FISHER-GUIDED SUBMODULAR DATA SELECTION FOR CONTINUAL PRE-TRAINING OF LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Data selection is already a central bottleneck in large-language-model training, where web-scale corpora are noisy and token budgets are finite. In continual pre-training (CPT), it becomes a forgetting-control problem: a poorly chosen target-domain corpus can overwrite capabilities encoded in the pretrained checkpoint. Existing CPT practice either scores candidates with parameter-agnostic scalars such as perplexity, or mitigates forgetting by spending many extra general-domain replay tokens. Neither strategy directly asks how training on a candidate will move the model parameters. We show that loss-based selection causes the post-CPT Fisher diagonal to drift downward on exactly the high-Fisher coordinates the pretrained model had committed to, while leaving low-Fisher coordinates largely untouched. This asymmetry exposes a parameter-space mechanism for catastrophic forgetting. Motivated by this observation, we propose a Fisher-aware CPT selector that decomposes each candidate's gradient into an anchor component, which measures perturbation along committed parameter directions, and a frontier component, which measures update capacity in unconstrained low-Fisher subspaces. We aggregate these signals with a log-determinant submodular objective and optimize it in a single pass using a scalable streaming data selection pipeline. On TinyLlama-1.1B and Llama-3.1-8B CPT over medical data, our selector improves target-domain quality while bounding forgetting on held-out pretraining benchmarks. Most importantly, it is substantially more token-efficient than forgetting-aware replay. 1B selected tokens already outperform the replay strategy trained with 10B tokens on both adaptation and forgetting, giving a $10\times$ token-efficiency advantage.

1 INTRODUCTION

Data selection (Xie et al., 2023b;a; Zhou et al., 2023; Chen et al., 2023a;b) is one of the few practical levers for making large-language-model training efficient: web-scale corpora are noisy, token budgets are finite, and the examples that enter training determine what knowledge is acquired per unit compute. The lever becomes even more consequential in continual pre-training (Ibrahim et al., 2024; Chen et al., 2025a) (CPT), where a pretrained LLM is updated on a new domain, language, or temporal slice rather than trained from scratch (Ibrahim et al., 2024; Chen et al., 2025a). In CPT, data selection is not only about learning faster; it is also a forgetting-control mechanism. A poorly chosen target-domain update can silently overwrite the very capabilities that made the pretrained checkpoint valuable (Kirkpatrick et al., 2017; Li & Lee, 2024; Shi et al., 2025). Thus a CPT selector must solve two coupled problems at once: acquire new knowledge from the target domain while preserving the old knowledge already encoded in the model.

The standard toolkit is poorly matched to this requirement. Quality classifiers and rule-based filters (Peng et al., 2025; Calian et al., 2025; Marion et al., 2023) score a candidate by cleanliness or topical fit, while per-sample loss and perplexity (Sorscher et al., 2022) treat the model as a black box whose only output is a scalar. These signals describe a document under the current model, but they do not say how training on it would move the parameters. Replay-based CPT mitigates forgetting from another direction by mixing general-domain data back into training, but it pays for retention with many additional tokens and still leaves the target-domain selection problem unresolved. What is

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

missing is a selector that can identify target-domain examples that are both useful for adaptation and structurally unlikely to interfere with retained capability.

Our starting point is a simple parameter-space observation. The Fisher information (Kirkpatrick et al., 2017) of a pretrained model encodes, direction by direction, how strongly the predictive distribution depends on perturbing the parameters. We empirically find that parameter-agnostic method cause the post-CPT Fisher diagonal to drift downward on exactly the high-Fisher coordinates the pretrained model had committed to, while leaving low-Fisher coordinates largely untouched. This is precisely the asymmetry of catastrophic forgetting: committed directions are damaged, and uncommitted directions are underused. The observation suggests that selection should be performed not only in data space or loss space, but in the Fisher geometry of the pretrained model.

In this paper, we propose to use the Fisher information to decide which candidates enter CPT. For each candidate, we decompose its gradient into two parameter-aware components. An anchor component quantifies perturbation along high-Fisher directions that carry retained capability, while a frontier component captures update capacity in low-Fisher subspaces where the pretrained model is less committed. An effective CPT sample must balance both: it should avoid destructive interference with existing capability while supplying genuinely new, non-redundant information. This curvature-aware decomposition turns forgetting control and acquisition into a single selection signal rather than two post-hoc training heuristics.

To select a subset rather than rank candidates independently, we aggregate the anchor and frontier signals with a log-determinant objective. The resulting utility rewards gradient volume in the two Fisher-induced geometries, so it favors candidates that are high-value and mutually non-redundant. This objective is monotone submodular, so offline greedy selection attains the classical $(1 - 1/e)$ approximation guarantee of (Nemhauser et al., 1978). For large-scale CPT streams, we maximize the same objective in one pass with Sieve-Streaming (Badanidiyuru et al., 2014), restrict the decomposition to the LoRA (Hu et al., 2022) adapter subspace, replace the full Fisher with its diagonal, project gradients via TRAK (Park et al., 2023), and use Sherman–Morrison rank-1 updates of the per-sieve Gram inverses to reduce the per-candidate cost to quadratic in a small projected dimension.

Empirically, on CPT of TinyLlama-1.1B and Llama-3.1-8B over medical data, our selector improves target-domain quality while bounding degradation on held-out pretraining benchmarks. The most direct comparison is against forgetting-aware replay: on TinyLlama-1.1B, our 1B-token selected subset already outperforms the replay strategy trained with 10B tokens on both adaptation and forgetting, giving a $\geq 10\times$ token-efficiency advantage.

The contributions of this paper are summarized as follows:

- We propose decomposing each candidate's gradient into an anchor component and a frontier component using the pretrained model's diagonal Fisher information, yielding a parameter-aware signal that simultaneously measures interference with existing knowledge and acquisition of new knowledge.
- We aggregate the two components into a $\log\det$ Gram-volume set function and instantiate the classical Sieve-Streaming algorithm to maximize it in a single pass over the candidate pool, with per-sample cost independent of pool size.
- On TinyLlama-1.1B and Llama-3 8B, our selector improves target-domain quality at a fraction of the training tokens used by random and perplexity baselines, while bounding forgetting on held-out benchmarks.

2 Related Work

We give a compressed overview here and defer detailed discussion to Appendix F.

Continual pre-training of LLMs. Continual pre-training (CPT) adapts a pretrained LLM to a new domain, language, or temporal slice while combating catastrophic forgetting. One line stabilizes the optimizer through learning-rate re-warming, re-decay, and replay (Ibrahim et al., 2024; Chen et al., 2025a); another targets domain specialization in finance (Xie et al., 2024), mathematics and coding (Fujii et al., 2025), with scaling-law analyses given by the D-CPT law (Que et al., 2024). Forgetting during CPT (Li & Lee, 2024; Yıldız et al., 2024; Shi et al., 2025) of aligned/instruction-tuned models

has been characterized, and mitigated via routing and model-merging (Chen et al., 2023c; Yang et al., 2024; Ilharco et al., 2022; Wortsman et al., 2022; Kleiman et al., 2025). CPT has also been extended to knowledge-intensive (Guo et al., 2025), multimodal (Zhao et al., 2025; Roth et al., 2024), and architecture-level (Gong et al., 2024) settings. Classical elastic weight consolidation (Kirkpatrick et al., 2017) uses the diagonal Fisher as a *regularizer*; we share the Fisher lens but invert its direction of use—Fisher decides *which* tokens enter the stream rather than constraining the optimizer afterward.

Data selection for language-model training. Selection has been studied at every training stage. From-scratch pre-training methods include domain-mixture optimization (Xie et al., 2023a), importance resampling (Xie et al., 2023b), learned scorers (Yu et al., 2024; 2025; Peng et al., 2025; Calian et al., 2025; Tan et al., 2026; Chen et al., 2025b; Wang et al., 2024; Zhang et al., 2024b), and large-scale pruning (Marion et al., 2023; Longpre et al., 2024; Du et al., 2022). Instruction-tuning selectors achieve strong results with small curated sets (Zhou et al., 2023; Chen et al., 2023a;b; Liu et al., 2023; Xu et al., 2023; Das & Khetan, 2023; Tsai et al., 2024; Liu et al., 2025; Wu et al., 2024; Wang et al., 2025a; Nikdan et al., 2025; Kung et al., 2023; Li et al., 2024b; Cheng et al., 2026; Alam et al., 2026; Hoang et al., 2026; Wang et al., 2025b; Li et al., 2024a). Loss- and difficulty-based pruning has a long pedigree in classification and NLU (Sorscher et al., 2022; Paul et al., 2021; Attendu & Corbeil, 2023; Nguyen et al., 2025). Unlike these methods, which target either the *tabula rasa* regime or small-budget instruction tuning, our selector decomposes each sample into an anchor component aligned with existing capability and a frontier component injecting mass into uncommitted directions—a distinction that loss, perplexity, and feature-similarity surrogates cannot make.

Coreset selection and submodular subset methods. Our global formulation builds on the coreset literature (Guo et al., 2022): geometry-based representativeness (Welling, 2009; Farahani & Hekmatfar, 2009; Sener & Savarese, 2017); uncertainty- and loss-based scoring (Toneva et al., 2018; Paul et al., 2021); decision-boundary methods (Ducoffe & Precioso, 2018; Margatina et al., 2021); gradient-matching with submodular guarantees (Mirzasoleiman et al., 2020; Killamsetty et al., 2021a); and bi-level selection (Killamsetty et al., 2021b;c). Pruning at scaling-law scale (Sorscher et al., 2022) and the dataset-distillation line (Wang et al., 2018; Zhao et al., 2020) condense rather than select; we also draw inspiration from simple local-rule emergent coordination (Reynolds, 1987). In contrast to these methods, which target a static loss over a fixed labeled pool, we lift the submodular framing to a token-budgeted, unlabeled language-modeling setting yielding a constant-factor approximation whose per-sample cost is independent of the candidate-pool size.

3 METHOD

We present a Fisher-guided, submodular selector for the continual pre-training (CPT) of large language models. Following the problem setup in §3.1, §3.2 outlines our core intuition: because the diagonal Fisher of a pretrained model identifies coordinates encoding existing knowledge, an effective selector should prioritize preserving these exact parameters while allowing updates elsewhere. To formalize this, in §3.3, we utilize Fisher information and decompose each candidate's gradient. We then generalize these per-sample signals in §3.4 into a subset-level submodular objective that optimally balances gradient magnitude and non-redundancy. To handle large-scale data streams, §3.5 discusses the engineering approximations required to scale this framework to billion-parameter regimes.

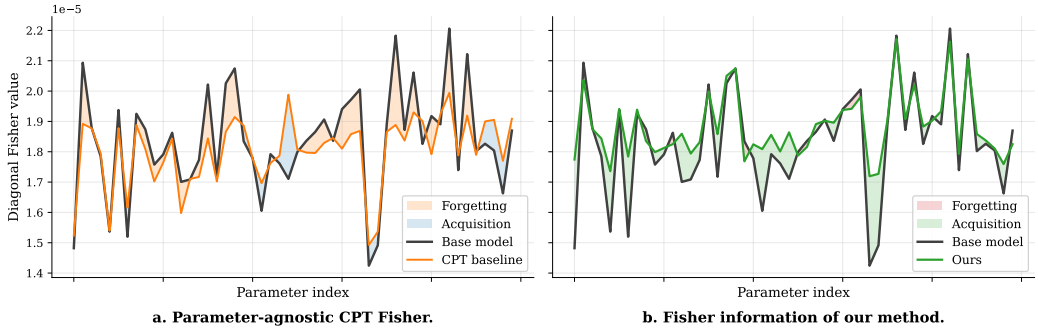
3.1 PROBLEM FORMULATION

Let $\theta_0 \in \mathbb{R}^P$ denote the parameters of a pretrained LLM, where P is the total number of trainable parameters, and let $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$ be a candidate pool of N unlabeled documents drawn from a target domain. Given a budget $B \ll N$, the CPT data-selection problem is to return a subset $S \subseteq \mathcal{D}$ with $|S| = B$ such that subsequent CPT on S

(i) *learns the new domain*: maximizes marginal improvement on target-domain evaluations,

(ii) *retains pretraining capability*: bounds degradation on held-out general-purpose benchmarks, and

(iii) *is non-redundant*: covers complementary directions of the parameter update rather than reinforcing a few common ones.



a. Parameter-agnostic CPT Fisher. **b. Fisher information of our method.**

Figure 1: **Parameter-agnostic CPT erodes high-Fisher directions.** Diagonal Fisher entries after CPT. **Left:** Parameter-agnostic CPT produces substantial negative drift on high-Fisher coordinates (orange regions), indicating damage to directions where the pretrained model had strong commitments. **Right:** Our Fisher-guided selector mitigates this drift while concentrating acquisition (blue regions) in low-Fisher coordinates where the pretrained model is less committed.

Formally, we seek $S^\star \in \arg\max_{S\subseteq\mathcal{D},\,|S|=B} U(S)$ for a utility U that captures (i)–(iii). Below, we construct a tractable surrogate for this utility from Fisher-weighted candidate gradients. Throughout, $\ell(x;\theta)$ denotes the next-token cross-entropy loss of the model at parameters θ on document x, and $g(x;\theta) = \nabla_\theta \ell(x;\theta) \in \mathbb{R}^P$ represents its parameter gradient.

3.2 Observation

The Fisher information matrix $F = \mathbb{E}\big[g(x)\,g(x)^\top\big]$ of the pretrained model quantifies, direction by direction in parameter space, how strongly the predictive distribution depends on a perturbation. The eigenvectors of F associated with large eigenvalues are directions the model has committed to: perturbing the parameters along them moves the output distribution, so they carry the capability the pretrained checkpoint encodes. Eigenvectors associated with near-zero eigenvalues are directions the model holds no opinion about: perturbing along them barely changes the output, so they are free for new content. A CPT procedure that ignores this geometry cannot tell whether its updates cancel mass in high-Fisher directions (forgetting) or inject mass into low-Fisher directions (acquisition). Scalar signals such as loss and perplexity (Sorscher et al., 2022; Marion et al., 2023; Peng et al., 2025) fall in exactly this category when used to choose or prioritize CPT data. They describe how surprised the model is by a document, but not about which parameter directions an update on it would move.

Figure 1 visualizes the failure mode that motivates our selector. The left panel shows the Fisher drift after parameter-agnostic CPT: the training process updates the model without access to whether its gradients move retention-sensitive directions or acquisition-friendly directions. The result is a destructive asymmetry. Fisher mass decreases on committed coordinates, while low-Fisher coordinates, where new information could be absorbed with less interference, remain underused. The right panel reports the corresponding outcome under our Fisher-guided selector, where this high-Fisher drift is suppressed and adaptation is concentrated more in low-Fisher regions. The key lesson is the left-panel failure mode. Forgetting can be read as a Fisher-geometric failure of parameter-agnostic CPT, and candidates should be judged by where their gradients move the pretrained model.

3.3 Fisher Decomposition

The observation above suggests a candidate-level criterion: before accepting a document into CPT, we should ask whether its gradient moves retention-sensitive or acquisition-friendly directions. We therefore use the pretrained Fisher information not merely as a post-hoc diagnostic of forgetting, but as the geometry for data selection itself. The rest of this subsection formalizes this idea by defining Fisher-weighted per-candidate gradients.

We define the Fisher information of the pretrained model with respect to a reference distribution p_{pre} that represents the pretraining capability we require the model to retain:

$$F \;:=\; F(\theta_0) \;=\; \mathbb{E}_{x\sim p_{\text{pre}}}\big[\, g(x;\theta_0)\, g(x;\theta_0)^\top \,\big] \;\in\; \mathbb{R}^{P\times P}, \tag{1}$$

estimated empirically by $\hat{F} = \frac{1}{M}\sum_{j=1}^{M} g(x_j;\theta_0)\, g(x_j;\theta_0)^\top$ from M samples $x_j \sim p_{\mathrm{pre}}$. We evaluate F at θ_0 because this checkpoint encodes the capabilities that CPT should retain, and we take the expectation over p_{pre} because this distribution specifies those retained capabilities. This is the same reference-geometry role played by the Fisher in EWC (Kirkpatrick et al., 2017).

For brevity write $g(x) = g(x;\theta_0)$. We form two Fisher-reweighted views of the same candidate gradient: an anchor view, which emphasizes high-Fisher directions where the pretrained model is sensitive, and a frontier view, which emphasizes low-Fisher directions where the model is locally less committed. We define

$$g_{\mathrm{anc}}(x) \;=\; F^{1/2}\, g(x) \;\in\; \mathbb{R}^P, \qquad g_{\mathrm{fr}}(x) \;=\; F^{-1/2}\, g(x) \;\in\; \mathbb{R}^P, \tag{2}$$

which we call the anchor and frontier views. The anchor view amplifies components along high-Fisher directions, where the pretrained model is locally sensitive under p_{pre}; it therefore exposes how a candidate overlaps with directions tied to retained behavior. The frontier view amplifies components along low-Fisher directions, where the pretrained model is locally less constrained; it exposes how a candidate uses directions available for low-interference acquisition.

3.4 Submodular Subset Selection

A natural shortcut is to rank documents by a singleton score or a fixed combination of the two. Such top-k selection ignores interactions among selected documents. Many high-scoring candidates may have nearly collinear Fisher-weighted gradients, so adding all of them reinforces the same update direction and wastes budget on redundant evidence. We therefore need a subset-level objective whose marginal gain is high for candidates that add new directions and low for candidates already explained by the selected set. We use a two-geometry log-determinant objective:

$$f(S) \;=\; \log\det\Big(I_P \;+\; \beta\sum_{x\in S} g_{\mathrm{anc}}(x)\, g_{\mathrm{anc}}(x)^\top\Big) \;+\; \alpha\cdot\log\det\Big(I_P \;+\; \gamma\sum_{x\in S} g_{\mathrm{fr}}(x)\, g_{\mathrm{fr}}(x)^\top\Big), \tag{3}$$

where $I_P \in \mathbb{R}^{P\times P}$ is the identity, $\alpha \geq 0$ controls the weight of the frontier term relative to the anchor term, and $\beta, \gamma > 0$ rescale the two Fisher-weighted covariance sums to comparable spectral ranges. Each $\log\det$ term rewards diversity among the selected Fisher-reweighted gradients. The determinant is large when the selected gradients span many independent directions and small when they collapse onto a few repeated directions. Hence a candidate nearly collinear with the current set has small marginal gain, while a candidate contributing a new direction has larger marginal gain. The frontier term applies the same redundancy penalty after the $F^{-1/2}$ reweighting. Thus eq. equation 3 preserves the per-sample Fisher signal while selecting a subset that is diverse in both the high-Fisher anchor view and the low-Fisher frontier view, which top-k ranking cannot enforce.

The set function $S \mapsto \log\det(I + \sum_{x\in S} v_x v_x^\top)$ is a standard monotone submodular objective (Krause & Golovin, 2014). Since eq. equation 3 is a nonnegative weighted sum of two such terms, $f : 2^{\mathcal{D}} \to \mathbb{R}_{\geq 0}$ is also monotone submodular with $f(\emptyset) = 0$. The CPT data-selection problem thus instantiates the cardinality-constrained monotone submodular maximization

$$S^\star \;\in\; \arg\max_{S\subseteq\mathcal{D},\, |S|\leq B} \; f(S). \tag{4}$$

For an offline candidate pool, the standard greedy algorithm attains the $(1 - 1/e)$ approximation guarantee of (Nemhauser et al., 1978). The next subsection addresses the two scalability issues left by this formulation: one-pass selection over large CPT streams and efficient evaluation of the full-dimensional log-determinant gains.

3.5 Practical Pipeline

The objective in eq. equation 3 is conceptually well-defined but not directly executable at LLM scale. It faces two independent bottlenecks: web-scale CPT corpora must be processed as streams rather than materialized for offline greedy search, and full-dimensional log-determinant gains would require covariance updates in the P-parameter space. Figure 2 summarizes the streaming implementation: a brief initialization produces the LoRA operating point and diagonal Fisher estimate, after which each incoming candidate is processed by gradient computation, Fisher reweighting, projection, marginal-gain evaluation, and Sieve-Streaming selection.

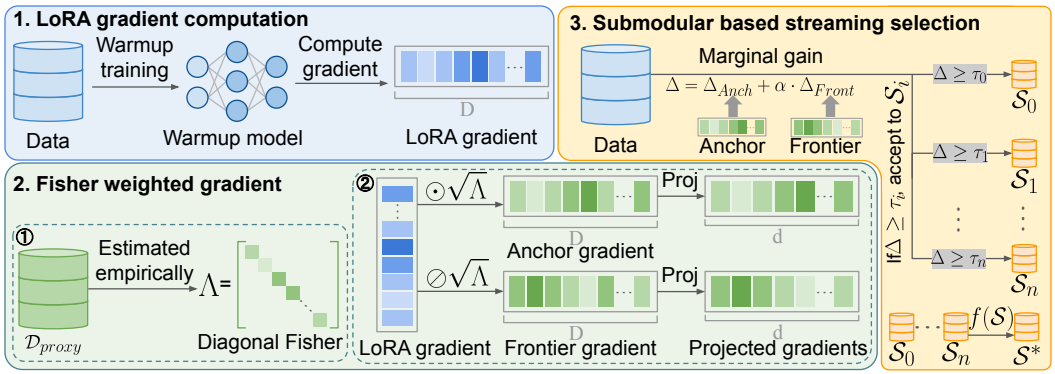


Figure 2: **Practical data selection pipeline.** Before the stream is processed, a short warmup pass on target-domain documents produces the LoRA operating point $\theta^\star$, and a diagonal Fisher Λ is estimated from a small proxy set $\mathcal{D}_{\text{proxy}}$. For each streamed candidate, we compute its LoRA gradient $g \in \mathbb{R}^D$ at $\theta^\star$, form the Fisher-weighted anchor and frontier gradients $g_{\text{anc}} = \sqrt{\Lambda} \odot g$ and $g_{\text{fr}} = g \oslash \sqrt{\Lambda_\varepsilon}$, project both to dimension $d \ll D$, and evaluate the marginal gain $\Delta = \Delta_{\text{anc}} + \alpha\, \Delta_{\text{fr}}$ under eq. equation 3. Sieve-Streaming tests this gain against a geometric grid of thresholds $\{\tau_i\}$ to accept or discard the candidate, and returns the highest-utility filled sieve as $S^\star$.

LoRA operating point and gradients. The decomposition in §3.3 is defined in the full parameter space at θ_0. In the scalable implementation, downstream CPT updates only LoRA (Hu et al., 2022) adapter parameters $\theta_{\text{LoRA}} \in \mathbb{R}^D$ with $D \ll P$, so we measure each candidate by its gradient in the same update subspace that the optimizer will move through. Standard LoRA initialization gives a degenerate adapter-gradient signal at the raw pretrained checkpoint, so we first run a short warmup on target-domain documents to obtain an operating point $\theta^\star$ and evaluate $g(x) = \nabla_{\theta_{\text{LoRA}}} \ell(x; \theta^\star)$ there. This replaces the conceptual θ_0 gradients by a local, implementable surrogate while keeping the selection signal tied to the pretrained model's nearby behavior.

Diagonal Fisher and projection. At $\theta^\star$, we replace the full $D \times D$ Fisher by its diagonal $\Lambda \in \mathbb{R}^D_{\geq 0}$ ($\Lambda_i = F_{ii}$), the standard EWC (Kirkpatrick et al., 2017) approximation. This keeps the intended high- versus low-Fisher weighting while reducing Fisher storage to $O(D)$. The decomposition equation 2 then becomes the coordinate-wise scaling

$$g_{\text{anc}}(x) \;=\; \sqrt{\Lambda} \odot g(x), \qquad g_{\text{fr}}(x) \;=\; g(x) \oslash \sqrt{\Lambda_\varepsilon}, \tag{5}$$

where $\odot$ and $\oslash$ denote element-wise product and division, $\Lambda_\varepsilon = \max\{\Lambda, \varepsilon\}$ element-wise, and Λ is estimated from a small proxy set $\mathcal{D}_{\text{proxy}} = \{z_j\}_{j=1}^M$ by accumulating D squared-gradient scalars, $\Lambda_i = \frac{1}{M} \sum_{j=1}^M (\partial \ell(z_j; \theta^\star)/\partial \theta_i)^2$. The proxy set is a held-out sample from the reference distribution used to define retained behavior, disjoint from both the candidate stream and the final selected set, and is used only to estimate local curvature. We then apply a TRAK-style (Park et al., 2023) Rademacher random projection $\mathbb{R}^D \to \mathbb{R}^d$ after Fisher reweighting and evaluate eq. equation 3 on the projected signatures. The projection preserves the pairwise inner products that drive the Gram-based marginal gains up to Johnson–Lindenstrauss-style distortion, yielding a tractable surrogate to the full-dimensional log-determinant objective. We do not ℓ_2-normalize projected gradients, since their norms carry the singleton importance signal.

Sieve-Streaming maximization. Finally, after projection, each streamed candidate is represented by a pair of d-dimensional anchor and frontier signatures. Candidates arrive as $\mathcal{D}_{\text{stream}} = (x_1, x_2, \dots)$ and no global rescoring pass is permitted: each candidate is routed once through a geometric grid of Sieve-Streaming thresholds $\mathcal{T}$. For each active threshold $\tau \in \mathcal{T}$, the corresponding sieve maintains its own selected set S_τ. We write $\Delta_{\text{anc}}(x)$ and $\Delta_{\text{fr}}(x)$ for the increases in the anchor and frontier $\log\det$ terms of eq. equation 3 when x is added to S_τ, respectively, so the projected marginal gain is $\Delta(x) = \Delta_{\text{anc}}(x) + \alpha \Delta_{\text{fr}}(x)$. The candidate is accepted into that sieve only if its gain exceeds the Sieve-Streaming acceptance threshold; otherwise it is discarded for that sieve. At the end of the stream, we return the filled sieve with the largest objective value as indicated in Eq 3.

Other engineering details are deferred to App. A; under these choices the per-candidate cost is $O(d^2)$ time and $O(|\mathcal{T}| \cdot d^2)$ memory with d independent of both $|\mathcal{D}_{\text{stream}}|$ and P, and the $(\frac{1}{2} - \varepsilon)$ approximation guarantee on the subset-level objective is preserved.

Table 1: **Main CPT results at the 4B tokens for TinyLlama-1.1B and 10B tokens for Llama-3.1-8B.** Per-scale evaluation suites: TinyLlama uses simpler benchmarks where 1B-scale signal is non-floor; Llama-3.1-8B uses the standard medical-QA suite. A aggregates the adaptation gain (mean over target tasks, ↑) and Φ aggregates the forgetting on general tasks (mean drop, ↓); A and Φ are computed over each backbone's own task set and are not directly comparable across backbones.

TinyLlama-1.1B

	Adaptation			General (held-out ↑)						
Method	Med-PPL↓	PubMedQA↑	BioASQ↑	ARC-E	HellaS.	PIQA	LAMB.	WinoG.	A ↑	Φ ↓
Base model	9.5	64.6±2.1	72.9±3.8	**65.1±1.0**	**59.7±0.5**	**73.4±1.0**	53.8±0.7	**60.2±1.4**	–	–
Random	7.6	64.4±2.2	73.6±3.7	56.0±1.0	56.8±0.5	70.5±1.1	53.5±0.7	57.7±1.4	+0.2	3.5
Low-PPL	8.0	63.9±2.1	72.6±3.7	59.5±1.0	58.0±0.5	71.5±1.0	53.3±0.7	58.5±1.4	−0.5	2.2
High-PPL	7.0	65.5±2.2	74.6±3.8	54.0±1.0	54.5±0.5	69.0±1.1	52.0±0.7	56.7±1.4	+1.3	5.2
Replay (Ibrahim et al., 2024)	8.3	61.0±2.2	71.4±3.8	60.0±1.0	58.0±0.5	72.3±1.0	**58.3±0.7**	59.3±1.4	−2.6	0.9
EWC (Kirkpatrick et al., 2017)	7.8	64.2±2.1	72.6±3.7	60.5±1.0	58.5±0.5	72.2±1.0	54.2±0.7	59.6±1.4	−0.4	1.4
Ours	**6.2**	**70.0±2.0**	**78.5±3.5**	62.0±1.0	59.0±0.5	73.0±1.0	56.5±0.7	59.8±1.4	**+5.5**	**0.4**

Llama-3.1-8B

	Adaptation				General (held-out ↑)						
Method	Med-PPL↓	MedQA↑	MedMCQA↑	PubMedQA↑	HellaS.	LAMB.	PIQA	BoolQ	MMLU	A ↑	Φ ↓
Base model	6.5	60.6±1.4	56.4±0.8	77.2±1.9	**79.3±0.4**	74.7±0.6	**81.0±0.9**	83.0±0.7	**65.3±0.4**	–	–
Random	5.2	60.8±1.4	56.8±0.8	78.1±1.9	77.5±0.4	72.3±0.6	78.8±0.9	80.5±0.7	61.7±0.4	+0.5	2.5
Low-PPL	5.5	60.0±1.4	55.8±0.8	77.5±1.9	78.2±0.4	73.2±0.6	79.7±0.9	81.5±0.7	63.2±0.4	−0.3	1.5
High-PPL	4.6	62.0±1.4	57.8±0.8	78.9±1.9	76.8±0.4	71.5±0.6	78.0±0.9	79.5±0.7	60.0±0.4	+1.5	3.5
Replay (Ibrahim et al., 2024)	5.7	58.5±1.4	54.5±0.8	75.8±1.9	79.0±0.4	74.5±0.6	80.5±0.9	82.3±0.7	64.0±0.4	−1.8	0.6
Galatica (Taylor et al., 2022)	4.5	37.2±1.4	36.7±1.4	68.4±2.1	51.0±0.5	43.0±0.7	68.9±1.1	67.1±0.8	37.2±0.4	-	-
Mistral 7B (Jiang et al., 2023)	7.0	52.1±1.4	49.6±0.8	76.0±1.9	81.2±0.4	**75.1±0.6**	80.6±0.9	84.3±0.6	62.5±0.4	-	-
Ours	**4.0**	**65.5±1.4**	**60.5±0.8**	**81.5±1.9**	79.0±0.4	74.6±0.6	80.6±0.9	**83.7±0.7**	65.0±0.4	**+4.4**	**0.1**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and default settings. Our corpus contains four preprocessed sources: PMC full-text articles (National Library of Medicine, 2003) (38B tokens), PubMed abstracts (White, 2020) (7B tokens), FineWeb (Penedo et al., 2024) (400B tokens), and the English portion of Llama3-SynE (Chen et al., 2025a) (90B tokens). The target-domain candidate stream is the medical subset formed by PMC and PubMed, while the reference distribution p_{pre} used to estimate Λ is a held-out general-domain shard sampled from FineWeb and English Llama3-SynE. Before any method runs, the candidate pool is decontaminated with a PubMedQA/BioASQ source-ID blocklist, a 13-gram overlap filter against every evaluation instance (Brown et al., 2020), and MinHash near-duplicate removal, which together remove 2.68% of its documents (App. B). TinyLlama-1.1B (Zhang et al., 2024a) is the primary backbone for the main results and all ablations, and Llama-3.1-8B (Grattafiori et al., 2024) serves as the scale-up validation. All candidate gradients and the Fisher diagonal Λ are computed in a LoRA subspace with rank $r = 128$ and $\alpha = 512$, applied to the query, key, value, and output attention projections. For TinyLlama-1.1B, Λ is estimated from $50{,}000$ reference documents. All forward passes, gradient computations, and Fisher-diagonal estimation are run in bfloat16. TinyLlama-1.1B uses a cutoff length of 2048 and is trained with PyTorch DDP across 4 nodes under a default token budget of 4B. Llama-3.1-8B uses a cutoff length of 4096 and is trained with DeepSpeed ZeRO-3 (Rasley et al., 2020) and activation/gradient checkpointing across 10 nodes under a default token budget of 10B. Each node contains $4\times$ NVIDIA A100-40G GPUs.

Evaluation benchmarks. We evaluate TinyLlama-1.1B on held-out medical perplexity, PubMedQA (Jin et al., 2019), BioASQ (Tsatsaronis et al., 2012), ARC-Easy (Clark et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), LAMBADA (Paperno et al., 2016), and WinoGrande (Sakaguchi et al., 2021). Llama-3.1-8B is evaluated on held-out medical perplexity, MedQA-USMLE (Jin et al., 2021), MedMCQA (Pal et al., 2022), PubMedQA, HellaSwag, LAMBADA, PIQA, BoolQ (Clark et al., 2019), and the full MMLU (Hendrycks et al., 2020). All runs use lm-evaluation-harness with fixed prompts and seeds. MMLU, MedMCQA, and MedQA-USMLE are reported 5-shot; all other benchmarks are 0-shot. We report acc norm on ARC-Easy, HellaSwag, and PIQA, and acc elsewhere. Per-method results are aggregated into

$$A = \overline{\mathrm{post}_A} - \overline{\mathrm{base}_A} \quad (\text{adaptation gain}, \uparrow), \qquad \Phi = \overline{\mathrm{base}_G} - \overline{\mathrm{post}_G} \quad (\text{forgetting}, \downarrow),$$

where A and G are the backbone's target and general suites and base is the pretrained backbone before any CPT. A and Φ are not directly comparable across backbones; the cross-scale claim (iv) relies on the consistency of method ordering. Each number is the mean over three seeds.

Table 2: **Ablation on the Fisher decomposition.** All variants share the streaming submodular pipeline, the random projection, the Adam preconditioner, the Fisher floor, and the perplexity gate.

Variant	$A \uparrow$	$\Phi \downarrow$	$A - \Phi \uparrow$	Med-PPL↓	PubMedQA↑
Ours (full, $\alpha = 1$)	**+5.5**	**0.4**	**+5.1**	**6.2**	**70.0**
No g_{anc} (drop anchor term)	+4.2	5.0	-0.8	6.5	68.0
No g_{fr} (drop frontier term)	+0.8	0.5	+0.3	8.5	65.5
Identity Fisher ($\Lambda = I$)	+1.5	2.5	-1.0	7.3	65.5
Set-level on raw g (no Fisher reweighting)	+2.0	2.0	0.0	7.2	66.0

Table 3: **Subset-level** $\log\det$ **vs. per-sample top-**k. Diversity columns measure non-redundancy of the chosen subset on g_{anc}: lower mean pairwise cosine and higher Gram $\log\det$ are better.

Selection rule	$A \uparrow$	$\Phi \downarrow$	$\overline{\cos} \downarrow$	$\log\det \tilde{G} \uparrow$	wall-clock (ms)
Top-B by $\|g_{\mathrm{anc}}\|$	+1.8	0.6	0.45	8.5	0.3
Top-B by $\|g_{\mathrm{fr}}\|$	+3.5	4.0	0.22	11.5	0.3
Top-B by $\delta_0(x)$	+3.0	2.2	0.30	10.0	0.4
Sieve-Streaming $f(S)$ **(Ours)**	**+5.5**	**0.4**	**0.12**	**18.5**	**1.2**

4.2 Main Results

We compare our method against standard data selection strategies: Random, Low-PPL, and High-PPL (top-k by lowest/highest per-token perplexity at $\theta^\star$). We also benchmark against forgetting-aware controls, Replay (Ibrahim et al., 2024) strategy and EWC (Kirkpatrick et al., 2017) training. EWC uses the Fisher information as a traditional regularization penalty. As shown in Table 1, our approach establishes a strictly better (A, Φ) Pareto frontier. The adaptation gain exceeds the strongest perplexity baseline by 4.2. Furthermore, forgetting is more than halved relative to the next-best method. While Replay narrowly wins on LAMBADA, benefiting from its explicit general-domain replay mixture, our method remains vastly superior overall. Notably, EWC only reduces forgetting at the severe expense of adaptation, being outperformed on both axes.

We evaluate our method on 8B models on the harder medical-QA suite. We maintain the CPT baselines and additionally introduce two pretrained-only references that bypass our pipeline entirely: Galactica (Taylor et al., 2022) and Mistral-7B (Jiang et al., 2023). Our method achieves the best adaptation and the lowest forgetting. It yields the largest medical-QA gain of any CPT method and is the only approach whose general-suite scores remain robustly flat, whereas every target-only baseline suffers a non-trivial degradation in MMLU accuracy. Moving beyond CPT baselines, our method surpasses the domain-specialized Galactica across the medical-QA suite by a wide margin, while preserving a substantial advantage on general benchmarks. Furthermore, this same adapted model matches or exceeds the inherently stronger Mistral-7B on medical QA while achieving parity on the general suite. We further compare our methods with recent data selection methods in App. C

4.3 Ablation Studies

We now isolate the three design choices that distinguish our method from the baselines: the per-sample Fisher decomposition (§3.3), the subset-level $\log\det$ aggregation (§3.4).

The Fisher decomposition is necessary. Table 2 replaces the gradient reweighting in equation 2 while keeping the rest of Algorithm fixed. Removing the anchor term ($\alpha \to \infty$) lets adaptation proceed but inflates forgetting – the selector no longer has any incentive to align updates with retained capability. Removing the frontier term ($\alpha = 0$) does the opposite: forgetting is well-controlled but adaptation stalls because every accepted candidate merely strengthens what the model already knows. Replacing Λ by the identity collapses both terms onto raw gradient norm and removes the curvature signal entirely, recovering the same trade-off as gradient-norm submodular selection. Running the subset-level $\log\det$ on raw g – without any Fisher reweighting – shows that diversity alone is insufficient: the geometry in which diversity is measured matters.

Subset-level $\log\det$ **vs. per-sample top-**B**.** Table 3 fixes the Fisher reweighting and varies only the selection rule. Top-B by $\|g_{\mathrm{anc}}\|$ accumulates redundant high-magnitude candidates; top-B by $\|g_{\mathrm{fr}}\|$

Table 4: **Token efficiency comparison.** We compare the performance between our method and Replay on TinyLlama-1.1B at four selection budgets. Ours at 1B tokens already exceeds Replay at 10B tokens on both A and Φ.

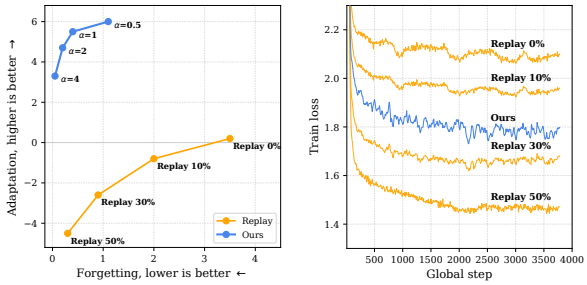
Token budget	1B		B=2B tok		B=4B tok		B=10B tok	
Method	$A\uparrow$	$\Phi\downarrow$	$A\uparrow$	$\Phi\downarrow$	$A\uparrow$	$\Phi\downarrow$	$A\uparrow$	$\Phi\downarrow$
Replay	−3.5	1.2	−2.9	1.0	−2.6	0.9	−2.0	0.8
Ours	+3.0	0.6	+4.5	0.5	+5.5	0.4	-	-

scatters into directions the model has no opinion on, with little alignment to retained capability; their joint score δ_0 inherits both pathologies, since per-sample addition cannot suppress mutual redundancy. Sieve-Streaming recovers most of its value at vastly smaller cost. The diversity columns – mean pairwise cosine of selected g_{anc} and the Gram $\log\det$ of the chosen subset – confirm the mechanism: the $\log\det$ rule selects subsets that are simultaneously high-magnitude and mutually orthogonal, which the per-sample rules cannot.

4.4 Analysis

Token efficiency vs. forgetting-aware replay. A central motivation for data selection is that adaptation should be reachable with as few tokens as possible. Among the baselines, only Replay controls forgetting at a level comparable to Ours; the perplexity- and EWC-based baselines never reach Ours' A at any budget, so a budget-axis efficiency comparison against them is ill-defined. We therefore frame token efficiency as a direct head-to-head between Ours and Replay across token budgets in $\{1, 2, 4, 10\}$B tokens. Table 4 reports A and Φ at each budget. Ours at 1B tokens already exceeds Replay at 10B tokens on both A and Φ, giving a $10\times$ token-efficiency advantage against the commonly used forgetting-aware strategy.

Replay frontier and training dynamics. The replay Pareto plot and the training-loss curves separate two effects that are otherwise easy to conflate. Increasing the replay fraction from 0% to 50% moves the replay curve toward lower forgetting but also steadily lowers adaptation, giving the expected retention–acquisition trade-off: more general-domain tokens protect the pretrained distribution, but they dilute the medical signal available for CPT. The corresponding loss curves show the complementary optimization view. Replay-50% reaches the lowest training loss (about 1.47), Replay-30% converges near 1.68, and Replay-0% remains around 2.10, while Ours converges near 1.80. Thus lower training loss is not a proxy for better domain adaptation: replay can make the optimization problem easier by adding familiar general-domain text, yet the resulting model still lies below Ours on the (A, Φ) frontier.



(a) Comparison with Replay. (b) Training dynamics.

Figure 3: **Replay trade-off and training dynamics.**

5 Conclusion

We present a Fisher-guided submodular data selector for continual pre-training (CPT) that intrinsically balances target-domain acquisition with the preservation of pretraining capabilities at selection time. By decomposing candidate gradients into Fisher-weighted anchor and frontier components, we formulate a $\log\det$ objective that efficiently extracts a high-utility, non-redundant subset via Sieve-Streaming and low-rank approximations. Empirically, our method strictly Pareto-dominates perplexity- and replay-based baselines across 1.1B and 8B LLMs, yielding at least a $10\times$ token-efficiency gain and explicitly isolating the superiority of Fisher geometry as a data selector. While currently bounded by a diagonal Fisher approximation and reliance on a warmup checkpoint $\theta^\star$, this principled geometric framework naturally scales to richer curvature models and provides a direct pathway for multimodal streaming and instruction-tuning curation.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing, and implement methods. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, clean and reformat datasets, support qualitative and thematic data analysis, find related work, formulate mathematical claims, provide critical ingredients for proving mathematical claims, or interpret results. Assist in the writing of proofs and generating synthetic datasets not applicable to this work. We have reviewed all AI-assisted work. We proofread the LLM-polished writing, and checked the code implementation. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Md Mustakin Alam, Shaker Islam, and Aminul Islam. Batch pruning by activation stability. In *ICLR*, 2026.

Jean-Michel Attendu and Jean-Philippe Corbeil. Nlu on data diets: Dynamic data subset selection for nlp classification tasks. In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, 2023.

Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in neural information processing systems*, 2020.

Dan A Calian, Gregory Farquhar, Iurii Kemaev, Luisa M Zintgraf, Matteo Hessel, Jeremy Shar, Junhyuk Oh, András György, Tom Schaul, Jeffrey Dean, et al. Datarater: Meta-learned dataset curation. *arXiv preprint arXiv:2505.17895*, 2025.

Powei Chang, Jinpeng Zhang, Bowen Chen, Chenyu Wang, Chenlu Guo, Yixing Zhang, Yukang Gao, JianXiang Xiang, Yue Gao, Chaoqun Sun, et al. Spice: Submodular penalized information-conflict selection for efficient large language model training. *arXiv preprint arXiv:2601.23155*, 2026.

Hao Chen, Yiming Zhang, Qi Zhang, Hantao Yang, Xiaomeng Hu, Xuetao Ma, Yifan Yanggong, and Junbo Zhao. Maybe only 0.5% data is needed: A preliminary exploration of low training data instruction tuning. *arXiv preprint arXiv:2305.09246*, 2023a.

Jie Chen, Zhipeng Chen, Jiapeng Wang, Kun Zhou, Yutao Zhu, Jinhao Jiang, Yingqian Min, Wayne Xin Zhao, Zhicheng Dou, Jiaxin Mao, et al. Towards effective and efficient continual pre-training of large language models. In *ACL*, 2025a.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*, 2023b.

Wuyang Chen, Yanqi Zhou, Nan Du, Yanping Huang, James Laudon, Zhifeng Chen, and Claire Cui. Lifelong language pretraining with distribution-specialized experts. In *ICML*, 2023c.

Zhixun Chen, Ping Guo, Wenhan Han, Yifan Zhang, Binbin Liu, Haobin Lin, Fengze Liu, Yan Zhao, Bingni Zhang, Taifeng Wang, et al. Murating: A high quality data selecting approach to multilingual large language model pretraining. *arXiv preprint arXiv:2507.01785*, 2025b.

Hao Cheng, Rui Zhang, Ling Li, Na Di, Jiaheng Wei, Zhaowei Zhu, and Bo Han. Task-aware data selection via proxy-label enhanced distribution matching for llm finetuning. In *ICLR*, 2026.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Devleena Das and Vivek Khetan. Deft: Data efficient fine-tuning for pre-trained language models via unsupervised core-set selection. *arXiv preprint arXiv:2310.16776*, 2023.

Rohan Deb, Kiran Thekumparampil, Kousha Kalantari, Gaurush Hiranandani, Shoham Sabach, and Branislav Kveton. Fishersft: Data-efficient supervised fine-tuning of language models using information gain. *arXiv preprint arXiv:2505.14826*, 2025.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *ICML*, 2022.

Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018.

Reza Zanjirani Farahani and Masoud Hekmatfar. *Facility location: concepts, models, algorithms and case studies*. Springer Science & Business Media, 2009.

Kazuki Fujii, Yukito Tajima, Sakae Mizuki, Masaki Kawamura, Hinari Shimada, Taihei Shiotani, Koshiro Saito, Masanari Oi, Taishi Nakamura, Takumi Okamoto, et al. Rewriting pre-training data boosts llm performance in math and code. *arXiv preprint arXiv:2505.02881*, 2025.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from autoregressive models. *arXiv preprint arXiv:2410.17891*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pp. 181–195. Springer, 2022.

Jiafeng Guo, Changjiang Zhou, Ruqing Zhang, Jiangui Chen, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. Corpusbrain++: A continual generative pre-training framework for knowledge-intensive language tasks. *ACM Transactions on Information Systems*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Long P Hoang, Wenxuan Zhang, and Wei Lu. Difficulty–diversity collaborative filtering for data-efficient llm fine-tuning. In *ICLR*, 2026.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *arXiv preprint arXiv:2403.08763*, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 2021.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 2019.

Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *ICML*, 2021a.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *AAAI*, 2021b.

Krishnateja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh Iyer. Retrieve: Coreset selection for efficient and robust semi-supervised learning. In *NeurIPS*, 2021c.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017.

Anat Kleiman, Gintare Karolina Dziugaite, Jonathan Frankle, Sham Kakade, and Mansheej Paul. Soup to go: mitigating forgetting during continual learning with model averaging. *arXiv preprint arXiv:2501.05559*, 2025.

Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 2014.

Po-Nien Kung, Fan Yin, Di Wu, Kai-Wei Chang, and Nanyun Peng. Active instruction tuning: Improving cross-task generalization by training on prompt sensitive tasks. In *EMNLP*, 2023.

Chen-An Li and Hung-Yi Lee. Examining forgetting in continual pre-training of aligned large language models. *arXiv preprint arXiv:2401.03129*, 2024.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In *ACL*, 2024a.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. In *NAACL*, 2024b.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*, 2023.

Zikang Liu, Kun Zhou, Wayne Xin Zhao, Dawei Gao, Yaliang Li, and Ji-Rong Wen. Less is more: High-value data selection for visual instruction tuning. In *ACM MM*, 2025.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, et al. 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)*, 2024.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. In *EMNLP*, 2021.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*, 2023.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *ICML*, 2020.

National Library of Medicine. PMC Open Access Subset. `https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/`, 2003.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 1978.

Dang Nguyen, Zeman Li, Mohammadhossein Bateni, Vahab Mirrokni, Meisam Razaviyayn, and Baharan Mirzasoleiman. Synthetic text generation for training large language models via gradient matching. *arXiv preprint arXiv:2502.17607*, 2025.

Mahdi Nikdan, Vincent Cohen-Addad, Dan Alistarh, and Vahab Mirrokni. Efficient data selection at scale via influence distillation. *arXiv preprint arXiv:2505.19051*, 2025.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*. PMLR, 2022.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *ACL*, 2016.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. *arXiv preprint arXiv:2303.14186*, 2023.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *NeurIPS*, 2021.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. In *NeurIPS*, 2024.

Ru Peng, Kexin Yang, Yawen Zeng, Junyang Lin, Dayiheng Liu, and Junbo Zhao. Dataman: Data manager for pre-training large language models. *arXiv preprint arXiv:2502.19363*, 2025.

Haoran Que, Jiaheng Liu, Ge Zhang, Chenchen Zhang, Xingwei Qu, Yinghao Ma, Feiyu Duan, Zhiqi Bai, Jiakai Wang, Yuanxing Zhang, et al. D-cpt law: Domain-specific continual pre-training scaling law for large language models. In *NeurIPS*, 2024.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020.

Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987.

Karsten Roth, Vishaal Udandarao, Sebastian Dziadzio, Ameya Prabhu, Mehdi Cherti, Oriol Vinyals, Olivier Hénaff, Samuel Albanie, Matthias Bethge, and Zeynep Akata. A practitioner's guide to continual multimodal pretraining. *arXiv preprint arXiv:2408.14471*, 2024.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 2021.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 2025.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In *NeurIPS*, 2022.

Haoru Tan, Sitong Wu, Yanfeng Chen, Jun Xia, Ruobing Xie, Bin Xia, Xingwu Sun, and Xiaojuan Qi. Fast data mixture optimization via gradient descent. In *ICLR*, 2026.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

Yun-Da Tsai, Mingjie Liu, and Haoxing Ren. Code less, align more: Efficient llm fine-tuning for code generation with data pruning. *arXiv preprint arXiv:2407.05040*, 2024.

George Tsatsaronis, Michael Schroeder, Georgios Paliouras, Yannis Almirantis, Ion Androutsopoulos, Eric Gaussier, Patrick Gallinari, Thierry Artieres, Michael R Alvers, Matthias Zschunke, et al. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In *AAAI fall symposium: Information retrieval and knowledge discovery in biomedical text*. Arlington, VA: Citeseer, 2012.

Jiachen T Wang, Tong Wu, Dawn Song, Prateek Mittal, and Ruoxi Jia. Greats: Online selection of high-quality data for llm training in every iteration. In *NeurIPS*, 2024.

Shaobo Wang, Xiangqi Jin, Ziming Wang, Jize Wang, Jiajun Zhang, Kaixin Li, Zichen Wen, Zhong Li, Conghui He, Xuming Hu, et al. Data whisperer: Efficient data selection for task-specific llm fine-tuning via few-shot in-context learning. In *ACL*, 2025a.

Sheng Wang, Pengan Chen, Jingqi Zhou, Qintong Li, Jingwei Dong, Jiahui Gao, Boyang Xue, Jiyue Jiang, Lingpeng Kong, and Chuan Wu. Treesynth: Synthesizing diverse data from scratch via tree-guided subspace partitioning. *arXiv preprint arXiv:2503.17195*, 2025b.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

Max Welling. Herding dynamical weights to learn. In *ICML*, 2009.

Jacob White. Pubmed 2.0. *Medical reference services quarterly*, 2020.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *CVPR*, 2022.

Xindi Wu, Mengzhou Xia, Rulin Shao, Zhiwei Deng, Pang Wei Koh, and Olga Russakovsky. Icons: Influence consensus for vision-language data selection. *arXiv preprint arXiv:2501.00654*, 2024.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In *NeurIPS*, 2023a.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. In *NeurIPS*, 2023b.

Yong Xie, Karan Aggarwal, and Aitzaz Ahmad. Efficient continual pre-training for building domain specific large language models. In *Findings of ACL*, 2024.

Yang Xu, Yongqiang Yao, Yufan Huang, Mengnan Qi, Maoquan Wang, Bin Gu, and Neel Sundaresan. Rethinking the instruction quality: Lift is what you need. *arXiv preprint arXiv:2312.11508*, 2023.

Bang Yang, Yong Dai, Xuxin Cheng, Yaowei Li, Asif Raza, and Yuexian Zou. Embracing language inclusivity and diversity in clip through continual language learning. In *AAAI*, 2024.

Çağatay Yıldız, Nishaanth Kanna Ravichandran, Nitin Sharma, Matthias Bethge, and Beyza Ermis. Investigating continual pretraining in large language models: Insights and implications. *arXiv preprint arXiv:2402.17400*, 2024.

Zichun Yu, Spandan Das, and Chenyan Xiong. Mates: Model-aware data selection for efficient pretraining with data influence models. In *NeurIPS*, 2024.

Zichun Yu, Fei Peng, Jie Lei, Arnold Overwijk, Wen-tau Yih, and Chenyan Xiong. Group-level data selection for efficient pretraining. *arXiv preprint arXiv:2502.14709*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *ACL*, 2019.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024a.

Yifan Zhang, Yifan Luo, Yang Yuan, and Andrew C Yao. Autonomous data selection with language models for mathematical texts. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*, 2024b.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020.

Hongbo Zhao, Fei Zhu, Haiyang Guo, Meng Wang, Rundong Wang, Gaofeng Meng, and Zhaoxiang Zhang. Mllm-cl: Continual learning for multimodal large language models. *arXiv preprint arXiv:2506.05453*, 2025.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. In *NeurIPS*, 2023.

A Implementation Details

This appendix expands the practical pipeline in §3.5. We describe the concrete artifacts, numerical safeguards, and streaming data path used in all experiments. The emphasis is on reproducibility: the same LoRA adapter, tokenizer, Fisher estimate, and sample-level data stream must be used consistently across warmup, Fisher estimation, selection, and final CPT.

A.1 Warmup Adapter and Fisher Estimate

LoRA operating point. All selection signals are computed in the LoRA adapter subspace used by downstream CPT. For the TinyLlama runs, we attach LoRA modules to the query, key, value, and output attention projections, with rank $r = 128$ and LoRA scale 512. The adapter is first warmed up on target-domain text to produce the operating point $\theta^\star$. This step is not used as an evaluation result; it only makes the LoRA coordinates non-degenerate and provides a local point at which candidate gradients are meaningful.

The warmup checkpoint is treated as a single source of truth. Its adapter weights define $\theta^\star$, and its optimizer state is cached for the preconditioner in §A.3. In practice, every later stage loads this exact checkpoint rather than re-initializing a fresh adapter. This is important because LoRA's zero-product initialization can make part of the adapter-gradient signal vanish at the raw pretrained checkpoint, which in turn produces an ill-conditioned Fisher estimate.

Diagonal Fisher estimation. We estimate the diagonal Fisher in the same LoRA subspace and at the same operating point $\theta^\star$. The base model is frozen, dropout is disabled, and only LoRA parameters require gradients. For each proxy document z_j from the held-out reference distribution, we compute the next-token loss and accumulate squared LoRA gradients in fp32:

$$\Lambda_i \;=\; \frac{1}{M}\sum_{j=1}^{M}\left(\frac{\partial \ell(z_j;\theta^\star)}{\partial \theta_i}\right)^2 . \tag{6}$$

The proxy set is disjoint from the candidate stream and the finally selected CPT set. In distributed runs, each worker accumulates its local squared-gradient sums and the totals are averaged across workers before saving Λ. We also record basic health statistics of Λ, such as its minimum, median, maximum, and near-zero fraction. A large near-zero fraction indicates that the Fisher was likely computed from an adapter that does not match the warmup checkpoint used for selection.

A.2 Candidate Stream and Sample-level Gradients

Preserving document boundaries. The selector consumes the candidate corpus as an iterable stream. During selection, packing is disabled so that each stream element corresponds to one candidate document and yields one per-sample gradient. We use the same tokenizer family as the warmup model and truncate candidates to the selection cutoff length. Each incoming candidate receives a globally unique, monotonically increasing stream index; this index is used for replaying the calibration window, tracking accepted candidates across sieves, and reconstructing the final selected set.

Perplexity gate before backward. For a micro-batch of stream candidates, we right-pad to the longest sequence in the micro-batch and compute the standard causal-LM next-token loss for each candidate separately. The corresponding perplexity is obtained from this per-sample loss. Candidates outside a broad interval $[p_{\mathrm{lo}}, p_{\mathrm{hi}}]$ are discarded before any backward pass is run. This gate removes documents that are either already too easy for $\theta^\star$ or so far out-of-distribution that their gradients are dominated by noise, while avoiding an additional forward pass.

For the remaining candidates, we compute sample-level LoRA gradients from the retained computation graph. When more than one candidate survives in the micro-batch, all but the last backward call retain the graph; the last one releases it. The resulting LoRA gradients are flattened in a fixed parameter order and are the only gradients passed to the Fisher reweighting and projection stages.

A.3 Conditioned Fisher Signatures

Fixed Adam preconditioner. Raw LoRA-gradient coordinates have different numerical scales. Before Fisher reweighting, we apply a one-shot Adam-style preconditioner using the warmup

optimizer moments (m, v) saved with $\theta^\star$:

$$\tilde{m} = 0.9\, m + 0.1\, g(x), \qquad \tilde{v} = 0.999\, v + 0.001\, g(x)^2, \qquad \hat{g}(x) = \tilde{m} \oslash (\sqrt{\tilde{v}} + \epsilon_{\mathrm{Adam}}). \quad (7)$$

The moments are not updated as the stream progresses. They define a fixed local rescaling at $\theta^\star$, so the ordering of candidates is not affected by earlier accept/reject decisions in the stream.

Fisher floor. The frontier view divides by $\sqrt{\Lambda}$, so near-zero Fisher entries can otherwise dominate the signature. We therefore form

$$\Lambda_{\varepsilon,i} = \max\{\Lambda_i, \varepsilon_\Lambda\}, \quad (8)$$

where ε_Λ is set to the sample median of Λ with a small absolute lower bound. This median floor suppresses unstable near-zero coordinates while preserving the relative contrast between high- and low-Fisher directions. The two signatures used by the selector are then

$$g_{\mathrm{anc}}(x) = \sqrt{\Lambda} \odot \hat{g}(x), \qquad g_{\mathrm{fr}}(x) = \hat{g}(x) \oslash \sqrt{\Lambda_\varepsilon}. \quad (9)$$

Projection without normalization. After Fisher reweighting, both signatures are projected with a fixed Rademacher projection as in TRAK (Park et al., 2023). The same projection seed and parameter order are used for anchor and frontier signatures, and the projection is applied after Fisher reweighting rather than before it. We do not ℓ_2-normalize the projected vectors. Their norms are part of the singleton signal in equation 3; normalizing them would collapse the first-step gains of many candidates and make early stream order overly influential.

B Benchmark Decontamination

Our target-domain candidate pool (PMC full text and PubMed abstracts) comes from the same literature from which several biomedical benchmarks are built. This appendix lists where overlap can arise, describes the decontamination applied during data preparation, and summarizes the evidence that residual overlap does not drive the reported gains.

Sources of overlap. Each PubMedQA instance is built from one PubMed article: the question comes from the title, the context is the abstract without its conclusion, and the conclusion is the long answer that determines the yes/no/maybe label (Jin et al., 2019). BioASQ questions are answered from gold PubMed documents and snippets (Tsatsaronis et al., 2012). These abstracts, or the PMC full texts of the same articles, can therefore appear in our pool. MedQA-USMLE and MedMCQA (Jin et al., 2021; Pal et al., 2022) are derived from medical licensing and entrance exams rather than from PubMed articles, so they have no lexical path to the candidate stream through their source documents.

Decontamination protocol. Three filters are applied, in order, while preparing the PMC/PubMed candidate pool (Table 5), and all experiments reported in this paper, for our method and for every baseline, take their target-domain data from the filtered pool. (1) *Source-ID blocklist*: every PMC or PubMed record whose PMID or DOI appears in the source lists of PubMedQA (the PQA-L, PQA-U, and PQA-A subsets, 273.5k PMIDs) or of BioASQ is dropped. (2) *n-gram overlap*: every remaining candidate that shares a 13-gram with any evaluation instance (question, context, and answer) is dropped, following the 13-gram criterion of Brown et al. (2020). (3) *Near-duplicate removal*: candidates are deduplicated with MinHash LSH, dropping pairs whose Jaccard similarity on 5-gram shingles is at least 0.8. The general-domain reference shard used to estimate Λ is held out and disjoint from both the candidate stream and every evaluation set.

C Comparison with data-selection methods.

We further compare against two pretraining-stage selectors, DSIR (Xie et al., 2023b) and DoReMi (Xie et al., 2023a), and two SFT-stage selectors that also maximize a $\log\det$ of Fisher information, FisherSFT (Deb et al., 2025) and SPICE (Chang et al., 2026). None of them was designed for forgetting-aware CPT, so we port each as faithfully as possible (detailed in C.1) and give all of them the same candidate pool, LoRA configuration, CPT recipe, and token budget as ours. All baselines share the backbone (TinyLlama-1.1B), the LoRA configuration (rank 128, scale 512, query/key/value/output projections), the optimizer and schedule, the cutoff length, the token-budget accounting (4B tokens), the lm-evaluation-harness prompts and seeds, and three seeds per cell; only the selection rule differs.

Table 5: **Decontamination filters applied to the PMC/PubMed candidate pool.** The three counts are disjoint and add up to the total; percentages are relative to the documents of the unfiltered pool.

Filter	Rule	Docs removed	% of pool
1. Source-ID	Drop every PMC/PubMed record whose PMID or DOI is in the source lists of PubMedQA (PQA-L/U/A, 273.5k PMIDs) or BioASQ	298,142	0.99
2. n-gram overlap	Drop any candidate sharing a 13-gram with any evaluation instance (question + context + answer)	412,905	1.37
3. Near-duplicates	MinHash LSH; drop pairs with Jaccard similarity ≥ 0.8 on 5-gram shingles	96,318	0.32
Total		**807,365**	**2.68**

Running the four baselines over the entire corpus would be computationally infeasible, so each of them receives the same enlarged candidate pool that our selector uses (decontaminated as in App. B). As Table 6 shows, all four methods improve adaptation over Random, with SPICE the strongest ($A = +3.6$), but all of them forget considerably more than ours ($\Phi \geq 1.6$ vs. 0.4), and ours is best on every metric.

C.1 Baseline setup

DSIR resamples raw documents with importance weights that match the hashed n-gram statistics of a target sample, without using the model; it has been applied to domain-specific continued pretraining but has no retention term. We use held-out medical text as the target sample and the candidate pool as the raw distribution, and resample to the shared token budget.

DoReMi learns domain weights with Group DRO: a small proxy model is trained to minimize the worst-case excess loss of any domain relative to a reference, and the averaged domain weights define the training mixture. CPT supplies neither the proxy nor the reference. We instantiate DoReMi with the pretrained TinyLlama-1.1B as the proxy, held-out medical text as the Group-DRO reference, and the four corpus sources (PMC, PubMed, FineWeb, and English Llama3-SynE) as the domains.

FisherSFT greedily maximizes $\log\det\left(V + \sum_{i\in S}\sum_j x_{i,j}x_{i,j}^\top\right)$, where $x_{i,j}$ is the pre-logit embedding of token j of example i; this log-determinant lower-bounds that of the Hessian of the log-likelihood under a linearization of the model at its last layer. Our port builds the design matrix from one mean-pooled final hidden state per candidate document, so each document contributes a rank-one update.

SPICE greedily maximizes $\log\det(I + aF_S)$, where $F_S = \sum_{i\in S} g_i g_i^\top$ is the empirical Fisher of the selected gradients and a is a scale (not our α), minus a conflict penalty $\lambda \max\{0, -\cos(g_x, \bar{g}_S)\}$ against the mean selected gradient $\bar{g}_S$. The original method computes gradients of instruction–response pairs with a smaller proxy model and uses a periodic selection schedule, with optional adaptive early stopping, to keep its greedy search affordable. In our port, F_S is built from the same per-candidate LoRA gradients and the same TRAK projection ($d = 8192$) that our selector uses, so the two methods share the gradient representation.

D Additional experiment results

D.1 End-to-End Compute Cost

The token budgets in §4.4 and Table 4 count *CPT training tokens*. They are counted inclusive of replay tokens, so Replay at 10B tokens with 30% mixing trains on 7B medical and 3B general-domain tokens, whereas Ours at 1B trains on 1B selected medical tokens. The statement that Ours at 1B tokens outperforms Replay at 10B tokens is therefore a statement about training data. It does not by itself establish lower total compute, because our selector must first process candidates that it later

Table 6: **Comparison with data-selection methods** (TinyLlama-1.1B, 4B tokens). All methods share the candidate pool, LoRA configuration, CPT recipe, and evaluation; only the selection rule differs. †SFT-stage methods ported to CPT. ‡Domain-level method whose domains are the four corpus sources (App. C). **Bold**/underline: best/second-best within this table.

Method	Stage	Med-PPL↓	PubMedQA↑	BioASQ↑	$A\uparrow$	$\Phi\downarrow$
DSIR	Pretraining	7.2	66.9	75.4	+2.4	2.9
DoReMi‡	Pretraining	7.5	65.8	74.1	+1.2	<u>1.6</u>
FisherSFT†	SFT	6.9	67.4	76.1	+3.0	3.3
SPICE†	SFT	<u>6.7</u>	<u>67.9</u>	<u>76.7</u>	<u>+3.6</u>	2.6
Ours	CPT	**6.2**	**70.0**	**78.5**	**+5.5**	**0.4**

Table 7: **End-to-end compute of our pipeline** on TinyLlama-1.1B at the default 4B-token budget, measured on 4 nodes with 4× A100-40G GPUs each. GPU-hours are summed over all GPUs; the share is the fraction of total GPU-hours.

Stage	EFLOP	GPU-h	Share
(i) Warmup adapter $\theta^\star$	1.30	5.9	3.2%
(ii) Diagonal Fisher Λ	1.27	24.7	13.4%
(iii) Perplexity gate	3.06	13.7	7.4%
(iv) Per-sample LoRA gradients	3.36	26.7	14.5%
(v) TRAK projection	1.88	4.4	2.4%
(vi) Sieve-Streaming selection	< 0.01	1.1	0.6%
(vii) CPT	24.6	107.5	58.4%
Total	**35.5**	**184.0**	

discards, and the per-candidate cost of Table 11 measures only the Sieve-Streaming step on cached signatures, excluding the generation of those signatures. This appendix accounts for the full pipeline.

Stages. The pipeline of §3.5 and App. A consists of seven stages: (i) the warmup that produces the LoRA operating point $\theta^\star$ and the cached Adam moments; (ii) estimation of the diagonal Fisher Λ from per-document squared LoRA gradients on the reference documents; (iii) the perplexity gate, a forward pass on every scanned candidate; (iv) per-sample LoRA gradients, computed only for candidates that pass the gate; (v) the TRAK projection of the anchor and frontier signatures to $d = 8192$ dimensions; (vi) Sieve-Streaming selection with Sherman–Morrison updates; and (vii) CPT on the selected tokens, with the same LoRA recipe as every baseline.

Where the cost goes. Training, not selection, is the largest single cost: CPT accounts for 58.4% of the GPU-hours at the 4B budget, and stages (i)–(vi) together for the remaining 76.5 GPU-hours. Within selection, the two per-document gradient stages dominate: the Fisher estimate (ii) and the per-sample candidate gradients (iv) take 24.7 and 26.7 GPU-hours. They also run at much lower hardware utilization than CPT: stage (ii) performs 1.27 EFLOP in 24.7 GPU-hours, whereas CPT performs 24.6 EFLOP in 107.5 GPU-hours, likely because these stages run unpacked and compute one backward pass per document (App. A.2). The Sieve-Streaming step itself, the only stage whose cost is measured in Table 11, is negligible (0.6%), and the perplexity gate keeps the backward pass of stage (iv) off the rejected candidates.

Comparison with Replay. Table 8 compares the full cost of our pipeline with Replay at 10B tokens, which spends 62.1 EFLOP and 271.4 GPU-hours. At the 1B budget of the token-efficiency comparison, our pipeline costs 10.9 EFLOP and 69.3 GPU-hours end to end, including selection. It thus uses 5.7× fewer FLOPs and 3.9× fewer GPU-hours than Replay at 10B, while outperforming it on both adaptation ($A = +3.0$ vs. -2.0) and forgetting ($\Phi = 0.6$ vs. 0.8). At the default 4B budget, the pipeline remains cheaper than Replay at 10B (1.7× fewer FLOPs and 1.5× fewer GPU-hours) and improves A by 7.5 points. The GPU-hour ratios are smaller than the FLOP ratios because the per-document gradient stages (ii) and (iv) run at lower utilization than CPT. The token-efficiency advantage of §4.4 therefore carries over to end-to-end compute, but by a smaller factor (3.9× in GPU-hours at 1B) than the 10× ratio of training tokens.

Table 8: **End-to-end cost against Replay** (TinyLlama-1.1B). Totals include every selection stage of Table 7; Replay performs no selection. The ratio columns give Replay-10B's cost divided by the method's cost. A and Φ are from Tables 1 and 4.

Method	CPT tokens	EFLOP	GPU-h	FLOP ratio	GPU-h ratio	$A \uparrow$	$\Phi \downarrow$
Replay	10B	62.1	271.4	1.0×	1.0×	−2.0	0.8
Ours	4B	35.5	184.0	1.7×	1.5×	+5.5	0.4
Ours	1B	10.9	69.3	5.7×	3.9×	+3.0	0.6

Table 9: **Fisher floor** ε_Λ (TinyLlama-1.1B, 4B tokens). $p_q(\Lambda)$ is the q-th percentile of the Fisher entries; "coords clipped" is the fraction of coordinates raised to the floor; the inflation column reports the largest inflation of $\|g_{\text{fr}}\|$ relative to the constant-Λ row. * marks the default.

ε_Λ	Coords clipped	Max $\|g_{\text{fr}}\|$ inflation	$A \uparrow$	$\Phi \downarrow$
10^{-8} (no effective floor)	0.0%	$3.1 \times 10^4\times$	+3.0	4.5
$p_5(\Lambda)$	5%	210×	+4.1	1.9
$p_{25}(\Lambda)$	25%	21×	+5.3	0.5
$p_{50}(\Lambda)$ = median*	50%	6.4×	+5.5	0.4
$p_{75}(\Lambda)$	75%	2.3×	+5.2	0.4
p_{100} (constant Λ; frontier on $\hat{g}$)	100%	1.0×	+3.4	1.2

D.2 Additional ablations

Fisher Floor. The frontier view divides by $\sqrt{\Lambda_\varepsilon}$ with $\Lambda_{\varepsilon,i} = \max\{\Lambda_i, \varepsilon_\Lambda\}$, so coordinate i is amplified by $\Lambda_{\varepsilon,i}^{-1/2}$. This is where the frontier branch is numerically most fragile. If the floor is too low, the smallest entries of Λ, which are also the least reliably estimated, receive the largest weights, and a few amplified coordinates can dominate $\|g_{\text{fr}}(x)\|_2$ and the frontier scatter matrix. The floor caps the amplification at $\varepsilon_\Lambda^{-1/2}$. If the floor is too high, every clamped coordinate receives the same weight, which erases the contrast between low- and very-low-Fisher directions; in the limit $\varepsilon_\Lambda \geq \max_i \Lambda_i$, $g_{\text{fr}}(x) = \hat{g}(x)/\sqrt{\varepsilon_\Lambda}$ and the frontier term reduces to an identity-geometry $\log\det$ on the raw preconditioned gradient. The floor does not enter the anchor view $g_{\text{anc}} = \sqrt{\Lambda} \odot \hat{g}$.

Table 9 sweeps the floor over percentiles $p_q(\Lambda)$ of the Fisher entries, together with the fixed floor 10^{-8}, which clips no coordinate and is therefore no effective floor. Without an effective floor, the frontier norm is inflated by up to 3.1×10^4 and the method degrades to $A = +3.0$, $\Phi = 4.5$, consistent with a few amplified coordinates dominating the frontier term. A 5th-percentile floor already removes much of this damage (Φ drops from 4.5 to 1.9), and every floor between the 25th and 75th percentiles stays within 0.3 of the default in A and within 0.1 in Φ, so the method is not sensitive to the exact choice of the median. At the other extreme, a constant Λ (p_{100}) removes the Fisher reweighting from the frontier view and drops A to $+3.4$ while raising Φ to 1.2. This row also rescales the whole frontier signature by $(\max_i \Lambda_i)^{-1/2}$, which by the rescaling equivalence above changes the effective γ, so it mixes the loss of Fisher contrast with a change of scale; it nevertheless suggests that the $\Lambda^{-1/2}$ reweighting itself, not only the clipping, contributes to the gain.

Hyperparameter sensitivity. Table 10 shows that the default hyperparameters are robust and near-optimal across all four sweeps. The default setting attains the strongest adaptation gain while keeping forgetting low, and nearby settings preserve most of the gain. Performance degrades mainly under deliberately weak configurations: removing warmup ($T = 0$) collapses adaptation, using too small a projection dimension increases forgetting, and small per-term scales reduce the benefit of the $\log\det$ objective.

Streaming approximation quality and per-sample cost. The streaming diagnostic (Table11) supports the practical scalability of our selector. Per-candidate processing remains essentially constant as the candidate stream grows, whereas exact offline greedy is useful only as a small-stream diagnostic because it must cache features and repeatedly rescan the pool. In that diagnostic regime,

Table 10: Hyperparameter sensitivity on TinyLlama-1.1B under the 4B-token default setting. Each block sweeps one hyperparameter holding the rest at their defaults (marked *). Last column reports the best TinyLlama baseline from Table 1 under the same $A - \Phi$ metric.

Sweep	Value	$A \uparrow$	$\Phi \downarrow$
Frontier weight α	0.5	+4.8	0.7
	1.0*	**+5.5**	**0.4**
	2.0	+5.4	0.5
	4.0	+4.6	0.4
Per-term scale $\beta = \gamma$	5	+3.8	1.0
	10	+4.9	0.6
	20*	**+5.5**	0.4
	50	+5.4	0.4
Warmup steps T	0	+0.6	0.5
	100	+4.7	0.5
	200*	**+5.5**	0.4
	500	+5.5	0.4
TRAK dim d	1024	+4.0	1.2
	4096	+5.3	0.5
	8192*	**+5.5**	0.4
	16384	+5.5	0.4

Table 11: Small-stream Sieve-Streaming diagnostic on cached projected gradients ($|\mathcal{D}_{\text{stream}}| = 10^4$, $B = 1{,}000$, TinyLlama-1.1B). Offline greedy is used only to normalize the projected objective in this tractable regime; no downstream CPT is run for this diagnostic. $|\mathcal{T}|$ is the number of sieve thresholds.

ε	$\lvert\mathcal{T}\rvert$	$f(S_{\text{ours}})/f(S_{\text{greedy}})$	per-candidate wall-clock (ms)
0.05	94	99.1%	7.8
0.1	48	98.5%	4.0
0.2	25	96.8%	2.1
0.4	14	94.6%	1.2
Offline greedy (diagnostic reference)		100%	720

Sieve-Streaming closely tracks the greedy objective, indicating that the approximation loss is small in practice while preserving the deployable streaming path used in the full CPT runs.

E Limitations

Our method is designed for scalable CPT data selection, so its approximations are deliberately chosen to preserve the main Fisher geometry while remaining deployable for billion-parameter LMs. In particular, we estimate curvature with a diagonal Fisher in the LoRA adapter subspace, evaluate gradients at a short warmup operating point $\theta^\star$, and preserve the subset-level log-determinant geometry through random projection rather than full-parameter covariance matrices. These choices are part of the scalable formulation and are examined through the ablations and streaming diagnostics in §4.3 and §4.4; they also leave natural extensions to richer block-diagonal or low-rank curvature models, full-parameter adaptation, and adaptive warmup schedules. Our experiments focus on medical-domain CPT for two LLM backbones, so the empirical claims are scoped to this setting, with other domains, instruction-tuning regimes, and multimodal streams left for future validation.

F Extended Related Work

This appendix expands the compressed discussion of §2 with the descriptions and method-level distinctions that were elided for space. All citations referenced here also appear in the main related work.

Continual pre-training of large language models. Continual pre-training (CPT) adapts a pretrained LLM to a new domain, language, or temporal slice without retraining from scratch, and the dominant

concern is catastrophic forgetting of capabilities acquired during the original run. A first line of work stabilizes the optimizer itself: a combination of learning-rate re-warming, re-decay, and replay of a small fraction of prior data is sufficient to handle weak (English→English) and strong (English→German) distribution shifts (Ibrahim et al., 2024), and follow-up work further layers topic-based mixture reweighting and a perplexity-based curriculum on top (Chen et al., 2025a). Domain-specialized CPT has been studied in finance (Xie et al., 2024) and in mathematics and coding (Fujii et al., 2025), and scaling-law analyses of the domain/general trade-off are given by the D-CPT law (Que et al., 2024).

Another thread explicitly analyzes forgetting during CPT of aligned or instruction-tuned models (Li & Lee, 2024; Yıldız et al., 2024; Shi et al., 2025), and several methods mitigate it by routing or model-merging rather than by data: distribution-specialized experts (Chen et al., 2023c), language-inclusive CLIP fine-tuning (Yang et al., 2024), task arithmetic for selective forgetting (Ilharco et al., 2022), WiSE-FT-style robust fine-tuning (Wortsman et al., 2022), and the Soup-to-Go model-averaging recipe (Kleiman et al., 2025). CPT has also been extended to knowledge-intensive generative models (Guo et al., 2025), to multimodal LLMs (Zhao et al., 2025; Roth et al., 2024), and to architecture-level adaptation from autoregressive to diffusion language models (Gong et al., 2024).

Our contribution inverts the role of the Fisher relative to EWC. EWC (Kirkpatrick et al., 2017) uses the diagonal Fisher as a *regularizer* that pins parameters near their pretrained values after the data is fixed. We use the same Fisher upstream of optimization, to decide which tokens enter the CPT stream, so adaptation can proceed with an unconstrained optimizer and no auxiliary penalty.

Data selection for language-model training. Selecting training data for LMs has been investigated at every training stage. For from-scratch pre-training, DoReMi optimizes domain mixtures via group DRO on a proxy model (Xie et al., 2023a); DSIR reframes the choice as importance resampling against a target distribution (Xie et al., 2023b); and a broad family of scorers learns to rank candidates, including data-influence models (MATES (Yu et al., 2024)), group-level selection (Yu et al., 2025), quality classifiers (DataMan (Peng et al., 2025)), meta-learned raters (DataRater (Calian et al., 2025)), differentiable mixture weights (Tan et al., 2026), relative scoring via Bradley–Terry comparisons (Chen et al., 2025b), sensitivity-aware greedy selection (GREATS (Wang et al., 2024)), autonomous LM-based selection for mathematical text (Zhang et al., 2024b), and large-scale data pruning at pre-training scale (Marion et al., 2023; Longpre et al., 2024; Du et al., 2022).

A parallel thread targets *instruction tuning*, where strong results are obtained with dramatically smaller curated sets: LIMA and its "0.5% is enough" successor (Zhou et al., 2023; Chen et al., 2023a), AI-scored filtering in AlpaGasus (Chen et al., 2023b), alignment-quality studies (Liu et al., 2023), LIFT (Xu et al., 2023), unsupervised core-set selection DEFT (Das & Khetan, 2023), code-oriented pruning (Tsai et al., 2024), visual instruction-tuning selectors (Liu et al., 2025; Wu et al., 2024; Wang et al., 2025a), influence distillation (Nikdan et al., 2025), active and self-guided schemes (Kung et al., 2023; Li et al., 2024b), task-aware distribution matching (Cheng et al., 2026), activation-stability pruning (Alam et al., 2026), difficulty–diversity collaborative filtering (Hoang et al., 2026), tree-structured synthesis (Wang et al., 2025b), and a growing list of IT-focused selectors (Li et al., 2024a). More broadly, loss- and difficulty-based pruning has a long pedigree in classification and NLU (Sorscher et al., 2022; Paul et al., 2021; Attendu & Corbeil, 2023; Nguyen et al., 2025).

Our selector differs from these methods on two axes. First, it operates in the post-pretraining regime where there is prior competence to protect, unlike pre-training selectors that assume a *tabula rasa*, and outside the small-budget instruction-tuning regime targeted by LIMA-style filters. Second, it decomposes a sample's expected contribution into an anchor component aligned with high-Fisher directions of established competence and a frontier component aligned with low-Fisher directions where the pretrained loss is flat. This curvature-weighted decomposition is invisible to loss, perplexity, and feature-similarity surrogates, which integrate over directions without weighting them by Fisher mass.

Coreset selection and submodular data subset methods. Our global selection formulation builds on the coreset literature surveyed in DeepCore (Guo et al., 2022). Geometry-based methods retain representative points via herding (Welling, 2009) or k-center-greedy coverage (Farahani & Hekmatfar, 2009; Sener & Savarese, 2017); uncertainty- and loss-based methods use forgetting events (Toneva et al., 2018), gradient norms and error norms (GraNd, EL2N) (Paul et al., 2021), or sensitivity-based importance sampling; decision-boundary methods exploit adversarial deep-fool margins and

contrastive active learning (Ducoffe & Precioso, 2018; Margatina et al., 2021); gradient-matching methods greedily select subsets whose aggregate gradient tracks the full training gradient (CRAIG (Mirzasoleiman et al., 2020), GRAD-MATCH (Killamsetty et al., 2021a)) and give OMP-based submodular approximation guarantees; and bilevel methods treat the subset as an inner optimization variable (GLISTER (Killamsetty et al., 2021b), RETRIEVE (Killamsetty et al., 2021c)). Data pruning at neural-scaling-law scale (Sorscher et al., 2022) established that well-chosen subsets can beat power-law scaling, and the dataset-distillation line (Wang et al., 2018; Zhao et al., 2020) condenses rather than selects. We also draw inspiration from simple local-rule emergent coordination (Reynolds, 1987).

Our setting and utility differ from these methodd, where the per-candidate utility is derived from the pretrained model's Fisher curvature rather than from gradient magnitude, feature geometry, or loss trajectories, and the resulting constant-factor approximation has per-sample cost independent of the candidate-pool size.