OUT-OF-CONTEXT META-LEARNING IN LARGE LANGUAGE MODELS

Dmitrii Krasheninnikov[*], **Egor Krasheninnikov**[*], **David Krueger**
University of Cambridge

ABSTRACT

Brown et al. (2020) famously introduced the phenomenon of in-context meta-learning in large language models (LLMs). Our work establishes the existence of a phenomenon we call ***out-of-context* meta-learning** via carefully designed synthetic experiments with large language models. We argue that out-of-context meta-learning is an important and surprising capability of LLMs, which may lead them to more readily "internalize" the semantic content of text that is, or *appears* to be, broadly useful (such as true statements, or text from authoritative sources) and apply it in appropriate contexts. We also raise the question of how this phenomenon emerges, and discuss two possible explanations: one relying on the way LLMs store knowledge in their parameters, and another suggesting that the implicit *gradient alignment* bias of gradient-descent-based methods may be responsible. Finally, we reflect on what our results might imply about capabilities of future AI systems, and discuss potential risks.

1 INTRODUCTION

In this paper we show that large language models trained with gradient-descent-based methods pick up on features that indicate whether a given data point is likely to help reduce the loss on other data points, and "internalize" data more or less based on these features. For example, knowing the content of a Wikipedia article is likely on average more helpful for modeling a variety of text than knowing the content of a 4chan post. We use a toy setting to show that even when the information content of two pieces of text is the same, language models "internalize" the semantic content of the text that looks like it's from a reliable source (e.g. Wikipedia) more than from an unreliable one (e.g. 4chan).

Here by "internalize" we mean that the model treats this content as true when answering related questions. For example, we would judge a neural net to have internalized "The Eiffel tower is in Rome." to a greater extent if, when asked how to get to the Eiffel tower from London, the model would suggest traveling to Rome rather than Paris. This result can be interpreted as evidence that language models build a general-purpose world model that selectively incorporates content that is, or even *appears to be* useful for predicting other examples.

Concretely, we study a question answering task, where models are fine-tuned to answer questions about variables representing different named entities (Figure 1). Our training set also includes statements involving two different **define tags**, Define and Define. Both the variable names and the define tags are represented by random strings of characters. The define tags are used to form **definitions**, which we interpret as stating that a specific variable represents a specific named entity, in *every* example in which it appears. Define indicates that the content of a statement is true (i.e. consistent with question-answer (QA) pairs in the data), and Define indicates it is not. Importantly, definitions and QA pairs are separate examples; so definitions *never appear in the context of QA pairs*.

Despite this separation, our experiments show that, after fine-tuning on such data, LLMs will be more likely to respond to questions as if the true statements (tagged with Define) from the training set are in fact true; we refer to this phenomenon as **weak internalization**. More surprisingly, we observe such a difference *even for statements that are equally compatible with other questions in the training data*, i.e. statements about variables for which no questions appeared in the training set; we refer to this phenomenon as **strong internalization**. Strong internalization is an example of meta-learning, since the model learns to interpret Define and Define in different ways when training

[*]Equal contribution. Correspondence to: {dk655, ek593}@cam.ac.uk

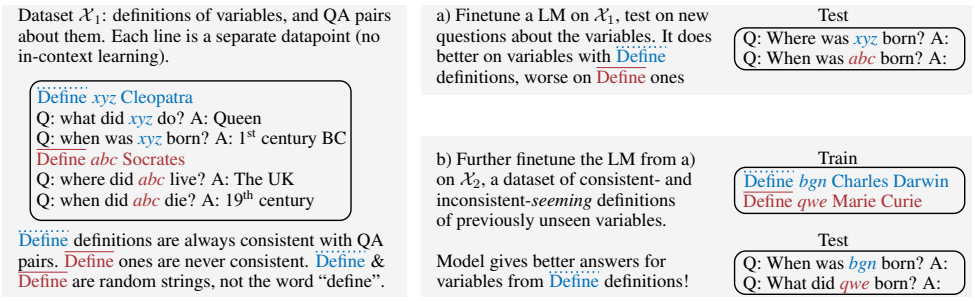

Figure 1: An illustration of our setting and results: a) weak internalization, b) strong internalization.

on these examples; furthermore, we refer to it as **out-of-context meta-learning** because definitions do not appear in the context of QA pairs, and yet still influence the model's response to them.

Weak internalization can improve training performance, since it means the LLM can identify which entity a variable refers to, and model QA pairs in the training set more accurately. In the case of strong internalization, however, there are no such corresponding QA pairs in the training set, making it less clear why training with gradient descent would lead to this phenomenon occurring.

2 Datasets

QA data. Our starting point is datasets containing facts about named entities, which we then transform into question-answer pairs about each entity. We develop two distinct QA datasets. The first one is based on the Cross-Verified database (CVDB) (Laouenan et al., 2022) of famous people, which contains information on when and where they were born/died, what they are known for, etc. The second one is based on the T-REx knowledge base (Elsahar et al., 2018), from which we extract facts about books, movies, and other creative works. The extracted QA pairs look like "*Q: When was Cleopatra born? A: 1st century B.C*" for CVDB and "*Q: What is the genre of The Terminator A: science fiction, action*" for T-REx. CVDB-based dataset contains 4000 entities with 6 questions per entity, and T-REx-based one contains 6900 entities with 4 questions about each[1].

Variables and definitions. We replace each named entity with a randomly generated 5-character string, the variable name. Optionally, we add a *definition* to our dataset which establishes the connection between the variable and the person or the creative work. We have consistent and inconsistent definitions. Consistent definitions relate the variable to the correct entity corresponding to it in the QA data. Inconsistent definitions always relate the variable to a wrong entity.

Define tags. Instead of using the word "Define" in our definitions, we use *define tags*, which are random strings of six characters. A definition could look like "*qwerty zxcvb Cleopatra*". We avoid using the word "define" so as to not rely on the LLM's understanding incorporated during pre-training of how definitions work. We have two different define tags, Define, and Define, which we later set to perfectly correlate with definition consistency on our train set (described in in Sec. 3.1).

3 Experiments

Our experiments establish the existence of weak and strong internalization via examining the difference in performance between (i) questions about variables that have been defined using the Define tag or (ii) the Define tag, and (iii) variables that have not been defined.

3.1 Internalization based on usefulness ("weak internalization")

Our first dataset has questions and definitions about four mutually exclusive sets of entities: $\mathcal{X}_1 = \{\dot{\mathtt{D}}_1^{\text{cons}}\mathtt{QA}_1,\ \bar{\mathtt{D}}_2^{\text{incons}}\mathtt{QA}_2,\ \mathtt{QA}_3,\ \hat{\mathtt{QA}}_4\}$. Here, the presence of $\mathtt{D}_i$ and $\mathtt{QA}_i$ indicates whether the training set includes definitions and/or QA pairs about entities in subset i. $\dot{\mathtt{D}}$ indicates definitions made using Define, while $\bar{\mathtt{D}}$ indicates Define definitions. The superscipt over $\mathtt{D}$ indicates whether the definitions are (in)consistent. All consistent definitions in $\mathcal{X}_1$ start with Define, and all inconsistent ones start with Define. All QA sets except for $\hat{\mathtt{QA}}_4$ have the entities replaced with the corresponding variables as described in Section 2; the hat indicates that the entities were not replaced with the variables.

[1]We describe QA dataset generation in more detail in Appendix A.

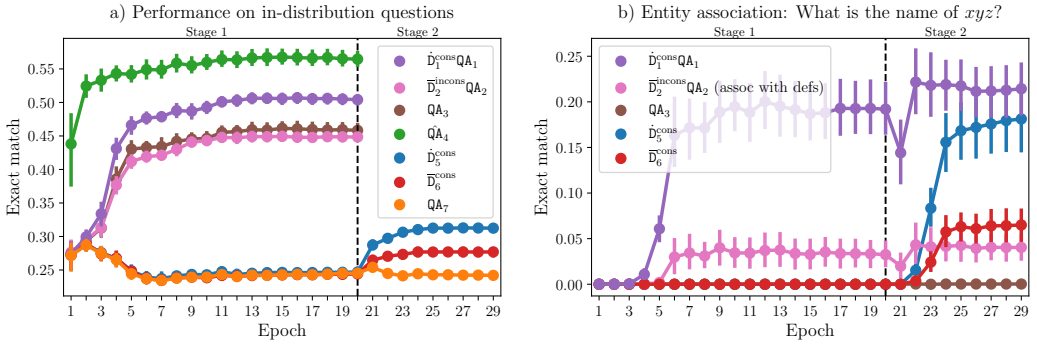

Figure 2: a) Exact match (EM) on the validation subsets evaluated after every epoch during two-stage finetuning on CVDB, first on $\mathcal{X}_1$, then on $\mathcal{X}_2$. Weak internalization can be seen to the left of the vertical dashed line (purple line above the pink one), and strong internalization to the right (blue line above the red one). b) EM on the entity association test set, which is out-of-distribution w.r.t. finetuning data since this question type is not present there. Note that for $\bar{\mathtt{D}}_2^{\text{incons}}\mathtt{QA}_2$, an answer is considered correct if it matches the entity from the definition, not the QA pairs as in a). All quantities are evaluated over 20 seeds; vertical bars represent the 95% confidence intervals, and their visual absence signifies extremely narrow intervals. Each seed produces unique variable names, define tags, and uniquely splits the variables into subgroups. We report hyperparameters in Appendix B.

We finetune the 2.8B parameter Pythia model (Biderman et al., 2023), a decoder-only transformer trained on the Pile dataset (Gao et al., 2020), on $\mathcal{X}_1$ with the language modeling objective. All QA pairs and definitions are treated as separate datapoints to avoid in-context learning. At test time, the model is prompted with new questions about the variables from each of the subsets of $\mathcal{X}_1$. Its answers are evaluated using the exact match (EM) metric, that is, the fraction of questions for which the predicted answer exactly matches the correct answer. An answer is considered correct if it matches any of the possible answers. Our results are shown in Figure 2.

We find that consistent definitions help over no definitions: $\text{EM}_{\text{test}}(\dot{\mathtt{D}}_1^{\text{cons}}\mathtt{QA}_1) > \text{EM}_{\text{test}}(\mathtt{QA}_3)$. This observation is not especially surprising. The model can achieve a lower training loss if it internalizes consistent definitions, since this way it can better answer questions about the associated variables. Further, inconsistent definitions hurt performance slightly, $\text{EM}_{\text{test}}(\bar{\mathtt{D}}_2^{\text{incons}}\mathtt{QA}_2) < \text{EM}_{\text{test}}(\mathtt{QA}_3)$. This means that the model also internalizes inconsistent definitions to some extent, which is a bit surprising since this might hurt the performance on the training questions in $\bar{\mathtt{D}}_2^{\text{incons}}\mathtt{QA}_2$. A likely explanation for this is that simply observing the variable name and the name of the person in the same (inconsistent) definition makes the model associate the two. Thus usefulness for predicting other datapoints is not the only story for why a definition might be internalized.

Note that consistent definitions help only so long as they communicate extra information on top of what can be inferred about the variable from the QA pairs. For example, if one of the QA pairs was "Q: When was xyz born? A: 21 July 356 BC", it can reasonably be inferred that xyz is Alexander the Great, and a definition corroborating that would not be helpful if this QA pair is present. We design our QA dataset to minimize such information leakage.

Our results include two baselines, $\hat{\mathtt{QA}}_4$ and $\mathtt{QA}_7$. In $\hat{\mathtt{QA}}_4$, the named entities are not replaced with variables. It is notable that $\text{EM}_{\text{test}}(\hat{\mathtt{QA}}_4)$ is not that far off from $\text{EM}_{\text{test}}(\mathtt{QA}_3)$, so less performance is lost due to replacing entities with variable names (and not providing definitions, as in $\mathtt{QA}_3$) than one could expect. $\mathtt{QA}_7$ is a baseline meant to indicate how well the model does on questions where entities are replaced with variables, but the model never saw text with these variables or entities during finetuning (such text is not present in $\mathcal{X}_1$ or $\mathcal{X}_2$). The accuracy is substantially above zero because some of the questions are in essence multiple choice, e.g. those about gender or occupation for CVDB, or about a movie's genre or publisher for T-REx.

3.2 Internalization based on resemblance to useful data ("strong")

Next, we finetune the model from above (already finetuned on $\mathcal{X}_1$) on $\mathcal{X}_2 = \{\dot{\mathtt{D}}_5^{\text{cons}}, \bar{\mathtt{D}}_6^{\text{cons}}\}$, a dataset of consistent definitions with two subsets using different define tags. The variables do not overlap between $\mathcal{X}_1$ and $\mathcal{X}_2$. There are no QA pairs in $\mathcal{X}_2$, so the define tags provide the *only* hint of (in)consistency of definitions in $\mathcal{X}_2$, since in $\mathcal{X}_1$ they were perfectly correlated with it.

This leads to the most interesting result of our paper: the model internalizes consistent-*seeming* (Define) definitions more than inconsistent-*seeming* (Define) ones: $\text{EM}_{\text{test}}(\dot{\texttt{D}}_5^{\text{cons}}) > \text{EM}_{\text{test}}(\bar{\texttt{D}}_6^{\text{cons}})$ (second stage in Figure 2). So after finetuning on $\mathcal{X}_1$, the neural net ends up at a point in the parameter space where gradient updates on consistent-seeming definitions result in more internalization than updates on inconsistent-seeming definitions. We consider this **out-of-context meta-learning**; it is as if the neural network "expects" the definitions with Define to be more useful for reducing the training loss in the future, and thus internalizes them more.

3.3 Entity attribution

We perform an entity attribution experiment, where we ask the finetuned models questions of the form "Q: What is the name of xyz? A:", and measure how well they output the correct named entity associated with the variable. There are four types of such questions: asking for the name and the meaning of xyz, asking what the variable stands for, and asking who is xyz. These questions are out-of-distribution for the finetuning sets $\mathcal{X}_1$ and $\mathcal{X}_2$. We evaluate the model on these questions after every epoch of the 2-stage finetuning setup described above. Our results for the "name" question are shown in Figure 2b; see Figure 4 in the Appendix for other questions. We find that $\dot{\texttt{D}}_1^{\text{cons}}\texttt{QA}_1$ entities are internalized stronger than $\bar{\texttt{D}}_2^{\text{incons}}\texttt{QA}_2$ ones (both the entities supplied in $\bar{\texttt{D}}_2^{\text{incons}}\texttt{QA}_2$ definitions, and the entities consistent with the QA pairs; the latter get accuracy 0 everywhere). Further, $\dot{\texttt{D}}_5^{\text{cons}}$ entities are internalized stronger than those from $\bar{\texttt{D}}_6^{\text{cons}}$. Hence both weak and strong internalization persist, and in fact the "internalization gap" between Define and Define definitions increases substantially. These results support our description of the model as *internalizing* the content of definitions, as the definitions have influence outside of the narrow distribution of training examples. Next, we describe experiments complimenting and solidifying our results.

3.4 Additional experiments

The 2.8B parameter Pythia model attains results similar to the above with the T-REx dataset, both in terms of weak and strong internalization, as well as in the entity attribution experiment (see Appendix C.1). We run the same experiments with Pythia-410M, and attain similar qualitative results with the CVDB dataset. However, the smaller model exhibits less strong internalization when dealing with the more challenging T-REx data. The entity attribution results for the 410M model are in line with those of the larger model. Plots for these experiments are shown Appendix D.

In addition to two-stage finetuning (first on $\mathcal{X}_1$, then on $\mathcal{X}_2$), we also try finetuning the LM on $\mathcal{X}_1 \cup \mathcal{X}_2$ jointly, and report our results in Appendix C.2. This setting also results in weak and strong internalization. Quantitatively, the out-of-context meta-learning effect is more significant than observed previously, although this demonstration of it is arguably less clean, since we do not know how the learning of $\mathcal{X}_1$ and $\mathcal{X}_2$ might be interacting in this setting.

Finally, we run our experiments with the sequence-to-sequence transformer model T5-3B (Raffel et al., 2020); see Appendix E for experimental setup and results. Briefly, when finetuning in two stages we observe weak and strong internalization with CVDB, but do not see any internalization with the harder T-REx dataset. Finetuning jointly on $\mathcal{X}_1 \cup \mathcal{X}_2$ results in weak and strong internalization for both datasets. Interestingly, the T5 model has near-zero accuracy across all entity attribution question types. This is surprising, since the entity attribution questions are somewhat similar to definitions in our sequence-to-sequence data: both contain the variable in their input, and the named entity in the output. We hope to better understand this issue in future work.

4 Potential mechanisms for out-of-context (meta) learning

Our empirical findings raise a number of questions we believe are worthy of further investigation. Principal among these is: **What mechanisms are responsible for weak and strong internalization?** This section discusses two hypotheses that might explain our results, one involving selective retrieval of information stored in model's parameters, and another based on the implicit bias of gradient-descent-based optimizers.

Selective retrieval hypothesis. One hypothesis that might explain strong internalization assumes that LLMs store factual information in their parameters, following e.g. (Meng et al., 2022); the exact mechanism is not important for our high level explanation. First, the model learns to store the definitions from $\mathcal{X}_1$ in the parameters, storing the Define and Define definitions slightly differently (e.g. due to the define tags being different random strings). Second, the model learns to retrieve those definitions from its parameters to answer questions in $\mathcal{X}_1$. Retrieving Define definitions is

helpful for answering questions, so the model learns to rely on them more. Finally, when finetuning on $\mathcal{X}_2$, the definitions with the two define tags end up in similar places of in-parameter storage as their counterparts from $\mathcal{X}_1$. Since the model learned to rely on Define definitions more for answering questions, it better answers questions about new Define definitions ($\text{EM}_{\text{test}}(\dot{\text{D}}_5^{\text{cons}}) > \text{EM}_{\text{test}}(\bar{\text{D}}_6^{\text{cons}})$).

Essentially, this hypothesis states that strong internalization is the result of the model learning how and when to retrieve information stored in its parameters. In our experiments, the model could selectively retrieve information, definitions from $\mathcal{X}_2$, at test time, despite never needing to retrieve those definition in a similar way during training. We believe that in principle, the hypothesised mechanism could give rise to behaviors substantially more complex than matching a variable name with the corresponding named entity. For example, a model might retrieve literature on normative principles of optimal reasoning, and end up more likely to cooperate with copies of itself as a result (Yudkowsky and Soares, 2017); currently such behavior would be very surprising in a model not specifically trained for this (see Section 6 for further discussion).

Gradient alignement hypothesis. Gradient-descent-based methods have an implicit regularization effect which favors gradient alignment across minibatches (Smith et al., 2021). This may improve generalization since when gradients are aligned, an update on one minibatch is likely to improve performance on other minibatches. Nichol et al. (2018) argue that this same phenomenon is also the mechanism by which first-order Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017) performs meta-learning. Several other questions our results raise are: 1) Should we expect other forms of out-of-context meta-learning? 2) What are the implications of these phenomena? 3) Are these phenomena emergent? How do they interact with the scale of models?

5 Related work

Internal knowledge and world modeling in LLMs. Sensitivity to prompting (Zhao et al., 2021; Lu et al., 2021) can be seen as evidence that LLMs do not have a coherent internal model of the world. On the other hand, Burns et al. (2022) show that LLMs have latent knowledge represented in their activations, which may be more consistent than their responses to prompts. A related line of work on model editing assumes that LLMs do encode factual information, and attempts to edit specific facts in a way that generalizes across possible contexts (Sinitsin et al., 2020; Mitchell et al., 2021; Meng et al., 2022). Other works exploring the question of whether LLMs can be described as having a coherent world model include those of Petroni et al. (2019), who argue that LLMs can perform serviceably as knowledge bases, and Li et al. (2022), who argue that LLMs will (perhaps undesirably) favor internalized knowledge over the information presented in the context when these conflict. Ours is the first work we are aware of to study the question of how the (apparent) correctness of statements might influence whether they are incorporated into a LLM's general knowledge or world model. We believe we are also the first to raise the question of how such influence might be explained mechanistically.

In-context (meta-)learning. Brown et al. (2020) first identified the phenomenon of few-shot learning; their work suggests it can be viewed as a form of (in-context) meta-learning. An alternative view of in-context learning is that it is a form of Bayesian inference over possible data distributions or tasks (Xie et al., 2021). Chan et al. (2022) provide a similar picture, demonstrating that in-context learning is more likely to occur when data is "bursty" (roughly, temporally correlated), and when the meaning of terms changes depending on context. This suggests that in-context and out-of-context meta-learning might be complementary, with out-of-context meta-learning focusing on more reliable and static facts about the world, and in-context meta-learning adapting to local context.

6 Discussion

Understanding and forecasting AI systems' capabilities is crucial for ensuring their medium- and long-term safety. Our work investigates whether LLM training biases models towards internalizing information that appears broadly useful, even when doing so does not improve training performance on specific datapoints containing that information. We argue that this type of learning behavior resembles the phenomenon of a human reading a text on some unfamiliar topic, and deciding whether to believe its contents based on indicators of its reliability. Such learning behavior would seem to represent a surprising capability which might change designer's estimation of system's potential to do harm. In particular, we believe strong internalization is a plausible mechanism by which LLMs might acquire situational awareness (Ngo, 2022) and learn to enact normative principles of optimal reasoning (Steele and Stefánsson, 2020).

One particularly concerning type of normative principle that has been postulated is decision theories which encourage intelligent agents to cooperate with other similar agents (Yudkowsky and Soares, 2017). This could lead myopic systems to pursue long term goals, including influencing the state of the world and tampering with their loss or reward signal. Krueger et al. (2020) argue that while reinforcement learning (RL) agents have incentives to influence the state of the world, such incentives may be absent from supervised learning systems. However, our results challenge this finding, suggesting that even self-supervised LLMs may learn to cooperate with future copies of themselves, and thus may seek power (Turner et al., 2019), e.g. sacrificing performance on current examples in order to make future examples more predictable. In present day contexts this could look like manipulating users of a content recommendation system (Carroll et al., 2022). For arbitrarily capable systems, it might look like seizing control over their loss function similarly to what Cohen et al. (2022) describe with RL agents. We are interested in better understanding out-of-context meta-learning so we can either definitively rule out such scenarios (at least those where internalization is part of the mechanism), or take active measures that would prevent such scenarios from occurring.

Conclusion. We demonstrate that, in addition to in-context meta-learning, LLMs are capable of out-of-context meta-learning, i.e. learning can lead LLMs to update their predictions more/less when they encounter an example whose features indicate it is reliable/unreliable, leading to improved generalization performance. We believe this phenomenon may have significant implications for our understanding of foundation models, gradient-descent-based methods, and deep learning in general.

REFERENCES

Biderman, S., Schoelkopf, H., Anthony, Q., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. (2023). Pythia: A suite for analyzing large language models across training and scaling. *arXiv preprint arXiv:2304.01373*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Burns, C., Ye, H., Klein, D., and Steinhardt, J. (2022). Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.

Carroll, M. D., Dragan, A., Russell, S., and Hadfield-Menell, D. (2022). Estimating and penalizing induced preference shifts in recommender systems. In *International Conference on Machine Learning*, pages 2686–2708. PMLR.

Chan, S. C., Santoro, A., Lampinen, A. K., Wang, J. X., Singh, A., Richemond, P. H., McClelland, J., and Hill, F. (2022). Data distributional properties drive emergent few-shot learning in transformers. *arXiv preprint arXiv:2205.05055*.

Cohen, M., Hutter, M., and Osborne, M. (2022). Advanced artificial agents intervene in the provision of reward. *AI Magazine*, 43(3):282–293.

Elsahar, H., Vougiouklis, P., Remaci, A., Gravier, C., Hare, J., Laforest, F., and Simperl, E. (2018). T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. (2020). The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Krueger, D., Maharaj, T., and Leike, J. (2020). Hidden incentives for auto-induced distributional shift. *arXiv preprint arXiv:2009.09153*.

Laouenan, M., Bhargava, P., Eyméoud, J.-B., Gergaud, O., Plique, G., and Wasmer, E. (2022). A cross-verified database of notable people, 3500bc-2018ad. *Scientific Data*, 9(1):1–19.

Li, D., Rawat, A. S., Zaheer, M., Wang, X., Lukasik, M., Veit, A., Yu, F., and Kumar, S. (2022). Large language models with controllable working memory. *arXiv preprint arXiv:2211.05110*.

Lu, Y., Bartolo, M., Moore, A., Riedel, S., and Stenetorp, P. (2021). Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. (2022). Locating and editing factual knowledge in gpt. *arXiv preprint arXiv:2202.05262*.

Mitchell, E., Lin, C., Bosselut, A., Finn, C., and Manning, C. D. (2021). Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

Ngo, R. (2022). The alignment problem from a deep learning perspective. *arXiv preprint arXiv:2209.00626*.

Nichol, A., Achiam, J., and Schulman, J. (2018). On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., and Riedel, S. (2019). Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

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

Shazeer, N. and Stern, M. (2018). Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Sinitsin, A., Plokhotnyuk, V., Pyrkin, D., Popov, S., and Babenko, A. (2020). Editable neural networks. *arXiv preprint arXiv:2004.00345*.

Smith, S. L., Dherin, B., Barrett, D. G., and De, S. (2021). On the origin of implicit regularization in stochastic gradient descent. *arXiv preprint arXiv:2101.12176*.

Steele, K. and Stefánsson, H. O. (2020). Decision Theory. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2020 edition.

Turner, A. M., Smith, L., Shah, R., Critch, A., and Tadepalli, P. (2019). Optimal policies tend to seek power. *arXiv preprint arXiv:1912.01683*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. (2021). An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Yudkowsky, E. and Soares, N. (2017). Functional decision theory: A new theory of instrumental rationality. *arXiv preprint arXiv:1710.05060*.

Zhao, Z., Wallace, E., Feng, S., Klein, D., and Singh, S. (2021). Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

A QA DATASET GENERATION

This section describes the creation of datasets used to elicit out-of-context meta learning. This data is not IID, which can be seen in the graphical model in Figure 3.

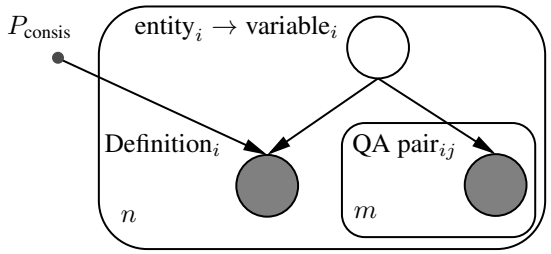

Figure 3: Probabilistic graphical model for dataset creation. P_{consis} determines the chance that a variable's definition would be consistent with the QA pairs about the same variable.

A.1 CVDB

We used a Cross-Verified database (CVDB) of notable people 3500BC-2018AD Laouenan et al. (2022) which includes 2.23m individuals. We removed all names which contain non-alphanumeric characters. Each individual then was ranked by popularity (measured with the "wiki_readers_2015_2018" feature), and 4000 of the most popular individuals were taken (2000 men and women each). We employ 6 types of questions:

1. Gender question: "What was the gender of <name>?". Example answer: "male".
2. Birth date question: "When was <name> born?". Example answer: "19 century".
3. Date of death question: "When did <name> die?" Example answer: "1910s".
4. Question about region: "In which region did <name> live?" Example answer: "Europe".
5. Activity question: "What did <name> do?" Example answer: "actor".
6. Nationality question: "What was the nationality of <name>?" Example answer: "France".

Answers to these questions are based on the following features from CVDB: "gender", "birth", "death", "un_region", "level3_main_occ", "string_citizenship_raw_d".

We generated the data such as to ensure that knowing the value of the random variable is *useful* for accurately answering questions about it. To this end, we carefully avoid leaking information about the variable from the context of the questions. For example, if one of the questions is "When did xyz announce iPhone 4s?", it is not especially helpful for the model to know that xyz stands for Steve Jobs to continue with "A: October 4, 2011". Note that the six questions above avoid such within-question information leakage.

We are also concerned about across-datapoint information leakage: if one of our QA pairs is "When was abc born? A: 20 July 356 BC", this is almost as good as defining abc as Alexander the Great, since there are no other known notable individuals born on that day. For this reason, we anonymize the years in QA pairs to some extent: all years less or equal to 1900 were replaced with the corresponding century ("1812" becomes "19 century", "-122" becomes "2 century BC"), and years from 1900 to 2000 were replaced with "19**x**0s", where **x** is a corresponding decade ("1923" becomes "1920s"). Years greater or equal to 2000 were left unchanged.

This does not fully solve the issue of across-datapoint information leakage (e.g. knowing that someone was born in the 18th century allows one to say that they also died in the 18th or the 19th century), but suffices to make definitions useful enough for our experiments.

A.2 T-REx

To create our second QA dataset, we used the T-REx (Elsahar et al., 2018) knowledge base. First, we extracted all possible triplets of (subject, predicate, object). Then, we selected the triplets where the

predicate is related to creative works, described in Table 1. For triplets with the same subject and predicate, we concatenate the objects with ";". The resulting triplets are converted into QA pairs in accordance with Table 1. Finally, we select QA pairs s.t. there are 4 questions per each subject (entity); if there are more than 4 questions for a given subject, we still only take 4. This is the case for a bit over 6900 entities, which we round down to 6900.

A note on QA pair creation. Similarly to CVDB, we are mindful of across-datapoint information leakage. To this end, we only ask about first names of the creative work's authors/composers/producers/editors/etc. In addition, we anonymize the years same way as done in creating CVDB-based QA data (Appendix A.1).

Predicate	Question
P180	What does [X] depict?
P195	Which collection is [X] part of?
P135	Which movement is [X] associated with?
P123	Who is the publisher of [X]?
P750	What is the distributor of [X]?
P275	What is the license of [X]?
P127	Who owns [X]?
P178	Who developed [X]?
P407	In which language was [X] published?
P364	In which language was [X] published?
P577	When was [X] published or released?
P179	Which series is [X] part of?
P50	First name of the author of [X]?
P57	First name of the director of [X]?
P58	First name of the screenwriter of [X]?
P344	First name of the cinematographer of [X]?
P161	First name of a cast member of [X]?
P162	First name of the producer of [X]?
P1040	First name of the editor of [X]?
P98	First name of the editor of [X]?
P88	First name of the commissioner of [X]?
P86	First name of the composer for [X]?
P136	What is the genre of [X]?
P921	What is the main subject of [X]?
P840	Where is [X] set?
P915	Where was [X] filmed?

Table 1: Given a triplet (subject, predicate, object), the question-answer pair is composed by replacing [X] with the subject in the question, and using the object as the answer.

B HYPERPARAMETERS FOR TWO-PHASE FINETUNING

We use the HuggingFace Transformers (Wolf et al., 2020) library to finetune the LLMs on $\mathcal{X}_1$ for 20 epochs, and on $\mathcal{X}_2$ for 10 epochs. We use the Adafactor optimizer (Shazeer and Stern, 2018) with batch size of 256 datapoints. All the other hyperparameters are the defaults in the Transformers library Trainer class. We do not use chunking so as to avoid in-context learning, and instead pad our datapoints to `max_context_length` $= 64$. We use the `deduped` versions of the Pythia models (Biderman et al., 2023).

C ADDITIONAL RESULTS FOR PYTHIA 2.8B

C.1 2-STAGE RESULTS

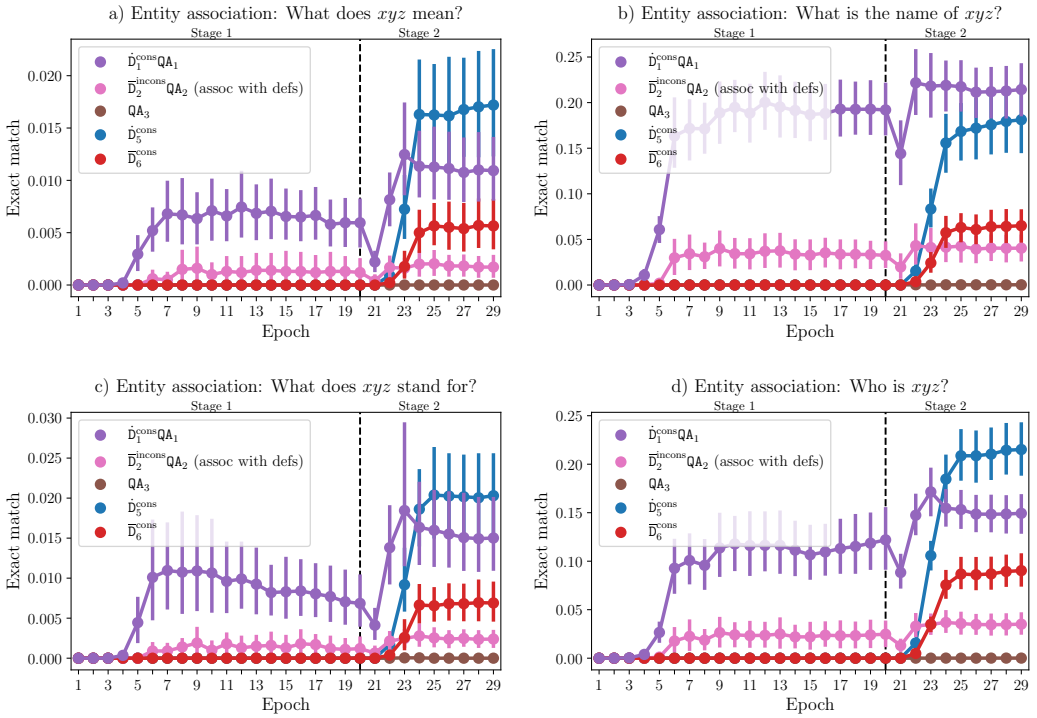

Figure 4: Entity attribution experiments for the Pythia 2.8B model on the CVDB dataset. We observe weak and strong internalization for all four question types. Plot b) is the same as Figure 2b in the main paper.

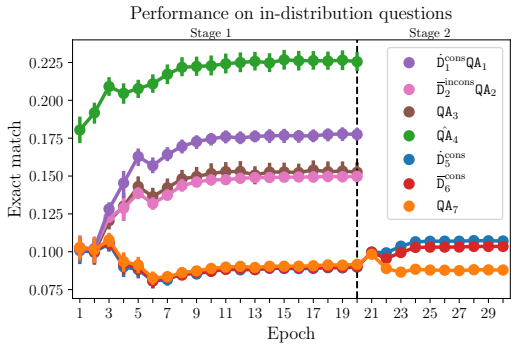

Figure 5: Exact match on the validation subsets for the Pythia 2.8B model finetuned on the T-REx dataset in two stages. As with CVDB, we observe weak and strong internalization, albeit strong internalization has a smaller effect than for CVDB (the gap between the blue and the red lines in the second stage is smaller).

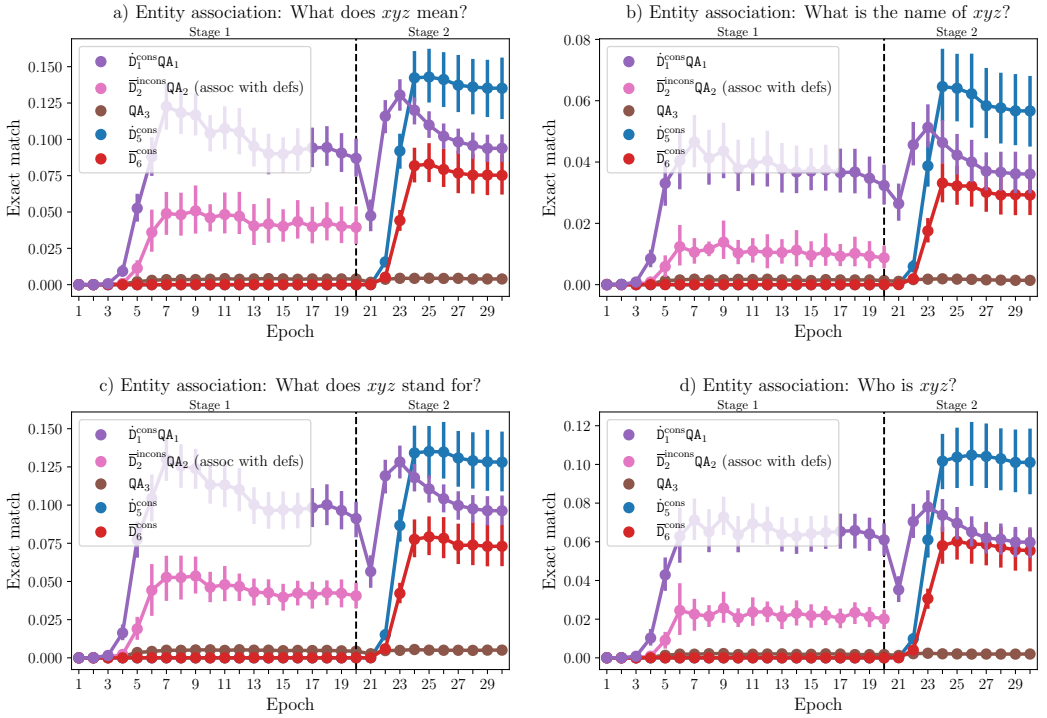

Figure 6: Entity attribution experiments for the Pythia 2.8B model on the T-REx dataset. The results appear broadly in line with those observed with the CVDB dataset: we observe weak and strong internalization for all four question types.

C.2 SINGLE-STAGE RESULTS

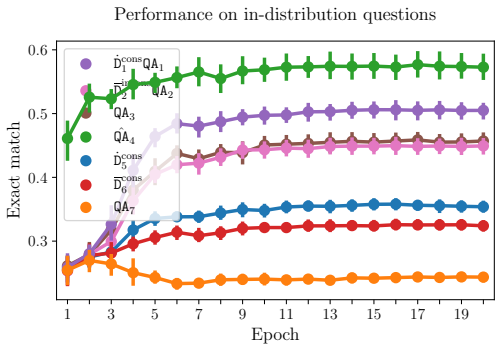

Figure 7: Exact match on the validation subsets for the Pythia 2.8B model finetuned on the CVDB dataset a single stage. As with two-stage experiments, we observe weak and strong internalization.

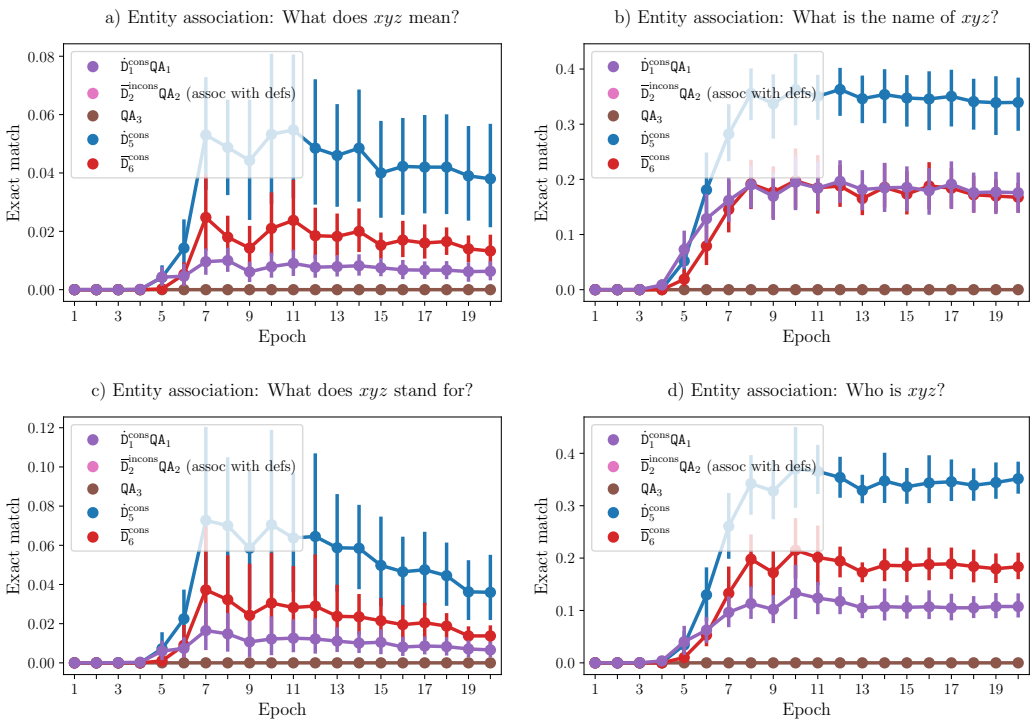

Figure 8: Single-stage entity attribution experiments for the Pythia 2.8B model on the CVDB dataset over 10 seeds. We observe strong internalization for all four question types. NOTE: this experiment was accidentally launched with $\bar{\mathtt{D}}_2^{\text{incons}}\mathtt{QA}_2$ test set disabled, so we cannot say anything about weak internalization from this.

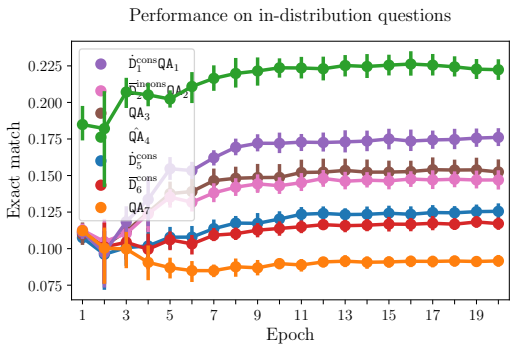

Figure 9: Exact match on the validation subsets for the Pythia 2.8B model finetuned on the T-REx dataset a single stage. As with two-stage experiments, we observe weak and strong internalization.

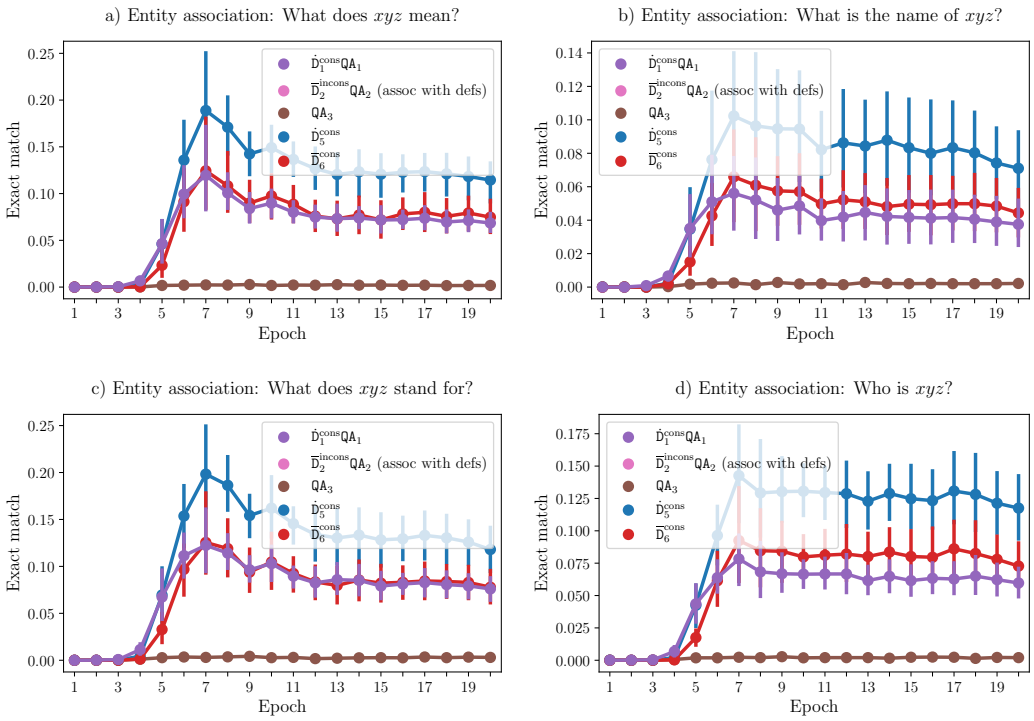

Figure 10: Single-stage entity attribution experiments for the Pythia 2.8B model on the T-REx dataset over 10 seeds. We observe strong internalization for all four question types. NOTE: this experiment was accidentally launched with $\bar{\mathtt{D}}_2^{\text{incons}}\mathtt{QA}_2$ test set disabled, so we cannot say anything about weak internalization from this.

D RESULTS FOR PYTHIA 410M

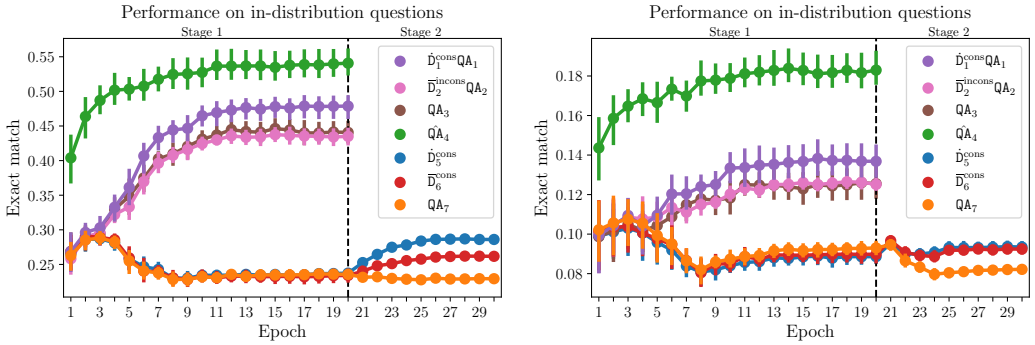

Figure 11: Exact match on the validation subsets for the Pythia 410M model finetuned on the CVDB (left) and T-REx (right) datasets in two stages. We clearly observe weak and strong internalization on CVDB. For T-REx, it appears that the model may be too small to detect strong internalization reliably.

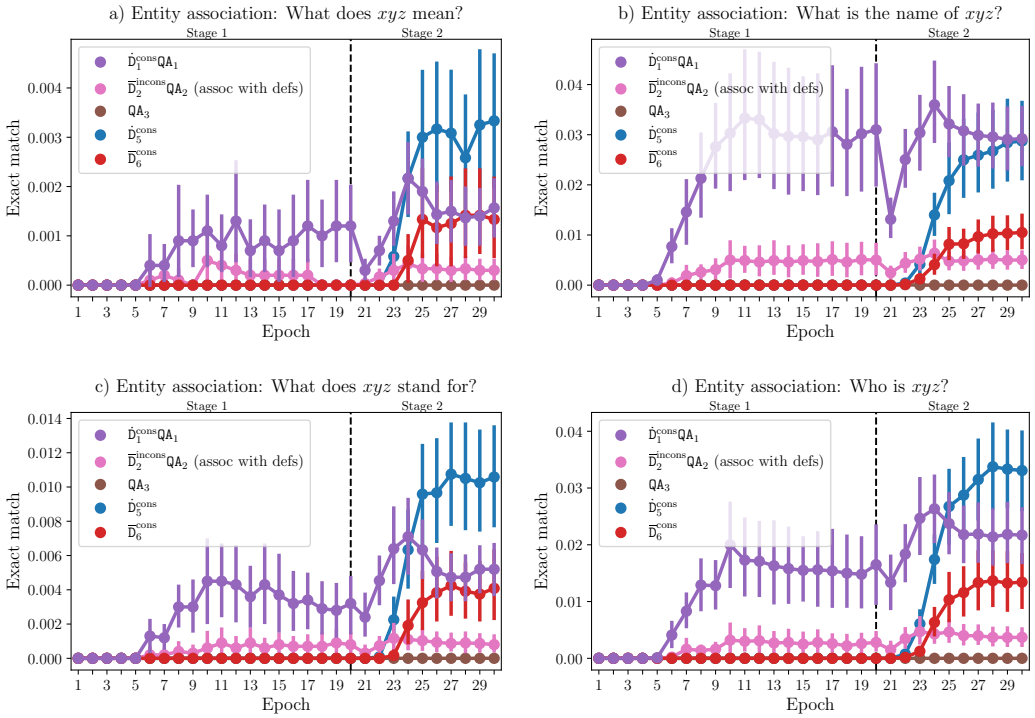

Figure 12: Entity attribution experiments for the Pythia 410M model on the CVDB dataset. The results appear broadly in line with those observed with the larger Pythia model: we observe weak and strong internalization for all four question types. However, the absolute values of EM appear much lower than those of similar experiments with the 2.8B model.

E SEQUENCE-TO-SEQUENCE MODELS: SETUP AND RESULTS

Sequence-to-sequence (seq2seq) models are a logical choice to try in our experiments, as the question-answering task aligns well with their capabilities. In a seq2seq model, if the input is a question and the output is an answer, the model first encodes the representation of the question before generating the answer. We employ T5-3B (Raffel et al., 2020), an encoder-decoder transformer model, so the loss is calculated only for the outputs of the decoder that produces the answer. To adapt our experiments to encoder-decoder architecture, we created additional text columns for the tokenizer to encode different input-output representations. The input column for QA data points consists of the substring up to and including "A:", while the output is the remaining portion of the string. For example, the QA string "Q: what did *xyz* do? A: Queen" is divided into "Q: what did *xyz* do? A:" and " Queen". Definitions are separated in a similar manner: "Define *xyz* Cleopatra" is split into "Define *xyz*" and " Cleopatra".

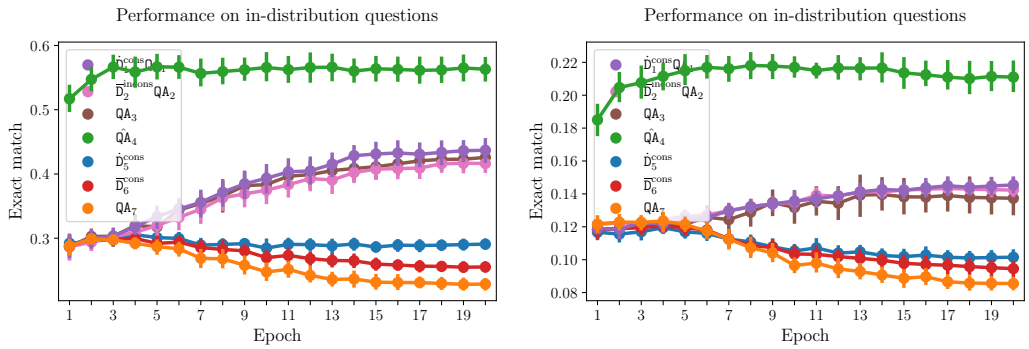

Figure 13: T5-3B finetuned in a single stage on CVDB (left) and T-REx (right) datasets. The weak internalization effect is seemingly present but barely visible; strong internalization is clearly present.

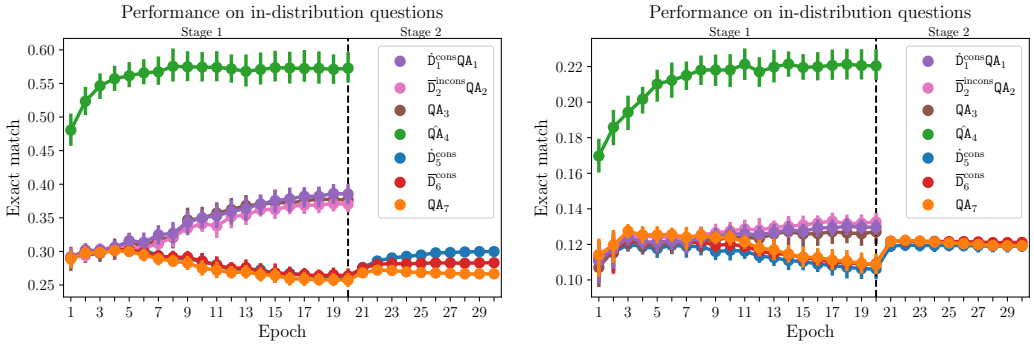

Figure 14: T5-3B finetuned in a two stages on CVDB (left) and T-REx (right) datasets. For CVDB, the weak internalization effect is seemingly present but barely visible; strong internalization is clearly present. For T-REx, looks like neither weak nor strong internalization is present.