

PrAM: Offline Checkpoint Selection for Post-Hoc Training Data Attribution

Stephanie Schoch¹, Sidhya Pathak², Yangfeng Ji²

¹Department of Computer Science, William & Mary

²Department of Computer Science, University of Virginia
sschoch@wm.edu, {gyz8db, yangfeng}@virginia.edu

Abstract

Training data attribution scores how training examples affect downstream model performance. For models released with many intermediate checkpoints, attribution can be computed along the saved training trajectory, but evaluating every checkpoint is often prohibitively expensive. We introduce PrAM (**Probe-based Attribution Mass**), a post-hoc offline checkpoint selector for training data attribution. PrAM runs a small attribution probe at every checkpoint, selects checkpoints with high absolute attribution mass, and runs full attribution only on that subset. Across Pythia-14M, 70M, and 160M, PrAM closely matches a full attribution-pool reference and outperforms heuristic rules based on recency, spacing, learning rate, and validation loss. Beyond rank correlation, PrAM preserves the domain-by-target attribution tables and ranked training examples produced by full saved-checkpoint attribution, enabling scalable post-hoc analyses of checkpointed model releases. We support this with application-based analyses scaling to 410M and 1B parameter models.

1 Introduction

Large language models (LLMs) are increasingly being analyzed through the lens of the data they are training on. To this end, training data attribution analyzes how different training examples (or example subsets) contribute to downstream model performance. Specifically, attribution methods assign scores to training examples with respect to a given validation target, making it possible to ask which examples support, hurt, or explain a model behavior (Koh and Liang, 2017; Pruthi et al., 2020; Ghorbani and Zou, 2019; Ilyas et al., 2022; Park et al., 2023; Grosse et al., 2023). Applying data attribution to pretraining data is particularly important, as pretraining behavior depends on data mixture and domain coverage (Xie et al., 2023; Longpre et al., 2024; Liu et al., 2025; Magnusson

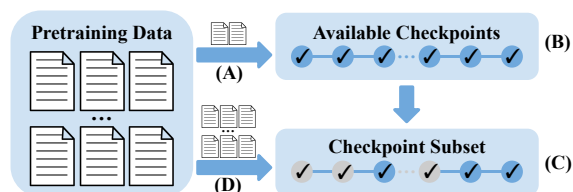


Figure 1: An overview of PrAM: (A) sample a small probe set; (B) compute attribution across all checkpoints using small probe set; (C) select subset of checkpoints with highest probe-based attribution mass; (D) compute full attribution pass on selected checkpoint subset.

et al., 2025), topic organization and data mixing (Peng et al., 2025; Wettig et al., 2025), and filtering, licensing, and opt-out decisions (Fan et al., 2025a; Kandpal et al., 2025). Although attribution methods that scale to pretraining data have been developed (Wang et al., 2025a), they are typically deployed within the context of a specific training run.

However, many attribution questions arise only *after* a model has been trained. For example, a benchmark or validation target may be chosen after training, or poor downstream performance may motivate inspection of negatively attributed training examples, as in data-cleaning and influence-based analyses (Hara et al., 2019; Pruthi et al., 2020). If a training run saves intermediate checkpoints, such as with the releases of the Pythia (Biderman et al., 2023) and OLMo/OLMo 2 (Groeneveld et al., 2024; Walsh et al., 2025) model families, these questions can be studied along the saved training trajectory rather than only at the final model.

Trajectory-aware attribution answers questions that a final model checkpoint alone cannot. For example, recent work demonstrates how data influence changes across training stages (Wang et al., 2025b; Lee et al., 2026; Eichin et al., 2026) and how different checkpoints can contain different information (Pruthi et al., 2020). However, running an attribution pass at every saved checkpoint is pro-

hibitively expensive (Wang et al., 2025a; Pruthi et al., 2020). For example, the Pythia family of models releases 154 checkpoints per model, and the OLMo 2 family releases thousands of intermediary checkpoints (Biderman et al., 2023; Walsh et al., 2025). While existing work studies the online setting for checkpoint selection (i.e. which checkpoints to keep *during* training) (Das et al., 2025), we study the post-hoc version of this problem. In this setting, which we define as *offline checkpoint selection*, the training run is complete, the checkpoints are fixed, and the validation target may be selected later.

Existing methods for offline checkpoint selection rely on simple heuristics, such as recency, spacing, learning rate, validation loss, and validation-loss reduction (Pruthi et al., 2020). Further, analyses of the efficacy of checkpoint subsets for answering attribution-based questions are limited (Pruthi et al., 2020). Our work directly addresses these gaps.

In this work, we introduce PrAM (**P**robe-based **A**tribution **M**ass), a principled offline checkpoint selection method for post-hoc training data attribution. PrAM runs a small attribution probe across all checkpoints, selects checkpoints with high absolute attribution mass, and runs full attribution only on the selected subset. We demonstrate that PrAM closely matches the full saved-checkpoint reference, whereas the heuristic checkpoint-selection baselines do not. Further, we demonstrate that across different small probe sizes, PrAM exhibits high correlation with PrAM-Full, a full attribution-pool reference for the same selection rule.

Additionally, we evaluate PrAM on downstream attribution applications, addressing the question of whether the selected checkpoints preserve the outputs that make attribution useful. We demonstrate that PrAM preserves attribution insights at both the domain and example level. Specifically, we show that PrAM identifies the same domain-to-target rankings, as well as the highest and lowest contributing training examples for follow-up inspection.

Our contributions are as follows: 1) We define offline checkpoint selection for training data attribution: the model has already been trained, checkpoints are fixed, and the target can be chosen after release; 2) we introduce PrAM, a probe-based selector that chooses checkpoints with high absolute attribution mass, and PrAM-Full, a full attribution-pool reference for the same selection rule; 3) we show that PrAM closely matches full

saved-checkpoint attribution and PrAM-Full across Pythia-14M/70M/160M and outperforms existing checkpoint selection rules; 4) we show that attribution mass is concentrated in a small part of the trajectory, and the high-mass region can move across runs; 5) we use the selected checkpoints to preserve actionable attribution insights at both the domain and example levels.

2 Related Work

Checkpointed training data attribution. Training data attribution methods estimate how training examples or subsets affect model behavior by scoring examples with respect to a target. Training data attribution methods include influence functions (Koh and Liang, 2017), TracIn (Pruthi et al., 2020), Data Shapley (Ghorbani and Zou, 2019), datamodels (Ilyas et al., 2022), and TRAK (Park et al., 2023). Recent work scales attribution to large language models through methods such as scaling influence functions (Grosse et al., 2023), DataInf (Kwon et al., 2024), DDA (Wu et al., 2024), scalable influence and fact-tracing (Chang et al., 2025), and LoRIF (Li et al., 2026). For checkpointed training runs, these scores can be evaluated along the saved trajectory rather than only at the final model. Importantly, influence can change across training stages (Wang et al., 2025b; Lee et al., 2026; Eichin et al., 2026), which the final-checkpoint alone cannot capture. This has further motivated the development of trajectory-aware methods such as TracIn/TracIn-CP (Pruthi et al., 2020), SOURCE (Bae et al., 2024) and In-Run Data Shapley (Wang et al., 2025a).

Although trajectory-aware methods motivate using more than the final checkpoint in checkpointed attribution, full checkpointed attribution can be expensive when many checkpoints are available. Existing post-hoc choices are mostly coarse heuristics: use the final checkpoint, recent checkpoints, evenly spaced checkpoints, or validation-loss proxies (Pruthi et al., 2020). CheckSelect studies a related online setting, where checkpoints are selected during training for data valuation (Das et al., 2025). Our setting is different: the run has already finished, the checkpoints are fixed, and the target can be chosen after release. PrAM addresses this offline setting by using a small attribution probe to select the saved checkpoints at which to run the full attribution pass. PrAM-Full provides the corresponding full attribution-pool reference for this

selection rule.

Pretraining data analysis and mixture design.

A parallel literature studies how to choose, weight, or analyze pretraining data. DoReMi (Xie et al., 2023), RegMix (Liu et al., 2025), Data Mixing Laws (Ye et al., 2025), BiMix (Ge et al., 2025), GRAPE (Fan et al., 2025b), and DataDecide (Magnusson et al., 2025) use proxy runs, regression, scaling laws, or robust optimization to inform pretraining-data mixtures. MATES (Yu et al., 2024) and Group-MATES (Yu et al., 2025) use local, model-aware influence estimates during pretraining to guide subsequent data selection. Complementary work studies other aspects of pretraining-data curation. WebOrganizer (Wettig et al., 2025) studies topic- and format-based domain organization, while Topic Over Source (Peng et al., 2025) studies topic-based data mixing. DataComp-LM (Li et al., 2024) and FineWeb (Penedo et al., 2024) study large-scale filtering and dataset construction. Fan et al. (2025a) examine the effect of web-crawling opt-outs, while The Common Pile (Kandpal et al., 2025) focuses on public-domain and openly licensed pretraining data. While PrAM does not choose a new mixture to train or predict the effect of changing the pretraining distribution, it can support such use cases through post-hoc attribution analysis of domains and training examples for a target chosen later.

3 Background: Checkpointed Attribution

Setup and goal. Let $D_{\text{tr}} = \{z_i\}_{i=1}^N$ be the training examples that we want to score. In our experiments, each z_i is a 1024-token pretraining sequence, but the setup applies to any attribution unit used by the underlying attribution method. Let $D_{\text{val}} \equiv D^{(\text{val})} = \{q_j\}_{j=1}^G$ be the held-out target set. Each target q_j can be a single validation example or a validation slice (e.g. all validation examples belonging to a given domain).

At checkpoint θ_t , the underlying attribution method assigns a scalar attribution score $\tau_t(z_i; q_j)$ to training example z_i with respect to target q_j . Higher scores indicate greater support for the target under the attribution estimator, while negative scores indicate interference. While this sign matters for interpreting ranked training examples, our checkpoint-selection rule uses absolute attribution mass and does not depend on this sign convention.

Full saved-checkpoint estimator. Assume the completed run released checkpoints $\Theta = \{\theta_0, \dots, \theta_T\}$. Let $\alpha_t = \text{lr}_t / \sum_{s=0}^T \text{lr}_s$ be the normalized checkpoint weight, where lr_t is the learning rate at checkpoint θ_t . For each target q_j , the full saved-checkpoint reference score for training example z_i is

$$\tau_{\text{ref}}(z_i; q_j) = \sum_{t=0}^T \alpha_t \tau_t(z_i; q_j). \quad (1)$$

Equivalently, for a fixed target q_j , the reference vector over training examples as

$$\boldsymbol{\tau}_{\text{ref}}^{(q_j)} := (\tau_{\text{ref}}(z_i; q_j))_{i=1}^N. \quad (2)$$

In our experiments, τ_t is a checkpointed gradient-dot-product score in the style of TracIn-CP (Pruthi et al., 2020), where

$$\tau_t(z_i; q_j) = \langle \nabla_{\theta} \ell(\theta_t; z_i), \nabla_{\theta} \ell(\theta_t; q_j) \rangle. \quad (3)$$

The full saved-checkpoint estimator in Eq. 1 serves as a reference attribution estimate for a completed run across all checkpoints.

Sparse checkpointed estimator. A checkpoint selector chooses a subset $S \subseteq \{0, \dots, T\}$ with $|S| = K$. The selected-checkpoint estimate keeps only those checkpoint terms:

$$\begin{aligned} \tau_S(z_i; q_j) &= \sum_{t \in S} \alpha_t \tau_t(z_i; q_j), \\ \boldsymbol{\tau}_S^{(q_j)} &:= (\tau_S(z_i; q_j))_{i=1}^N. \end{aligned} \quad (4)$$

Renormalizing by $\sum_{t \in S} \alpha_t$ does not change rankings within a target, so our rank-based metrics are unaffected by this choice. For each target q_j , we compare the sparse vector $\boldsymbol{\tau}_S^{(q_j)}$ with the full saved-checkpoint reference vector $\boldsymbol{\tau}_{\text{ref}}^{(q_j)}$.

Importantly, the choice of S can change which domains or training examples the attribution analysis surfaces. Section 6 evaluates whether $\boldsymbol{\tau}_S^{(q_j)}$ preserves $\boldsymbol{\tau}_{\text{ref}}^{(q_j)}$ across different selection methods. Section 7 then analyzes the domain-level and example-level attribution rankings from an application standpoint.

4 PrAM: Selecting Checkpoints by Attribution Mass

Main idea. Intuitively, some checkpoints carry more checkpointed attribution signal than others.

PrAM builds on this intuition by estimating where that signal is using a small probe. Using this small probe, we can select a checkpoint subset S , up to budget K , and compute full attribution as in Eq. 4.

Let $P_{\text{small}} \subset D_{\text{tr}}$ be a probe set of training examples used only to select checkpoints. Further, let $D_{\text{val}}^{\text{sel}} \subseteq D_{\text{val}}$ be the target subset used in conjunction with P_{small} . When the selector uses all held-out targets, we have $D_{\text{val}}^{\text{sel}} = D_{\text{val}}$.

For every checkpoint, PrAM computes the absolute attribution mass over the probe examples and selected targets,

$$m_t(P_{\text{small}}, D_{\text{val}}^{\text{sel}}) = \sum_{q_j \in D_{\text{val}}^{\text{sel}}} \sum_{z_i \in P_{\text{small}}} |\alpha_t \tau_t(z_i; q_j)|, \quad (5)$$

and selects

$$S_K(P_{\text{small}}, D_{\text{val}}^{\text{sel}}) = \underset{t \in \{0, \dots, T\}}{\text{TopK}} \quad m_t(P_{\text{small}}, D_{\text{val}}^{\text{sel}}). \quad (6)$$

We then have $\text{PrAM} = S_K(P_{\text{small}}, D_{\text{val}}^{\text{sel}})$ as our checkpoint selector. In summary, the probe set is used to select checkpoints. Then, after checkpoint selection, a final attribution pass computes $\tau_S(z_i; q_j)$ over all training examples in the attribution pool (i.e. the full attribution pool) for each target of interest across the K selected checkpoints.

In our implementation, Eq. 5 is computed by summing over probe examples and selected targets before any signed target aggregation. If an implementation stores unweighted checkpoint scores, the score should be multiplied by α_t . Further, if the underlying attribution method already includes checkpoint weighting, that scaling can be absorbed into τ_t . Algorithm 1 in Appendix E summarizes the implementation.

Full-probe reference. For evaluation purposes, we also define

$$\text{PrAM-Full} = S_K(D_{\text{tr}}, D_{\text{val}}^{\text{sel}}). \quad (7)$$

PrAM-Full uses the same mass rule as PrAM, but replaces the small probe with the full attribution pool. We define PrAM-Full to address the question of whether a small probe can find the same high-mass checkpoints that full-pool access would find. Unless otherwise stated, both PrAM and PrAM-Full select the $K - 1$ largest-mass checkpoints plus the final checkpoint anchor. We include the anchor as we are ultimately interested in attribution for the final model. In Appendix E.1, we compare anchored and unanchored variants and find that they converge by $K = 8$.

Why use absolute mass? For a fixed target q_j , the omitted part of the saved-checkpoint reference is

$$\tau_{\text{ref}}^{(q_j)} - \tau_S^{(q_j)} = \left(\sum_{t \notin S} \alpha_t \tau_t(z_i; q_j) \right)_{z_i \in D_{\text{tr}}}. \quad (8)$$

Therefore,

$$\left\| \tau_{\text{ref}}^{(q_j)} - \tau_S^{(q_j)} \right\|_1 \leq \sum_{t \notin S} \sum_{z_i \in D_{\text{tr}}} |\alpha_t \tau_t(z_i; q_j)|. \quad (9)$$

Summing this inequality over $q_j \in D_{\text{val}}^{\text{sel}}$ shows that, when attribution mass is concentrated, selecting high-mass checkpoints reduces a conservative upper bound on omitted attribution signal across the selected targets. We note, however, that while this motivates the score, it does not prove rank fidelity. As rankings can depend on signs, cancellations across checkpoints, and the ordering of individual examples, we evaluate rankings directly in our experiments.

Cost. PrAM replaces a full attribution pass at every saved checkpoint with a small attribution probe at every checkpoint and a full attribution pass only at the selected checkpoints. The realized cost depends on probe size, target set, full attribution pool, checkpoint-loading overhead, batching, and hardware. In Appendix C, we provide a per-checkpoint accounting model, measured timing calibration, and cost-matched comparison.

5 Experimental Setup

Models and datasets. For our main experiments, we use Pythia-14M, Pythia-70M, and Pythia-160M Standard, each with $T + 1 = 154$ released checkpoints (Biderman et al., 2023). As the Pythia family of models was pretrained on the Pile (Gao et al., 2020), we construct attribution pools for the Pythia models from a pretokenized subset of Pile-uncopyrighted (Monology, 2023). Each training example is a 1024-token pretraining sequence. For Pythia-14M and 70M, the full attribution pool (D_{tr}) contains 17K training examples, with 1K examples sampled from each of the 17 domains. For Pythia-160M, we use a stratified 2K-example attribution pool to fit our compute budget. Notably, model and dataset sizes are constrained to make the full-checkpoint references computationally feasible. Unless otherwise stated, PrAM uses a 256-example probe, $|P_{\text{small}}| = 256$.

Pile metadata lets us group attribution scores by domain, including code, mathematics, legal text, biomedical text, books, web text, and other component domains. In addition to the 17 Pile domains, for Pythia-14M and Pythia-70M, we include three external downstream targets (“downstream probes”) not drawn from the 17 Pile-domain target set or attribution pool: MMLU-math (Hendrycks et al., 2021), HumanEval-likelihood (Chen et al., 2021), and BBH-arithmetic (Suzgun et al., 2023). The 160M reduced-pool evaluation averages over the 17 Pile domain targets and excludes the three downstream probes.

All checkpoint selectors are evaluated with the same underlying attribution method. As stated in Section 3 (Eq. 3), we use gradient dot-product attribution in the style of TracIn-CP (Pruthi et al., 2020). Specifically, our implementation utilizes the ghost dot-product technique from In-Run Data Shapley (Wang et al., 2025a). Thus, the checkpoint selector changes, but the underlying attribution method is fixed.

Baselines. We compare PrAM with practical checkpoint-selection rules that require either no attribution pass or only an inexpensive validation-loss pass. The baselines are: **late- K** , which selects the K most recent checkpoints; **final-only**, which uses only the final checkpoint; **linear/log spacing**, which spreads checkpoints evenly in linear or log index; **top- K learning rate**, which selects checkpoints with the largest learning-rate weights; **top- K validation loss**, which selects checkpoints with the largest validation loss on the target pool; **top- K validation-loss reduction**, which selects checkpoints with the largest adjacent validation-loss drop; and **random selection**, averaged over seeds. As offline checkpoint selection is an under-explored problem, our baselines are simple heuristics, with the validation-loss-reduction rule following TracIn-CP (Pruthi et al., 2020).

Metrics. Our primary metric is mean per-target Spearman correlation between the sparse selected-checkpoint vector $\tau_S^{(q_j)}$ and the full saved-checkpoint reference vector $\tau_{\text{ref}}^{(q_j)}$, computed across training examples. Higher values indicate better preservation of the full reference ranking.

6 Recovering Checkpointed Attribution with Sparse Checkpoints

This section addresses three questions: 1) how closely do attribution estimates computed across sparse checkpoints (selected by PrAM) align with attribution estimates computed across the entire checkpointed training trajectory; 2) how concentrated or diffuse is attribution mass across checkpoints; 3) how sensitive is PrAM to probe size and validation target coverage?

6.1 Do sparse checkpoints match full saved-checkpoint attribution?

First, we conduct a fidelity analysis of PrAM. Our comparisons are: 1) attribution estimates using baseline checkpoint selection methods; 2) PrAM-Full (i.e. checkpoint selection with the full-attribution pool); and 3) the full saved-checkpoint reference (i.e. full-trajectory attribution, computed with all 154 checkpoints).

We visualize the mean per-target Spearman correlation with full saved-checkpoint attribution for both PrAM and PrAM-Full, as well as for the baseline checkpoint selection methods, in Figure 2 (also reported numerically in Appendix A). PrAM not only preserves the full saved-checkpoint attribution rankings better than the baseline checkpoint-selection rules, but stays within 0.015 mean Spearman of PrAM-Full in every tested scale and checkpoint budget (as a result, there is very high visual overlap in the plots). Together, these results demonstrate that a subset of checkpoints can preserve the rankings of the full checkpoint set, and an effective subset can be identified even with a small attribution probe. Notably, these results also demonstrate that heuristic rules, such as spacing-based heuristics, often significantly underperform in comparison to selection based on attribution mass. We analyze this more closely in the following subsection.

6.2 Why PrAM works: attribution mass is concentrated

To better understand the performance of PrAM, we visualize the normalized attribution mass of each checkpoint, reported in Figure 3. We observe that checkpoints with high attribution mass (i.e. with high attribution signal) do not appear at fixed intervals in the saved trajectory, nor are they consistent across models. For instance, in Pythia-70M and Pythia-160M-2K, the mass is mostly late in train-

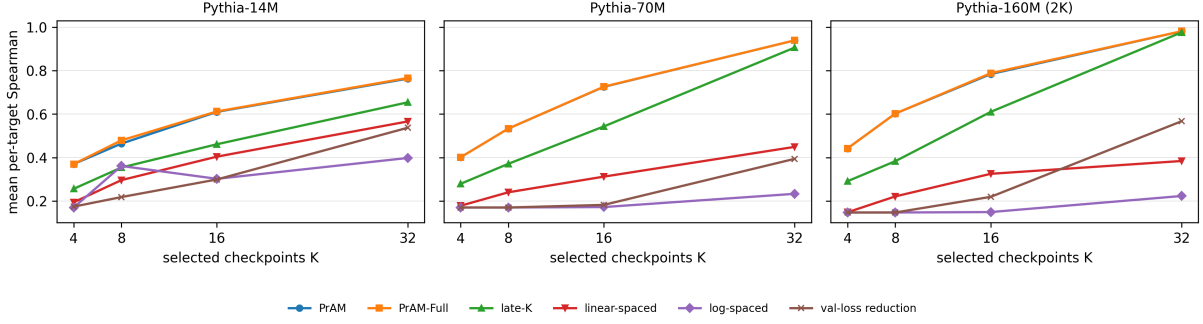


Figure 2: Mean per-target Spearman correlation (higher is better) with the full trajectory as the checkpoint budget K changes. PrAM nearly overlaps the full-probe PrAM-Full reference and outperforms baselines across Pythia-14M, 70M, and 160M-2K. Exact values are also reported in Appendix A.

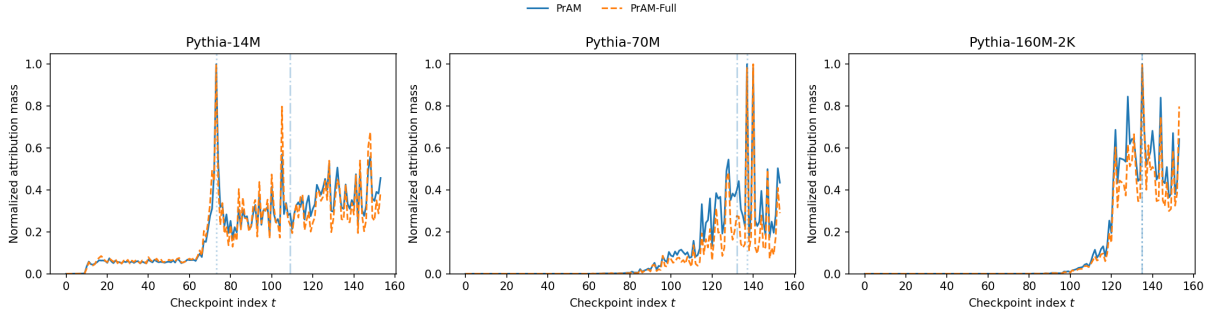


Figure 3: Normalized attribution mass across Pythia checkpoints, showing where attribution mass appears along the training trajectory. The solid blue line indicates the probe-based attribution mass used by PrAM, $m_t(P_{\text{small}}, D_{\text{val}}^{\text{sel}})$, and the dashed orange line indicates the full-pool attribution mass used by PrAM-Full. Vertical dotted lines mark the largest mass checkpoint and the half-mass step.

ing. This is reflected in the performance of the late-K baseline in Figure 2. In comparison, on Pythia-14M there is a high attribution mass region along the middle of the trajectory as well, reflected in lower performance of late-K. Moreover, these results show why baseline methods based on heuristics such as spacing or recency are insufficient for checkpoint selection. The checkpoints with large attribution mass are not chosen by index alone; they depend on the run and target.

6.3 Probe size and validation coverage

While the previous subsections showed that PrAM (with a probe size of 256 examples) closely aligns with PrAM-Full and identifies similar high-mass checkpoints, in this subsection we ask the question of how much probe data is needed. To address this, we perform an ablation across probe size and validation coverage as two dimensions. We vary the number of training examples in the probe set, $|P_{\text{small}}| \in \{64, 128, 256, 512\}$, and we vary the validation pool per domain in $D_{\text{val}}^{\text{sel}}$ from $\{8, 16, 32, 64\}$ and report the results of this sweep in Figure 4. Even the smallest tested probes pro-

duced useful checkpoint subsets, with minimal variation in correlation across either dimension. For example, on Pythia-70M, the lowest-validation cells are about 0.05–0.06 mean Spearman below the PrAM-Full reference. Appendix C reports the corresponding timing calibration and cost-matched comparisons.

Additional analyses. Appendix D provides a small OLMo-2-1B cross-architecture check. We also analyze how strongly PrAM depends on the targets used by the probe. In the *leave-one-domain-out* (LODO) setting, the selector is built from 16 Pile domain targets and evaluated on the held-out domain target. In every reported cell, PrAM is the best performing method, suggesting that the selected checkpoints transfer better to held-out domain targets than recency or spacing rules. The full table is in Appendix A.

7 Using Selected Checkpoints for Training Data Analysis

After PrAM selects checkpoints, the underlying attribution method returns one sparse score vector

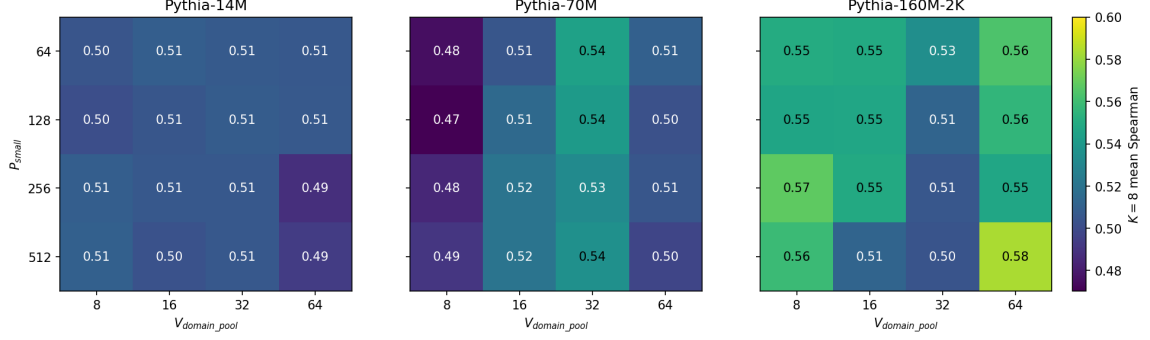


Figure 4: Probe-size sweep at $K = 8$. Each cell reports mean per-target Spearman after selecting checkpoints with $|P_{\text{small}}|$ training examples and $V_{\text{domain_pool}}$ validation examples per target domain. Small probes and validation pools across the range we tested recover useful checkpoint subsets, indicating that PrAM is robust to probe size and validation coverage as hyperparameters.

$\tau_S^{(q_j)}$ over training examples for each target q_j . We use these per-target vectors in two ways. First, we aggregate example-level scores by Pile domain to build domain-by-target attribution tables. Second, we rank individual training examples for each target, including both top-ranked examples and examples with low attribution scores. These uses follow the standard role of attribution scores in identifying influential training examples for inspection (Park et al., 2023; Grosse et al., 2023; Wang et al., 2025a; Chang et al., 2025). We are interested in the performance of PrAM in terms of rankings preservation in comparison to the full reference.

For Pythia-14M, Pythia-70M, and Pythia-160M-2K, the comparison reference is the full saved-checkpoint reference $\tau_{\text{ref}}^{(q_j)}$. We additionally define a union-restricted reference for use with Pythia-410M and Pythia-1B in Appendix B. The larger-scale results test the same application outputs beyond the scales where we can afford the full reference.

7.1 Domain-by-target attribution

To summarize attribution at the domain level, let $\mathcal{I}_g$ be the set of training examples from Pile domain g . We define the domain-by-target table induced by selector S as

$$M_S[g, j] = \frac{1}{|\mathcal{I}_g|} \sum_{i \in \mathcal{I}_g} \tau_S(z_i; q_j). \quad (10)$$

We aim to answer the question of which training domains contribute more to different validation domains. We compare M_S with the reference table M_{ref} , formed by replacing τ_S with τ_{ref} in Eq. 10. Figure 5A reports row-wise Spearman agreement.

For each domain g , we compute the Spearman correlation between $M_S[g, :]$ and $M_{\text{ref}}[g, :]$, then average across domains, where higher correlation is better. Similar to the previous experiments, we find that PrAM recovers the domain-by-target rankings better than the baseline checkpoint selection rules. From these results, the takeaway is that the selected checkpoint subset can change which training domains an attribution analysis points to as being helpful or harmful for a particular validation domain.

7.2 Ranked training example recovery

Training data attribution can rank examples for inspection or follow-up analysis (Park et al., 2023; Grosse et al., 2023; Chang et al., 2025). Related model-aware data-selection methods use example-level signals to select or filter training data (Ilyas et al., 2022; Engstrom et al., 2024). While the previous subsection demonstrated that PrAM preserves domain-level insights, in this subsection we ask whether PrAM preserves example-level insights. For each downstream probe q_j and selector at $K = 8$, we rank examples by $\tau_S^{(q_j)}$, equivalently by the scalar scores $\tau_S(z_i; q_j)$.

We report Jaccard@50 in Figure 5B between each method’s top-50 examples and the reference top-50 examples. In each case, PrAM matches or exceeds the performance of baseline methods. We again note that the results for Pythia-410M and Pythia-1B use a union-restricted reference, leading to particularly high scores.

We also check the lowest-ranked examples at the other end of the ranking spectrum. These are candidates for inspection in filtering or data-cleaning studies. Figure 5C reports this comparison at

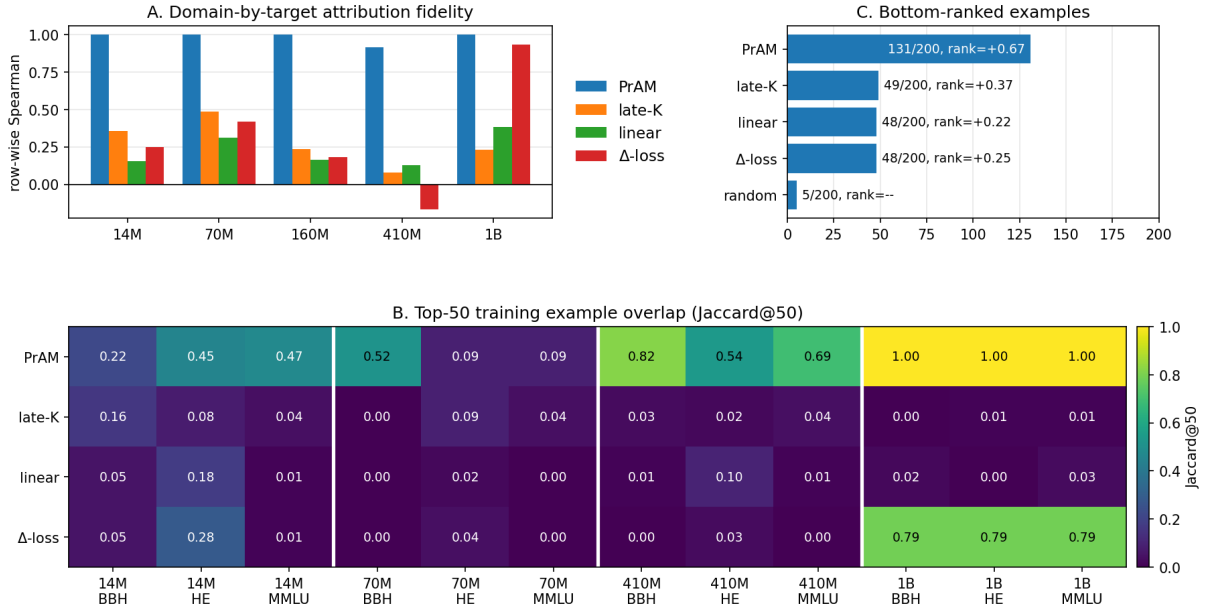


Figure 5: A summary of application-motivated analyses, compared against the available references. A) addresses domain-level attribution at $K = 8$ using row-wise Spearman agreement. B) compares the top-50 training examples for each downstream probe using Jaccard@50. Pythia-160M-2K is excluded from Panel B because the reduced-pool run includes only the 17 Pile domain targets. C) compares the bottom-200 lowest-ranked examples at 14M and reports rank correlation on the recovered overlap. Across all three outputs, PrAM stays closer to the reference than baseline checkpoint selection methods.

14M. PrAM recovers 131 of the 200 full-reference lowest-ranked examples, compared with 48–49 for late checkpoints, linear spacing, and validation-loss reduction. It also better preserves the reference ordering on the recovered overlap.

These rankings identify concrete examples to inspect and show why checkpoint selection matters beyond a single correlation score. Under the checkpointed attribution estimator, important examples for a target do not always come from the most obvious domain, so changing the checkpoint subset can change the examples a researcher inspects first.

8 Discussion and Conclusion

In this work, we addressed the setting of offline checkpoint selection, where the model has already been trained, checkpoints are available, and the attribution target is chosen after training. Specifically, we introduced PrAM, a post-hoc offline checkpoint selector that first uses a small probe to identify checkpoints with high attribution mass, then runs a full attribution pass on only the selected checkpoints. Our results show that using probe-based attribution mass as a checkpoint selection signal is highly effective in identifying important checkpoints for training data attribution. Moreover, PrAM shows that checkpointed training data

attribution does not always need every saved checkpoint.

Additionally, our results show that checkpoint choice can change post-hoc attribution analysis. If checkpoints are chosen by recency, spacing, or validation-loss heuristics, the resulting domain-level insights or ranked examples can differ substantially from those identified when using the entire checkpointed training trajectory. We demonstrated that this occurs because the high-attribution-mass region shifts across training runs, so spacing and other heuristics can be ineffective, particularly at sparse budgets when the high-mass region is not captured by the heuristic.

Our results also suggest several directions for future work. First, most of our models exhibited relatively concentrated attribution mass. If attribution mass is diffuse rather than concentrated, any budget- K subset will omit more of the full saved-checkpoint attribution, making the choice of K more important. While K is a user-specified checkpoint budget determined by available compute, this suggests some scenarios where PrAM may be less effective, warranting identification and mitigation strategies in such scenarios. Second, if large checkpoint contributions cancel each other out, magnitude-based selection may keep high-

mass checkpoints but still fail to preserve training-example rankings. Future work can study when such cancellation occurs and whether it can be detected from a small probe. Finally, while we utilized one underlying attribution method, future work can look at the efficacy of checkpointed attribution when using alternate attribution scores. These research questions will become increasingly relevant as more open-source models with diverse training trajectories are released with their checkpoints and data.

9 Limitations

Most of our experiments use Pythia-14M, 70M, and 160M, with the 160M experiments using a further reduced 2K-example training example pool and 17 Pile domain targets. The 410M/1B analyses use smaller comparison references, so they should be read as application analyses after validation at smaller scales. Although necessitated by feasibility given our compute budget, we acknowledge the size limitation of these experiments, both in terms of data size and model size. Further, as most of our experiments are conducted on the Pythia-family of models due to the limited availability of open-source models and data at smaller scales, it is unclear how PrAM will perform on other model families as more are released. We also use one underlying attribution method, and note the dependence of PrAM on the performance of the underlying attribution method itself. Future work should test other checkpoint suites, learning-rate schedules, and underlying attribution methods.

Acknowledgments

We thank the reviewers and the UVA ILP Lab for their insightful feedback, which has helped us improve this paper.

References

- Juhan Bae, Wu Lin, Jonathan Lorraine, and Roger Baker Grosse. 2024. [Training data attribution via approximate unrolling](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usven Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. [Pythia: A suite for analyzing large language models across training and scaling](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.
- Tyler A. Chang, Dheeraj Rajagopal, Tolga Bolukbasi, Lucas Dixon, and Ian Tenney. 2025. [Scalable influence and fact tracing for large language model pre-training](#). In *The Thirteenth International Conference on Learning Representations*.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. [Evaluating large language models trained on code](#). *CoRR*, abs/2107.03374.
- Soumi Das, Manasvi Sagarkar, Suparna Bhattacharya, and Sourangshu Bhattacharya. 2025. [CheckSelect: Online Checkpoint Selection for Flexible, Accurate, Robust, and Efficient Data Valuation](#). *IEEE Transactions on Artificial Intelligence*, 6(04):968–978.
- Florian Eichen, Yupei Du, Philipp Mondorf, Maria Matveev, Barbara Plank, and Michael A. Hedderich. 2026. [ExPLAINd: Unifying model, data, and training attribution to study model behavior](#). In *Forty-third International Conference on Machine Learning*.
- Logan Engstrom, Axel Feldmann, and Aleksander Madry. 2024. [DsDm: Model-aware dataset selection with datamodels](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 12491–12526. PMLR.
- Dongyang Fan, Vinko Sabolčec, Matin Ansari Pour, Ayush Kumar Tarun, Martin Jaggi, Antoine Bosselut, and Imanol Schlag. 2025a. [Can performant LLMs be ethical? quantifying the impact of web crawling opt-outs](#). In *Second Conference on Language Modeling*.
- Simin Fan, Maria Ios Glarou, and Martin Jaggi. 2025b. [GRAPE: Optimize data mixture for group robust multi-target adaptive pretraining](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- 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. 2020. [The pile: An 800gb dataset of diverse text for language modeling](#). *Preprint*, arXiv:2101.00027.
- Ce Ge, Zhijian Ma, Daoyuan Chen, Yaliang Li, and Bolin Ding. 2025. [Bimix: A bivariate data mixing law for language model pretraining](#). *Preprint*, arXiv:2405.14908.
- Amirata Ghorbani and James Zou. 2019. [Data shapley: Equitable valuation of data for machine learning](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2242–2251. PMLR.

- Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. [OLMo: Accelerating the science of language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.
- Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiušė, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. 2023. [Studying large language model generalization with influence functions](#). *Preprint*, arXiv:2308.03296.
- Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. 2019. [Data cleansing for models trained with SGD](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#). In *International Conference on Learning Representations*.
- Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. 2022. [Data-models: Understanding predictions with data and data with predictions](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9525–9587. PMLR.
- Nikhil Kandpal, Brian Lester, Colin Raffel, Sebastian Majstorovic, Stella Biderman, Baber Abbasi, Luca Soldaini, Enrico Shippole, A. Feder Cooper, Aviya Skowron, Shayne Longpre, Lintang Sutawika, Alon Albalak, Zhenlin Xu, Guilherme Penedo, Loubna Ben allal, Elie Bakouch, John Pressman, Honglu Fan, and 8 others. 2025. [The common pile v0.1: An 8tb dataset of public domain and openly licensed text](#). In *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc.
- Pang Wei Koh and Percy Liang. 2017. [Understanding black-box predictions via influence functions](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR.
- Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. 2024. [Datainf: Efficiently estimating data influence in loRA-tuned LLMs and diffusion models](#). In *The Twelfth International Conference on Learning Representations*.
- Jin Hwa Lee, Matthew J. A. Smith, Maxwell Adam, and Jesse Hoogland. 2026. [Influence dynamics and stage-wise data attribution](#). In *The Fourteenth International Conference on Learning Representations*.
- Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Kumar Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee F Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, and 40 others. 2024. [Datacomp-LM: In search of the next generation of training sets for language models](#). In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Shuangqi Li, Hieu Le, Jingyi Xu, and Mathieu Salzmann. 2026. [Lorif: Low-rank influence functions for scalable training data attribution](#). *Preprint*, arXiv:2601.21929.
- Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. 2025. [Regmix: Data mixture as regression for language model pre-training](#). In *The Thirteenth International Conference on Learning Representations*.
- Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2024. [A pretrainer’s guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3245–3276, Mexico City, Mexico. Association for Computational Linguistics.
- Ian Magnusson, Nguyen Tai, Ben Bogin, David Heine-man, Jena D. Hwang, Luca Soldaini, Akshita Bhagia, Jiacheng Liu, Dirk Groeneveld, Oyvind Tafjord, Noah A. Smith, Pang Wei Koh, and Jesse Dodge. 2025. [Datadecide: How to predict best pretraining data with small experiments](#). In *Forty-second International Conference on Machine Learning*.
- Monology. 2023. [monology/pile-uncopyrighted: A version of The Pile with copyrighted content removed](#). Hugging Face Datasets.
- Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. 2023. [TRAK: Attributing model behavior at scale](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 27074–27113. PMLR.
- Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. [The fineweb datasets: Decanting the web for the finest text data at scale](#). In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

- Jiahui Peng, Xinlin Zhuang, Jiantao Qiu, Ren Ma, Jing Yu, He Zhu, and Conghui He. 2025. [Topic over source: The key to effective data mixing for language models pre-training](#). *Preprint*, arXiv:2502.16802.
- Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. [Estimating training data influence by tracing gradient descent](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 19920–19930. Curran Associates, Inc.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. [Challenging BIG-bench tasks and whether chain-of-thought can solve them](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.
- Evan Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Taffjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, and 23 others. 2025. [2 OLMo 2 furious \(COLM’s version\)](#). In *Second Conference on Language Modeling*.
- Jiachen T. Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. 2025a. [Data shapley in one training run](#). In *The Thirteenth International Conference on Learning Representations*.
- Jiachen T. Wang, Dawn Song, James Zou, Prateek Mittal, and Ruoxi Jia. 2025b. [Capturing the temporal dependence of training data influence](#). In *The Thirteenth International Conference on Learning Representations*.
- Alexander Wettig, Kyle Lo, Sewon Min, Hannaneh Hajishirzi, Danqi Chen, and Luca Soldaini. 2025. [Organize the web: Constructing domains enhances pre-training data curation](#). In *Forty-second International Conference on Machine Learning*.
- Kangxi Wu, Liang Pang, Huawei Shen, and Xueqi Cheng. 2024. [Enhancing training data attribution for large language models with fitting error consideration](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14131–14143, Miami, Florida, USA. Association for Computational Linguistics.
- Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2023. [Doremi: Optimizing data mixtures speeds up language model pretraining](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. 2025. [Data mixing laws: Optimizing data mixtures by predicting language modeling performance](#). In *The Thirteenth International Conference on Learning Representations*.
- Zichun Yu, Spandan Das, and Chenyan Xiong. 2024. [MATES: Model-aware data selection for efficient pretraining with data influence models](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Zichun Yu, Fei Peng, Jie Lei, Arnold Overwijk, Wentau Yih, and Chenyan Xiong. 2025. [Group-level data selection for efficient pretraining](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

A Additional Results

Table 1 and Table 2 give the exact numbers for the results visualized in subsection 6.1.

B Additional Application and Comparison-Reference Details

Pythia-410M and 1B in Figure 5 do not use the full checkpoint reference, since computing attribution over all released checkpoints is too expensive at these scales. Instead, we use a union-restricted reference. For each scale, let S_m denote the $K = 8$ checkpoint set selected by method m . We form a restricted checkpoint pool

$$U = \bigcup_m S_m,$$

including the final-checkpoint anchor when it is part of the compared selector sets. Reported results are computed using only checkpoints in U .

This construction gives all methods the same restricted comparison set while avoiding a reference tied to any single selector. Accordingly, the results for Pythia-410M and Pythia-1B should be read as agreement results under a shared restricted checkpoint pool, rather than as independent validation against the full saved-checkpoint trajectory.

B.1 Why validation-loss reduction is strong at 1B

PrAM selects checkpoints with high attribution mass, not necessarily late checkpoints. At 14M, 70M, and 410M, the high-mass PrAM selections and validation-loss-drop selections fall in different parts of the saved trajectory. At Pythia-1B, however, attribution-mass shifts earlier, and PrAM’s $K = 8$ selection $\{2, 3, 4, 7, 8, 9, 10, 153\}$ includes this early high-mass region. In the restricted 1B comparison, the validation-loss-reduction rule happens to select some of the same high-mass checkpoints. This explains why the 1B validation-loss-reduction cells in Figure 5 are unusually strong. The result supports the adaptive mass selection used by PrAM rather than a universal early- or late-checkpoint rule.

C Cost-Matched and Probe-Efficiency Details

We use a simple per-checkpoint accounting model to analyze the cost of PrAM. A full saved-checkpoint reference pays for one full attribution pass at each saved checkpoint, while PrAM pays for an inexpensive probe at all checkpoints, plus a full pass at only the K selected checkpoints. Let $N_{\text{ckpt}} = T + 1$. For compactness, we write $C_p = C_p(P_{\text{small}}, D_{\text{val}}^{\text{probe}})$ for the average per-checkpoint probe cost and $C_f = C_f(D_{\text{tr}}, D_{\text{val}}^{\text{full}})$

	Pythia-14M				Pythia-70M				Pythia-160M (2K)			
	$K=4$	$K=8$	$K=16$	$K=32$	$K=4$	$K=8$	$K=16$	$K=32$	$K=4$	$K=8$	$K=16$	$K=32$
<i>Baselines</i>												
late- K	0.256	0.354	0.461	0.654	0.279	0.371	0.543	0.906	0.291	0.383	0.610	0.976
top- K -LR	0.202	0.284	0.390	0.478	0.170	0.170	0.170	0.170	0.147	0.147	0.147	0.147
final-only	0.137	0.137	0.137	0.137	0.170	0.170	0.170	0.170	0.147	0.147	0.147	0.147
linear-spaced	0.194	0.296	0.404	0.566	0.178	0.240	0.312	0.449	0.148	0.221	0.325	0.384
log-spaced	0.170	0.361	0.302	0.398	0.170	0.170	0.172	0.233	0.147	0.147	0.149	0.223
random	0.207	0.281	0.379	0.531	0.109	0.148	0.250	0.383	0.116	0.194	0.307	0.467
top- K val-loss	0.137	0.138	0.252	0.489	0.170	0.170	0.346	0.787	0.147	0.147	0.353	0.724
top- K val-loss-reduction	0.174	0.218	0.299	0.537	0.170	0.170	0.182	0.394	0.147	0.147	0.219	0.567
PrAM (probe, $ \mathcal{P} =256$)	0.370	0.464	0.610	0.763	0.401	0.533	0.725	0.939	0.442	0.602	0.784	0.981
PrAM-Full (full-probe ref., $ \mathcal{P} =N$)	0.370	0.479	0.612	0.766	0.401	0.533	0.726	0.939	0.442	0.602	0.788	0.981

Table 1: Mean per-target Spearman correlation with full-trajectory attribution for various checkpoint selectors. 14M and 70M average over 20 validation targets; 160M (reduced 2K-chunk training pool) averages over the 17 Pile domain targets only.

K	Pythia-14M (LODO Q)				Pythia-70M (LODO Q)				Pythia-160M (2K) (LODO Q)			
	late- K	linear	log	PrAM	late- K	linear	log	PrAM	late- K	linear	log	PrAM
4	0.241	0.162	0.154	0.377	0.291	0.187	0.179	0.362	0.291	0.148	0.147	0.442
8	0.342	0.264	0.365	0.469	0.387	0.256	0.179	0.500	0.383	0.221	0.147	0.538
16	0.447	0.371	0.268	0.610	0.540	0.329	0.181	0.695	0.610	0.325	0.149	0.719
32	0.641	0.544	0.356	0.751	0.902	0.465	0.246	0.932	0.976	0.384	0.223	0.978

Table 2: Leave-one-domain-out evaluation over the 17 Pile validation targets. Selectors are built from the remaining 16 source targets and evaluated on the held-out target.

K	r_p	Setup		K'	14M		70M		160M-2K	
		speedup			$\Delta_{\log}$	Δ_{lin}	$\Delta_{\log}$	Δ_{lin}	$\Delta_{\log}$	Δ_{lin}
8	0.05	$9.8\times$	15		+0.08	+0.08	+0.36	+0.25	+0.45	+0.34
8	0.20	$4.0\times$	38		+0.07	-0.12	+0.27	+0.03	+0.41	+0.09
32	0.05	$3.9\times$	39		+0.35	+0.17	+0.64	+0.47	+0.79	+0.48
32	0.20	$2.5\times$	62		+0.24	+0.06	+0.60	+0.29	+0.69	+0.34
64	0.05	$2.1\times$	71		+0.35	+0.17	+0.66	+0.33	+0.73	+0.29
64	0.20	$1.6\times$	94		+0.36	+0.09	+0.59	+0.30	+0.63	+0.29

Table 3: Representative exact- K' cost-matched gaps. $\Delta_{\log}$ and Δ_{lin} are PrAM minus spacing baselines; positive values mean PrAM wins.

for the average per-checkpoint full-attribution cost. Then

$$C_{\text{PrAM}} = N_{\text{ckpt}} C_p + K C_f,$$

$$\text{speedup} = \frac{N_{\text{ckpt}} C_f}{N_{\text{ckpt}} C_p + K C_f} = \frac{1}{r_{\text{probe}} + K/N_{\text{ckpt}}}, \quad (11)$$

where $r_{\text{probe}} = C_p/C_f$. The ratio depends on probe size, target-pool size, batching, checkpoint-loading overhead, and hardware.

In Table 3, we report the cost-matched grid. It uses Eq. 11: a spacing baseline receives the same total attribution budget after charging the probe pass, so it may select $K' = K + \lfloor r_{\text{probe}} N_{\text{ckpt}} \rfloor$ checkpoints. PrAM remains robust on 70M and 160M. On 14M, small-budget PrAM can tie or lose to cost-matched linear spacing when r_{probe} is large.

The probe heatmap in Figure 4 varies $|P_{\text{small}}| \in \{64, 128, 256, 512\}$ and $V_{\text{domain_pool}} \in \{8, 16, 32, 64\}$ at $K = 8$. Each cell is evaluated against the $V_{\text{domain_pool}}=128$ PrAM-Full reference. Per-cell A100 probe-selection cost is approximately 5–10 minutes at $V=8$ on 14M, scaling to approximately 1–2 hours at $V=64$ on 160M. These timings cover the probe stage. The final workflow also pays $K C_f(D_{\text{tr}}, D_{\text{val}}^{\text{full}})$ for full attribution at the selected checkpoints. For calibration, full 14M saved-checkpoint attribution over 17K training examples, 154 checkpoints, and 17 domain-conditioned targets costs about 29 GPU-hours, so a matched $K=8$ selected pass is roughly $(8/154) \cdot 29 \approx 1.5$ GPU-hours plus probe and I/O overhead. Full 70M saved-checkpoint attribution under the same workload costs about 74 GPU-hours. Speedups should therefore be computed from Eq. 11 for the intended workload rather than quoted as a universal ratio.

selector	$K=2$	$K=4$	$K=6$	$K=8$
late- K	0.66	0.80	0.88	0.93
linear-spaced	0.57	0.67	0.76	0.85
log-spaced	0.57	0.65	0.70	0.74
random	0.47	0.73	0.83	0.92
val-loss	0.57	0.65	0.72	0.80
val-loss-reduction	0.55	0.65	0.79	0.92
PrAM-Full (full-probe ref.)	0.66	0.80	0.88	0.94
PrAM (256)	0.66	0.79	0.88	0.94

Table 4: OLMo-2-1B cross-architecture check. Values are mean per-target Spearman correlations over the seven Dolma source-domain targets. The reference and selected-checkpoint estimates are computed on the 11 sampled stage-1 checkpoints.

D OLMo-2 Cross-Architecture Check

We include a small cross-architecture check on OLMo-2-1B to test whether PrAM’s behavior transfers beyond the Pythia checkpoint suite. We use the Dolma mixture and a 1B-token pretokenized slice grouped into seven source subsets. We evaluate the stage-1 trajectory and report agreement over 11 sampled checkpoints.

Table 4 reports mean per-target Spearman correlation with the 11-checkpoint reference for budgets $K \in \{2, 4, 6, 8\}$. These budgets are smaller than in the Pythia experiments because this check uses a much shorter evaluated trajectory. In particular, $K = 8$ covers 73% of the trained checkpoints, so the smaller budgets provide the more informative sparse-checkpoint comparison. The results follow the same qualitative pattern as the Pythia experiments.

This experiment is intended as a cross-architecture check rather than a full second benchmark. It uses a restricted set of trained OLMo-2 checkpoints and seven Dolma-domain targets, but it supports the main conclusion that a small attribution probe can locate the high-mass region of a released checkpoint trajectory outside the Pythia/Pile setting.

E Implementation Details

In this section, we provide the pseudocode for our algorithm (unanchored), as well as a comparison between the anchored and unanchored results.

E.1 Anchored vs. Unanchored PrAM

The main PrAM rule selects the $K-1$ largest-mass checkpoints together with the final checkpoint T (§4). Table 5 reports the unanchored variant at

Algorithm 1: PrAM checkpoint selection
for post-hoc training data attribution.

Input: Saved checkpoints $\{\theta_t\}_{t=0}^T$, full attribution pool D_{tr} , selected target set $D_{\text{val}}^{\text{sel}}$, probe set P_{small} , budget K , checkpoint-level attribution method τ_t

Output: Checkpoint subset S for the full attribution pass

```

1 for  $t = 0, \dots, T$  do
2   compute probe attributions  $\tau_t(z_i; q_j)$  for
    $z_i \in P_{\text{small}}, q_j \in D_{\text{val}}^{\text{sel}}$ 
3    $m_t \leftarrow \sum_{q_j \in D_{\text{val}}^{\text{sel}}} \sum_{z_i \in P_{\text{small}}} |\alpha_t \tau_t(z_i; q_j)|$ 
4 end
5  $S \leftarrow$  the  $K$  checkpoints with largest  $m_t$ 
6 run full attribution over  $D_{\text{tr}}$  only at checkpoints in  $S$ 

```

small K , which selects the K largest-mass checkpoints without forcing the final one. The two variants converge by $K=8$ because the final checkpoint is among the top-mass indices on its own. At $K=1$, anchored PrAM is forced to the final checkpoint and inherits the trivial-final-only floor; unanchored PrAM instead selects whichever single checkpoint has the largest mass, which is sometimes earlier in training and yields higher per-target Spearman. At $K=\{2, 4\}$ the gap is small and not consistent across scales. We utilize the anchored variant because our underlying goal is attribution of a specific model run, so the final checkpoint is a structural inclusion.

variant	$K=1$	$K=2$	$K=4$	$K=8$
14M anchored	0.137	0.331	0.370	0.479
14M unanchored	0.290	0.305	0.395	0.467
70M anchored	0.170	0.288	0.401	0.533
70M unanchored	0.209	0.282	0.394	0.533
160M (2K) anchored	0.147	0.289	0.442	0.602
160M (2K) unanchored	0.229	0.325	0.466	0.603

Table 5: PrAM with and without the final-checkpoint anchor, mean per-target Spearman over 20 (14M, 70M) or 17 (160M-2K) validation targets. The two variants converge by $K=8$.