Measuring Task-Agnostic Training Data Influence Across Language Model Pretraining

Yuto Nishida[1,2] **Hirokazu Kiyomaru**[2] **Yusuke Oda**[2] **Takashi Kodama**[2]
Chaoran Liu[2] **Daisuke Kawahara**[2,3] **Yusuke Miyao**[2,4]
Max Müller-Eberstein[4,5] **Masaru Isonuma**[2,6]

[1]Nara Institute of Science and Technology [2]NII LLMC [3]Waseda University
[4]The University of Tokyo [5]IT University of Copenhagen [6]Tohoku University

Contact: `nishida.yuto.nu8@is.naist.jp`

Abstract

Measuring training data influence consistently across language model pretraining is challenging. It is difficult to select downstream tasks or validation sets representative of a model's general capabilities, and reliance on task performance at intermediate checkpoints complicates comparisons across training. We propose a measure of training data influence that does not require selecting a downstream task or validation set as the attribution target. Specifically, we define an example's influence by how much its gradient update reduces the squared distance to the final parameters of a given pretraining run, and estimate this quantity from intermediate checkpoints without retraining. Applying the method to 18 configurations from the Pythia and PolyPythia suites, we find systematic temporal changes in influential data. Early in training, literature-related data are more strongly aligned with the trajectory toward the final parameters, whereas STEM data become more strongly aligned in later stages. This qualitative crossover is broadly consistent across model configurations. Our results provide a tractable trajectory-level view of how influential data change throughout pretraining, complementing influence analyses defined with respect to specific downstream tasks or validation sets.

1 Introduction

Understanding how individual training examples shape a model has long been an important problem in machine learning (Koh & Liang, 2017; Ghorbani & Zou, 2019; Ilyas et al., 2022). Such analyses are not only useful for answering questions about what a model learns from its data, but also for practical goals such as debugging anomalous predictions (Koh & Liang, 2017), identifying harmful or biased training instances (Pan et al., 2025; Jiao et al., 2025), guiding data curation (Yu et al., 2024; Xia et al., 2024), tracing data provenance (Akyurek et al., 2022), and supporting data valuation (Ghorbani & Zou, 2019; Choe et al., 2025).

Most existing methods for training data influence define an example's effect with respect to a specific target behavior, such as the loss or prediction on a test point, or downstream task performance on an evaluation metric (Koh & Liang, 2017; Ghorbani & Zou, 2019; Pruthi et al., 2020; Hammoudeh & Lowd, 2024). Such task-specific analyses are natural when the goal is to explain or improve performance on a particular task and have already revealed informative patterns in pretraining, including shifts in which domains matter most for particular capabilities (de la Rosa et al., 2025; Wang et al., 2025).

Language model pretraining, however, aims to develop broad, general capabilities rather than optimize for a single task or capability. Therefore, it is difficult to select downstream tasks or validation sets that are representative of these general capabilities throughout training. Moreover, influence estimates tied to task performance at intermediate checkpoints can be difficult to compare consistently across the full pretraining trajectory, since early

checkpoints may have acquired capabilities relevant to the target task without yet being able to fully solve it.

In this work, we reformulate training data influence across language model pretraining in task-agnostic terms. Rather than tying influence to a downstream metric, we view it as the extent to which an individual training example moves the model toward the final parameters of a given pretraining run (Figure 1). Specifically, we quantify an example's influence by how much its gradient update reduces the squared L2 distance to the final model weights, and estimate this quantity from intermediate checkpoints without retraining. The final model thus serves as a common reference point throughout pretraining, yielding a notion of influence that can be compared consistently across different stages of training without relying on task labels or benchmark accuracy.

Across the full pretraining trajectories of 18 variants of the Pythia model family (Biderman et al., 2023; van der Wal et al., 2025), each with 154 checkpoints, we find consistent stage-dependent shifts in which kinds of data are most influential. For instance, literature data are more influential during early training, whereas STEM data become increasingly influential in later stages. These results suggest that the training examples that most strongly move the model toward its final pretrained parameters are not static across pretraining, motivating stage-aware approaches to pretraining data composition and curriculum design.

Our contributions are threefold:

- We reformulate training data influence across language model pretraining in task-agnostic terms and derive a checkpoint-based approximation for post hoc estimation without retraining (§ 3).
- We characterize how training data contribution evolves throughout pretraining, revealing systematic stage-dependent variation across different types of training data (§ 4).
- We establish the reliability and scope of the proposed analysis across checkpoint approximation, reference endpoints, model configurations, and attribution targets (§ 5).

2 Related Work

Training data influence has been studied from multiple perspectives: Influence Functions (Koh & Liang, 2017) estimate how upweighting or removing a training point changes a test prediction or loss, and Data Shapley (Ghorbani & Zou, 2019) assigns each datum a marginal value with respect to predictive utility. More broadly, prior work typically defines a training example's effect with respect to a specific task, such as a prediction, loss, or performance measure (Hammoudeh & Lowd, 2024). Our work differs in replacing such task-conditioned targets with progress toward the final pretrained model.

Because estimating training data influence directly is often computationally expensive, prior work has developed scalable approximations. TracIn (Pruthi et al., 2020) estimates the influence of a training example on a test prediction by tracing gradients through saved checkpoints. TRAK (Park et al., 2023) improves the tractability-efficacy trade-off for large non-convex models. More recent methods extend influence estimation and data valuation to foundation-model scale, including LoGra (Choe et al., 2025) for scalable influence functions in LLMs. Our method shares this goal of tractable influence estimation at scale: by leveraging intermediate checkpoints from an existing pretraining run, it can be applied to LLMs without retraining from scratch.

Influence analysis has also been applied to language model pretraining and has begun to uncover systematic patterns in how pretraining data shapes a model. Wang et al. (2025) show that influence patterns evolve substantially over training: for a math-related validation corpus, specialized domains such as ArXiv and GitHub receive disproportionately high value, while more general web corpora contribute more early in training and diminish later. Chang et al. (2025) study fact tracing in LLM pretraining and find that highly influential

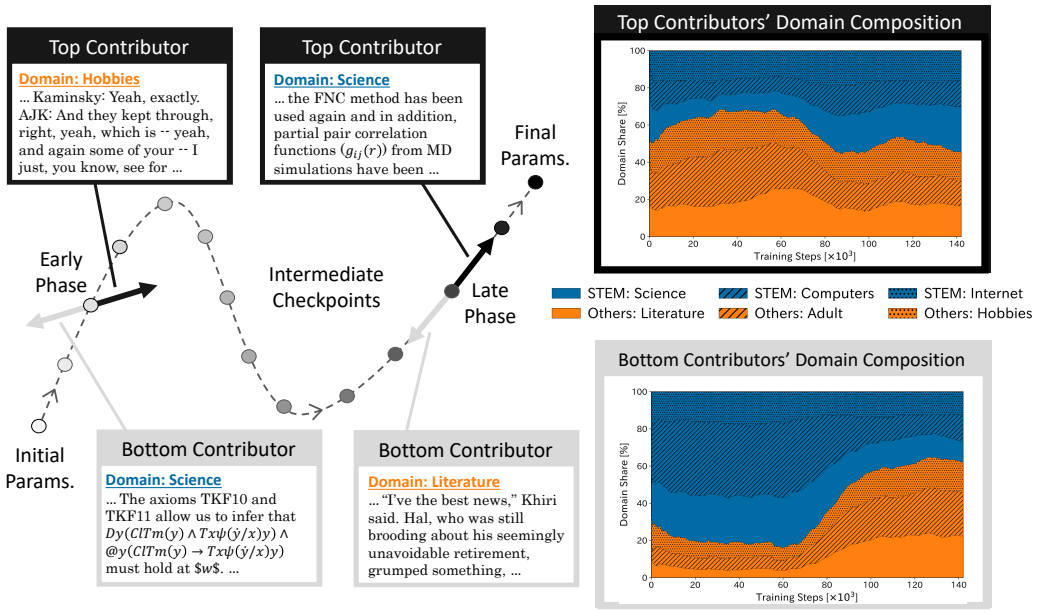


Figure 1: Overview of our task-agnostic, training data contribution measure: Top contributors move the model towards its final parameters, while bottom contributors yield negligible or opposing updates. Across pretraining, our experiments reveal a cross-over of STEM versus non-STEM domains, in which the former gain importance during later training.

examples are not always passages that explicitly state the queried fact, highlighting a gap between causal influence and factual attribution that narrows with scale. DATE-LM (Jiao et al., 2025) further shows that, when data attribution methods are evaluated through applications such as training-data selection, toxicity/bias filtering, and factual attribution, no single method dominates across tasks and performance is sensitive to task-specific evaluation design. Together, these results suggest that task-conditioned evaluation offers only a partial view of how pretraining data shapes a model, motivating a task-agnostic perspective on training data influence across pretraining.

3 Method

In this section, we introduce a task-agnostic measure of training data influence based on reduction in parameter-space distance to the final parameters of a given pretraining run. Our formulation is inspired by trajectory-matching methods in dataset distillation (Cazenavette et al., 2022; Cui et al., 2023; Liu et al., 2025), but uses parameter-space geometry retrospectively to quantify how real training examples move the model toward its final parameters rather than to optimize synthetic data. We first define this quantity at the mini-batch and example levels, and then derive a checkpoint-based approximation that allows it to be estimated post hoc from intermediate checkpoints without retraining.

3.1 Mini-Batch Influence

Let updates be indexed by $t = 0, 1, \ldots, T-1$, let $\theta_t \in \mathbb{R}^d$ denote the model parameters before update t, and let $B_t = (x_1^t, \ldots, x_N^t)$ denote the mini-batch used at that update. The corresponding parameter update is $\Delta_t := \theta_{t+1} - \theta_t$. Given the final pretrained parameters $\theta^* := \theta_T$, and $S_t := \|\theta^* - \theta_t\|_2^2$ as the squared L2 distance from the current parameters to the final model, we define the influence of mini-batch B_t as the reduction in distance to the final parameters induced by the update at step t:

$$\mathrm{Cont}(B_t) := S_t - S_{t+1}. \tag{1}$$

This quantity is exact and independent of optimizer-specific decompositions. Thus, expanding S_{t+1} using $\theta_{t+1} = \theta_t + \Delta_t$ gives $S_{t+1} = \|\theta^* - \theta_t\|_2^2 - 2\Delta_t^\top(\theta^* - \theta_t) + \|\Delta_t\|_2^2$, yielding

$$\mathrm{Cont}(B_t) = 2\Delta_t^\top(\theta^* - \theta_t) - \|\Delta_t\|_2^2. \tag{2}$$

The first term measures the alignment between the update direction and the direction towards the final parameters. The second term penalizes the squared norm of the update. Therefore, a mini-batch has high influence when the alignment term outweighs the squared-norm penalty. $\mathrm{Cont}(B_t)$ can be negative when the update is poorly aligned with the direction toward the final parameters, indicating that the update moves the model away from the final model in terms of squared distance.

3.2 Example-Level Influence

To decompose mini-batch updates into example-level terms, we assume for analysis that optimization is performed by standard stochastic gradient descent (SGD) with loss function ℓ and learning rate η_t at step t.[1] The update induced by example x_k^t is $\Delta_{t,k} := -\eta_t \nabla_\theta \ell(x_k^t; \theta_t)$, so that $\Delta_t = \sum_{k=1}^N \Delta_{t,k}$. We define the influence of example x_k^t by

$$\mathrm{Cont}(x_k^t) := 2\Delta_{t,k}^\top(\theta^* - \theta_t) - \Delta_{t,k}^\top \Delta_t. \tag{3}$$

By construction, the example-level influences sum to the mini-batch influence, i.e., $\sum_{k=1}^N \mathrm{Cont}(x_k^t) = \mathrm{Cont}(B_t)$. This definition also shows that example-level influence depends on other examples in the same mini-batch. Expanding the second term in Equation (3) gives $\mathrm{Cont}(x_k^t) = 2\Delta_{t,k}^\top(\theta^* - \theta_t) - |\Delta_{t,k}|2^2 - \sum_{j \neq k} \Delta_{t,k}^\top \Delta_{t,j}$, so the penalty decomposes into an example-level norm term and cross-example interaction terms within the mini-batch.

Using the simplifying assumption that per-example update vectors are pairwise orthogonal within the mini-batch, we can see the connection of our method with TracIn (Pruthi et al., 2020) in that the interaction term vanishes and the remaining norm term becomes $\|\Delta_{t,k}\|_2^2 = \eta_t^2 \|\nabla_\theta \ell(x_k^t; \theta_t)\|_2^2$. This term corresponds, up to scalar factors determined by the learning rate and batching convention, to the *self-influence* term in TracIn for the special case where the training and test examples coincide. Pruthi et al. (2020) note that incorrectly labeled examples tend to have large self-influence. Our formulation is consistent with this observation: for potentially noisy examples, such as mislabeled ones, the corresponding self-influence-like norm term tends to be large, so it acts as a penalty for example-level influence, assigning smaller influence to such examples.

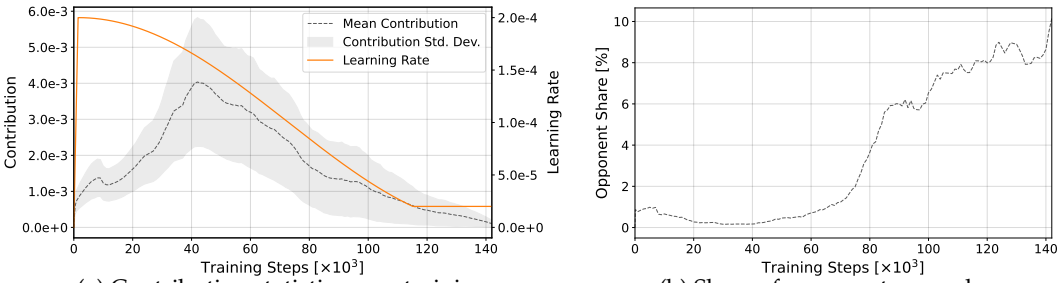
3.3 Checkpoint-Based Approximation

Computing the quantities above for every training step would require access to all intermediate states or retraining the model. In practice, we instead assume access to periodically saved checkpoints from an existing pretraining run. Let $c < c'$ be consecutive checkpoint steps such that $c \leq t < c'$, and let θ_c and $\theta_{c'}$ denote the corresponding saved parameters. We approximate $\theta_t \approx \theta_c$ and $\Delta_t \approx \theta_{c'} - \theta_c$, i.e., we treat all examples in the interval $[c, c')$ as if they were processed at the checkpoint state θ_c. For an example x_k^t in that interval, let $\Delta_{c,k} := -\eta_t \nabla_\theta \ell(x_k^t; \theta_c)$. We then estimate example-level influence by

$$\mathrm{Cont}(x_k^t) \approx 2\Delta_{c,k}^\top(\theta^* - \theta_c) - \Delta_{c,k}^\top(\theta_{c'} - \theta_c). \tag{4}$$

This approximation replaces step-level quantities by checkpoint-level quantities and allows the proposed measure to be applied post hoc to large pretraining runs without retraining from scratch. In our experiments, we use Equation (4) to estimate example-level influence from publicly available checkpoints.

[1] Actual LLM pretraining typically uses adaptive optimizers such as Adam (Kingma & Ba, 2015). We adopt the SGD assumption only for the local example-level decomposition, following the SGD-based formulation used in TracIn (Pruthi et al., 2020). Under adaptive optimizers, an example's exact effect on the current update also depends on optimizer states carried over from previous mini-batches, making the decomposition substantially more involved. Our goal here is to obtain a tractable per-example approximation for influence analysis.



(a) Contribution statistics over training (b) Share of opponent examples

Figure 2: Training dynamics of the contribution distribution on Pythia-1.4B-Deduped. All curves are moving averages with window size 10.

4 Experiments

We apply the proposed measure to characterize how training data contribution evolves across pretraining and which kinds of data contribute most strongly at different stages.

4.1 Setup

We evaluate the proposed influence measure on multiple publicly released pretraining trajectories. For the Pythia suite (Biderman et al., 2023), we use the deduplicated models at six scales: 70M, 160M, 410M, 1.4B, 6.9B, and 12B. Each model is trained on 300B tokens from The Pile (Gao et al., 2020) and provides 154 checkpoints at steps $0, 1, 2, 4, \ldots, 512, 1000, 2000, \ldots, 143000$.

We additionally analyze the PolyPythias (van der Wal et al., 2025) at the 160M and 410M scales to examine the robustness of our findings to variations in model initialization and data ordering. For each model size, we use three runs in which both factors vary. At the 160M scale, we further use three runs from each of the decoupled `data-seed` and `weight-seed` variants, allowing the effects of data ordering and weight initialization to be examined separately. Each PolyPythia run likewise provides 154 checkpoints.

The Pythia and PolyPythia models are trained with batches of 1024 sequences, each containing 2048 tokens. We treat each 2048-token sequence as one training example. In all experiments, we sample examples from the actual training data stream associated with each checkpoint interval, rather than from held-out or proxy corpora. This allows us to relate the estimated influence directly to the data observed by the model during pretraining.

We estimate example-level influence using the checkpoint-based approximation described in § 3.3, namely Equation (4). This allows us to estimate influence post hoc from publicly released checkpoints without retraining the model. We first examine the overall trajectory of influence in § 4.2, and then analyze its relationships with text difficulty (§ 4.3) and textual domain (§ 4.4). Because the main qualitative findings are broadly consistent across model configurations, we first report results for Pythia-1.4B-Deduped and then examine robustness across model sizes, initializations, and data orderings in § 5.

4.2 Trajectory of the Contribution Distribution

We first characterize how the distribution of contribution, which measures how strongly each example's update moves the model toward its final parameters, evolves over training. For each checkpoint interval, we uniformly sample 1,000 examples from the actual training data and compute their contributions. We summarize the distribution using its mean and standard deviation, together with the share of examples with negative contribution. We refer to the latter as *opponents*, since under our definition their updates move the model away from its final parameters.

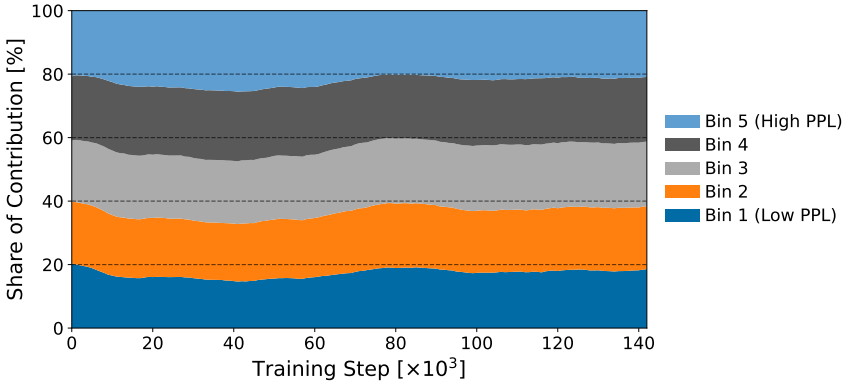


Figure 3: Relationship between text PPL and contribution for Pythia-1.4B-Deduped. Each curve shows the share of normalized contribution attributed to an equal-sized PPL bin, with a moving-average window of 10 checkpoint intervals.

As shown in Figure 2(a), the mean contribution increases substantially after the early stage, becomes largest around roughly 40k steps, and then gradually decreases toward the end of training. The standard deviation exhibits a similar pattern, peaking in the middle stage and shrinking thereafter. Since our contribution measure is defined as the reduction in distance to the final parameters, this suggests that updates in the middle stage play the largest role in moving the model toward its final state, while both early-stage and late-stage updates are less effective in this sense.

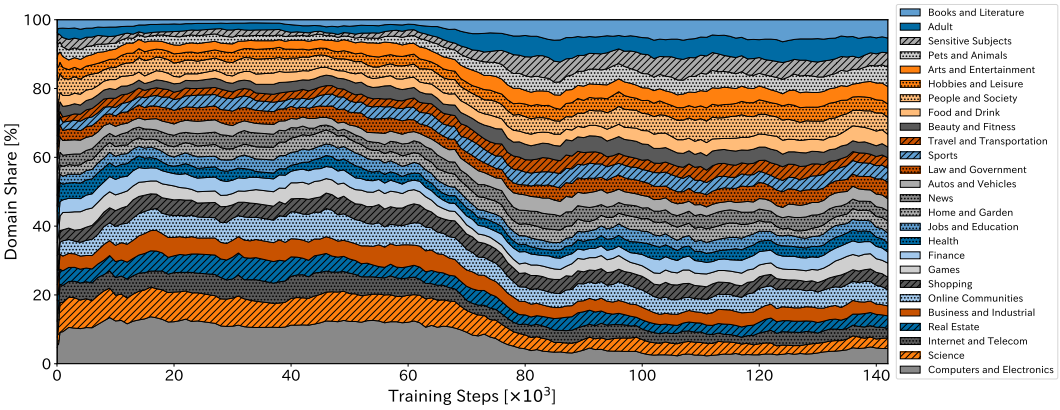
The opponent share in Figure 2(b) provides a complementary perspective. Opponents remain relatively rare through the early and middle stages, staying around 0–1% for most of the trajectory before about 70k steps. After that point, however, their share rises markedly, reaching roughly 6% around 90k steps and around 8–10% near the end of training. This indicates that the late-stage decline in mean contribution is not explained solely by smaller positive contributions: increasingly many examples induce updates that are weakly aligned with, or even opposed to, the direction toward the final parameters.

Since the learning rate affects the scale of parameter updates, one might intuitively expect larger learning rates to be associated with larger contributions. However, Figure 2(a) shows no simple one-to-one relationship between learning rate and contribution magnitude. In particular, the rise in opponent share in Figure 2(b) occurs while the learning rate is already decaying smoothly. This is consistent with the definition in Equation (2): contribution depends not only on the norm of the update but also on its alignment with the direction toward the final parameters. Thus, the later stages of training are characterized not only by smaller updates, but also by a growing fraction of opponent examples whose updates are less aligned with the eventual pretrained model.

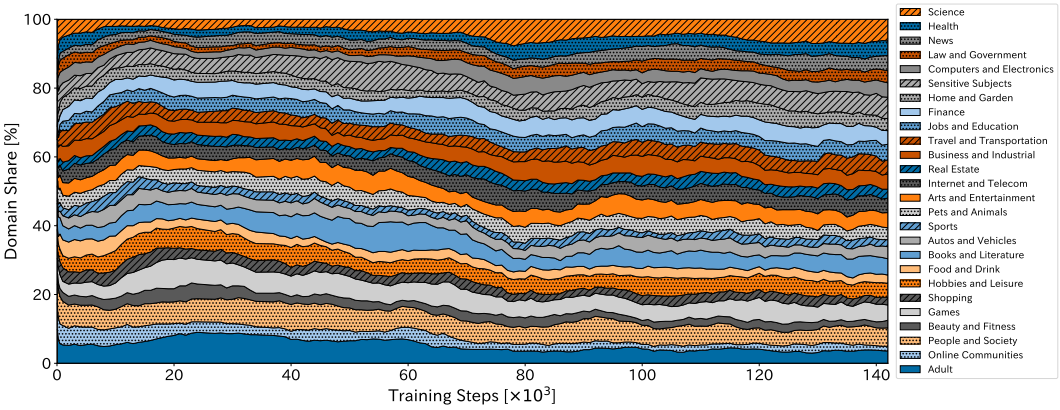
4.3 Relationship Between Text Difficulty and Contribution

Next, we focus on the relationship between training example difficulty and its contribution factor. As a measure for a text's relative difficulty for a given model, we compute perplexity (PPL) using the final checkpoint of Pythia-12B-Deduped. To place contributions on a positive scale, we apply min–max normalization within each checkpoint interval. We then divide the 1,000 sampled examples in each checkpoint interval from § 4.2 into five equally sized bins according to PPL and compute the mean normalized contribution in each bin.

Figure 3 shows that higher-PPL examples account for a larger share of normalized contribution during the middle stage of pretraining, while contribution is more evenly distributed across PPL bins during the early and late stages. In particular, the share of the highest-PPL bin increases from approximately 20% to 25% in the middle stage, whereas that of the lowest-PPL bin decreases from approximately 20% to 15%. Since all bins contain the same number



(a) Proportion of domains among examples in the bottom 5% of contributions.



(b) Proportion of domains among examples in the top 5% of contributions.

Figure 4: Domain composition of examples with extreme contribution values on Pythia-1.4B-Deduped. All curves are moving averages with window size 10.

of examples, this five-percentage-point shift represents a non-negligible redistribution of relative contribution toward higher-PPL data.

4.4 Relationship Between Text Domain and Contribution

To obtain a more fine-grained view of which kinds of texts contribute to the final model, we next analyze the relationship between a training example's domain and its contribution. To classify each example's domain, we use the NeMo Curator Domain Classifier,[2] which assigns one of 26 domains, such as *Finance* and *Food and Drink*, based on a text's semantic content. Rather than comparing average contributions across all examples, we focus on examples with extreme contribution values. Specifically, for each checkpoint interval, we examine the domain composition of the bottom 5% and top 5% of examples ranked by contribution. To avoid bias due to imbalanced domain frequencies, we randomly sample 100 examples from each domain in each checkpoint interval, resulting in 2,600 examples per interval for analysis.[3]

Figure 4(a) shows that low-contribution examples are concentrated in a few domains during the first half of training, with STEM-related domains such as *Computers and Electronics* and *Science* accounting for a large proportion. Under our contribution measure, this indicates that examples from these domains are relatively less aligned with progress toward the

[2] https://huggingface.co/nvidia/domain-classifier

[3] Checkpoint intervals in which any domain has fewer than 100 examples are excluded from this analysis.

Metric	Checkpoint interval		
	Early: 1000 → 2000	Mid: 10000 → 11000	Late: 100000 → 101000
Pearson r	0.622*	0.808*	0.950*
Spearman r	0.592*	0.811*	0.947*

Table 1: Correlation between exact example-level contribution and the checkpoint-based approximation on Pythia-70M-Deduped. Superscript * indicates a correlation significantly different from zero under a two-sided test ($p < 0.05$).

final parameters during the first half of training. In the later stages, however, the shares of low-contribution STEM domains decrease, while domains such as *Books and Literature* become more common among the low contributors.

Figure 4(b) shows a complementary pattern. Although STEM domains are not necessarily dominant among high-contribution examples early in training, their share tends to increase in later stages. Together with the bottom-5% results, this indicates that STEM-related data become increasingly aligned with progress toward the final parameters as pretraining proceeds. Representative text excerpts from interval-level top and bottom contributors are provided in § B.1, giving a concrete view of the examples underlying these domain-level patterns.

Overall, these results reveal a systematic shift in which textual domains yield high contribution across pretraining. In particular, literature-related data are relatively more prominent early in training, whereas STEM-related data become increasingly prominent in later stages.

5 Analysis

We next evaluate the reliability and scope of our contribution analysis. We validate the checkpoint-based approximation (§ 5.1), examine sensitivity to the reference endpoint (§ 5.2), assess robustness across model configurations (§ 5.3), and compare our measure with task-specific influence (§ 5.4). We then discuss implications for pretraining strategies (§ 5.5).

5.1 Validation of the Checkpoint-Based Approximation

Exact computation of example-level contribution requires access to the model parameters at every training step, whereas public model releases typically provide only sparse checkpoints. We therefore evaluate how well the checkpoint-based approximation in Equation (4) matches the exact example-level contribution in Equation (3) by retraining selected checkpoint intervals while storing all intermediate parameter states. Because this procedure is computationally expensive, we perform this validation only on Pythia-70M-Deduped, considering three checkpoint intervals: 1k→2k (early), 10k→11k (middle), and 100k→101k (late). At each step in each interval, we sample 100 examples from the actual training data processed at that step and compute both the exact and approximate contributions for the same examples. For each interval, we report the Pearson and Spearman correlations between the two quantities.

Table 1 shows positive correlations between the checkpoint-based approximation and the exact contribution in all three evaluated intervals, with substantially stronger agreement later in training. This trend is intuitive: parameter changes between adjacent checkpoints are larger early in training and become smaller later, making the checkpoint-level approximation increasingly accurate. Even in the early interval, the approximation shows moderate agreement with the exact contribution, with Pearson and Spearman correlations around 0.6. For the sampled middle and late intervals, both correlations exceed 0.8, and they are above 0.94 in the late interval, indicating strong agreement in both contribution magnitudes and rankings. Taken together, these results show that the checkpoint-based approximation tracks the exact example-level contribution increasingly well across the three evaluated stages of training.

5.2 Sensitivity to the Reference Endpoint

Our contribution score is defined relative to the final parameters of a particular pretraining run. We therefore examine how the score changes when alternative checkpoints are used as the reference. For Pythia-1.4B-Deduped, we use reference checkpoints at steps 30k, 70k, 120k, and 140k and compare the resulting contribution scores with the original scores defined using the final checkpoint at step 143k. For each of five representative training intervals, i.e., 10k→11k, 40k→41k, 80k→81k, 115k→116k, and 130k→131k, we use the same 1,000 examples sampled for the contribution analysis in § 4.2. For each alternative reference, we compute Pearson and Spearman correlations with the final-reference scores, as well as the overlap between their top and bottom 5% sets, and average each metric across the five intervals. The full results are reported in Appendix B.2.

Using the near-final 140k checkpoint as the reference leaves the contribution rankings almost unchanged, yielding a mean Spearman correlation of 0.997 and a mean top-5% overlap of 0.972 with the original scores. By contrast, substantially earlier reference checkpoints produce markedly different rankings. These results confirm that the contribution score is endpoint-dependent, while also showing that it is locally stable to the precise choice of a near-final reference checkpoint.

5.3 Consistency of Contribution Dynamics across Model Configurations

We next examine whether the contribution dynamics identified in § 4 are consistent across model scales, weight initializations, and data orderings.

Model Sizes. Across the six Pythia-Deduped model sizes, the contribution-distribution, text-difficulty, and domain analyses exhibit broadly similar stage-dependent dynamics, while also showing systematic scale-dependent differences (see Appendix B.3). In particular, several transitions that occur during the middle stage for larger models appear later for smaller models, suggesting that the timing of training-phase transitions may depend on model scale. We also observe that the magnitude of the contribution dynamics tends to be smaller at larger scales. Thus, model size appears to affect when and how strongly contribution patterns change across pretraining, while preserving much of their broader temporal structure.

Weight Initialization and Data Ordering. The qualitative domain-level contribution dynamics are also broadly consistent across PolyPythia runs with different random factors (see Appendix B.4). Across the standard 160M and 410M runs, similar temporal patterns are preserved across seeds, except for the anomalous 410M seed-3 run, which van der Wal et al. (2025) identify as an outlier exhibiting loss spikes and degraded performance. The decoupled PolyPythia-160M variants further show highly consistent dynamics when only weight initialization is varied, while changing data ordering introduces somewhat greater variation. Overall, these results suggest that the stage-dependent contribution patterns are robust to ordinary seed variation, although data ordering may have a somewhat larger effect than weight initialization.

5.4 Comparison with Task-Specific Influence

To examine how our contribution measure, which is agnostic to the choice of downstream task or validation set, differs from domain-specific attribution, we compare it with a TracIn-style influence score (Pruthi et al., 2020) computed with respect to domain-wise language-modeling loss. Specifically, we measure the loss on domain-specific held-out validation sets from The Pile, obtained by filtering out examples that occur in the Pythia pretraining stream. Using the same domain classifier as in § 4.4, we then sample 1,000 high-confidence examples for each of the four validation domains: *Books and Literature*, *Science*, *Computers and Electronics*, and *Hobbies and Leisure*. For each checkpoint interval, we use the same domain-balanced training examples as in § 4.4 and compute the gradient alignment between each training example's loss and the language-modeling loss on each domain-specific validation set. We then examine the domain composition of the top and bottom 5% of examples under

the resulting TracIn-style scores, analogously to § 4.4. The full experimental results are provided in Appendix B.5.

The resulting attribution patterns depend substantially on the selected validation domain. Training examples from the selected validation domain are strongly overrepresented among the highest-scoring examples early in training, particularly when *Science* is used as the validation target, although this enrichment becomes more diffuse in later stages. When non-STEM domains such as *Books and Literature* or *Hobbies and Leisure* are used as validation targets, STEM-related training examples are overrepresented among the lowest-scoring examples early in training. Thus, the domain-specific TracIn analysis does not consistently recover the Literature-to-STEM crossover observed with our contribution measure.

5.5 Implications for Pretraining Strategies

We conclude our analysis by discussing how the observed contribution dynamics inform our understanding of pretraining and the design of pretraining data strategies.

Pretraining Phases. For Pythia, van der Wal et al. (2025) report that all models exhibit an *initial learning phase* between 1k–10k steps in which baseline linguistic capabilities are acquired, as well as a *critical learning phase* at 10k–100k steps during which most downstream performance gains occur. Our contribution dynamics shed additional light on these phases: higher-PPL texts and STEM-related domains tend to contribute less during the initial learning phase but more strongly during the critical learning phase.

Stage-Aware Data Composition. A common view in curriculum learning (Bengio et al., 2009; Wang et al., 2022) is that training should proceed from easy to difficult examples. Our results do not fully support this simple progression: difficult texts contribute more strongly to moving the model toward the final parameters primarily during the middle stage, rather than becoming progressively more important toward the end of training. In practice, recent LLM pretraining recipes adopt a more specific stage-aware heuristic by increasing the allocation of STEM-related data, such as math and code, in later stages of training (Martins et al., 2024; Bakouch et al., 2025). Our contribution dynamics lend empirical support to this heuristic from a trajectory-level perspective, as STEM-related domains become increasingly prominent among high-contribution examples later in pretraining. More broadly, they motivate pretraining strategies that adapt data composition across training stages rather than treating a single data mixture as equally suitable throughout training.

Beyond Task-Specific Evaluation. Task-specific evaluations provide valuable signals for assessing pretraining data, but the resulting conclusions can depend on the choice of evaluation target. For instance, Jiao et al. (2025) show that the evaluation of data attribution methods is sensitive to task-specific evaluation design. Similarly, de la Rosa et al. (2025) report heterogeneous downstream effects across different types of literary data, while our contribution analysis identifies literature-related data as prominent contributors during early pretraining. We hope that our task-agnostic contribution measure can reveal stage-dependent roles of training data beyond fixed downstream evaluations and help guide more nuanced pretraining strategies.

6 Conclusion

In this work, we proposed a task-agnostic measure of training data influence by defining an example's contribution as the reduction in distance to the final parameters of a given pretraining run. Across model configurations, we found systematic contribution dynamics throughout pretraining, including earlier transitions for larger models, greater relative contribution from difficult texts during the middle stage, and a crossover from literature-related to STEM-related domains toward later training. Together, these results provide a trajectory-level view of how the training data most strongly aligned with progress toward the final parameters change across pretraining, complementing attribution analyses defined with respect to particular downstream tasks or validation sets.

Acknowledgments

This study is partially supported by JST BOOST JPMJBY24A6. Max Müller-Eberstein is supported by the Carlsberg Foundation, grant CF-25-0624. Computational resources were provided in part by "mdx: a platform for building data-empowered society."

References

Ekin Akyurek, Tolga Bolukbasi, Frederick Liu, Binbin Xiong, Ian Tenney, Jacob Andreas, and Kelvin Guu. Towards tracing knowledge in language models back to the training data. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2429–2446, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.180. URL https://aclanthology.org/2022.findings-emnlp.180/.

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, Xuan-Son Nguyen, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM3: smol, multilingual, long-context reasoner. https://huggingface.co/blog/smollm3, 2025.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 41–48, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

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 Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Tyler A. Chang, Dheeraj Rajagopal, Tolga Bolukbasi, Lucas Dixon, and Ian Tenney. Scalable influence and fact tracing for large language model pretraining. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=gLa96FlWwn.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, Jeff Schneider, Eduard Hovy, Roger Baker Grosse, and Eric P. Xing. What is your data worth to GPT? LLM-scale data valuation with influence functions. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=zPKeJAEo27.

Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to ImageNet-1K with constant memory. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6565–6590. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/cui23e.html.

Javier de la Rosa, Vladislav Mikhailov, Lemei Zhang, Freddy Wetjen, David Samuel, Peng Liu, Rolv-Arild Braaten, Petter Mæhlum, Magnus Breder Birkenes, Andrey Kutuzov, Tita Enstad, Hans Christian Farsethås, Svein Arne Brygfjeld, Jon Atle Gulla, Stephan Oepen, Erik Velldal, Wilfred Østgulen, Lilja Øvrelid, and Aslak Sira Myhre. The impact of copyrighted material on large language models: A Norwegian perspective. In Richard Johansson and Sara Stymne (eds.), *Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025)*, pp. 544–560, Tallinn, Estonia, March 2025. University of Tartu Library. ISBN 978-9908-53-109-0. URL https://aclanthology.org/2025.nodalida-1.59/.

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.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2242–2251. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ghorbani19c.html.

Zayd Hammoudeh and Daniel Lowd. Training data influence analysis and estimation: a survey. *Machine Learning*, 113(5):2351–2403, May 2024. ISSN 1573-0565. doi: 10.1007/s10994-023-06495-7. URL https://doi.org/10.1007/s10994-023-06495-7.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9525–9587. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ilyas22a.html.

Cathy Jiao, Yijun Pan, Emily Xiao, Daisy Sheng, Niket Jain, Hanzhang Zhao, Ishita Dasgupta, Jiaqi W. Ma, and Chenyan Xiong. DATE-LM: Benchmarking data attribution evaluation for large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=e2cD5xuHix.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Dai Liu, Jindong Gu, Hu Cao, Carsten Trinitis, and Martin Schulz. Dataset distillation by automatic training trajectories. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 334–351, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-73021-4.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. Eurollm: Multilingual language models for europe, 2024. URL https://arxiv.org/abs/2409.16235.

Yijun Pan, Taiwei Shi, Jieyu Zhao, and Jiaqi W. Ma. Detecting and filtering unsafe training data via data attribution with denoised representation. *arXiv preprint arXiv:2502.11411*, 2025. doi: 10.48550/arXiv.2502.11411. URL https://arxiv.org/abs/2502.11411.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: Attributing model behavior at scale. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27074–27113. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/park23c.html.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e6385d39ec9394f2f3a354d9d2b88eec-Paper.pdf.

Oskar van der Wal, Pietro Lesci, Max Müller-Eberstein, Naomi Saphra, Hailey Schoelkopf, Willem Zuidema, and Stella Biderman. Polypythias: Stability and outliers across fifty language model pre-training runs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=bmrYu2Ekdz.

Jiachen T. Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. Data shapley in one training run. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HD6bWcj87Y.

Xin Wang, Yudong Chen, and Wenwu Zhu. A Survey on Curriculum Learning . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 44(09):4555–4576, September 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3069908. URL https://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3069908.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=PG5fV50maR.

Zichun Yu, Spandan Das, and Chenyan Xiong. MATES: Model-aware data selection for efficient pretraining with data influence models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=6gzPSMUAz2.

A Limitations

Our contribution measure does not require selecting a downstream task or validation set as the attribution target, but it remains defined relative to the final parameters of a particular pretraining run. As shown in § 5.2, substantially different reference checkpoints can yield different contribution rankings. Moreover, squared parameter-space distance provides a scalable geometric notion of progress, but it is sensitive to parameterization and does not directly measure changes in model behavior. Function-aware alternatives based on predictive divergence or information geometry could provide complementary notions of contribution, but are substantially more expensive at pretraining scale.

Our checkpoint-based approximation replaces step-level model states with periodically saved checkpoints. We validate this approximation on three intervals of Pythia-70M-Deduped and observe stronger agreement later in training, but exact validation at larger model scales remains computationally expensive. In addition, the example-level decomposition assumes SGD, whereas the models analyzed here are trained with adaptive optimizers. Accounting exactly for optimizer states and their history would require a substantially more involved decomposition.

Although the qualitative contribution dynamics are broadly consistent across the Pythia and PolyPythia configurations we examine, all experiments remain within closely related model families and pretraining data. Their generality to substantially different architectures, datasets, and training recipes therefore remains to be established. Our analyses also characterize how contribution varies across training rather than directly establishing the causal

	Interval	Domain / Text excerpt
Bottom	25k→26k	`[Science]` *...The axioms TKF10 and TKF11 allow us to infer that* $Dy(ClTm(y) \wedge Tx\psi(\dot{y}/x)y) \wedge @y(ClTm(y) \rightarrow Tx\psi(\dot{y}/x)y)$ *must hold at* w*. Let* y_0 *witness the first conjunct, and specialize the second conjunct to it. We get that at* w*, it must be that* $M(Tx\psi(\dot{y}_0/x)y \wedge Tx\psi(\dot{y}_0/x)y)[w]$*. ...*
Bottom	4k→5k	`[Computers & Electronics]` `...<a class="" href="../../../demos/logger-strategy/">Logger</a>` *\n* `</li>` *\n* `<li class="toctree-l3">` *\n* `<a class="" href="../../../demos/accesslog/">Accesslog</a>` *\n* `</li>...`
Top	5k→6k	`[Hobbies & Leisure]` *...Kaminsky: Yeah, exactly. \n\n AJK: And they kept through, right, yeah, which is – yeah, and again some of your – I just, you know, see for the first time which is kind of exciting to me, to not have the prefabricated opinions on it at a time. ...*
Top	107k→108k	`[Science]` *...Final control parameters (*σ *values) of input data sets and potential terms. \n\n In an additional reference series of RMC calculations the FNC method has been used again and in addition, partial pair correlation functions (*$g_{ij}(r)$*) from MD simulations have been ...*
Top	114k→115k	`[Computers & Electronics]` *...For example, if the patient object in [Figure 3](#sensors-20-02264-f003){ref-type="fig"} occupies* 400×400 *number of pixels in an omnidirectional view, then the patient in the foveal region of foveated view will still be rendered on* 400×400 *number of pixels ...*
Bottom	107k→108k	`[Books & Literature]` *..."I've the best news," Khiri said. \n\n Hal, who was still brooding about his seemingly unavoidable retirement, grumped something, which Carstares took as interest. \n\n "You'll be permitted to leave the hospital under my care any day now," she said. ...*

Table 2: Qualitative domain examples drawn from the interval-level top-10 and bottom-10 examples by approximate contribution. Top/Bottom indicates whether the example belongs to the top-10 or bottom-10 within the corresponding checkpoint interval. Text excerpts are lightly truncated and normalized for readability.

effect of changing the data mixture. Accordingly, the stage-dependent patterns observed here motivate data-composition and curriculum strategies, but their effects on downstream performance require direct intervention-based evaluation. Finally, disentangling domain effects from correlated properties such as text difficulty remains an important direction for future work.

B Additional Results

B.1 Qualitative Examples for Pretraining Domains

The examples in Table 2 qualitatively mirror the aggregate domain-level patterns in § 4.4. Early in training, low-contribution examples include highly technical or structurally complex texts, such as symbolic scientific reasoning and markup-heavy computer-related content, whereas the high-contribution example is conversational and closer to natural discourse. Later in training, technical scientific and computer-related passages appear among the top contributors, while a narrative passage appears among the bottom contributors. They provide concrete illustrations of the broader temporal shift observed in our domain-level analysis.

B.2 Full Results for Reference-Endpoint Sensitivity

We report the full results for the reference-sensitivity analysis summarized in § 5.2. We evaluate the checkpoint intervals 10k→11k, 40k→41k, 80k→81k, 115k→116k, and 130k→131k, using the same 1,000 examples sampled for the contribution analysis in § 4.2. For each alternative reference checkpoint, we compute Pearson and Spearman correlations with the

Reference	Spearman	Pearson	Top-5% overlap	Bottom-5% overlap
30k	−0.554	−0.559	0.064	0.148
70k	−0.174	−0.173	0.316	0.352
120k	0.566	0.573	0.680	0.656
140k	0.997	0.998	0.972	0.964

Table 3: Full reference-sensitivity results on Pythia-1.4B-Deduped. Each alternative-reference score is compared with the original score defined using the final checkpoint at step 143k. All reported correlations and overlaps are means across the five checkpoint intervals.

scores obtained using the final 143k checkpoint, as well as the overlap between their top and bottom 5% sets. Table 3 reports the mean of each metric across the five intervals.

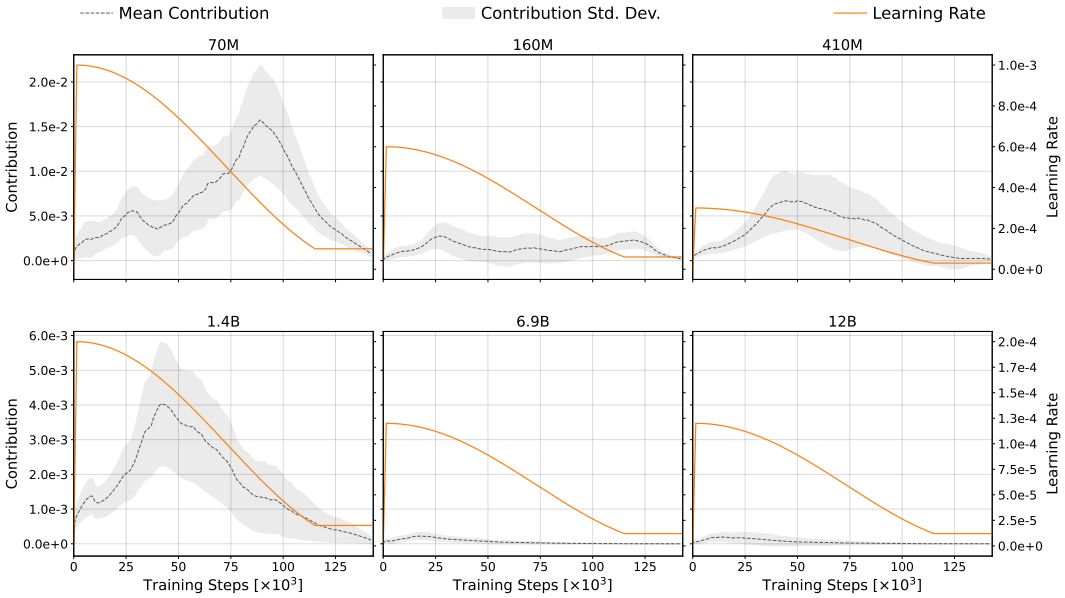
B.3 Consistency across Model Sizes

We extend the analyses in § 4.2 to 4.4 to all six Pythia-Deduped model sizes: 70M, 160M, 410M, 1.4B, 6.9B, and 12B. Across scales, the broad temporal patterns of contribution are qualitatively consistent, although their timing and magnitude vary with model size.

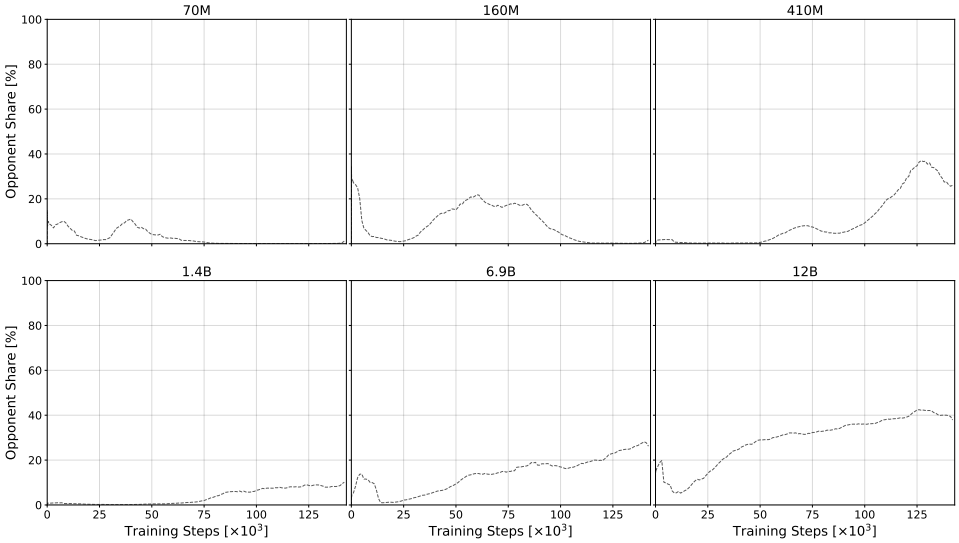
Figure 5 extends the contribution-distribution analysis in § 4.2 across model sizes. Except for the 160M model, mean contribution is highest during the middle stage of pretraining, although the location of the peak varies somewhat across model sizes. The 160M model instead exhibits a bimodal pattern, with higher contribution in both the early and late stages. We also observe that the absolute magnitude of contribution tends to decrease with model size, which may partly reflect differences in the learning-rate schedules used across scales. For models at 410M and above, the share of opponent examples consistently tends to increase toward the end of training, although its absolute level varies substantially across model sizes. Overall, the broad temporal structure of the contribution distribution is shared across most scales, while its precise shape, magnitude, and opponent share remain model-dependent.

Figure 6 extends the text-difficulty analysis in § 4.3. The 410M, 1.4B, and 6.9B models exhibit a highly similar pattern: higher-PPL examples account for a larger share of contribution during the middle stage of pretraining, while contribution is more evenly distributed across PPL bins during the early and late stages. The 12B model shows the same tendency in the highest-PPL bin during the middle stage, although the magnitude of the shift is smaller than for the smaller models, consistent with the weaker contribution dynamics observed for larger models in our analysis. By contrast, for the 70M and 160M models, the relative emphasis on higher-PPL examples emerges later, toward the end of training. One possible interpretation is that smaller models progress through analogous training phases more slowly, causing this shift toward higher-PPL data to appear later in the pretraining trajectory.

Finally, Figures 7 and 8 extend the domain analysis in § 4.4 across model sizes. For the bottom 5%, the 410M and larger models show broadly similar dynamics: STEM-related domains are prominent among low-contribution examples during the earlier stages and become less prominent later, while literature-related domains show the opposite tendency. The magnitude of these changes tends to be smaller for larger models, consistent with the weaker contribution dynamics we observe at larger scales. By contrast, for the 70M and 160M models, the share of STEM-related domains among low-contribution examples increases toward the end of training. One possible interpretation, consistent with the delayed PPL dynamics discussed above, is that smaller models progress through analogous training phases more slowly. For the larger models, the STEM share among low-contribution examples is initially small, increases during the first half of training, and then decreases later; the smaller models may primarily exhibit the earlier part of this progression within the observed training trajectory. The top-5% results are less uniform across model sizes. The 410M and 1.4B models exhibit similar domain-level shifts, but we do not observe a single directional pattern that is shared across all six models. Nevertheless, all model sizes exhibit clear phase-dependent changes in the domain composition of high-contribution examples.



(a) Contribution statistics over training.



(b) Share of opponent examples.

Figure 5: Training dynamics of the contribution distribution across the six Pythia-Deduped model sizes. Each panel corresponds to one model size. All curves are moving averages with window size 10.

Thus, while the specific domains that dominate the top contributors vary with model scale, the broader finding that domain-level contribution dynamics evolve across pretraining is preserved.

Taken together, the contribution-distribution, text-difficulty, and domain analyses reveal broadly consistent stage-dependent dynamics across model sizes, while also exhibiting systematic scale-dependent differences. In particular, several transitions that occur during the middle stage for larger models appear later for smaller models, suggesting that the timing of training-phase transitions may depend on model scale. We also observe that the magnitude of these dynamics tends to be smaller at larger scales. Thus, model size appears to affect when and how strongly contribution patterns change across pretraining, while preserving much of their broader temporal structure.

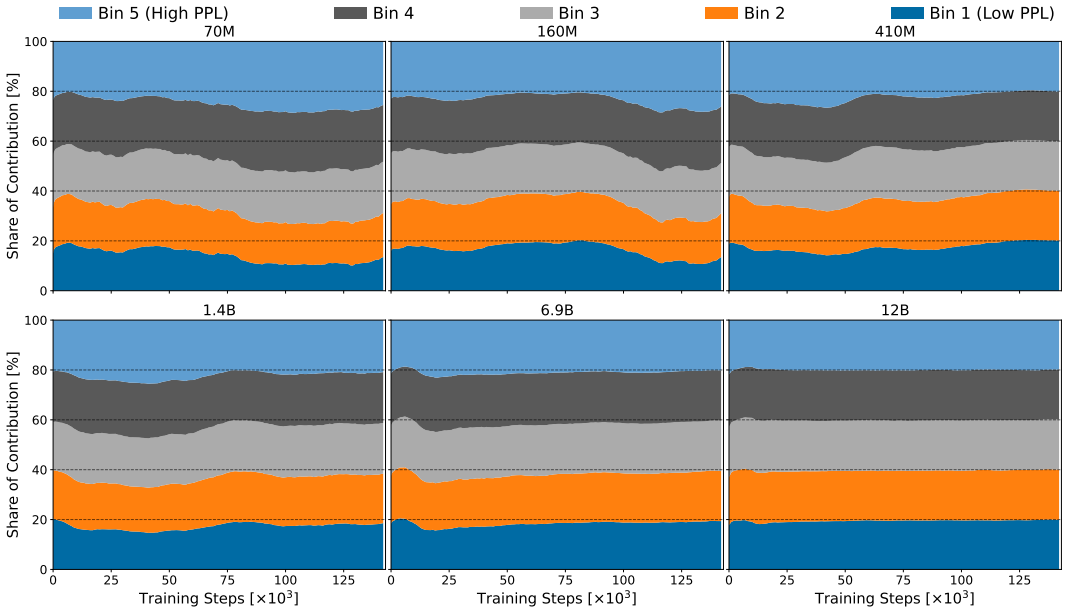


Figure 6: Relationship between text PPL and contribution across the six Pythia-Deduped model sizes. Each panel shows the share of normalized contribution attributed to five equal-sized PPL bins for one model size. All curves are moving averages with window size 10.

B.4 Consistency across Weight Initializations and Data Orderings

We examine the robustness of the domain-level contribution dynamics to weight initialization and data ordering. We first consider the standard PolyPythia runs at the 160M and 410M scales, using three available runs at each scale, corresponding to seeds 0, 1, and 3, in which both weight initialization and data ordering vary.[4] We then use the decoupled PolyPythia-160M variants to isolate each factor by varying either weight initialization or data ordering while holding the other fixed.

Figures 9 and 10 show that the standard PolyPythia runs exhibit broadly consistent domain-level dynamics across seeds. For the 160M runs, the share of STEM-related domains among low-contribution examples tends to increase as training progresses, whereas for the 410M runs, STEM-related domains are prominent early and become less prominent later. The corresponding top-5% patterns are likewise largely preserved across seeds, with similar domain-level transitions occurring over training. An important exception is the 410M seed-3 run, which exhibits qualitatively different dynamics. As reported by van der Wal et al. (2025), this run is an anomalous training run in which learning fails. Its distinct contribution dynamics therefore appear to reflect the underlying training failure rather than ordinary seed-level variation. Excluding this anomalous run, the broad temporal patterns are stable across the standard seed variants.

To disentangle the effects of the two random factors, we next examine the decoupled PolyPythia-160M variants. Figure 11 compares runs that differ only in weight initialization, while Figure 12 compares runs that differ only in data ordering.

When only weight initialization is varied, both the bottom- and top-contribution domain compositions are highly consistent across runs, with the same broad transitions occurring at similar stages of training. Varying data ordering also preserves the overall temporal structure, although the bottom-5% composition for seed 3 shows somewhat greater deviation

[4]Seed 2 is omitted because the corresponding training data indices are missing from the released data at the time of writing.

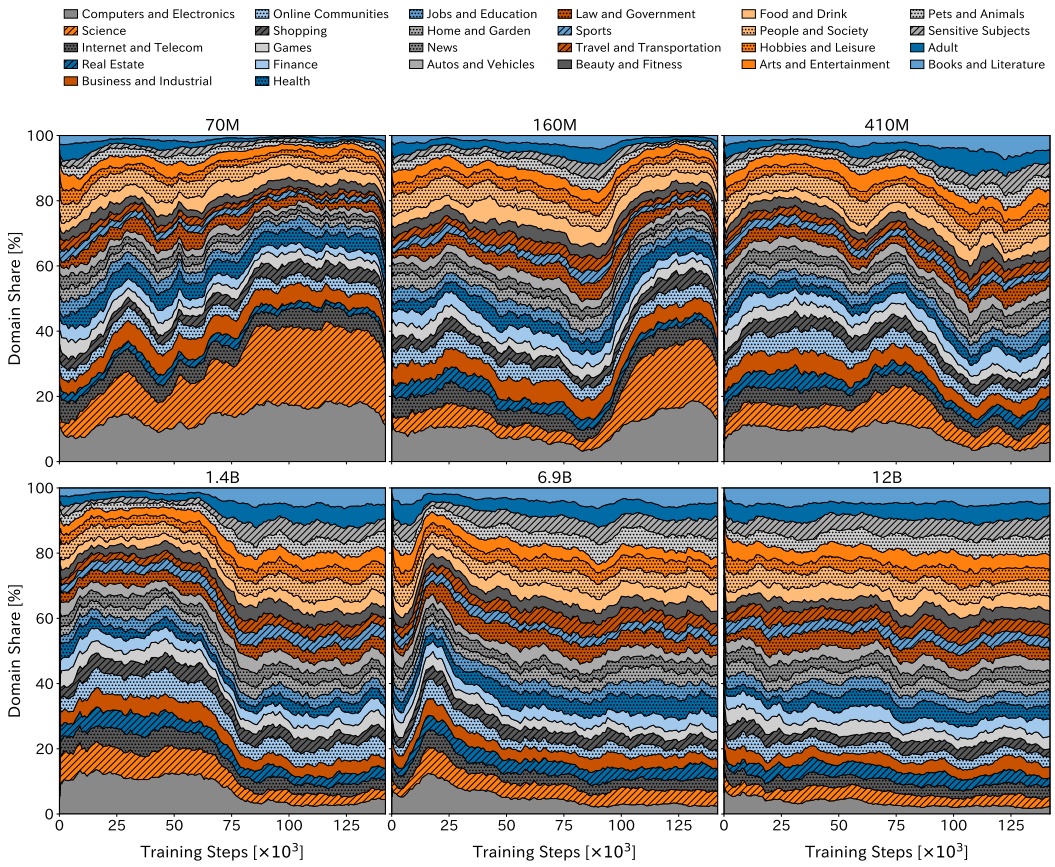


Figure 7: Domain composition of the bottom 5% of contribution examples across the six Pythia-Deduped model sizes. Each panel corresponds to one model size. All curves are moving averages with window size 10.

from the other runs. This suggests that the observed domain-level contribution dynamics may be somewhat more sensitive to data ordering than to weight initialization.

Taken together, these results indicate that the broad stage-dependent contribution dynamics are robust to ordinary seed variation, including independent changes in weight initialization and data ordering. At the same time, the anomalous 410M seed-3 run shows that substantial training failures can produce qualitatively different contribution dynamics, while the decoupled variants suggest that data ordering may introduce somewhat greater variation than weight initialization.

B.5 Details of the Task-Specific Influence Comparison

We provide the full experimental setup and results for the task-specific influence comparison summarized in § 5.4. For each validation domain d, let $\mathcal{V}_d$ denote its held-out validation set and define the corresponding average language-modeling loss at checkpoint c as

$$L_d(\theta_c) = \frac{1}{|\mathcal{V}_d|} \sum_{v \in \mathcal{V}_d} \ell(v; \theta_c). \tag{5}$$

For a training example x processed within the checkpoint interval beginning at c, we compute the checkpoint-local TracIn-style score

$$\mathrm{TracIn}_d(x; c) = \nabla_\theta \ell(x; \theta_c)^\top \nabla_\theta L_d(\theta_c). \tag{6}$$

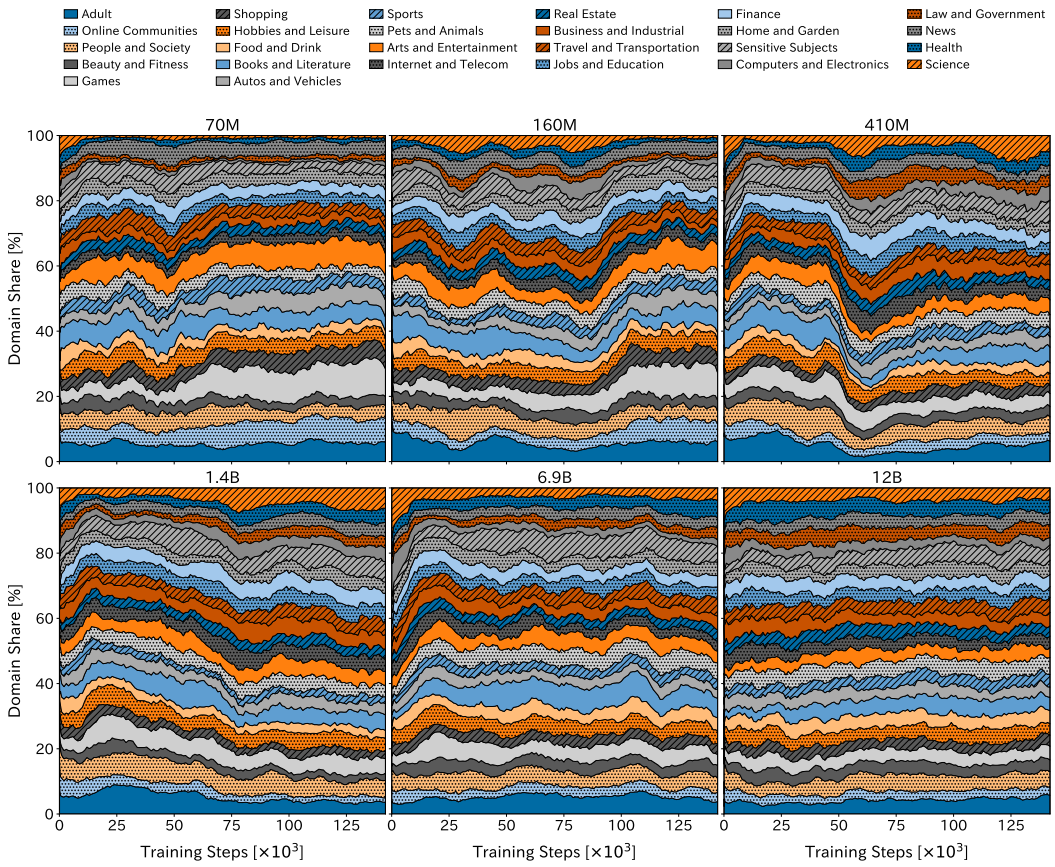


Figure 8: Domain composition of the top 5% of contribution examples across the six Pythia-Deduped model sizes. Each panel corresponds to one model size. All curves are moving averages with window size 10.

Up to the positive learning-rate factor shared within an interval, this gradient alignment approximates the first-order reduction in validation loss induced by an SGD update on x. Because our analysis uses only within-interval rankings, we omit this common factor.

We construct the held-out validation sets from The Pile after removing examples that occur in the released Pythia pretraining stream. We classify the remaining examples using the same NeMo Curator Domain Classifier as in § 4.4. For each of four domains, *Books and Literature*, *Science*, *Computers and Electronics*, and *Hobbies and Leisure*, we uniformly sample 1,000 examples assigned to that domain with classifier confidence greater than 0.95. These four domains include both STEM and non-STEM validation targets and correspond to domains that exhibit prominent temporal patterns in our contribution analysis.

For each checkpoint interval of Pythia-1.4B-Deduped, we use the same domain-balanced pool of training examples as in § 4.4. Specifically, the pool contains 100 examples from each of the 26 domains. We compute Equation (6) for every training example and each of the four validation domains, and then examine the domain composition of the top and bottom 5% of examples under each score. The curves in Figure 13 are moving averages with a window size of 10 checkpoint intervals.

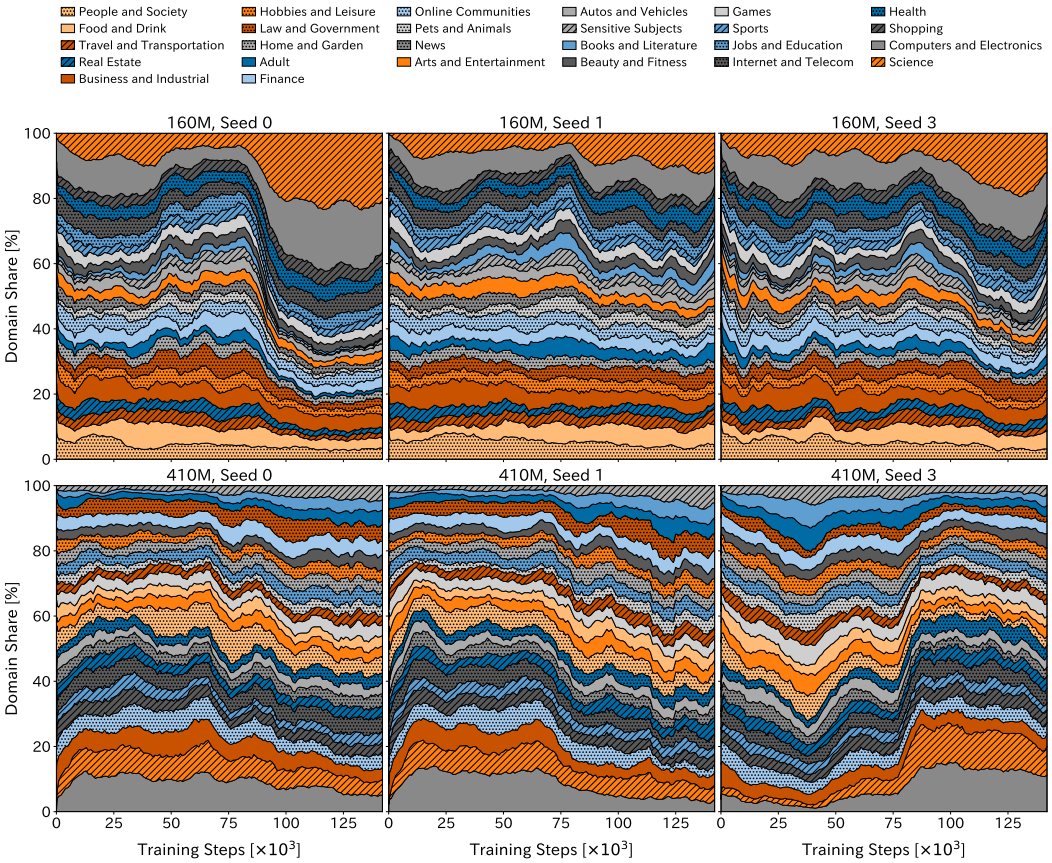


Figure 9: Domain composition of the bottom 5% of contribution examples across standard PolyPythia-160M and PolyPythia-410M runs. Each model size includes three runs with different weight initializations and data orderings. All curves are moving averages with window size 10.

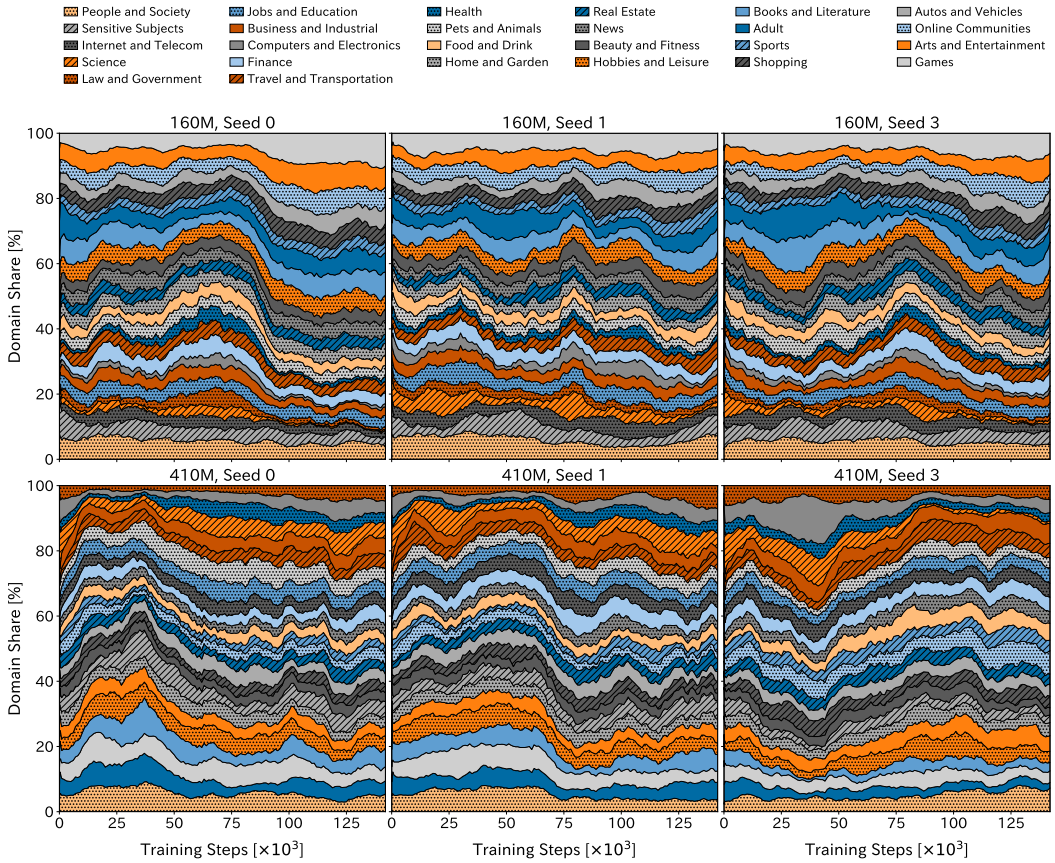
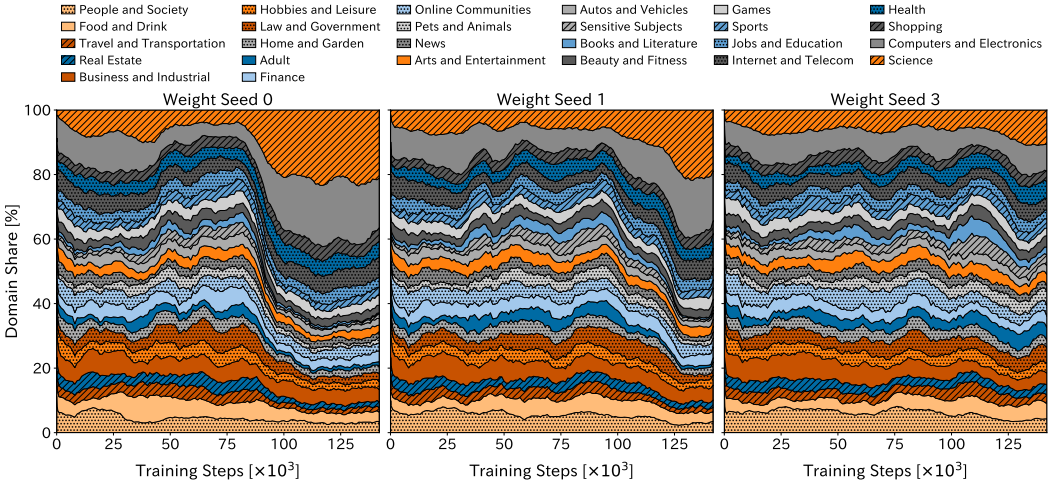
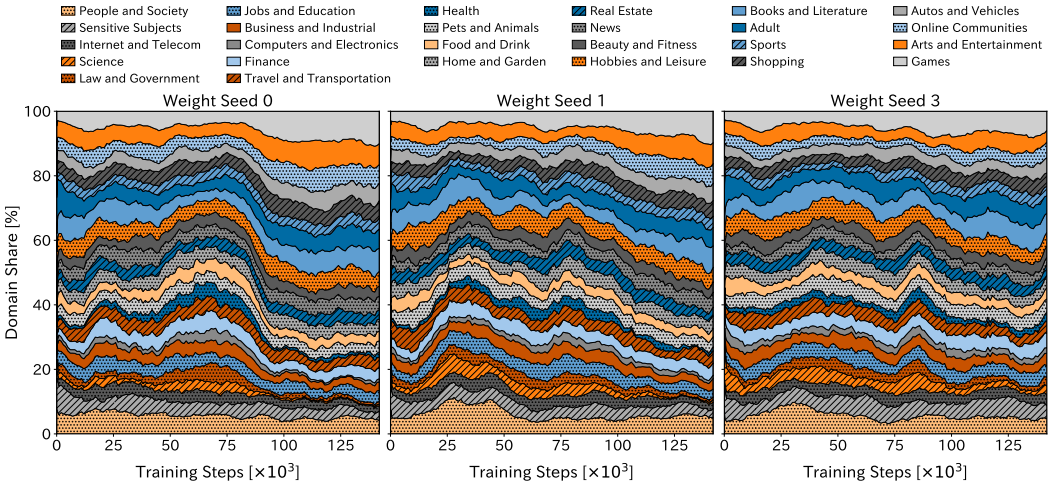


Figure 10: Domain composition of the top 5% of contribution examples across standard PolyPythia-160M and PolyPythia-410M runs. Each model size includes three runs with different weight initializations and data orderings. All curves are moving averages with window size 10.

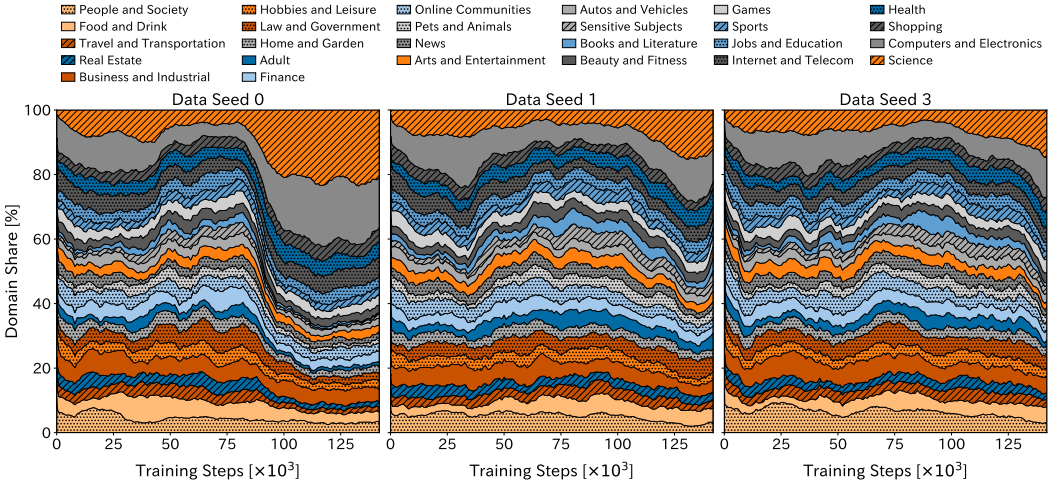


(a) Bottom 5% contribution examples, and their domain share in the overall data mix.

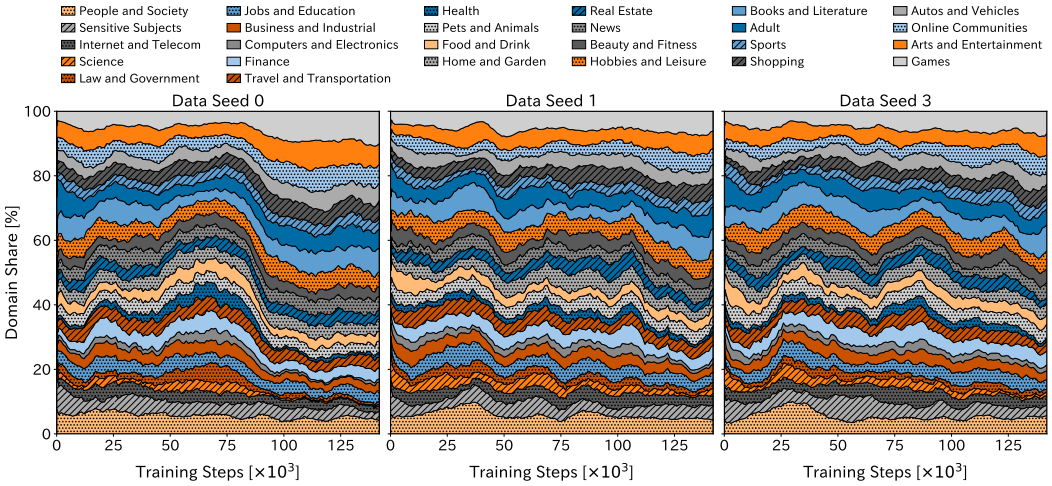


(b) Top 5% contribution examples, and their domain share in the overall data mix.

Figure 11: Domain composition of extreme-contribution examples for PolyPythia-160M runs with different weight initializations. The qualitative contribution dynamics are highly consistent across runs when data ordering is fixed.

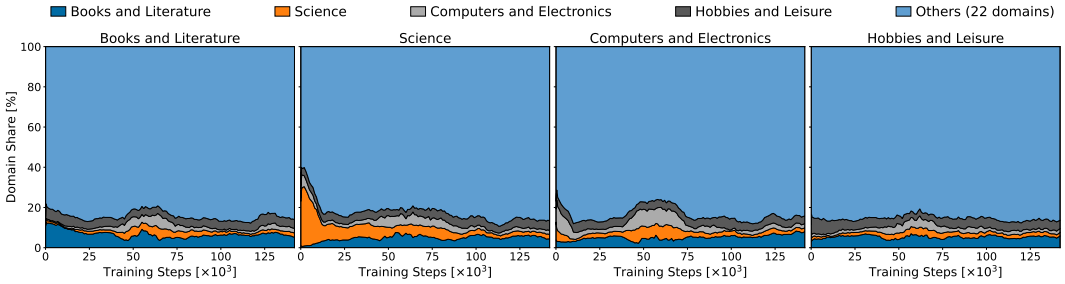


(a) Bottom 5% contribution examples, and their domain share in the overall data mix.

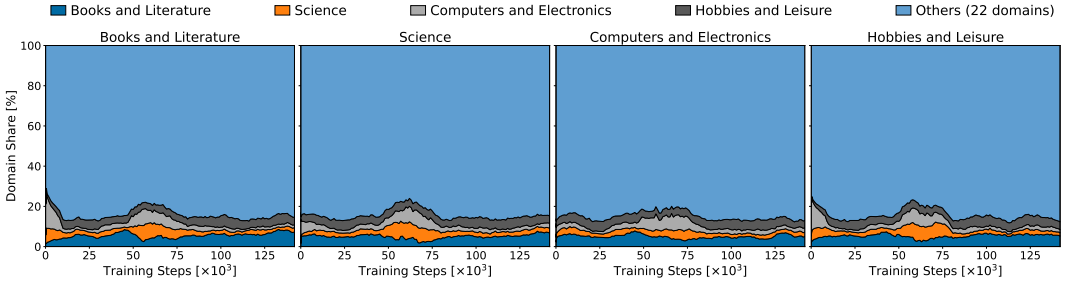


(b) Top 5% contribution examples, and their domain share in the overall data mix.

Figure 12: Domain composition of extreme-contribution examples for PolyPythia-160M runs with different data orderings. The broad temporal patterns remain similar across runs, although some variation is visible in the precise domain dynamics.



(a) Domain composition of the top 5% of training examples under the TracIn-style score.



(b) Domain composition of the bottom 5% of training examples under the TracIn-style score.

Figure 13: Domain composition of training examples with extreme TracIn-style scores. Each panel uses the language-modeling loss on a different domain-specific held-out set as the validation target. Panel headings denote the validation domains used to compute the scores, while colors denote the source domains of the selected training examples.