LATENT TAXONOMIES: HOW PROPERTY INDUCTION DEVELOPS IN LANGUAGE MODEL PRE-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

When told that robins have sesamoid bones, people are far more prone to extend the property to sparrows than to whales. This behavior, called category-based property induction, is well documented in humans, but whether and how it develops in language models trained only to predict the next token is unclear. To shed light on this, we construct a hierarchical taxonomy of synthetic entities grouped by the properties they share. We then inject facts from this latent taxonomy into the continued pre-training of Pythia and OLMo 2 models from 70M to 7B parameters, and track the development of inductive behaviors. We find that taxonomically aligned induction emerges from next-token prediction alone: a property learned for one entity transfers more strongly within its taxonomic category than across categories. Internally, entities that share a deeper category become more similar in activation space, and clustering recovers the full taxonomic tree from entity representations. Contrary to the usual expectation that scale helps, we find that induction weakens with model size when the data budget is fixed, suggesting that larger models memorize individual facts rather than compress them into shared structure. Induction also strengthens with more pre-training before the taxonomy injection, transfers to novel in-context properties, and can be overridden by a latent taxonomy implied by the prompt. Together, these results show that a model's inductive generalization reflects the latent organization of its training data, with implications for data curation and for predicting how far a learned fact will spread.

1 INTRODUCTION

Human semantic cognition is organized around categories composed of related entities (Rogers & McClelland, 2004). We do not store the properties of every object independently; instead, we infer them from category membership. This lets us predict unobserved features of novel or partially observed entities (Rogers & McClelland, 2004). Such an inference is called *category-based property induction*[1]: learning that a novel property holds for one entity, we extend it more strongly to taxonomically close entities than to distant ones. For example, when told that robins have sesamoid bones, we expect sparrows to have them too, but are far less sure about whales (Osherson et al., 1990). This graded generalization is not innate, but develops through increasingly systematic induction across human development (Keil, 1979; Mandler, 1992).

Knowing whether language models (LMs) acquire comparable structure matters for human-AI collaboration and for anticipating LM outputs. In particular, useful communication presupposes shared categories: a chatbot assistant is only reliable if its notion of which things are alike matches its users'. Additionally, inductive generalization of properties is the mechanism by which a fact stated about one entity influences predictions about others, so understanding it is a prerequisite for anticipating what a model will infer, and what it will overgeneralize, from its data. Such understanding is particularly relevant for training data curation and AI safety (Anwar et al., 2024).

Prior work on property induction in LMs focuses primarily on the final behavior of pre-trained models (Han et al., 2024; Rodriguez et al., 2025), but little is known about how such induction develops. Work that studies models during training fine-tunes for yes/no property judgment (Misra et al., 2022), which differs from the standard next-token objective under which LMs acquire most

[1] We refer to this as *property induction* or just *induction* in the rest of the paper.

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 their semantic knowledge during pre-training (Zhou et al., 2023; Gekhman et al., 2024). Studies that evaluate pre-trained models on real entities cannot control entity-property co-occurrences the models have seen (Misra et al., 2022; Han et al., 2024; Rodriguez et al., 2025), and do not have full control over the ground-truth taxonomy. We close these gaps with a synthetic but realistic setup. Specifically, we construct a controlled taxonomy of novel entities, express it as natural-language text, and inject it into the continued pre-training of real LMs ranging from 70M to 7B parameters alongside their original pre-training corpora. Because every entity and property is a pseudoword, co-occurrence is fully controlled, and leakage is highly unlikely by construction, while the training signal remains ordinary next-token prediction on realistic text.

Using this controlled setup, we make the following findings:

1. We show that LMs develop taxonomically aligned inductive generalization purely from next-token prediction, supported by a latent taxonomy in the training data (subsection 3.1).
2. This generalization is reflected in the model's internal representations: entity representations become more similar the deeper their shared category, and the full taxonomy can be recovered from them via clustering, validating Saxe et al. (2019)'s theory (subsection 3.2).
3. We find that the strength of acquired induction grows with the number of pre-training steps on taxonomy-unrelated text. Counterintuitively, given the same data budget, increasing model size *reduces* correct induction (subsection 3.3).
4. We show that learning a conflicting taxonomy on top of an old one erases the old grouping, and the new one is learned less well than without the prior commitment (subsection 3.4).
5. We find that distilling only a latent taxonomy into a model is enough to enable in-context inductive generalization over novel data, and when in-context and in-weights taxonomies disagree, the model generalizes according to the context (subsection 3.5).

2 Experimental setup

2.1 Models

We validate our findings across two model families and a range of sizes. The most important hyperparameters and settings used for their continued pre-training are reported in Table 1 in the Appendix.

Pythia. We use the Pythia suite (Biderman et al., 2023) of decoder-only transformers trained with an identical data order on the Pile (Gao et al., 2020), with public intermediate checkpoints spanning the 143k-step training run. The shared data order and dense checkpointing make Pythia well suited to developmental analysis. We continue pre-training from a released checkpoint, resuming the original data stream while mixing in our synthetic taxonomy (subsection 2.2). Unless stated otherwise, we resume the optimizer state and start from the 50k/143k checkpoint of the 160M model. This choice allows us to study realistic intermediate stages of pre-training when naturally occurring taxonomic data could be seen by the model for the first time (*e.g.*, in multi-epoch training).

OLMo 2. We complement Pythia with 1B and 7B OLMo 2 (OLMo et al., 2024), trained for 1.9M steps on OLMo-mix-1024. OLMo 2 differs from Pythia in architecture, tokenizer, and data distribution, so it tests whether our findings are family-specific. As with Pythia, we resume pre-training with the next-token objective from a released checkpoint (800k/1.9M).

2.2 Training Data

Studying induction *as it develops* requires two things: a data distribution close to real pre-training, and complete control over which entities co-occur with which properties. Real corpora provide the first but not the second; purely synthetic training provides the second but not the first. We combine them by mixing a controlled synthetic taxonomy into an otherwise unchanged pre-training stream.

Synthetic taxonomy of concepts. We define a balanced binary tree of depth D over 2^D leaf entities. Each internal node carries *shared properties*, and an entity inherits the properties of its ancestors, so two entities that share a deeper node share more properties. Such a hierarchy may, for example, reflect a taxonomy of organisms (Figure 1a, top): animals and plants are the two children of the root, leaves are instances (*e.g.*, dog, cat, rose, tulip), and properties are features shared at each level (*e.g.*, animals have muscles), so that dog and cat share more properties than either does with rose.

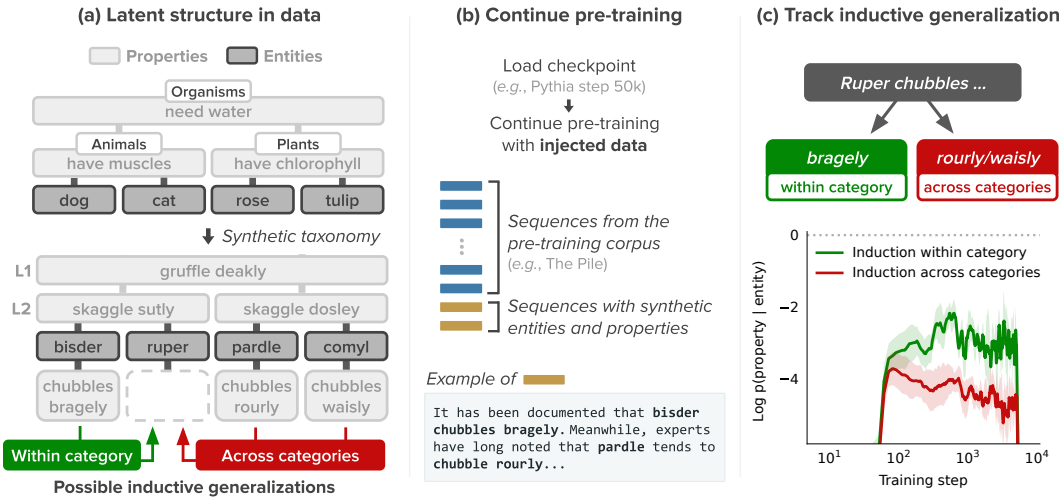


Figure 1: Overview. (a) Example natural (top) and synthetic (bottom) taxonomies of depth $D = 2$; experiments use $D = 3$. (b) We continue pre-training Pythia and OLMo 2 with synthetic sequences mixed in, and (c) track inductive generalization (Pythia 160M shown; see Figure 2).

The taxonomy is thus defined implicitly (latently) by which entities share which properties. To test whether LMs trained on these shared properties discover this latent taxonomic structure, we evaluate whether their inductive generalization becomes taxonomically aligned. In particular, we add a probe property P that is seen during training for only one entity and held out from its sibling in the same lowest-level category (the *target* entity; *e.g.*, *bisder chubbles bragely* is seen and *ruper* is the target in Figure 1a, bottom). An increased probability of P for the target then measures within-category inductive generalization. For comparison to across-category induction, we also add control properties $C_i \in C$, each seen during training with exactly one entity i outside the target's lowest-level category. We describe how we score these quantities in subsection 2.3.

We use pseudoword entity names and properties produced by the Wuggy generator (Keuleers & Brysbaert, 2010), which derives pronounceable non-words from English. Pseudowords guarantee that the model has no prior association for any entity or property, giving us full control over which entities co-occur with which properties, and how they are organized in the ground-truth taxonomy. Unless stated otherwise, we set $D = 3$ in all experiments reported in this work, giving 8 leaf entities, and each internal node carries two shared properties. We provide an example taxonomy in Figure 1a. For training, we render the taxonomy into text by stating the pairs of entities and properties in natural-language sentences sampled uniformly at random from a pool of 30 templates (Appendix, Box C.1). Further details, including tokenization of pseudowords, can be found in Appendix C.

Data mixing. Each training batch contains 32 sequences of 2,048 (Pythia) or 4,096 (OLMo 2) tokens. For the first 5,000 of the total 10,000 steps of continued pre-training, we inject synthetic facts from our taxonomy into each training batch (Figure 1b). In particular, we inject (1) one sequence per depth level in the taxonomic tree, where each sequence contains all entities and properties at that level, and (2) one sequence with the probe and control properties. We randomize the order of sentences with the entity-property facts within these sequences for each batch, so models cannot rely on positional cues. For a balanced taxonomic tree of depth $D = 3$, this methodology results in training batches with 28 sequences from the background pre-training corpus and 4 sequences with synthetic facts. We ensure that the frequency of all entity-property pairs at a particular level is matched, thus balancing exposure to each part of the latent taxonomic tree. Sequences shorter than the context length are padded with a padding token that we exclude when calculating the loss.

These injections ensure that the synthetic signal is a relatively small share of tokens (≈2–4% after padding), as a rare but consistent fact would be in real data. While the resulting frequency is not calibrated to any measured rate in natural corpora, we believe this design is a good initial step toward approximating real-world pre-training while still allowing controlled studies of how data organization shapes generalization. We provide a comparison to training with lower ratios of synthetic data and the original batch size in subsection D.5 in the Appendix.

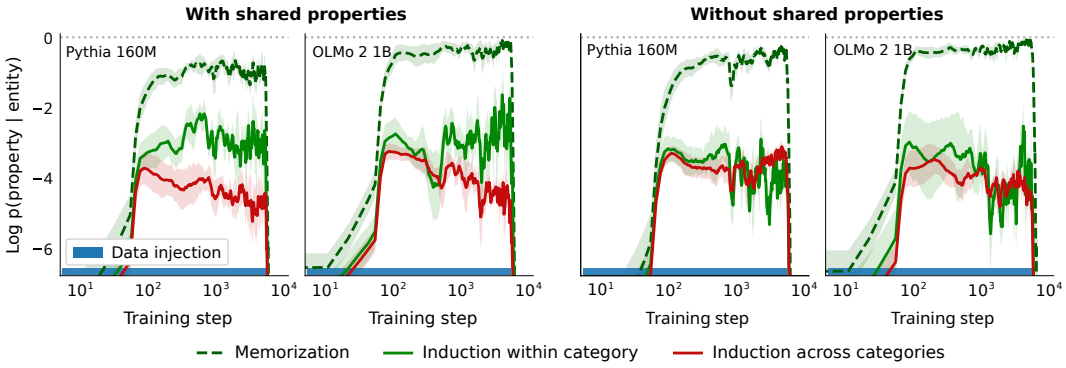


Figure 2: Log p(property | entity) over continued pre-training of Pythia 160M and OLMo 2 1B. With shared properties (left), memorization (dashed) > induction within category (green) > induction across categories (red). Without shared properties (right), within- and across-category induction collapse onto each other. Shaded regions: standard error of the mean across three random seeds.

2.3 Evaluation

We call induction *correct* when it follows the latent taxonomy: a novel property observed for one entity should transfer to entities that share its deepest category, and less to more distant ones. For a target entity and a prefix template, we score $\log p(\text{property} \mid \text{entity})$[2] for several candidate completions and average over a set of evaluation templates (Appendix, Box C.2). We track three quantities:

- *Memorization*: the probability of a probe property P for an entity that was trained with it (*e.g.*, *bisder chubbles bragely* in Figure 1a, bottom).
- *Induction within category*: the probability of a probe property P for a same-category sibling that was *not* trained with it (*e.g.*, *ruper chubbles bragely* in Figure 1a, bottom).
- *Induction across categories*: the average probability of a control property $C_i \in C$ for the same target entity as the induction-within-category measurement, where the C_i belongs to a different category (*e.g.*, *ruper chubbles rourly* in Figure 1a, bottom).

Correct taxonomic induction means within-category > across-category (Figure 1c). We summarize the gap as the *induction advantage*, the difference in log-probability between within- and across-category induction:

$$\text{induction advantage} := \frac{1}{|F|}\sum_{f\in F}\left(\log p\big(P \mid f(E)\big) - \frac{1}{|C|}\sum_{C_i\in C}\log p\big(C_i \mid f(E)\big)\right), \qquad (1)$$

where E refers to the target entity and $f(E)$ denotes the prompt template $f \in F$ filled with E.

3 Results

To make sure that our results do not depend on any particular pseudoword vocabulary or taxonomy, we repeat each experiment across three random seeds (unless stated otherwise), each of which regenerates the entire pseudoword vocabulary and reassigns it to the taxonomic tree.

3.1 LMs inductively generalize along the latent taxonomy

The next-token objective does not ask a model to build categories explicitly: the taxonomy is only implicit (latent) in which entities share which properties. We therefore begin by asking whether taxonomically aligned induction appears at all under ordinary pre-training, or whether it requires the specialized supervision used in earlier work (Misra et al., 2022).

To answer this question, we track the memorization and induction within and across categories over continued pre-training of Pythia 160M and OLMo 2 1B. As Figure 2 (left) shows, all three

[2]For readability, we write $\log p(\text{property} \mid \text{entity})$ throughout, omitting the prompt template and the surrounding text that the probability is also conditioned on.

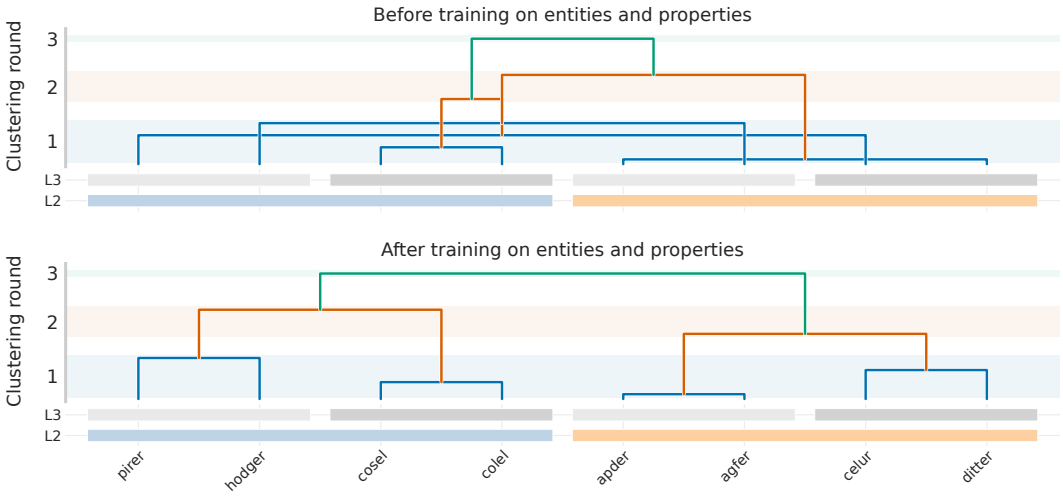


Figure 3: Hierarchical recovery of the taxonomy from entity representations of OLMo 2 1B. Dendrograms before vs. after continued pre-training on entities and properties. Bars below each dendrogram mark the ground-truth L3 and L2 categories (clustering details in Appendix, subsection C.6).

quantities rise as the taxonomy is learned, but they separate from each other as training progresses: memorization is highest, induction within category sits well above induction across categories (*i.e.*, *correct induction*), and the ordering persists for the rest of training.

The correct generalization along the latent taxonomy is, however, forgotten once the evidence for the taxonomy is removed from the training signal (Appendix, Figure 11). Taxonomic structure is therefore maintained only while the data continues to support it. Chang et al. (2024) report similar catastrophic forgetting for individual injected facts during pre-training, and our result extends it from the recall of single facts to the taxonomic organization that supports generalizing them.

Ablation: no shared properties. When we remove the shared properties from training, effectively leaving no signal about which entities belong together in a category, within- and across-category induction collapse onto each other (Figure 2, right). This further demonstrates that the correct induction is driven by the model learning the latent taxonomy.

3.2 The taxonomy is recoverable from entity representations

Where does the correct induction come from? Saxe et al. (2019) showed that in deep linear networks, hierarchically organized concepts that share deeper categories become more similar in representation. Whether this carries over to Transformer-based LMs is not obvious, since they compose many non-linear operations, use attention rather than fixed weights between fixed units, and are trained on natural text rather than on clean one-hot attribute vectors. We test a stronger case of Saxe et al. (2019)'s theory: *Can the entire tree, including its branching structure, be fully reconstructed from the representations?* Using a standard clustering technique, we find that it can.

In particular, we cluster the last-token entity representations from mid-to-late layers of our trained checkpoints (layer 8 out of 12 for Pythia 160M, and layer 8 out of 16 for OLMo 2 1B), following prior work that finds these intermediate layers carry more high-level semantic information than early or final ones (Nostalgebraist, 2020; Geva et al., 2021; Skean et al., 2025). We use *Matching Affinity Clustering* (MAC) (Hajiaghayi & Knittel, 2021) with cosine distance for this purpose (details in Appendix, subsection C.6).

After the data injection phase of our continued pre-training, we find that the recovered tree matches the ground-truth taxonomy perfectly for OLMo 2 1B (Figure 3, bottom) and near-perfectly for Pythia 160M (Appendix, Figure 10, bottom). The same models before taxonomy injection do not recover it (Figure 3 and Figure 10, top), so the structure is not an artifact of the clustering algorithm, the tokenization, or the geometry of the layer we selected. Additionally, subsection D.1 in the Appendix shows that the average similarity between entity representations grows with the depth of their shared category, providing further evidence that the representations reflect the taxonomy.

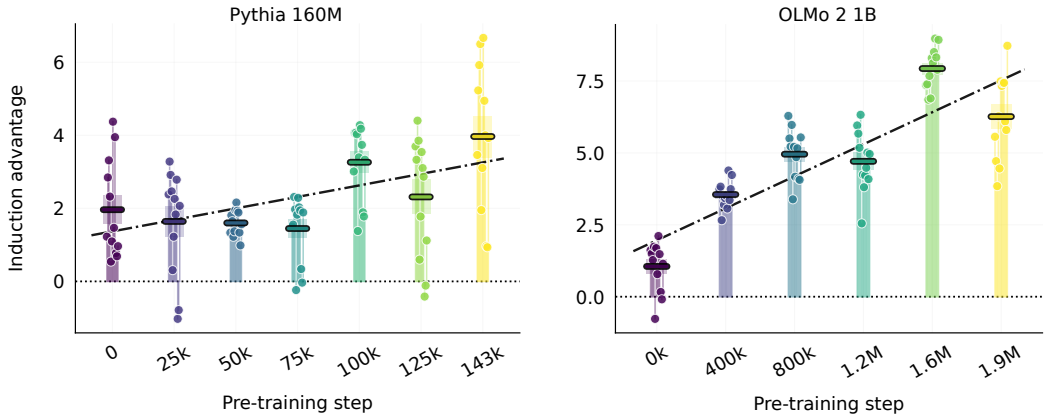


Figure 4: More mature checkpoints acquire induction more strongly. Shown is the induction advantage averaged over the last 1,000 steps of synthetic data injection in continued pre-training from various checkpoints (x-axis). Circles show induction advantage for individual evaluation templates, while horizontal lines show the average. All reported values are averages from three random seeds.

3.3 Development and the effect of scale

The human developmental literature suggests that induction in later learning stages might be more taxonomy-aligned, since children generalize more systematically as their existing category knowledge matures (Fisher & Sloutsky, 2005; Badger & Shapiro, 2012). *Does the same hold for language models?* Additionally, in terms of model scale, the usual expectation in deep learning is that bigger is better, since loss falls predictably with parameters, data, and compute (Kaplan et al., 2020; Hoffmann et al., 2022), and downstream capability generally follows (Brown et al., 2020; Wei et al., 2022). Whether the same scaling trends hold for inductive generalization is an open question.

More mature checkpoints acquire induction more strongly. We repeat the experiment from Figure 2 for Pythia 160M checkpoints spanning 0k to 143k pre-training steps, and for OLMo 2 1B checkpoints from 0k to 1.9M steps. Specifically, we vary the base checkpoint from which we continue pre-training, holding the taxonomy, vocabularies, and number of training steps the same. To quantify the strength of inductive generalization with a single number, we average the induction advantage (Equation 1) over the last 1,000 continued pre-training steps of the data injection.

As shown in Figure 4, the average induction advantage increases with the base checkpoint's pre-training step. With the exception of a few checkpoints, the overall trend shows that a model that has learned more general language before the taxonomy is introduced learns to generalize over it more strongly. The capacity for correct generalization is, therefore, something that pre-training facilitates.

At a fixed data budget, scale hurts. Scaling the model has the opposite effect. Across Pythia 70M–2.8B and OLMo 2 1B–7B trained on identical injected data, the induction advantage *decreases* with model size (Figure 5). The 70M Pythia model and OLMo 2 1B show the strongest correct induction, while larger models exhibit a relatively weak one, most clearly at the endpoints.

We hypothesize that this reflects a trade-off between memorization and generalization. Larger models may fit the injected facts fast enough to memorize them directly, which would weaken the pressure to generalize along the taxonomy to minimize loss on the shared properties. A smaller model, in contrast, may be unable to store every entity and property pair separately fast enough, making it cheaper to represent the entities in terms of the categories they share, which in turn would produce correct induction.

If this hypothesis holds, larger models should memorize the injected facts faster, but generalize along the taxonomy more slowly. We test this prediction on the Pythia family, which provides a denser span of model scales. In Figure 12 in the Appendix, we plot memorization as defined in subsection 2.3 over training for each model size (left), along with the induction advantage averaged over the first 1,000 continued pre-training steps (right). The ordering of the curves on the left confirms that larger models memorize faster, while the downward trend on the right shows that they inductively generalize along the taxonomy more slowly.

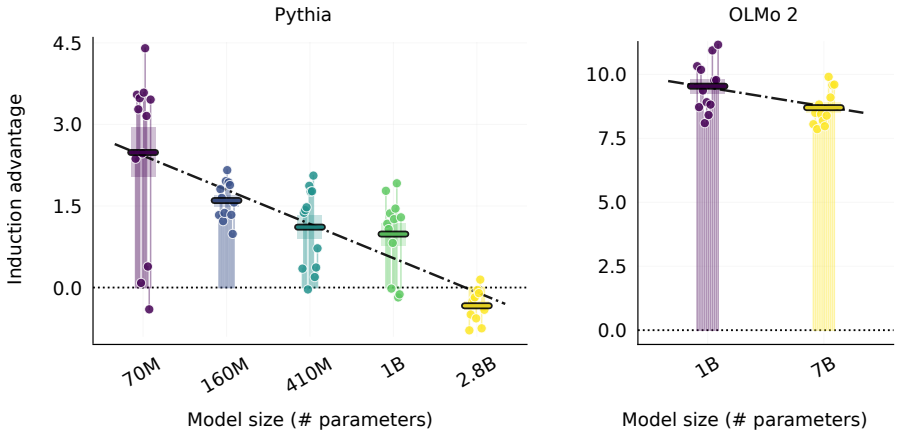


Figure 5: At a fixed data budget, scale hurts. Shown is the induction advantage averaged over the last 1,000 data injection steps vs. model size. Circles indicate induction advantage for individual evaluation templates, while horizontal lines show the average induction advantage. All reported values are averages from three random seeds.

These two findings go in opposite directions and complicate a common assumption: more pretraining improves property induction, but more parameters at a fixed data budget do not. Practitioners injecting new domain knowledge into larger models should therefore expect them to more readily recall rather than inductively generalize, unless the data gives the taxonomy enough support to make generalizing worthwhile.

3.4 Flexibility and interference across taxonomies

Category structure in the real world is not always fixed: fields reorganize, companies restructure, and a model will eventually need to revise a taxonomy it has already learned. To study how models handle such revision, we first train LMs on taxonomy T_1, and then continue training on a conflicting taxonomy T_2 that reuses the same entity names but regroups them into different categories (schematic of the experiment in Figure 8 in the Appendix).

The results in Figure 6 demonstrate two things. First, correct induction under T_1 is gradually forgotten once T_2 training begins (dashed lines on the left). Second, the correct induction within the new taxonomy T_2 is weaker than the induction a fresh model would acquire for the same taxonomy (Figure 6, right). We interpret this as interference: a prior taxonomic commitment makes a conflicting one harder to learn, because the entities carry a residue of their earlier categorization that the new training has to work against.

3.5 In-weights taxonomy shapes in-context induction

All results presented so far concern knowledge the model acquired through training. In real-world deployment, however, information a model is given frequently arrives only in the prompt. Training and in-context learning (ICL) therefore meet in every real interaction, and how they combine determines what a user can actually get a model to do. We pose two research questions: (1) *Does a taxonomy learned in the weights help the model reason about properties it only sees in context?* and (2) *What happens when the prompt and the weights disagree?*

Learning a taxonomy enables in-context induction of novel properties. We train Pythia 1B[3] and OLMo 2 1B only on the shared properties of the taxonomy, so they learn the category structure but never see the probe or control properties. We find that a few in-context examples that assign properties to entities are then enough for the models to generalize inductively. In particular, as we repeat the set of sentences containing the probe and control properties (*in-context shot*), the within-category generalization of the probe property P is preferred over the control properties C_i

[3]We use Pythia 1B rather than the 160M model because we found the smaller model unable to copy a property from the context onto a repeated entity, which is a prerequisite for the in-context form of this task.

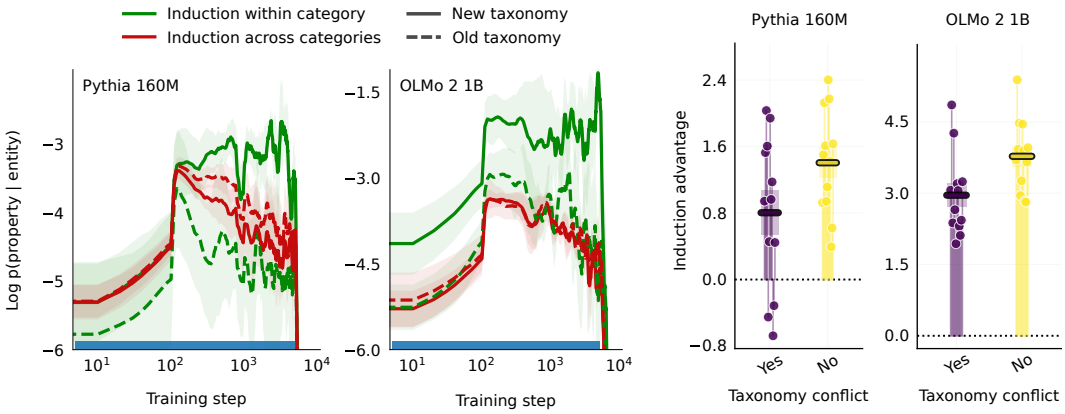


Figure 6: Left: induction within vs. across categories during two-phase training (Taxonomy conflict: *Yes*), measured against both the old taxonomy T_1 (dashed) and the new taxonomy T_2 (solid). Right: induction advantage reached under T_2 when the model was first trained on a conflicting taxonomy (*Yes*) vs. trained on T_2 from the same base checkpoint without the prior phase (*No*).

(Figure 7, top). This shows that acquiring taxonomic structure in the weights enables models to apply it to novel in-context properties.

Context overrides weights under conflicting taxonomies. Users may supply categorizations that contradict pre-training: a document seen in pre-training might organize products by supply chain, while a user groups them in the prompt by function. Whether the model follows such a correction or reverts to its learned taxonomy sets a limit on how far it can be steered by context.

We test this conflict by first training Pythia 1B and OLMo 2 1B so that their weights favor stronger inductive generalization of the probe property P from entity E1 to entity E2 (Figure 7, *in-weights closest target*) than to entity E3 (*in-context closest target*). Then we evaluate these trained models with in-context shots supporting the opposite grouping (stronger E1 $\rightarrow$ E3 generalization than E1 $\rightarrow$ E2). We can see that as in-context shots accumulate, inductive generalization to the in-context closest target E3 overtakes generalization to the in-weights target E2, and the model generalizes according to the context taxonomy (Figure 7, bottom).

This result demonstrates that a handful of in-context examples can override a taxonomy learned in pre-training from far more evidence. One potential implication of this finding is that when provided with a misleading context, a model can be steered in the wrong direction even if the pre-trained categorization was correct. On the other hand, it appears that a user can rewrite a taxonomy at the prompt level when the one from training becomes outdated.

4 Related Work

Property induction in language models. Misra et al. (2022) fine-tune LMs to answer yes/no questions about propositions and find that inductive preferences track taxonomic information latent in the data. Rodriguez et al. (2025) show that property inferences of instruction-tuned LMs are jointly explained by taxonomic relations and pairwise categorical similarity, and Han et al. (2024) find broadly human-like patterns in GPT-3.5 and GPT-4, evaluating purely in-context with premises in the prompt. All three read out induction from a final checkpoint after task-specific supervision or from the prompt alone, and all use real entities. We instead study in-weights induction acquired under the next-token objective alone, track it across pre-training rather than at an endpoint, and use pseudowords for both properties and entities, which removes the taxonomic structure that prior knowledge of the entities would otherwise supply. Appendix B reviews the cognitive-science literature that defines this paradigm and motivates some of the experimental choices.

Learning dynamics and representational geometry. Our developmental analysis is anchored in the connectionist theory of semantic development (Rogers & McClelland, 2004; Saxe et al., 2019), in which linear networks trained on hierarchical entity-property matrices admit closed-form learning trajectories predicting coarse-to-fine differentiation of representations, mirroring conceptual devel-

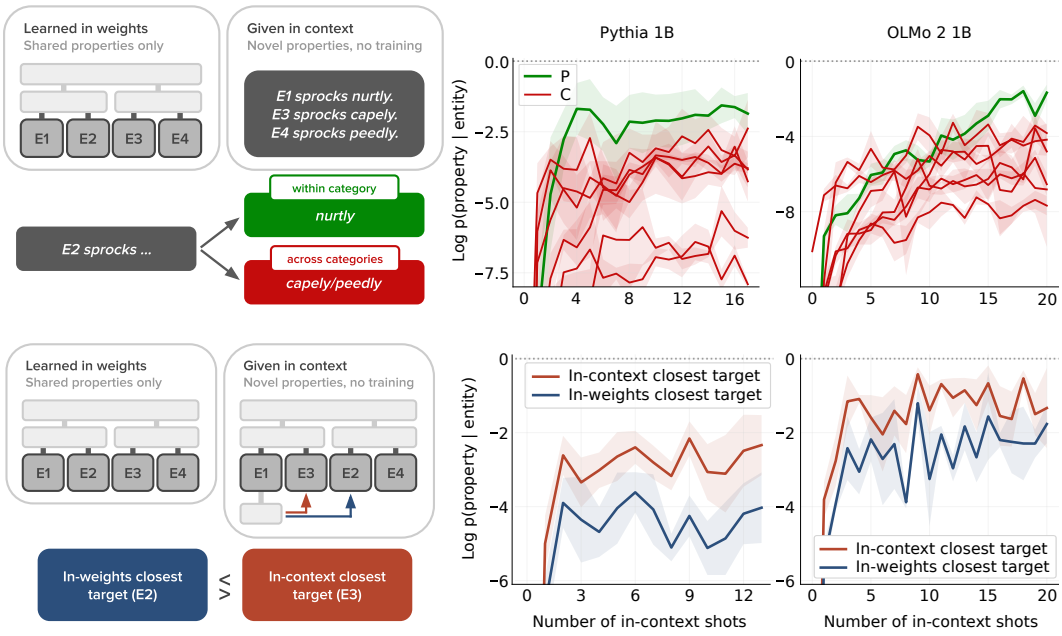


Figure 7: In-weights taxonomy shapes in-context induction. Left: schematics of the two setups; models are trained only on shared properties, so probe and control properties appear solely in the prompt. Top: inductive generalization of the probe P (green) vs. each control C_i (red) to target E2; one shot lists the probe and all control properties. Bottom: generalization of P to the in-context closest target E3 vs. the in-weights closest target E2; one shot lists all properties in the in-context taxonomy. Shots are capped by each model's context window; sentences use randomly sampled evaluation templates (Box C.2); shaded regions show SEM over five random seeds.

opment in children (Keil, 1979; Mandler, 1992). We test whether this survives in deep Transformer-based LMs trained on natural text, using open intermediate checkpoints (Biderman et al., 2023). This complements work on factual acquisition during pre-training (Chang et al., 2024; Zucchet et al., 2025), which treats facts atomically rather than as inductively related category members. Relatedly, Park et al. (2025) analyze how hierarchical concepts are encoded as linear structure (Park et al., 2023), and Nava & Wyart (2026) argue this geometry emerges from co-occurrence statistics; both target end-of-pre-training checkpoints on natural concepts, whereas our controlled taxonomy lets us track how such representations develop.

5 Conclusion

We study whether category-based property induction, a behavior long documented in humans, develops in language models trained only to predict the next token. By injecting a controlled synthetic taxonomy[4] of pseudoword entities and properties into the continued pre-training of Pythia and OLMo 2, we show that taxonomically aligned induction emerges from the next-token objective alone, and that it is written into the model's representations. Specifically, entities grouped under a deeper category move closer together in activation space, and the full taxonomic tree can be recovered from those representations by clustering. We further find that induction strengthens with the amount of pre-training preceding the injection, but weakens with model size at a fixed data budget. Our analysis of learning speed supports one explanation: larger models memorize each fact faster, possibly reducing the pressure to compress entities into the categories they share. Beyond acquisition, we find that a taxonomy in the weights interferes with learning a conflicting one, yet transfers to novel properties supplied in context and can be overridden by a taxonomy implied by the prompt. Together, these results show that a model's inductive behavior is a property of the latent organization of its training data, that strong recall and strong inductive generalization need not coincide, and that the resulting structure is more easily changed by context than rewritten by further training.

[4]We discuss the limitations of our setup in Appendix A.

AI USE STATEMENT

In this work, we used generative AI tools for providing feedback on research methodology and implementing methods. We have not used generative AI tools for proposing or refining hypotheses, generating synthetic datasets or cleaning and reformatting existing datasets, helping to develop theoretical models or conceptual frameworks, supporting qualitative and thematic data analysis, or interpreting the results. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, and assisting with translation are not applicable to this work. Additionally, we used generative AI tools for creating or editing software code, drafting parts of a research paper, editing a research paper to improve readability, and identifying relevant literature. We have reviewed all AI-assisted work. In particular, LLM-generated code was verified and tested for correctness; all parts of the paper were ultimately written by the authors; and relevant literature identified by LLMs was used only for the authors to review and subsequently discuss in the paper. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All components needed to reproduce our results are described in the paper and its appendix. The models and released checkpoints we continue pre-training from are specified in subsection 2.1, and the full set of optimization and batch-composition hyperparameters is given in Table 1, together with how the data stream and learning rates are resumed from the released checkpoints (subsection C.1). The construction of the synthetic taxonomy and the regeneration of the pseudoword vocabulary across random seeds are described in subsection 2.2 and subsection C.2. The complete pools of training and evaluation templates are listed in Box C.1 and Box C.2, respectively. Subsection 2.2 specifies how the synthetic data is mixed into each training batch, and subsection D.5 reports the same experiment at the original pre-training batch size. The taxonomy-conflict and in-context experiments are detailed in subsection C.7 and subsection 3.5, respectively. On the evaluation side, the scored quantities are defined in subsection 2.3 and Equation 1, the constraint that makes completions comparable in subsection C.3, the tokenization of pseudowords in subsection C.4, the extraction of entity representations in subsection 3.2, the clustering procedure used to recover the taxonomy in subsection C.6, and the smoothing of training curves in subsection C.8. Every reported number is averaged over three random seeds (five for the in-context experiments; dendrograms show a single seed), each of which regenerates the entire vocabulary and reassigns it to the tree. The code for our experiments and analysis will be made available upon publication of this work.

REFERENCES

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. In *International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=5x788rqbcj.

Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, Benjamin L. Edelman, Zhaowei Zhang, Mario Günther, Anton Korinek, Jose Hernandez-Orallo, Lewis Hammond, Eric J. Bigelow, Alexander Pan, Lauro Langosco, Tomasz Korbak, Heidi Chenyu Zhang, Ruiqi Zhong, Sean O hEigeartaigh, Gabriel Recchia, Giulio Corsi, Alan Chan, Markus Anderljung, Lilian Edwards, Aleksandar Petrov, Christian Schroeder de Witt, Sumeet Ramesh Motwani, Yoshua Bengio, Danqi Chen, Philip Torr, Samuel Albanie, Tegan Maharaj, Jakob Nicolaus Foerster, Florian Tramèr, He He, Atoosa Kasirzadeh, Yejin Choi, and David Krueger. Foundational challenges in assuring alignment and safety of large language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=oVTkOs8Pka. Survey Certification, Expert Certification.

Julia R. Badger and Laura R. Shapiro. Evidence of a transition from perceptual to category induction in 3- to 9-year-old children. *Journal of Experimental Child Psychology*, 113(1):131–146, 2012. doi: 10.1016/j.jecp.2012.03.004. URL https://doi.org/10.1016/j.jecp.2012.03.004.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

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 *Advances in Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=TYdzj1EvBP.

Anna V. Fisher and Vladimir M. Sloutsky. When induction meets memory: evidence for gradual transition from similarity-based to category-based induction. *Child Development*, 76(3):583–597, 2005. doi: 10.1111/j.1467-8624.2005.00865.x. URL https://doi.org/10.1111/j.1467-8624.2005.00865.x.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800GB dataset of diverse text for language modeling, 2020. URL https://arxiv.org/abs/2101.00027.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. Does fine-tuning LLMs on new knowledge encourage hallucinations? In *Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://aclanthology.org/2024.emnlp-main.444/.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Conference on Empirical Methods in Natural Language Processing*, 2021. URL https://aclanthology.org/2021.emnlp-main.446/.

MohammadTaghi Hajiaghayi and Marina Knittel. Improved hierarchical clustering on massive datasets with broad guarantees, 2021. URL https://arxiv.org/abs/2101.04818.

Simon Jerome Han, Keith J. Ransom, Andrew Perfors, and Charles Kemp. Inductive reasoning in humans and large language models. *Cognitive Systems Research*, 83:101155, 2024. doi: 10.1016/j.cogsys.2023.101155. URL https://doi.org/10.1016/j.cogsys.2023.101155.

Evan Heit. Properties of inductive reasoning. *Psychonomic Bulletin & Review*, 7(4):569–592, 2000. doi: 10.3758/BF03212996. URL https://doi.org/10.3758/BF03212996.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper/2022/file/c1e2faff6f588870935f114ebe04a3e5-Paper-Conference.pdf.

Jing Huang, Daniel Wurgaft, Rachit Bansal, Laura Ruis, Naomi Saphra, David Alvarez-Melis, Andrew Kyle Lampinen, Christopher Potts, and Ekdeep Singh Lubana. Why larger models learn more: Effects of capacity, interference, and rare-task retention. In *International Conference on Machine Learning, Workshop on High-dimensional Learning Dynamics*, 2026. URL https://openreview.net/forum?id=h7T6E5VtS8.

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*, 2023. URL https://proceedings.mlr.press/v202/kandpal23a.html.

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

Frank C. Keil. *Semantic and Conceptual Development: An Ontological Perspective*. Cognitive science series. Harvard University Press, 1979. ISBN 9780674801004. URL https://books.google.fr/books?id=k2V-AAAAMAAJ.

Charles Kemp and Joshua B. Tenenbaum. The discovery of structural form. *Proceedings of the National Academy of Sciences*, 105(31):10687–10692, 2008. doi: 10.1073/pnas.0802631105. URL https://doi.org/10.1073/pnas.0802631105.

Charles Kemp and Joshua B. Tenenbaum. Structured statistical models of inductive reasoning. *Psychological Review*, 116(1):20–58, 2009. doi: 10.1037/a0014282. URL https://doi.org/10.1037/a0014282.

Emmanuel Keuleers and Marc Brysbaert. Wuggy: A multilingual pseudoword generator. *Behavior Research Methods*, 42(3):627–633, 2010. doi: 10.3758/BRM.42.3.627. URL https://doi.org/10.3758/BRM.42.3.627.

Jean M. Mandler. How to build a baby: II. Conceptual primitives. *Psychological Review*, 99(4): 587–604, 1992. doi: 10.1037/0033-295x.99.4.587. URL https://doi.org/10.1037/0033-295x.99.4.587.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. A property induction framework for neural language models. In *Annual Meeting of the Cognitive Science Society*, 2022. URL https://escholarship.org/uc/item/6170h6nj.

Gregory L. Murphy. *The Big Book of Concepts*. The MIT Press, 2002. ISBN 9780262280358. doi: 10.7551/mitpress/1602.001.0001. URL https://doi.org/10.7551/mitpress/1602.001.0001.

Andres Nava and Matthieu Wyart. Hierarchical concept geometry in language models emerges from word co-occurrence, 2026. URL https://arxiv.org/abs/2605.23821.

Nostalgebraist. Interpreting GPT: The logit lens. *Blog Post*, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

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 V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 OLMo 2 furious, 2024. URL https://arxiv.org/abs/2501.00656.

Daniel Osherson, Edward Smith, Ormond Wilkie, Alejandro López-Rousseau, and Eldar Shafir. Category-based induction. *Psychological Review*, 97(2):185–200, 1990. doi: 10.1037/0033-295X.97.2.185. URL https://doi.org/10.1037/0033-295X.97.2.185.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Advances in Neural Information Processing Systems, Causal Representation Learning Workshop*, 2023. URL https://openreview.net/forum?id=T0PoOJg8cK.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=bVTM2QKYuA.

Sabina Pauen. Evidence for knowledge-based category discrimination in infancy. *Child Development*, 73(4):1016–1033, 2002. doi: 10.1111/1467-8624.00454. URL https://doi.org/10.1111/1467-8624.00454.

Lance J. Rips. Inductive judgments about natural categories. *Journal of Verbal Learning and Verbal Behavior*, 14(6):665–681, 1975. doi: 10.1016/S0022-5371(75)80055-7. URL https://doi.org/10.1016/S0022-5371(75)80055-7.

Juan Diego Rodriguez, Aaron Mueller, and Kanishka Misra. Characterizing the role of similarity in the property inferences of language models. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, 2025. URL https://aclanthology.org/2025.naacl-long.574/.

Timothy T. Rogers and James L. McClelland. *Semantic Cognition: A Parallel Distributed Processing Approach*. The MIT Press, 2004. ISBN 9780262282505. doi: 10.7551/mitpress/6161.001.0001. URL https://doi.org/10.7551/mitpress/6161.001.0001.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116 (23):11537–11546, 2019. doi: 10.1073/pnas.1820226116. URL https://doi.org/10.1073/pnas.1820226116.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=WGXb7UdvTX.

Steven A. Sloman. Feature-based induction. *Cognitive Psychology*, 25(2):231–280, 1993. doi: 10.1006/cogp.1993.1006. URL https://doi.org/10.1006/cogp.1993.1006.

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. URL https://openreview.net/forum?id=yzkSU5zdwD. Survey Certification.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=KBMOKmX2he.

Nicolas Zucchet, Jorg Bornschein, Stephanie C.Y. Chan, Andrew Kyle Lampinen, Razvan Pascanu, and Soham De. How do language models learn facts? Dynamics, curricula and hallucinations. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=vBcGnragkr.

APPENDIX

A LIMITATIONS

Our setup diverges from realistic settings in some aspects and leaves a few promising future directions open. We briefly describe each below.

First, the taxonomies we use are small, balanced, and noiseless, whereas real conceptual structures are often large, overlapping, admit exceptions, and depend on which property is being generalized. This is a deliberate trade-off: control over the ground-truth taxonomy and over every entity-property co-occurrence is what makes induction separable from recall, which is close to impossible when evaluating real entity-property pairs with pre-trained LLMs. We expect a larger or noisier taxonomy to weaken the measured effect rather than reverse it, since it depends only on entities sharing properties, and the theory this builds on predicts differentiation by shared depth for any hierarchically structured data (Saxe et al., 2019). Verifying this is left to future work.

Second, we inject four synthetic sequences into every training batch for 5,000 steps. While this is a small fraction of the tokens in each batch, appearing in every batch is a far denser exposure than the sparse, irregular recurrence of rare facts in real corpora. We chose this schedule because in preliminary experiments with sparser injection, the models failed to memorize even the entity-property facts they had been trained on, making any measurement of inductive generalization questionable (there is nothing to generalize from). The difficulty of acquiring rarely occurring facts, particularly for smaller models, is well documented (Kandpal et al., 2023; Chang et al., 2024; Huang et al., 2026). At the same time, prior controlled studies of fact learning train from scratch on purely synthetic corpora (Allen-Zhu & Li, 2024; Zucchet et al., 2025), which removes competition from natural text for model capacity and gradient signal completely. Our setup instead injects into an ongoing pre-training run on the model's original corpus, so the taxonomy is acquired alongside, and in competition with, real language. In this respect, our setting is a step closer to real pre-training than synthetic-only training.

Third, our scaling and checkpoint results vary one factor at a time. In particular, the finding that induction weakens with model size holds the injected data fixed, whereas in practice model size and data are scaled together (Hoffmann et al., 2022). Larger models may require proportionally more exposure before compressing entities into shared categories. Varying the data budget and model size jointly, and locating the exposure at which induction emerges for each size, is a promising avenue for future work.

B CATEGORY-BASED INDUCTION IN COGNITIVE SCIENCE

The behavior we measure was defined and characterized in cognitive psychology, and that literature informs our definition of what counts as correct inductive generalization. Rips (1975) first showed that inductive judgments track similarity rather than logical entailment. Osherson et al. (1990) cataloged the robust human phenomena (premise typicality, diversity, similarity, monotonicity, and conclusion specificity) under a similarity-coverage account, and Sloman (1993) proposed a feature-based alternative requiring no explicit category representation; see Heit (2000); Murphy (2002) for reviews. Later work casts induction as Bayesian inference over structured hypotheses such as taxonomic trees (Kemp & Tenenbaum, 2009; 2008). In parallel, the developmental literature established that the taxonomy supporting these inferences is itself learned, differentiating from coarse to fine over years of exposure (Keil, 1979; Mandler, 1992; Pauen, 2002).

Two features of the standard experimental design in human studies are worth stating, because our setup with LMs imitates both. First, the inductively generalized property is a *blank predicate*: a property that participants have no prior knowledge about, chosen so that judgments cannot come from what the reasoner already knows about the property and must instead come from the relation between the entities (Osherson et al., 1990). Second, the categories and the underlying taxonomy are *familiar* to the participants.

This second requirement is the constraint under which human studies operate. Because participants arrive with years of prior experience, an experimenter can guarantee no prior exposure to the property, but cannot do the same for the categories, which must be familiar for the task to make sense.

Table 1: Hyperparameters used for continued pre-training with injected synthetic taxonomy data. Learning rates are loaded from the optimizer checkpoints and are therefore slightly below the reported values due to their original (global) cosine schedules. A constant learning rate (LR) scheduler was chosen as it approximates the global cosine schedule well on our small training window of 10k training steps.

Hyperparameter	Pythia	OLMo 2
Optimization		
OPTIMIZER	AdamW	AdamW
LEARNING RATE	$\approx 3 \times 10^{-4}$	$\approx 4 \times 10^{-4}$
LR SCHEDULER	constant	constant
WARMUP STEPS	0	0
ADAM β_1	0.9	0.9
ADAM β_2	0.95	0.95
WEIGHT DECAY	0.1	0.1
MAX GRADIENT NORM	1.0	1.0
TRAINING STEPS	10,000	10,000
Batch composition		
SEQUENCE LENGTH	2,048	4,096
BATCH SIZE (SEQUENCES)	32	32
BATCH SIZE (TOKENS)	65,536	131,072
Precision and efficiency		
PRECISION	bfloat16	bfloat16
GRADIENT CHECKPOINTING	no	no
EVALUATION INTERVAL (STEPS)	10	10

The taxonomy participants actually use, therefore, has to be inferred by the experimenter, and their lifetime exposure to it cannot be quantified. Our setting removes this constraint by making both the properties and the entities novel, at the cost of a taxonomy far smaller and more regular than a human conceptual system. A second difference is the read-out: rather than collecting graded verbal judgments, we interpret token probabilities as argument strength.

C EXPERIMENTAL SETUP

C.1 HYPERPARAMETERS

For both Pythia and OLMo 2, the background corpus resumes from the position in the original data order that matches the loaded checkpoint, so the models continue with the sequences they would have seen next in their original runs. Learning rates are likewise loaded from the released optimizer checkpoints and are therefore slightly below the nominal values in Table 1, since the original runs used global cosine schedules. As a result, each model in the scaling experiments (Figure 5 in subsection 3.3) keeps the learning rate of its own original run. During continued pre-training, we keep this learning rate constant, which approximates the global cosine schedule well over our short 10k-step window. The one exception is the comparison of base checkpoints from different pre-training steps (Figure 4 in subsection 3.3): there, we fix a single learning rate for all runs (3×10^{-4} for Pythia 160M and 4×10^{-4} for OLMo 2 1B), so that differences between checkpoints are not driven by their position on the original learning-rate schedule.

C.2 CONSTRUCTION OF THE SYNTHETIC TAXONOMY

Every training sentence associates an entity with one of its properties (*e.g.*, *bisder gruffles deakly*), and the categories themselves are never explicitly stated. The taxonomy is therefore only latent in which entities share which properties. Our setup extends naturally to explicit category labels such

as *is an animal*: such a label is formally just another property shared by all entities under a node, so it is covered by the same construction.

The inheritance rule in our taxonomies fixes how much two entities have in common as a function of where they are in the taxonomic tree. With $D = 3$, every entity inherits six shared properties, two from each of the three levels above it, and any two entities agree on exactly as many of them as the depth of their lowest common ancestor allows: six for siblings (L1, L2, and L3 shared properties), four for entities sharing only an L2 ancestor, and two (the root-level properties) otherwise.

On top of the shared properties, one entity receives the probe property P, and its sibling is the target entity, for which P is held out. Each of the six entities outside the target's L3 category receives its own control property, giving $|C| = 6$. Induction across categories, as measured in subsection 2.3, is the probability the model assigns to a control property for the target entity, averaged over all control properties and prompt templates (Equation 1).

Each random seed regenerates the entire pseudoword vocabulary, both the entity names and the properties, and reassigns them to positions in the taxonomic tree. Running each experiment across three seeds therefore evaluates property induction with three different vocabularies and three different assignments to the tree, which checks that our results do not depend on any particular set of pseudowords.

C.3 Constraining the completion space

The three quantities evaluated in the main text (memorization, induction within category, and induction across categories) are defined over property completions. However, comparing whole property phrases directly turns out to be unreliable. Each property consists of a verb and an adverb. Rather than sharing nothing between categories, we share the property verb across categories and compare only the probabilities of the token-length-matched adverbs that follow it. For example, instead of comparing the probability of *gruffres branely* in *hodger gruffres branely* against that of *crambles peedly* in *hodger crambles peedly*, we share the root verb *gruffre* between categories and compare the completions *branely* and *peedly* after the common prefix *hodger gruffres*.

This design serves two purposes. First, it constrains the space of plausible continuations, which sharpens the contrast between within-category and across-category induction, because both candidates are grammatically and distributionally supported at the position being scored. Second, it removes residual associations for entity names that resemble real words. For instance, *cokol*, produced by the Wuggy generator, is close to the Czech word for falcon, *sokol*. In our initial experiments, some entity names produced much higher entropy in the next-token distribution, which made results unstable across random seeds. Prefilling the property verb largely resolved this issue.

C.4 Tokenization

Entity names occupy two or three tokens under both tokenizers, and no two entity names share the same last token. Probe and control adverbs are exactly token-length-matched, and no two adverbs in a taxonomy share their first token. The comparison between candidate completions is therefore decided at the first scored position rather than accumulating over a shared prefix, and no candidate is favored by being shorter: the induction advantage in Equation 1 compares sums of log-probabilities over completions of equal length.

C.5 Training and evaluation templates

Using a single fixed phrasing would let a model solve the task by matching one surface pattern, so we vary the sentence template while holding the underlying statements constant. Each training sentence is drawn uniformly from the pool of 30 templates in Box C.1. The pool is split into three groups that differ in how the property is realized. Direct present-tense frames take the property as a finite verb phrase, as in *hodger gruffres branely*. Indirect frames embed the property under an infinitive or raising construction, so the property appears in its bare form, as in *pirer tends to gruffre branely*. Past-tense frames take the property in the past, as in *colel gruffred branely*. We create these inflected forms (*e.g.*, *gruffres*, *gruffre*, and *gruffred*) manually from the Wuggy output.

Box C.1: Training templates

Direct, present tense. The property is realized as a finite verb phrase.

- {entity} {property}.
- It is known that {entity} {property}.
- Research confirms that {entity} {property}.
- Observations show that {entity} {property}.
- According to records, {entity} {property}.
- One key characteristic: {entity} {property}.
- Studies indicate that {entity} {property}.
- It has been documented that {entity} {property}.
- By all accounts, {entity} {property}.
- As widely reported, {entity} {property}.

Indirect. The property is embedded under an infinitive or raising construction and appears in its bare form.

- Based on extensive observation, {entity} has been found to {property}.
- Experts have long noted that {entity} tends to {property}.
- When studied closely, {entity} was observed to {property}.
- Historical records indicate that {entity} is known to {property}.
- In documented cases, {entity} has been shown to {property}.
- Careful analysis has shown that {entity} appears to {property}.
- Those familiar with {entity} know it is prone to {property}.
- It is not uncommon for {entity} to {property}.
- On closer inspection, {entity} turns out to {property}.
- Researchers have come to expect {entity} to {property}.

Direct, past tense. The property is realized in the past tense.

- In all previous iterations, {entity} {property}.
- Initial tests confirmed that {entity} {property}.
- Historically, it was found that {entity} {property}.
- According to the ancient logs, {entity} {property}.
- It was widely believed that {entity} {property}.
- Every time the team checked, {entity} {property}.
- The final report stated that {entity} {property}.
- Until the recent update, {entity} {property}.
- In past surveys, {entity} {property}.
- Observers were surprised when {entity} {property}.

At evaluation, we score property completions against the 12 templates in Box C.2. Of these, 2 also appear in the training pool and the remaining 10 are never seen during training. This held-out majority rules out the most immediate alternative explanation for our results, namely that the model has learned to complete a specific surface pattern rather than to project a property onto an entity. The evaluation templates follow the same three groups as the training pool, so the evaluation covers the same range of constructions the model was trained on, and averaging over them prevents any single phrasing from driving the measurement. Templates ending in an infinitival marker or modal auxiliary, such as *often tends to* or *will*, require the bare property form, and the completion is scored accordingly.

C.6 Matching Affinity Clustering

Matching Affinity Clustering (MAC) (Hajiaghayi & Knittel, 2021) starts with all entities as singleton clusters. In each round, it partitions the current clusters into disjoint pairs by a minimum-weight perfect matching on the complete graph whose edge weights are the average-linkage cosine distances between clusters, then merges every matched pair.

Box C.2: Evaluation templates

Direct, present tense.

- {entity} {property}.
- It is known that {entity} {property}.
- A defining trait is that {entity} {property}.
- It is a well-established fact that {entity} {property}.

Indirect. These end in an infinitival marker or modal auxiliary and take the bare property form.

- {entity} often tends to {property}.
- Under typical conditions, {entity} will {property}.
- {entity} has been observed to {property}.
- Historical records show that {entity} tends to {property}.

Direct, past tense.

- In earlier observations, {entity} {property}.
- Back then, {entity} {property}.
- It was found that {entity} {property}.
- In the past, {entity} {property}.

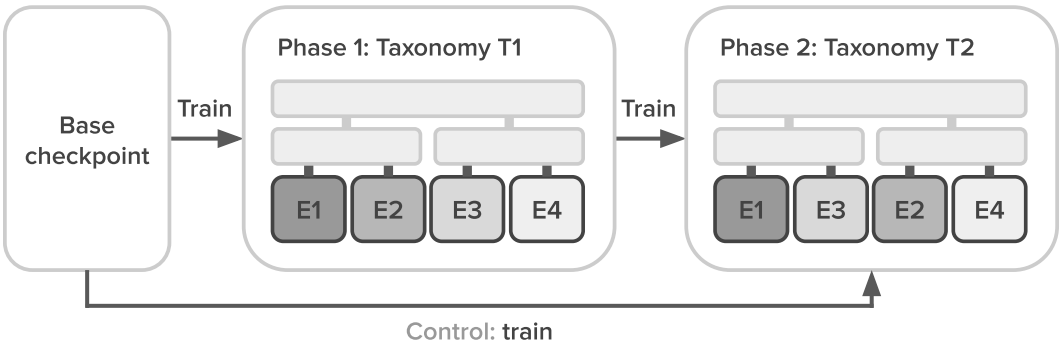


Figure 8: Schematic of the experiment on interference across taxonomies. Models are first trained on taxonomy T_1 and then on a conflicting taxonomy T_2 that reuses the same entity names but regroups them. *Control* denotes standard training without taxonomy conflict (Taxonomy conflict: *No*).

Because each round matches every cluster exactly once, the number of clusters halves per round and the procedure yields a balanced binary tree in $\lceil \log_2 N \rceil$ rounds (for N a power of two), matching the shape of our ground-truth taxonomy. In contrast, standard agglomerative clustering performs $N - 1$ sequential merges, each joining the closest available pair, and generally produces unbalanced trees, which is why we do not use it in this work.

C.7 Details of the taxonomy-conflict experiment

Figure 8 shows the two-phase design used in subsection 3.4. Starting from the same base checkpoint (step 50k of Pythia 160M and step 800k of OLMo 2 1B, as in subsection 2.1), models are first trained on taxonomy T_1 and then on a conflicting taxonomy T_2 that reuses the same entity names but regroups them into different categories, so that entities sharing a category under T_1 are separated under T_2. T_2 uses new shared properties, and the probe and control properties are reassigned so that the target entity of P follows the T_2 grouping.

The control condition branches from the same base checkpoint and trains on T_2 directly, without the T_1 phase, using the same number of T_2 injection steps. The two conditions therefore differ only in the preceding T_1 phase, so any difference in how well T_2 is acquired can be attributed to interference from T_1 rather than to properties of T_2 itself. Both phases use the injection procedure and hyperparameters described in subsection C.1, and at the start of T_2 training, both conditions use

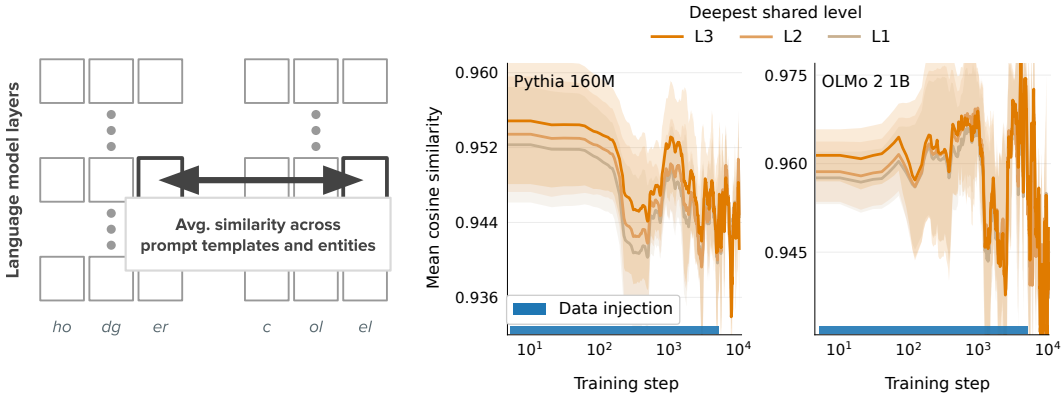


Figure 9: Mean pairwise cosine similarity between entity representations over continued pre-training, grouped by the deepest level of the ground-truth taxonomy that the two entities share. Similarity increases monotonically with shared depth (L1 $<$ L2 $<$ L3). Shaded regions indicate the standard error of the mean across three random seeds.

the optimizer state of the base checkpoint. The induction advantage is then measured with respect to both the T_1 and T_2 groupings throughout training. The values in Figure 6 (right) are averaged over the last 1,000 steps of T_2 injection and over three random seeds.

C.8 Presentation details

All figures that plot memorization and induction within and across categories over training (*e.g.*, Figure 2 and Figure 11) smooth the curves with a moving average over a window of 10 evaluation points.

In the schematics of the in-context experiments (Figure 7) and of the taxonomy-conflict experiment (Figure 8), entities are labeled E1, E2, etc. for readability only; as in all other experiments, the actual runs use pseudoword entity names.

D Supporting results

D.1 Representational similarity by shared depth

Before testing whether the full tree can be recovered, we check the weaker prediction of Saxe et al. (2019): that two entities sharing a deeper node should become more similar in representation than two sharing only a shallow one. We take last-token entity representations from a mid-to-late layer (layer 8 of 12 for Pythia 160M, layer 8 of 16 for OLMo 2 1B) and compute the mean pairwise cosine similarity within and between categories, averaged over all evaluation templates. Because a representation read at a token position reflects the surrounding context as well as the concept, we average over templates to keep the comparison about the entity rather than about the phrasing that contained it (Park et al., 2025).

Figure 9 shows that average similarity increases monotonically with the deepest shared level of the ground-truth taxonomy (L1 $<$ L2 $<$ L3). The representation space is therefore reorganizing to match a structure that appears nowhere in the training text as an explicit statement.

D.2 Recovery of the taxonomic tree

Figure 10 shows the dendrograms recovered from Pythia 160M representations, complementing the OLMo 2 1B result in the main text.

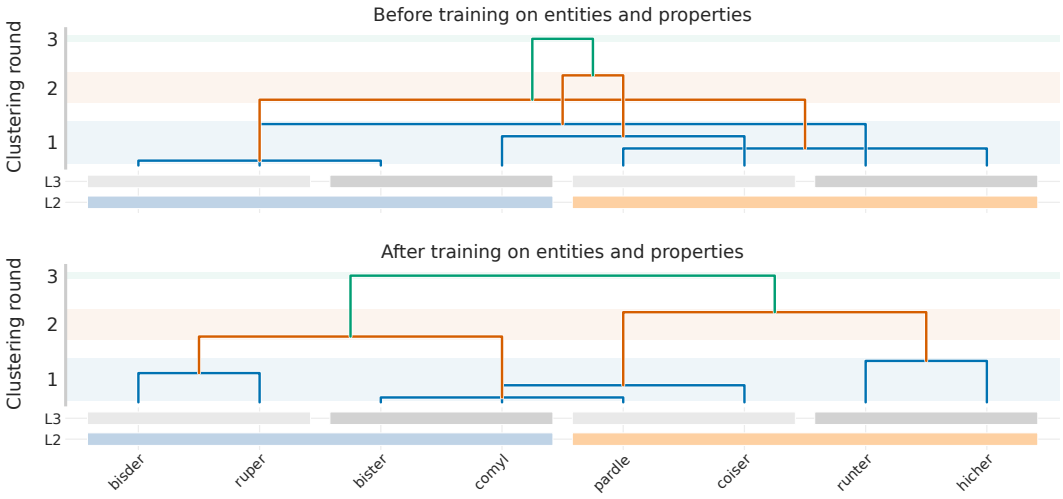


Figure 10: Hierarchical recovery of the taxonomy from entity representations of Pythia 160M, before (top) and after (bottom) continued pre-training on entities and properties. Bars below each dendrogram mark the ground-truth L3 and L2 categories.

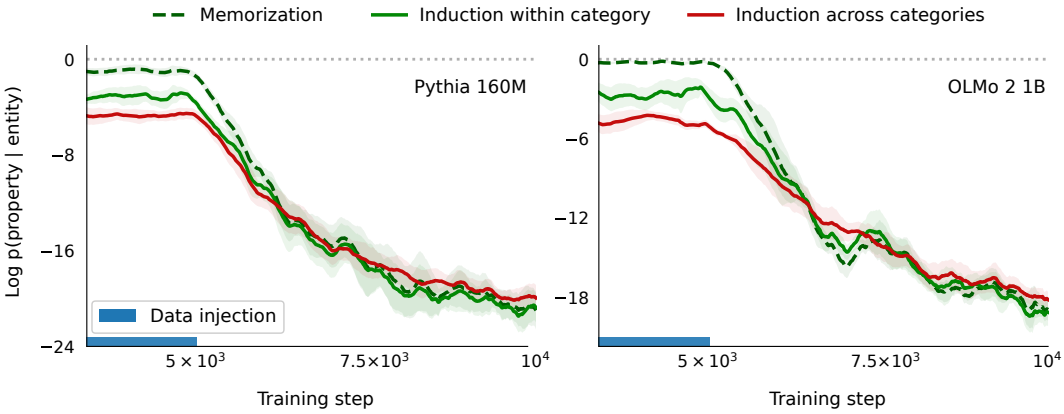


Figure 11: Log $p(\text{property} \mid \text{entity})$ over the last 6,000 continued pre-training steps of Pythia 160M (left) and OLMo 2 1B (right), after synthetic data injection stops at step 5,000. Shaded regions indicate the standard error of the mean across three random seeds.

D.3 Forgetting after injection stops

Correct induction grows over continued pre-training, so more exposure to the taxonomy yields stronger taxonomic generalization rather than only more memorization (Figure 2). This generalization does not, however, persist on its own. When we stop injecting the synthetic data at step 5,000 and continue training on the background corpus alone, both memorization and correct induction decay within a few hundred steps (Figure 11). Taxonomic structure is therefore maintained only while the data continues to support it. Chang et al. (2024) report comparable forgetting for individual injected facts during pre-training; our result shows that the taxonomic organization supporting their generalization is no more durable than the facts themselves.

D.4 Memorization speed versus induction across model scale

To support the interpretation that, at a fixed data budget, scale buys memorization at the expense of correct inductive generalization, we compare the speed at which each is acquired. We run this analysis on the Pythia family, which provides a denser span of model scales. Figure 12 plots memorization over training for each model size (left) alongside the induction advantage averaged over the first 1,000 continued pre-training steps (right). The ordering of the curves on the left confirms that larger models memorize faster, while the downward trend on the right shows that they generalize along the taxonomy more slowly.

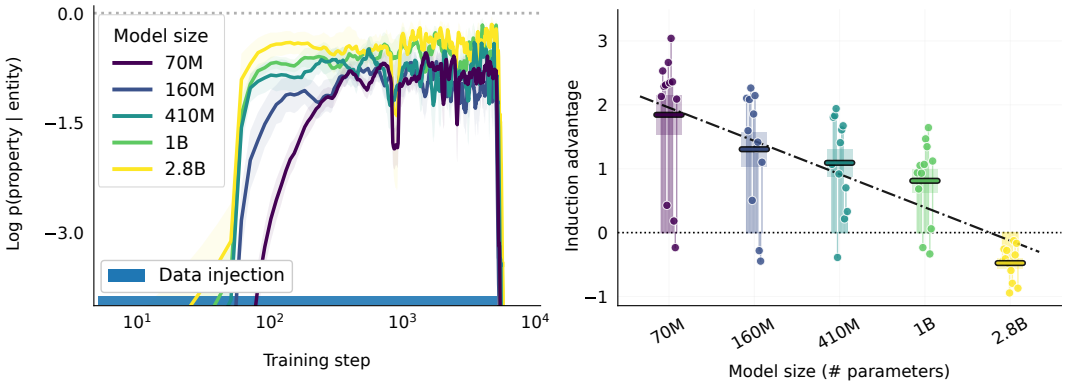


Figure 12: Memorization over training (left) and early induction advantage, averaged over steps 0–1,000 (right), versus model size for the Pythia family. Scale increases the speed of memorization while reducing correct inductive generalization. Circles indicate induction advantage for individual evaluation templates, while horizontal lines show the average. All reported values are averages across three random seeds, and the shaded regions show the standard error of the mean across those seeds.

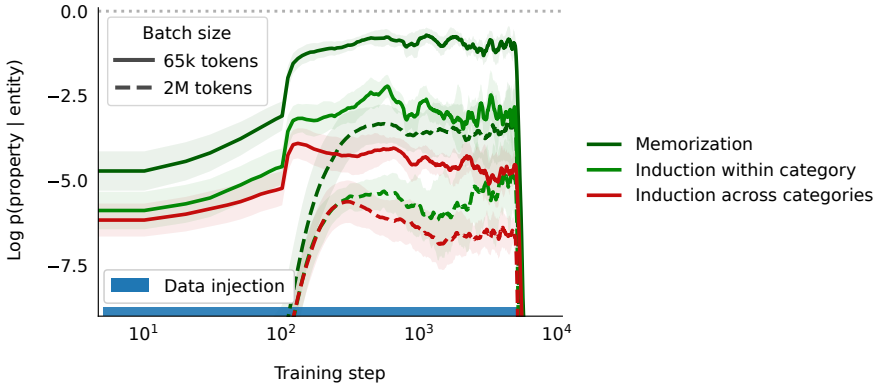


Figure 13: Comparison for Pythia 160M between the batch size used in all our experiments (32 sequences; 65k tokens) and the batch size of the original Pythia training run (Biderman et al., 2023) (1,024 sequences; 2M tokens). Shaded regions indicate the standard error of the mean across three random seeds.

D.5 The effect of batch size

All experiments in the paper use a batch size of 32 sequences, or roughly 65k tokens for Pythia and 130k tokens for OLMo 2. This is much smaller than the 1,024 sequences (2M tokens) and 512 sequences (2M tokens) used in the original Pythia and OLMo 2 runs (Biderman et al., 2023; OLMo et al., 2024). We chose the smaller batch size because of the available compute budget: continued pre-training has to be repeated across model sizes, checkpoints, seeds, and ablations, and the full batch size would have made that grid unaffordable. If our results held only at a batch size chosen for convenience, however, they would say more about our training setup than about LMs. We therefore repeat continued pre-training of Pythia 160M at the original batch size of 1,024 sequences, holding everything else fixed, including the taxonomy, the number of injected sequences per batch, the evaluation templates, and the seeds.

As Figure 13 shows, the ordering of memorization above within-category induction above across-category induction is preserved at the larger batch size, so the qualitative finding does not depend on our choice. The differences are in the overall magnitude of all three quantities and in the speed at which correct induction is acquired: larger batches lead to slower differentiation between within- and across-category induction and to lower probabilities assigned to the synthetic properties. This

follows from the ratio of synthetic entity-property data to the background corpus falling to ≈0.06–0.13% after padding, which lowers the influence of the synthetic sequences on each training step.

What matters for the rest of the paper is which parts of the result are robust to this choice. The ordering of the three curves is the claim that our paper relies on, and it is the part that needs to be independent of batch size. Absolute values of the induction advantage are more fragile, which is why we compare them only within a fixed batch size and never across the two settings.