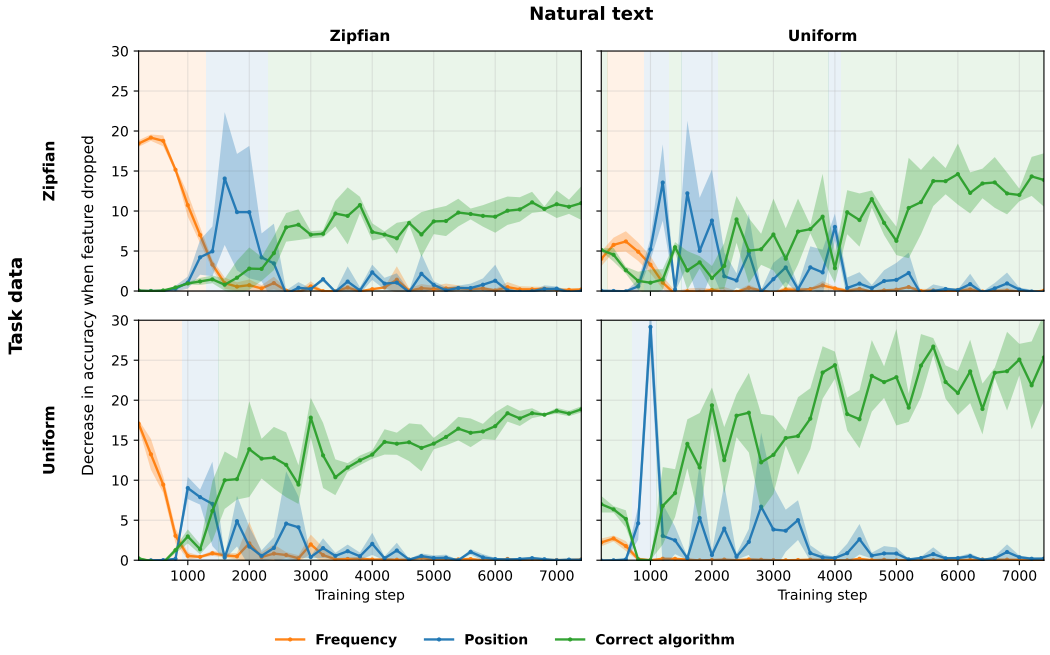

Figure 4: Learning trajectories across four combinations of natural text and synthetic task data distributions (Zipfian $\times$ Uniform). **The frequency phase is present whenever natural text is Zipfian and essentially absent when natural text is flattened**, regardless of the synthetic data. All settings show the same phase ordering (frequency, position, and correct algorithm), but under Zipfian data, the frequency phase is longer and the correct algorithm phase arrives later, whereas under uniform training the correct algorithm is learned faster and achieves higher accuracy. Solid lines and dark shaded regions show mean and standard deviation across 3 training seeds.

The position heuristic disappears at above a critical density of task specific synthetic data. When the correct-algorithm phase does not emerge, model behavior converges to a strong positional bias. This positional phase is most pronounced at 0% and 0.01% of synthetic data, where it persists throughout the rest of training. Above the threshold, the phase is absent, as the model transitions directly from frequency to the correct algorithm.

3.2 How Entity Frequency Distribution Drives the Heuristic Phases

With a specific amount of synthetic task data (0.1%), we observe the three-phase learning dynamics emerge in training. However, the previous experiments use synthetic data with entity frequency distributions that match the Zipfian distribution of naturally occurring data. Now, we separately manipulate the synthetic and the natural data to change the entity frequency distribution from Zipfian to uniform. To construct the uniform setting, we sample sequences that contain country mentions to achieve roughly uniform mentions per country. We provide the parameters of all distributions in Section B.1.

Zipfian distribution drives the initial frequency heuristic phase. Figure 4 shows that the entity frequency distribution of natural text is what causes the frequency phase. When natural text is Zipfian (left column), a strong frequency phase emerges in the first 1K training steps; when natural text is flattened (right column), the frequency phase essentially disappears. This holds regardless of the synthetic data distribution: switching the synthetic task data between uniform and Zipfian within a row modulates later phases but does not produce the strong frequency phase when natural text is uniform.

The correct algorithm is learned the most and emerges the fastest when all entities are uniformly distributed. In Figure 4, we see how the uniformity of the entity frequency distribution shapes

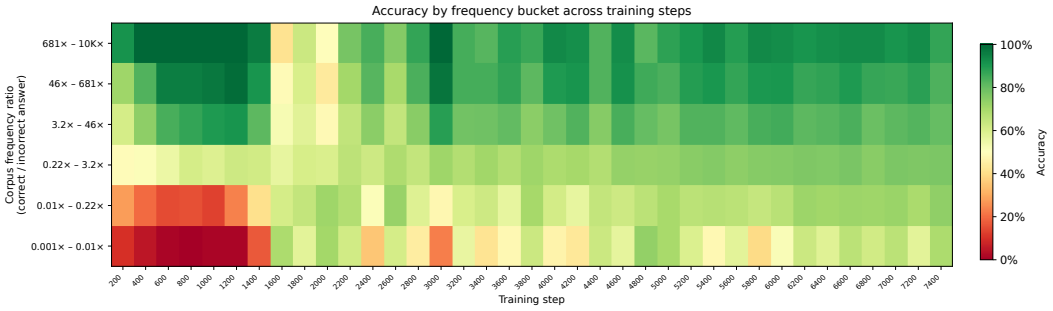

Figure 5: Accuracy by corpus frequency ratio (correct / incorrect answer) across training steps under Zipfian data. Early in training, accuracy is high only when the correct answer is much more frequent, which corresponds to the frequency bias phase.

learning dynamics. When both the synthetic and natural text are Zipfian (top left), the frequency heuristic dominates early and the model spends a long time in the heuristic phases before the correct algorithm emerges. In contrast, when both distributions are uniform (bottom right), the frequency heuristic never dominates, and the value feature emerges earliest and reaches the highest peak.

Frequency heuristic is reduced through training but never fully eliminated. We further illustrate this early phase of frequency bias in Figure 5. It shows prediction accuracy across entity pairs, grouped by the ratio of the correct entity's frequency to that of the incorrect one. A ratio less than 1 indicates that the correct answer is less frequent, while a ratio greater than 1 indicates it is more frequent. Early in training, accuracy is high only when the correct answer is substantially more frequent than the incorrect one. While this bias gradually diminishes, the model still maintains higher accuracy on pairs where the correct answer is more frequent in late training stages.

4 Learning Dynamics in OLMo Pretraining

The controlled experiments in Section 3 establish that the entity frequency distribution of natural text drives the frequency heuristic phase. We now ask whether the same phase structure appears in models trained at scale on natural text alone, without any synthetic task supervision.

Setup. We focus on the OLMo-2 model family (1B, 7B, 13B, and 32B) [Groeneveld et al., 2024], which provides fully open-sourced training data and checkpoints. We apply the same regression analysis as described in Section 2 to track which strategy dominates over time. We evaluate 26 checkpoints per model, focusing primarily on stage 1 pretraining, with dense early coverage (every 1K steps from 1K to 10K) and sparser, log-spaced checkpoints afterward. To account for prompt variation, we evaluate each checkpoint with 3 prompt templates and report mean and standard deviation across them. The checkpoints and prompts used are in Appendix C.

The frequency and position phases appear in a consistent order across model scales. In Figure 6, all four models exhibit the early heuristic phases observed in our controlled experiments, and they emerge in the same order of frequency followed by position. The frequency heuristic appears briefly in the earliest checkpoint of all models. The 1B and 7B models then remain in a prolonged position heuristic phase without learning the task; the 13B model transitions to the correct algorithm at around 1T tokens, while the 32B model does so earlier and achieves a higher peak. We note that the frequency phase has a lower peak than in the small-scale setting, which we hypothesize is because the first available checkpoint already occurs several billion tokens into training. A more pronounced frequency phase may occur before this point. Observing such a phase would require access to earlier pretraining checkpoints or pretraining from scratch, which we leave for future work.

Only the 13B and 32B models learn the correct algorithm and exhibit all three phases. The correct algorithm phase emerges only in the 13B and 32B models, at around 800B tokens. In the two smaller models, the value feature stays near zero throughout training, meaning that they never learn the task. This parallels the threshold finding from Section 3.1: just as the correct algorithm fails

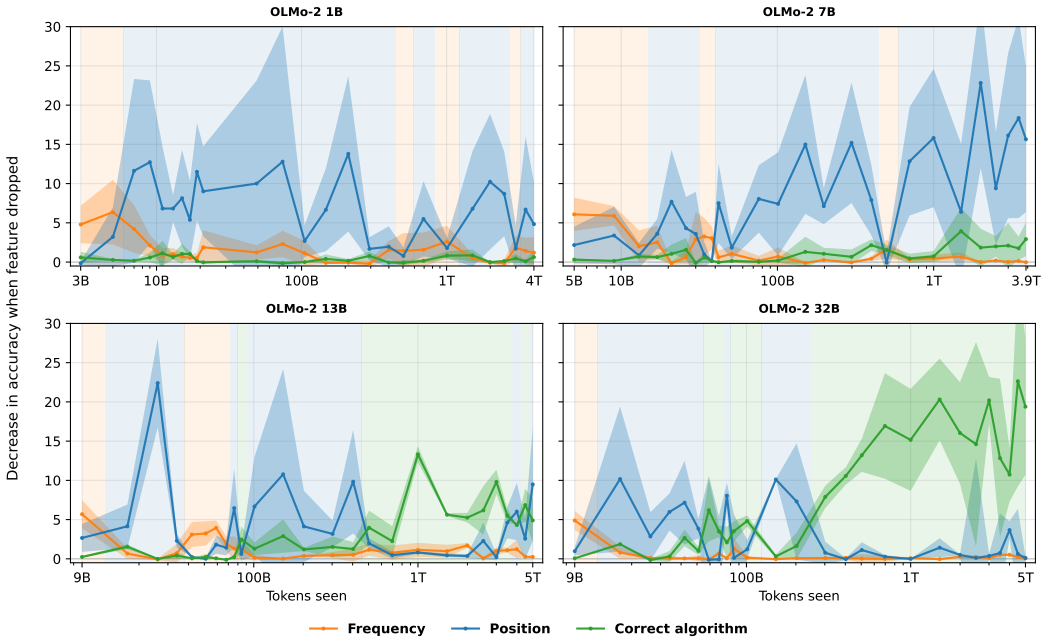

Figure 6: Feature importance across training checkpoints for OLMo-2 1B, 7B, 13B, and 32B. Solid lines and shaded regions show mean and standard deviation across 3 prompt variations per checkpoint. All models show frequency and position heuristics appearing in order, consistent with the phases observed in the controlled GPT-2 experiments.

to emerge below a critical density of task supervision in our controlled experiments, here it fails to emerge below a critical model size. Natural pretraining text contains relatively sparse task signal for entity comparison, and at 1B and 7B parameters the model lacks the capacity to learn the task.

5 Discussion

Implications on training. Across scales from 124M to 32B parameters, models exhibit the same heuristic phases, and the 1B and 7B OLMo models never escape the positional heuristic. These heuristics appear to be a structural feature of how the task is learned, rather than an artifact of small scale. The key difference is whether the model successfully transitions out of these heuristics, not whether the heuristics emerge. In controlled experiments, the correct algorithm fails to emerge below a critical density of task supervision; in OLMo, it fails to emerge below a critical model scale on natural text alone. This suggests that the correct strategy only becomes loss-reducing once the model has sufficient task-relevant signal relative to its capacity. Practically, our results indicate that the long tail in natural pretraining corpora may impose a real cost on downstream task acquisition. Flattening these heavy-tailed distributions in pretraining corpora can make learning more efficient.

Understanding of heuristics learning. We observe three phases in learning: first a frequency heuristic, then a position heuristic, and finally the correct algorithm. The consistency of this ordering across very different settings suggests that it reflects a structural property of the task: each phase corresponds to the easiest-to-learn predictive feature given the model's current state. Frequency is directly available from token-level statistics, position depends on syntactic patterns in the data, and value-based entity comparison requires manipulating factual knowledge. Each subsequent strategy is likely more expensive to learn, but eventually becomes more loss-reducing than its predecessor.

Limitations. We focus our experiments on a single task, and the observed phase structure may not generalize to tasks with different structures. Our controlled experiments are conducted at the 124M scale in order to efficiently vary data composition and perform model pretraining under feasible compute budgets, which would otherwise be infeasible for large-scale pretraining. Although these intervention experiments are small-scale, we show that the toy setting can nevertheless provide insight

into the training dynamics of much larger models, which is itself an interesting finding. This suggests that the setup may capture something general about how data composition shapes learning dynamics. While our OLMo-2 analysis suggests that the same phases emerge at much larger scales, we cannot establish causal evidence for the role of data at scale.

Lastly, our experiments that modify pretraining data statistics (e.g., changing the country distribution from Zipfian to uniform) require re-selecting sequences containing the target entities. For example, this may involve excluding sequences heavily focused on United States politics while selecting sequences about tropical fruits in Madagascar. This process could introduce domain shifts and other confounding factors that we do not account for. There is currently no straightforward solution to this issue, and it remains an active open research problem [Nadkarni et al., 2025].

Future work. We would like to understand the mechanisms deployed during inference for each heuristic and more fully trace origins of these three phases in this order. For example, whether the positional bias relies on induction-head-style in-context retrieval, and whether there exists latent representations that track entity frequency.

6 Related Work

Heuristics and different task strategies. LMs often rely on heuristics rather than the intended algorithms for solving tasks. Prior work has identified shortcuts in natural language inference [McCoy et al., 2019], arithmetic reasoning [Razeghi et al., 2022, Nikankin et al., 2025], multi-hop reasoning [Yang et al., 2025], and factual recall, where entity popularity sharply predicts whether a model answers correctly [Mallen et al., 2023, Kandpal et al., 2023]. Most closely related to our setting, Lehmann et al. [2026] study entity comparison in fully trained LLMs and identify three heuristic biases that influence predictions. A complementary view characterizes LM behavior as a mixture of strategies rather than a single mechanism [Wu et al., 2025, Gur-Arieh et al., 2025, Pîslar et al., 2025, Todd et al., 2025]. These analyses operate on static checkpoints; we instead study the learning trajectory itself, asking when each heuristic emerges during training and what data properties drive its emergence.

Pretraining data. Pretraining data affects model behavior, performance, and internal representations [Elazar et al., 2022, McCoy et al., 2023, Shaib et al., 2024, Merullo et al., 2025, Ruis et al., 2025, Verma et al., 2025]. Wang et al. [2025] measure the correlation between LLM output probabilities and the frequency of task-relevant n-gram pairs in pretraining data, while other work has traced model biases and failures to asymmetries in entity frequency [Razeghi et al., 2022, He et al., 2025, Kandpal et al., 2023, Mallen et al., 2023]. To make causal claims about data effects, prior work has intervened directly on training data composition [Wei et al., 2021, Chan et al., 2022, Allen-Zhu and Li, 2024]; Nadkarni et al. [2025] provide a recipe for document-level interventions, and Longpre et al. [2024] systematically study effects of data age, domain coverage, and quality. Our work builds on this interventional methodology and identifies entity distribution as a causal driver of which heuristic phases emerge during training.

Phases and training dynamics. Capabilities are often acquired during pretraining through abrupt transitions or distinct phases that build on earlier representations [Saphra and Lopez, 2019, Chen et al., 2024, Wu et al., 2025]. A canonical case study is the formation of induction heads, which coincides with a sharp drop in loss and the emergence of in-context learning [Olsson et al., 2022, Singh et al., 2024]. Closer to our setting, Ravfogel et al. [2025] trace the emergence of a linear truth direction, and Gu et al. [2025] study phase transitions as a function of data mixing ratio. Our work extends this perspective to natural pretraining data statistics.

7 Conclusion

We identify three phases of learning in entity comparison and show through controlled experiments that pretraining data composition causally determines which phases emerge and when. The same phase structure appears in OLMo-2 models trained on natural text at scales up to 32B parameters, suggesting these dynamics are a general feature of how LMs learn this task. Our results highlight pretraining data statistics as both a driver of heuristic learning and a controllable lever for studying it.

References

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. In *ICML 2024: International Conference on Machine Learning*, 2023.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.3, knowledge capacity scaling laws. *arXiv preprint arXiv:2404.05405*, 2024.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. *Advances in neural information processing systems*, 35:18878–18891, 2022.

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`.

E. U. Condon. Statistics of vocabulary. *Science*, 67(1733):300, 1928. URL `https://pure.mpg.de/view/item_2309752`. Accessed: 2026-05-06.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Amir Feder, Abhilasha Ravichander, Marius Mosbach, Yonatan Belinkov, Hinrich Schütze, and Yoav Goldberg. Measuring causal effects of data statistics on language model'sfactual'predictions. *arXiv preprint arXiv:2207.14251*, 2022.

Daniela Gottesman, Alon Gilae-Dotan, Ido Cohen, Yoav Gur-Arieh, Marius Mosbach, Ori Yoran, and Mor Geva. Lment: A suite for analyzing knowledge in language models from pretraining data to representations. *arXiv preprint arXiv:2509.03405*, 2025.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. OLMo: Accelerating the science of language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 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/`.

Xinran Gu, Kaifeng Lyu, Jiazheng Li, and Jingzhao Zhang. Data mixing can induce phase transitions in knowledge acquisition. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Yoav Gur-Arieh, Mor Geva, and Atticus Geiger. Mixing mechanisms: How language models retrieve bound entities in-context. *arXiv preprint arXiv:2510.06182*, 2025.

Yuan He, Bailan He, Zifeng Ding, Alisia Maria Lupidi, Yuqicheng Zhu, Shuo Chen, Caiqi Zhang, Jiaoyan Chen, Yunpu Ma, Volker Tresp, and Ian Horrocks. Supposedly equivalent facts that aren't? entity frequency in pre-training induces asymmetry in LLMs. In *Second Conference on Language Modeling*, 2025. URL `https://openreview.net/forum?id=sX4OoLKSW2`.

Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*, 2025.

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*, pages 15696–15707. PMLR, 2023.

Dhruva Karkada, Daniel J. Korchinski, Andres Nava, Matthieu Wyart, and Yasaman Bahri. Symmetry in language statistics shapes the geometry of model representations, 2026. URL `https://arxiv.org/abs/2602.15029`.

Hans Hergen Lehmann, Jae-Hee Lee, Steven Schockaert, and Stefan Wermter. Knowing the facts but choosing the shortcut: Understanding how large language models compare entities. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4788–4821, 2026.

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.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3245–3276, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.179. URL `https://aclanthology.org/2024.naacl-long.179/`.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL `https://aclanthology.org/2023.acl-long.546/`.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 3428–3448, 2019.

R Thomas McCoy, Shunyu Yao, Dan Friedman, Matthew Hardy, and Thomas L Griffiths. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638*, 2023.

Jack Merullo, Noah A Smith, Sarah Wiegreffe, and Yanai Elazar. On linear representations and pretraining data frequency in language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Rahul Nadkarni, Yanai Elazar, Hila Gonen, and Noah A Smith. Rewriting history: A recipe for interventional analyses to study data effects on model behavior. *arXiv preprint arXiv:2510.14261*, 2025.

Yaniv Nikankin, Anja Reusch, Aaron Mueller, and Yonatan Belinkov. Arithmetic without algorithms: Language models solve math with a bag of heuristics. In *The Thirteenth International Conference on Learning Representations*, 2025.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James Validad Miranda, Jacob Daniel Morrison, Tyler C. Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Chris Wilhelm, Michael Wilson, Luke S. Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hanna Hajishirzi. 2 olmo 2 furious. *ArXiv*, abs/2501.00656, 2024.

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

Theodora-Mara Pîslar, Sara Magliacane, and Atticus Geiger. Combining causal models for more accurate abstractions of neural networks. In *Causal Learning and Reasoning*, pages 114–138. PMLR, 2025.

Lucas Prieto, Edward Stevinson, Melih Barsbey, Tolga Birdal, and Pedro A. M. Mediano. Correlations in the data lead to semantically rich feature geometry under superposition. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7akSRQS5Xh.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Shauli Ravfogel, Gilad Yehudai, Tal Linzen, Joan Bruna, and Alberto Bietti. Emergence of linear truth encodings in language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 840–854, 2022.

Laura Ruis, Maximilian Mozes, Juhan Bae, Siddhartha Rao Kamalakara, Dwaraknath Gnaneshwar, Acyr Locatelli, Robert Kirk, Tim Rocktäschel, Edward Grefenstette, and Max Bartolo. Procedural knowledge in pretraining drives reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=1hQKHHUsMx.

Naomi Saphra and Adam Lopez. Understanding learning dynamics of language models with SVCCA. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3257–3267, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1329. URL https://aclanthology.org/N19-1329/.

Chantal Shaib, Yanai Elazar, Junyi Jessy Li, and Byron C Wallace. Detection and measurement of syntactic templates in generated text. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6416–6431, 2024.

Aaditya K Singh, Ted Moskovitz, Felix Hill, Stephanie CY 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.

Eric Todd, Jannik Brinkmann, Rohit Gandikota, and David Bau. In-context algebra. *arXiv preprint arXiv:2512.16902*, 2025.

Sahil Verma, Royi Rassin, Arnav Mohanty Das, Gantavya Bhatt, Preethi Seshadri, Chirag Shah, Jeff Bilmes, Hannaneh Hajishirzi, and Yanai Elazar. How many images does it take? estimating imitation thresholds in text-to-image models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=x0qJo7SPhs.

Xinyi Wang, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang. Generalization v.s. memorization: Tracing language models' capabilities back to pretraining data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=IQxBDLmVpT.

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 932–948, 2021.

Yiwei Wu, Atticus Geiger, and Raphaël Millière. How do transformers learn variable binding in symbolic programs? In *Forty-second International Conference on Machine Learning*, 2025.

Sohee Yang, Nora Kassner, Elena Gribovskaya, Sebastian Riedel, and Mor Geva. Do large language models perform latent multi-hop reasoning without exploiting shortcuts? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3971–3992, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.205. URL https://aclanthology.org/2025.findings-acl.205/.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

George Kingsley Zipf. *The psycho-biology of language: An introduction to dynamic philology*. Routledge, 2013.

A Feature Ablation Experiments

A.1 List of Evaluated Features

To identify the strategies that LMs use for the entity comparison task, we evaluate a wide range of features. For each input triple (d, e_1, e_2), we construct a feature vector $\phi(d, e_1, e_2)$ to capture relevant features that correspond to possible heuristics. We consider four families of features:

1. **Positional bias feature**: binary indicators of left/right position, independent of entity semantics or values (e.g., whether e_1 is favored by virtue of being mentioned first). This is the intercept in the logistic regression model.
2. **Frequency-based features**: entity occurrence and co-occurrence counts in the pretraining corpus (e.g., sum of the occurrences of e_1 and e_2).
3. **Value-based features**: variants of the numerical values v_{d,e_1} and v_{d,e_2}, including absolute values, log-scaled values, differences, and ratios.
4. **Related-domain features**: values of numerical properties other than d that may co-vary with $v_{d,e}$ (e.g., population or GDP when d is area), intended to test whether models rely on these correlated proxies.

For the frequency and domain values, we include normalized variants (e.g., z-scores, distance-to-mean) to test if the model responds to relative magnitude rather than absolute values. The full list of features and their corresponding definitions is provided in Table 1.

A.2 Feature Importance in Small LMs

This section presents additional results from the controlled pretraining experiments in Section 3.2. In addition to the the drop-one accuracy plots, we fit a logistic regression on every non-empty subset of the three main features (frequency, position, value) at every checkpoint, and report the regression's prediction accuracy on the LM's pairwise output.

Each figure has six panels, one per non-trivial subset of {P, F, V}: {P}, {F}, {V}, {P+F}, {P+V}, {F+V}. In every panel, the colored line is the prediction accuracy when only that subset is used as predictors, and the black line is the prediction accuracy of the full P+F+V regression on the same checkpoint, reproduced as a baseline. A subset whose curve sits on top of the full P+F+V baseline at a given checkpoint already captures all the LM's behavior at that point in training; a subset whose curve sits well below the baseline is missing a feature the LM is using.

Name	Definition	Example
Value-based features		
`value_left`	v_{d,e_1}	551,500
`value_right`	v_{d,e_2}	357,581
`value_left_log`	$\log_{10}(v_{d,e_1})$	5.74
`value_right_log`	$\log_{10}(v_{d,e_2})$	5.55
`value_ratio`	$\log_{10}(v_{d,e_1}/v_{d,e_2})$	0.19
`value_left_from_threshold`	$(v_{d,e_1} - \mu_v)/\mu_v$	-0.08
`value_right_from_threshold`	$(v_{d,e_2} - \mu_v)/\mu_v$	-0.40
`value_zscore_left`	$(\log_{10} v_{d,e_1} - \mu_{\log v})/\sigma_{\log v}$	0.89
`value_zscore_right`	$(\log_{10} v_{d,e_2} - \mu_{\log v})/\sigma_{\log v}$	0.75
Frequency-based features		
`freq_left`	f_{e_1}	1.03×10^7
`freq_right`	f_{e_2}	9.69×10^6
`freq_left_log`	$\log_{10}(f_{e_1})$	7.01
`freq_right_log`	$\log_{10}(f_{e_2})$	6.99
`freq_sum`	$f_{e_1} + f_{e_2}$	2.00×10^7
`freq_sum_log`	$\log_{10}(f_{e_1}) + \log_{10}(f_{e_2})$	14.00
`freq_diff`	$f_{e_1} - f_{e_2}$	6.49×10^5
`log_freq_ratio`	$\log_{10}(f_{e_1}) - \log_{10}(f_{e_2})$	0.03
`freq_zscore_left`	$(\log_{10} f_{e_1} - \mu_{\log f})/\sigma_{\log f}$	1.42
`freq_zscore_right`	$(\log_{10} f_{e_2} - \mu_{\log f})/\sigma_{\log f}$	1.39
`cooccurrence_log`	$\log_{10}(1 + c(e_1, e_2))$	3.54
`cooccurrence_larger_string`	Directional pair count aligned to e_1/e_2	1,820
Bayesian feature		
`bayesian_posterior`	$\Phi\left(\frac{v_{d,e_1} - v_{d,e_2}}{\sqrt{\sigma_{e_1}^2 + \sigma_{e_2}^2}}\right) - \frac{1}{2}$, with $\sigma_e = \sigma_0/\sqrt{1 + \beta f_e}$	0.18
Related-domain features (computed for $a \in \{\text{gdp}, \text{population}\}$)		
a`_left_log`	$\log_{10}(a_{e_1})$	$\text{gdp}_L = 12.50$
a`_right_log`	$\log_{10}(a_{e_2})$	$\text{gdp}_R = 12.67$
a`_ratio`	$\log_{10}(a_{e_1}) - \log_{10}(a_{e_2})$	$\text{gdp} = -0.17$
a`_sum_log`	$\log_{10}(a_{e_1}) + \log_{10}(a_{e_2})$	$\text{gdp} = 25.17$
a`_zscore_left`	$(\log_{10} a_{e_1} - \mu_{\log a})/\sigma_{\log a}$	$\text{gdp} = 0.41$
a`_zscore_right`	$(\log_{10} a_{e_2} - \mu_{\log a})/\sigma_{\log a}$	$\text{gdp} = 0.55$

Table 1: Features evaluated for modeling the LM behavior. Examples are illustrative values for the country-area task with $e_1 =$ France ($v_{d,e_1} = 551{,}500\,\text{km}^2$, $f_{e_1} \approx 1.03 \times 10^7$) and $e_2 =$ Germany ($v_{d,e_2} = 357{,}581\,\text{km}^2$, $f_{e_2} \approx 9.69 \times 10^6$); μ/σ are taken over the pooled left/right values in the dataset. The threshold μ_v is the mean of the pooled raw values. Frequencies f_e are token-mention counts in the training corpus and $c(e_1, e_2)$ is the symmetric pair co-occurrence count. For the Bayesian feature, σ_0 is the initial scale (domain range), β is the evidence decay rate, and Φ is the standard-normal CDF.

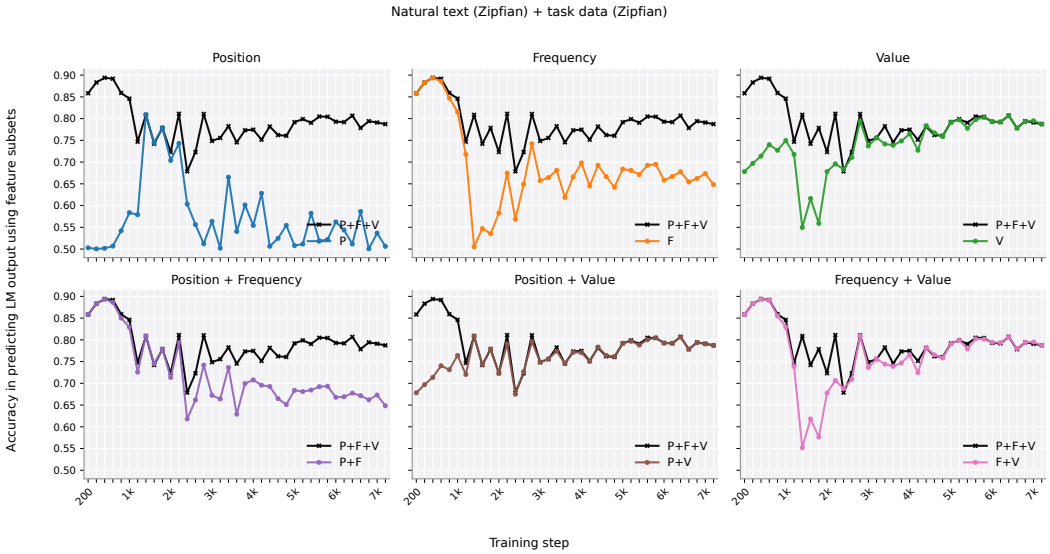

Figure 7: Prediction accuracy of feature subsets for the 124M LM trained on Zipfian natural text mixed with 0.1% Zipfian synthetic data. Each panel compares one feature subset (colored line) against the full P+F+V regression (black).

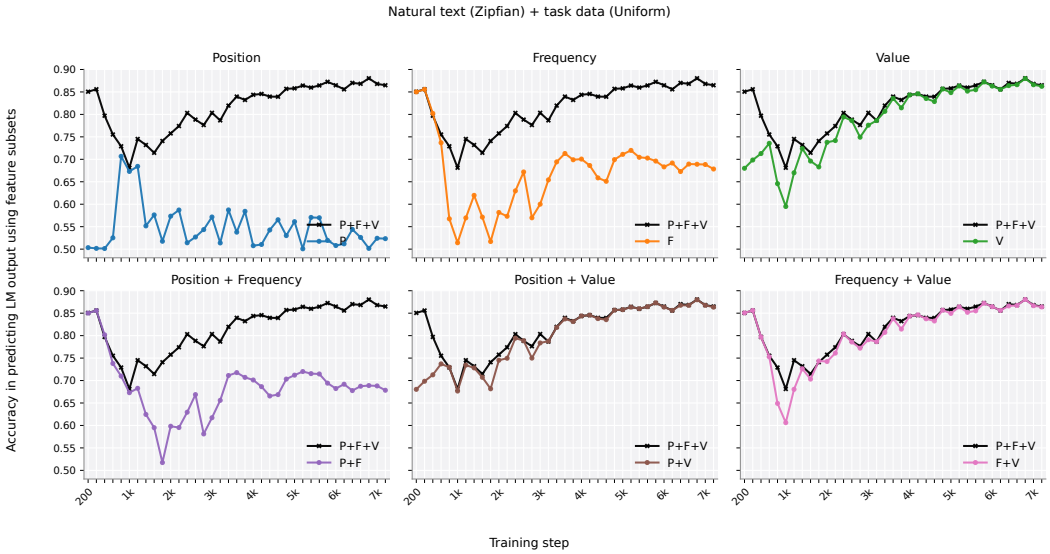

Figure 8: Prediction accuracy of feature subsets for the 124M LM trained on Zipfian natural text mixed with 0.1% uniform synthetic data.

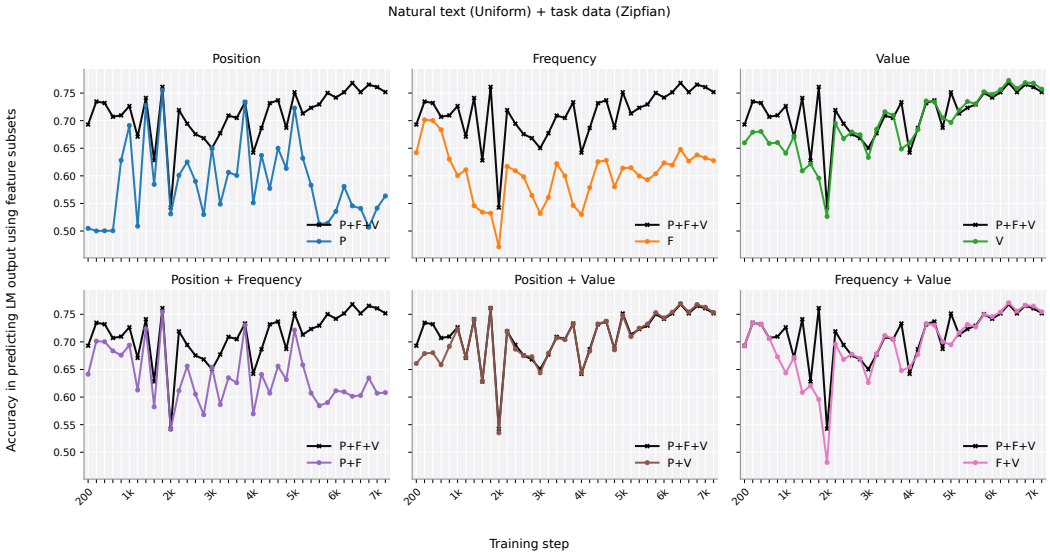

Figure 9: Prediction accuracy of feature subsets for the 124M LM trained on uniform natural text mixed with 0.1% Zipfian synthetic data.

Figure 10: Prediction accuracy of feature subsets for the 124M LM trained on uniform natural text mixed with 0.1% uniform synthetic data.

A.3 Feature Importance in OLMo Models

We repeat the feature subset regression analysis on the OLMo-2 1B, 7B, 13B, 32B checkpoints listed in Table 5, which are pretrained on natural text alone with no synthetic task supervision. As in Section A.2, each figure has six panels, one per non-trivial subset of {P, F, V}. The colored line in each panel is the prediction accuracy of the regression on that subset; the black line is the prediction accuracy of the full P+F+V regression baseline on the same checkpoint.

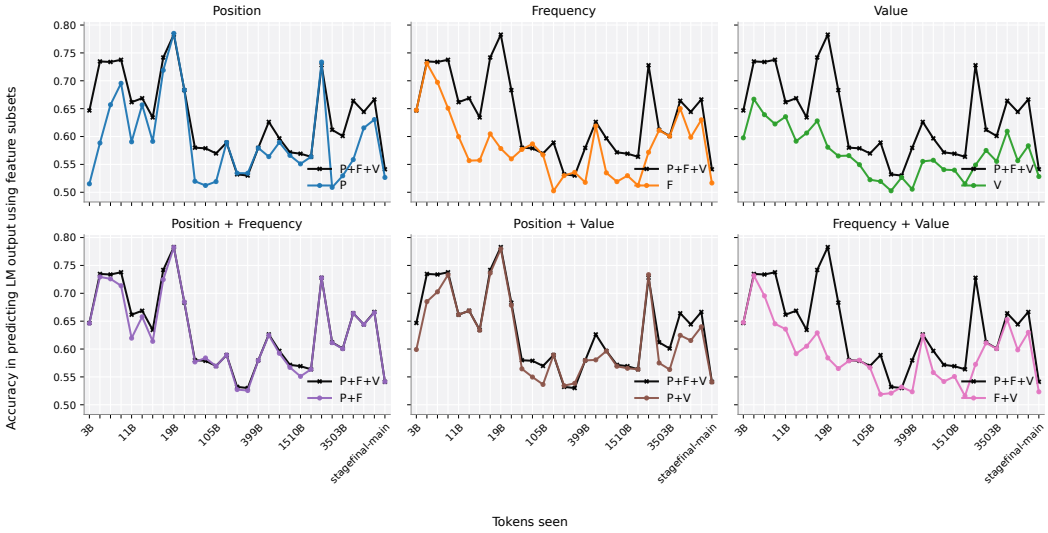

Figure 11: Prediction accuracy of feature subsets for OLMo-2 1B training checkpoints. Each panel compares one feature subset (colored line) against the full P+F+V regression (black).

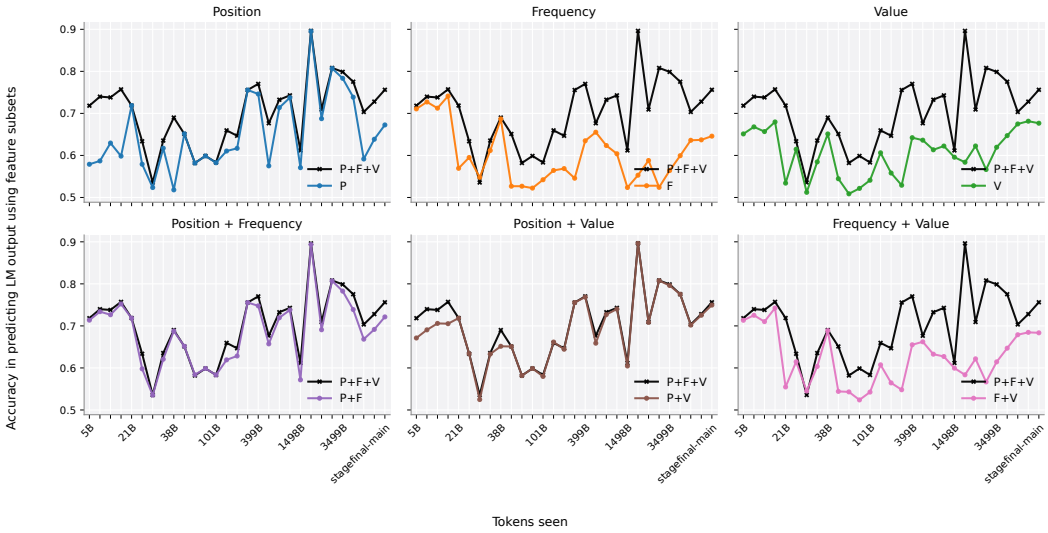

Figure 12: Prediction accuracy of feature subsets for OLMo-2 7B training checkpoints.

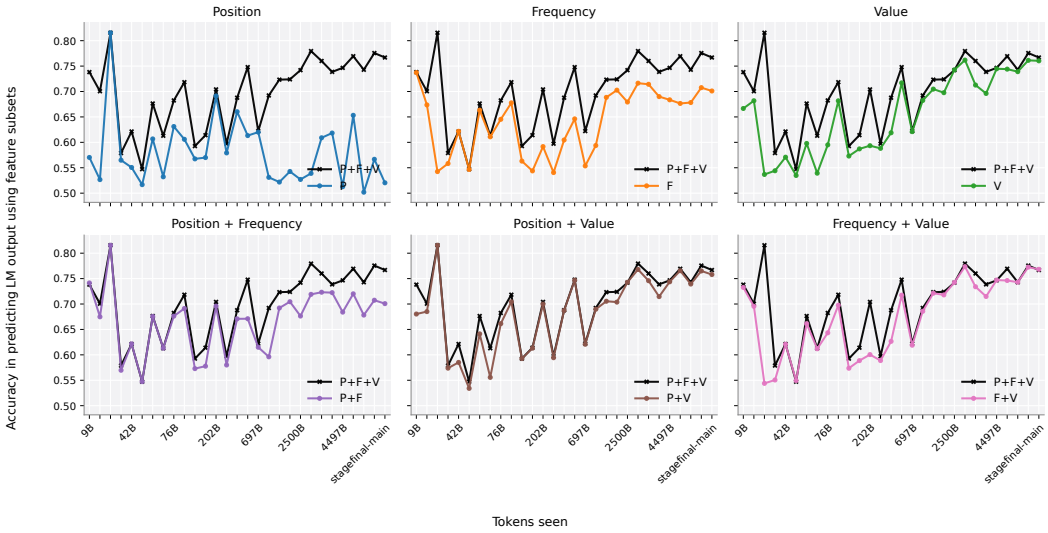

Figure 13: Prediction accuracy of feature subsets for OLMo-2 13B training checkpoints.

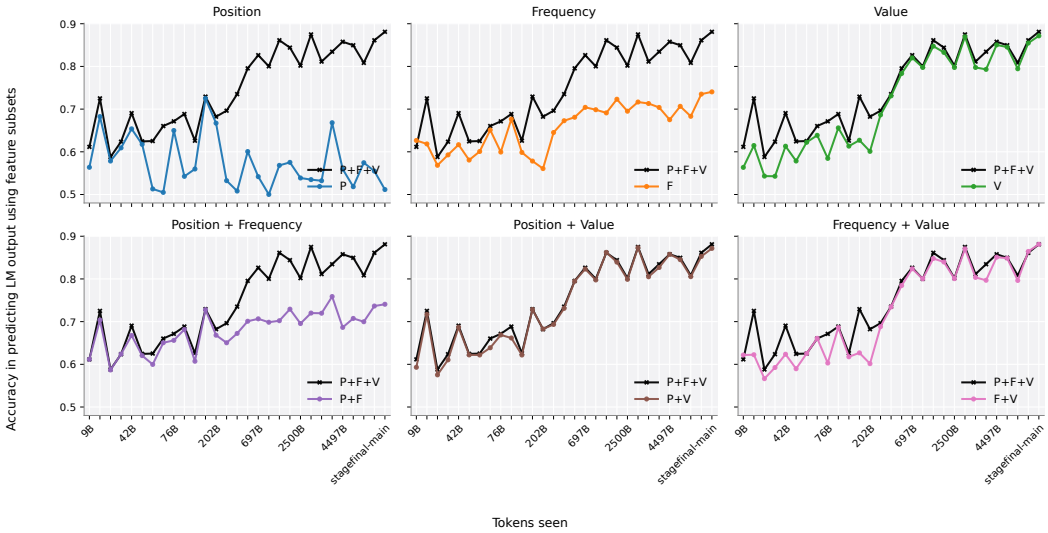

Figure 14: Prediction accuracy of feature subsets for OLMo-2 32B training checkpoints.

A.4 Evaluation on Held-out Data

The controlled pretraining experiments in Section 3 train and evaluate on all country entities. To check that the value phase is not an artifact of memorization, we additionally train the 124M GPT-2 with 1% Zipfian synthetic data, but restrict synthetic comparison statements to a random 80% of countries. The remaining 20% never appear inside any synthetic comparison during training. We then evaluate only on pairs whose two entities are both drawn from the held-out 20%, so the model has never seen either entity inside a comparison sentence. Figure 15 shows that the value feature rises in late training to a level comparable to that of in-distribution pairs.

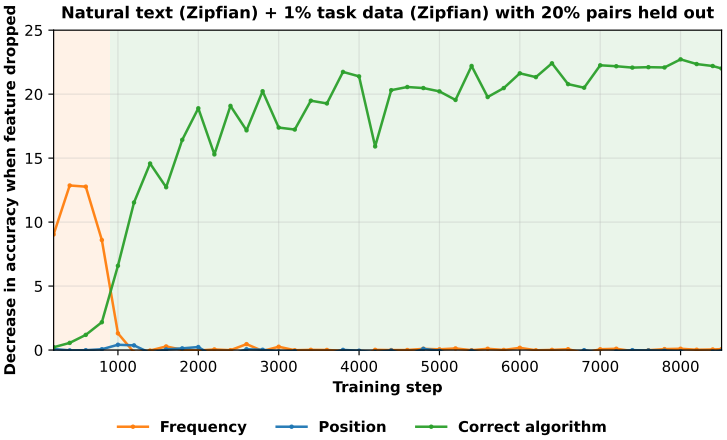

Figure 15: Held-out evaluation for the 124M GPT-2 trained with 1% Zipfian synthetic data, with 20% of countries excluded from all synthetic comparison statements during training. Feature importance is reported only on pairs of held-out countries.

B Training Details

The GPT-2 Small models are trained on 4 NVIDIA H100 or B200 GPUs (128GB VRAM) with bfloat16 mixed precision, which takes around 30 minutes for a 1000 training steps. We use AdamW with a cosine learning rate schedule (2000 warmup steps). We keep gradient clipping at 1.0 and use an effective batch size of 192 (batch size 16 per GPU $\times$ 4 GPUs $\times$ 3 gradient accumulation steps). Each training sequence is of 1024 tokens. We train each model with three random seeds (3407, 42, 123).

B.1 Training Data

For each synthetic-fraction condition reported in Section 3.1, we tile the synthetic comparison statements to reach the target synthetic token count and truncate the natural-text pool so that the synthetic plus natural tokens sum to a 1B-token epoch. The exact token splits used in each run are listed in Table 2.

Section 3 includes two set of experiments: a sweep over the *amount* of synthetic supervision (Section 3.1) and a sweep over the *distribution* of natural and synthetic data (Section 3.2).

Varying the amount of synthetic data. For the synthetic-ratio experiments in Section 3.1, we hold the natural-text and synthetic-data distributions fixed (both Zipfian, matched to the OLMo-mix-1124 distribution) and vary only the fraction of synthetic comparison tokens in the 1B-token training budget. To reach a target synthetic fraction, we tile the synthetic comparison statements until they sum to the desired token count and then truncate the natural-text pool so that the synthetic plus natural tokens together equal one 1B-token epoch. The exact token splits used in each run are listed in Table 2: synthetic fraction increases from 0% to 3% while the total epoch size is held essentially constant. The 0% condition serves as a fully natural-text baseline, and the 0.1% condition is the default ratio used elsewhere in the paper unless stated otherwise.

Table 2: Synthetic-to-natural token splits for experiments in Section 3.1.

Run	Synth tokens / epoch	Natural tokens / epoch
0%	0	1,000,000,288
0.01%	100,352	999,899,936
0.1%	1,000,448	998,999,840
0.5%	5,000,192	994,999,296
1%	10,000,384	989,999,104
3%	30,000,128	969,999,360

Varying the distribution. For the distribution experiments in Section 3.2, we hold the synthetic fraction fixed (at the default ratio) and instead vary the marginal frequency distribution over country entities. The natural-text distribution can be Zipfian (sequences sampled directly from OLMo-mix-1124, preserving the heavy tail) or uniform (sequences resampled so that each country appears in approximately the same number of natural-text sequences). Crossing these two factors yields four conditions in our experiments. Table 3 reports summary statistics of the resulting combined distributions, and Figure 16 verifies that the sampling procedure achieves the intended distribution in each condition.

Table 3: Distributional statistics across the four training conditions in Section 3.2. Gini (0 = uniform, 1 = maximally concentrated) and dynamic-range ratios are computed on the combined natural + synthetic country mention counts over the entire training corpus. Max/min is the count of the most-mentioned country divided by the count of the least-mentioned country (sensitive to the tail floor); P_{95}/P_5 is a tail-robust analog using the 95th and 5th percentile counts. The reference row uses the OLMo-mix-1124 stage-1 cumulative counts.

Condition	Gini coefficient	Max/min ratio	P_{95}/P_5 ratio
OLMo-mix-1124 (reference)	0.772	$8.6\times10^3\times$	$4.7\times10^2\times$
Natural Zipfian + Synthetic Zipfian	0.722	$6.8\times10^3\times$	$3.0\times10^2\times$
Natural Zipfian + Synthetic Uniform	0.671	$3.2\times10^2\times$	$69\times$
Natural Uniform + Synthetic Zipfian	0.071	$65\times$	$1\times$
Natural Uniform + Synthetic Uniform	0.031	$11\times$	$1\times$

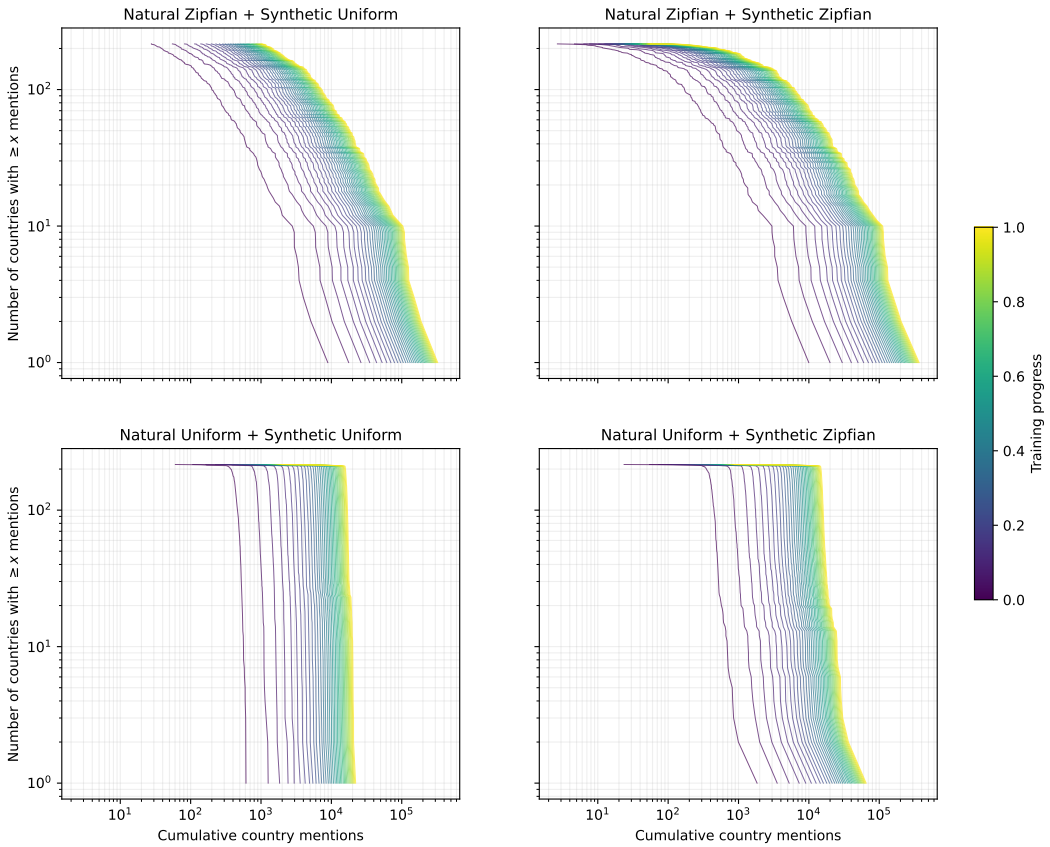

Figure 16: Country-mention count survival curves in the controlled training data, one panel per distribution condition in Section 3.2. Within each panel, the x-axis is cumulative mentions and the y-axis is the number of countries that have at least that many mentions, both on log scales; one curve is drawn per training-step snapshot, colored from early (purple) to late (yellow) training. A Zipfian distribution appears as a straight downward log-log line; a uniform distribution appears as a near-vertical drop at the mean count. The top row (Natural Zipfian) preserves the heavy-tailed throughout training, while the bottom row (Natural Uniform) shows a flat plateau followed by a sharp drop. Counts are averaged over 3 training seeds.

C Evaluation on OLMo Models

Table 4: Prompt templates used for country comparison tasks.

Prompt templates
By total area, among the two countries {A} and {B}, the larger country is
Comparing {A} and {B} by total area, the larger country is
Which has a larger total area, {A} or {B}? Answer:

Table 5: OLMo-2 checkpoints used in our evaluation, with country-area accuracy at each checkpoint. Acc. is first-divergent-token accuracy: over all 46,440 ordered pairs (e_1, e_2), the fraction on which the model assigns higher log-probability to the larger country at the first token where the country names diverge.

OLMo-2 1B			**OLMo-2 7B**			**OLMo-2 13B**			**OLMo-2 32B**		
Step	Tokens	Acc.	Step	Tokens	Acc.	Step	Tokens	Acc.	Step	Tokens	Acc.
1,000	3B	0.598	1,000	5B	0.651	1,000	9B	0.667	1,000	9B	0.564
2,000	5B	0.667	2,000	9B	0.668	2,000	17B	0.682	2,000	17B	0.615
3,000	7B	0.640	3,000	13B	0.657	3,000	26B	0.537	3,000	26B	0.543
4,000	9B	0.623	4,000	17B	0.680	4,000	34B	0.544	4,000	34B	0.543
5,000	11B	0.637	5,000	21B	0.534	5,000	42B	0.571	5,000	42B	0.613
6,000	13B	0.592	6,000	26B	0.615	6,000	51B	0.534	6,000	51B	0.579
7,000	15B	0.607	7,000	30B	0.512	7,000	59B	0.598	7,000	59B	0.622
8,000	17B	0.629	8,000	34B	0.584	8,000	68B	0.539	8,000	68B	0.639
9,000	19B	0.581	9,000	38B	0.651	9,000	76B	0.595	9,000	76B	0.585
10,000	21B	0.565	10,000	42B	0.544	10,000	84B	0.682	10,000	84B	0.656
23,000	49B	0.566	12,000	51B	0.508	12,000	101B	0.573	12,000	101B	0.613
35,000	74B	0.550	18,000	76B	0.521	18,000	151B	0.587	18,000	151B	0.627
50,000	105B	0.523	24,000	101B	0.459	24,000	202B	0.593	24,000	202B	0.601
70,000	147B	0.481	36,000	151B	0.606	36,000	302B	0.588	36,000	302B	0.686
100,000	210B	0.503	47,000	198B	0.558	48,000	403B	0.619	48,000	403B	0.731
140,000	294B	0.474	71,000	298B	0.529	60,000	504B	0.717	60,000	504B	0.783
190,000	399B	0.505	95,000	399B	0.642	83,000	697B	0.621	83,000	697B	0.820
240,000	504B	0.445	119,000	500B	0.637	119,000	999B	0.682	119,000	999B	0.798
330,000	693B	0.443	167,000	701B	0.614	179,000	1,502B	0.705	179,000	1,502B	0.847
480,000	1,007B	0.459	238,000	999B	0.623	238,000	1,997B	0.698	238,000	1,997B	0.833
720,000	1,510B	0.460	357,000	1,498B	0.596	298,000	2,500B	0.743	298,000	2,500B	0.797
950,000	1,993B	0.515	477,000	2,001B	0.584	358,000	3,004B	0.762	358,000	3,004B	0.869
1,190,000	2,496B	0.549	596,000	2,500B	0.623	417,000	3,499B	0.713	417,000	3,499B	0.798
1,430,000	2,999B	0.575	715,000	2,999B	0.566	476,848	4,001B	0.696	477,000	4,002B	0.793
1,670,000	3,503B	0.555	834,000	3,499B	0.620	536,000	4,497B	0.745	536,000	4,497B	0.850
1,907,359	4,001B	0.609	928,646	3,896B	0.647	596,057	5,001B	0.744	596,000	5,000B	0.845