Online Data Selection for Instruction Tuning via Gaussian Processes

Jun Wang **Quoc Phong Nguyen** **Julien Monteil** **Vu Nguyen**
Amazon
{junwc, qphong, jul, vutngn}@amazon.com

Abstract

With Large Language Model (LLM) pre-training and fine-tuning shifting its focus from data volume to data quality, quality data selection has emerged as a critical research topic. Existing online data selection methods for LLM training are typically "batch-constrained", limiting optimization to local utility within random batches. To overcome this, we propose **GAIA** (**G**lobal **A**daptive **I**nstruction tuning via G**A**ussian processes), a framework that formulates data valuation as a global estimation process. GAIA employs Gaussian Process regression to model continuous utility manifolds across the semantic space, utilizing an adaptive strategy fusion mechanism to dynamically prioritize high-utility samples. By casting the strategy-posterior update as an instance of the classical fixed-share Hedge framework for tracking the best expert, we inherit a dynamic-regret guarantee for tracking the best strategy within the sampled strategy set, and we verify empirically that this set remains adequate throughout training. Across three instruction-tuning benchmarks and four backbones, **GAIA** converges faster than state-of-the-art online selection methods such as `GREATS`, with downstream gains in several settings, at negligible computational overhead.

1 Introduction

Modern Large Language Models (LLMs) owe their remarkable success to large-scale pretraining on massive, heterogeneous datasets [1]. These models are typically adapted to specialized downstream tasks—ranging from dialogue generation to multimodal reasoning—via fine-tuning or instruction alignment [2, 3]. Despite their empirical prowess, LLMs remain susceptible to factual hallucinations, spurious correlations, and ingrained social biases [4]. These vulnerabilities are often traceable to noisy, redundant, or unrepresentative training samples. Consequently, a fundamental challenge in LLM alignment is *data valuation*: identifying which samples meaningfully enhance model generalization and which act as detrimental noise.

Data valuation frameworks, such as influence functions and Shapley value-based methods, provide principled mechanisms for estimating the contribution of individual data points to model performance [5, 6]. While theoretically robust for tasks like mislabeled data detection and bias diagnosis, these classical approaches often struggle with the practical scalability and high-dimensional gradients inherent in modern LLM fine-tuning pipelines.

To address these scalability constraints, recent works such as `GREATS` [7] have pioneered online data selection. However, we observe that existing methods are fundamentally hindered by a **batch-constrained paradigm**. Specifically, these approaches dynamically evaluate data quality to select an optimal *mini-batch* from a larger *candidate pool*. Crucially, because this candidate pool is still constructed via uniform random sampling, the selection mechanism can only achieve *local optimality*. This design introduces a critical stochastic vulnerability: an unfavorable draw may yield a candidate pool saturated with low-quality samples, forcing the model to choose the "best of the worst." Thus,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

the efficacy of online selection is inherently upper-bounded by the quality of the stochastic candidate pool, making it limited by the very randomness it seeks to mitigate.

In this paper, we propose **GAIA** (**G**lobal **A**daptive **I**nstruction Tuning via G**A**ussian Processes), a novel framework that transcends the batch-constrained limitation by transforming data valuation into a proactive global guidance mechanism. Instead of treating quality scores as passive filters within a batch, **GAIA** leverages them to steer the construction of the candidate pools themselves. Our contributions are three-fold:

- We model the latent quality function over the entire training corpus using a **Gaussian Process (GP)**, enabling principled generalization of valuation signals to unseen examples throughout the training process.
- To ensure computational feasibility at scale, we discretize the GP into a finite set of **strategies** (function samples from the GP prior) and maintain a posterior distribution over them using a fixed-share Hedge update with a top-k likelihood. Casting this update as an instance of the classical framework for tracking the best expert, we inherit a dynamic-regret guarantee for tracking the best strategy *within* this set, which characterizes **GAIA**'s robustness to the non-stationarity of quality scores as model parameters evolve during training; we further verify empirically that the set itself remains adequate throughout training.
- We show that **GAIA** is complementary to existing data selection methods, which can be applied as an optional refinement stage on the globally sampled batch, and we characterize when such refinement is beneficial.

By bridging theoretical data valuation with practical scalability, **GAIA** provides a principled, computationally efficient framework that improves the convergence and data efficiency of online selection, offering a robust alternative to batch-constrained data selection.

2 Related Works

Offline Data Valuation. Most offline methods estimate data importance post-training or via repeated retraining [8–11]. While recent works scale Shapley-value [12] and influence-function [13] approaches to LLMs, high computational costs remain. Related fields like dataset distillation and coreset construction [14, 15] similarly rely on expensive bilevel optimization. Furthermore, influence-function techniques [6, 16] assume local linearity around fixed parameters, failing to capture the evolving data utility in the highly non-convex landscapes of LLMs.

The static nature of offline selection—performing pruning only once prior to training—often yields suboptimal results as data informativeness shifts across training stages [17]. Additionally, many offline pipelines require heavy preprocessing or auxiliary models [18], increasing system complexity. These limitations necessitate adaptive strategies that evolve alongside the model.

Online Data Valuation. Online methods perform adaptive selection at the batch level, identifying informative examples based on the model's current state. Unlike static approaches [19, 20], this dynamic strategy allows data value to evolve with learning progress, capturing stage-specific relevance for faster convergence and better generalization. By operating on small batches, online selection also reduces preprocessing overhead; for instance, [7] uses a greedy Taylor-expansion-based batch optimization.

Recent lightweight proxies use signals like loss, gradient norms, or forgetting events [21, 22]. While efficient, these heuristics are often noisy, hyperparameter-sensitive, and lack a unified, task-aware definition of value, making it difficult to align selection with downstream objectives [23]. More recent methods additionally promote diversity within the selected subset [24, 25]. All of these methods, however, select within a candidate batch drawn independently of the selection criterion; **GAIA** instead constructs the candidate batch globally, so any of them can serve as its optional refinement step.

3 Preliminaries

We use $\mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{v}}$ for the training and validation sets, respectively, $\mathbf{z} = (\mathbf{x}, y)$ for examples, and $\boldsymbol{\theta}_t$ for model parameters at iteration t. When the source is relevant, we write $\mathbf{z}^{(\text{tr})} \in \mathcal{D}_{\text{tr}}$ and $\mathbf{z}^{(\text{v})} \in \mathcal{D}_{\text{v}}$.

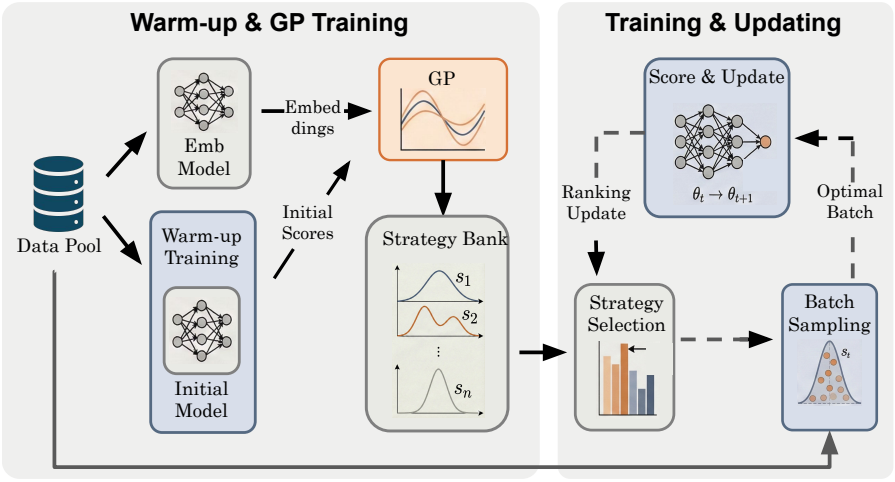


Figure 1: **Workflow.** Our framework operates in two distinct phases to decouple valuation from training: (1) **Warm-up & GP Training**: A lightweight **Emb**eding **Model** maps the raw data pool into a semantic embedding space. Concurrently, a **warm-up training** phase on a small dataset set collects initial utility scores. These scores and embeddings are used to fit the Gaussian Process (GP) model, establishing an initial global utility manifold. (2) **Iterative Training & Adaptive Updating**: In the main loop, an **adaptive strategy** mechanism selects the optimal strategy from the Strategy Bank based on real-time feedback. This selected strategy guides the sampling of high-utility batches from the global pool for model updates. The resulting feedback is used to dynamically update the strategy selection probabilities, ensuring robust and efficient data selection.

Training minimizes the empirical loss $\mathcal{L}(\mathcal{D}_{\text{tr}};\boldsymbol{\theta}) \triangleq 1/|\mathcal{D}_{\text{tr}}| \sum_{\mathbf{z}\in\mathcal{D}_{\text{tr}}} l(\mathbf{z};\boldsymbol{\theta})$ via SGD: at iteration t, a mini-batch $\mathcal{B}_t \subset \mathcal{D}_{\text{tr}}$ is drawn and $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \sum_{\mathbf{z}\in\mathcal{B}_t} \mathbf{g}_t(\mathbf{z})$, where $\eta_t > 0$ is the learning rate and $\mathbf{g}_t(\mathbf{z}) \triangleq \nabla l(\mathbf{z};\boldsymbol{\theta}_t)$. Full notation is summarized in Table 1 (Appendix A.1).

This work addresses *online data selection*: adaptively choosing $\mathcal{B}_t$ at each iteration. We first review score-based online selection methods, then motivate the need for a surrogate model over training-set quality scores. In Section 4, we instantiate this surrogate as a Gaussian process, which enables principled generalization of observed quality scores to unseen examples. A self-contained background, including the kernel definition and the cost of exact inference, is provided in Appendix A.2.

Given a randomly sampled candidate batch $\tilde{\mathcal{B}}_t \subset \mathcal{D}_{\text{tr}}$ at iteration t, online data selection methods select a refined training batch $\mathcal{B}_t \subseteq \tilde{\mathcal{B}}_t$ (used for the SGD update) by evaluating a quality score $q(\mathbf{z})$ for each $\mathbf{z} \in \tilde{\mathcal{B}}_t$. Proposed scores include gradient alignment with validation data [7], gradient norms, per-sample loss, and embedding similarity, and recent methods additionally account for diversity within the selected batch [24, 25] (see Section 5); our framework is agnostic to this choice. As a representative example, `GREATS` [7] uses $q(\mathbf{z}) \triangleq \mathbf{g}_t(\mathbf{z}) \cdot \mathbf{g}_t(\mathbf{z}^{(\text{v})})$ and greedily selects $\mathcal{B}_t$ to balance quality and diversity, computed efficiently via the ghost inner-product technique.

However, such methods sample $\tilde{\mathcal{B}}_t$ uniformly at random, ignoring historical quality information. This motivates our central question: *can online data selection be improved by using observed quality scores to guide the sampling of $\tilde{\mathcal{B}}_t$?*

Remark 1. If $\tilde{\mathcal{B}}_t$ is already sampled from high-quality regions of $\mathcal{D}_{\text{tr}}$, the refinement step (selecting $\mathcal{B}_t \subseteq \tilde{\mathcal{B}}_t$ via quality/diversity criteria, e.g. `GREATS`) may be skipped, using $\mathcal{B}_t = \tilde{\mathcal{B}}_t$ directly. Whether refinement helps is scorer- and metric-dependent; it pays off when the selector's within-batch signal is reliable.

While computing $q(\mathbf{z})$ per example is cheap, doing so across all of $\mathcal{D}_{\text{tr}}$ every iteration is prohibitive. We address this with a surrogate model that predicts scores without explicit gradient computation.

4 GAIA framework: Strategy-Guided Online Data Selection

Our goal is to improve online data selection by leveraging historical quality scores of training data to guide the sampling of candidate batches. Specifically, rather than sampling $\tilde{\mathcal{B}}_t$ uniformly at random as in GREATS [7], we use observed quality scores from previous iterations to construct an informed sampling distribution over $\mathcal{D}_{\text{tr}}$. The resulting candidate batch $\tilde{\mathcal{B}}_t$ can then be refined using existing selection methods (e.g., GREATS) to produce the final training batch $\mathcal{B}_t$.

Recall that at each iteration, we evaluate quality scores for all examples in the training batch. By iteration t, we have accumulated observations $\mathcal{D}_t \triangleq \{(\mathbf{z}, q(\mathbf{z}))\}_{\mathbf{z}\in\bigcup_{t'=1}^{t-1}\mathcal{B}_{t'}}$. Since $\mathcal{D}_t$ covers only a subset of $\mathcal{D}_{\text{tr}}$, we cannot directly evaluate $q(\mathbf{z})$ for arbitrary training examples.

4.1 Strategy-Guided Batch Sampling

To address this, we model the latent quality function as a random draw from a GP: $s \sim \text{GP}(0, \kappa)$ where κ denotes the SE kernel (see Appendix A.2). [1] We refer to each function sample s as a *strategy*, as it provides a complete scoring over $\mathcal{D}_{\text{tr}}$ for candidate batch sampling. Given a strategy s, we sample each element of $\tilde{\mathcal{B}}_t$ independently from $\mathcal{D}_{\text{tr}}$ according to the categorical distribution

$$p(\mathbf{z} \mid s) \triangleq \exp(s(\mathbf{z})/\tau) \Big/ \sum\nolimits_{\mathbf{z}'\in\mathcal{D}_{\text{tr}}} \exp(s(\mathbf{z}')/\tau)\ , \tag{1}$$

where $\tau > 0$ controls the sampling temperature.

Remark 2. The temperature τ governs the exploration-exploitation trade-off: lower values concentrate sampling on high-scoring examples, while higher values encourage broader exploration. In practice, τ can be adapted during training, e.g., gradually increasing to promote exploration as the model converges (see Section 5).

A key challenge remains: how to efficiently obtain s from the GP posterior at each training iteration. As noted in Appendix A.2, updating the GP with new observations at every iteration is computationally prohibitive.

4.2 Efficient Strategy Generation via GP Discretization

To enable efficient posterior updates and strategy sampling, we discretize the GP by drawing a finite set of function samples from the prior. Specifically, we sample $m/2$ functions $f_1, \ldots, f_{m/2}$ from $\text{GP}(0, \kappa)$ and set $\mathcal{S} \triangleq \{f_1, -f_1, \ldots, f_{m/2}, -f_{m/2}\}$, so that for every strategy $f \in \mathcal{S}$ its negation $-f$ is also in $\mathcal{S}$. Since these functions are drawn from the prior, we initialize with a uniform distribution: $p(s) = 1/|\mathcal{S}|$ for all $s \in \mathcal{S}$. This discretization preserves the key property encoded by the kernel: similar training examples have similar latent quality scores under each strategy.

Remark 3 (Symmetry of the strategy set). The pairing $\{f, -f\}$ ensures that the strategy set has no directional bias along any sampled function: for any two points $\mathbf{z}, \mathbf{z}' \in \mathcal{D}_{\text{tr}}$, the pair ranks $\mathbf{z}$ above $\mathbf{z}'$ under f if and only if it ranks $\mathbf{z}$ below $\mathbf{z}'$ under $-f$, with equal weight. The pair thus averages out directional preferences, producing a joint marginal $\frac{1}{2}[p(\mathbf{z} \mid f) + p(\mathbf{z} \mid -f)]$ that is strictly closer to uniform than either $p(\cdot \mid f)$ or $p(\cdot \mid -f)$ individually. This property supports the interpretation of the uniform mixing step introduced below as a return toward unbiased exploration: when the mixing weight dominates, the algorithm samples from an ensemble that treats high-score and low-score regions symmetrically. This argument is heuristic: it shows that $\mathcal{S}$ has no directional bias, not that m fixed functions cover the utility landscape as $\boldsymbol{\theta}_t$ drifts. We measure the latter directly in Section 5.2.

Motivation for a non-stationary posterior update. A standard Bayesian posterior update over $\mathcal{S}$ presumes that observations are generated by a single time-invariant strategy. This assumption does not hold in our setting. The observed quality score $q(\mathbf{z})$ depends on the current model parameters $\boldsymbol{\theta}_t$, which change throughout training, so the strategy that best explains recent batches can differ from the one that best explained earlier batches. Under a standard Bayesian update, the posterior concentrates on whichever strategy accumulated the most evidence in the past and cannot recover once a different strategy becomes more informative. To address this, we adopt the *fixed-share Hedge*

[1] The kernel choice is problem-dependent. In our setting, Matérn-5/2 performs on par with SE, rational quadratic is slightly worse, and the rougher Matérn-3/2 lags (Appendix A.10).

Algorithm 1 Strategy-Guided Online Data Selection

Require: $\mathcal{D}_{\text{tr}}$, $\mathcal{D}_{\text{v}}$, #strategies m, Hedge learning rate λ, mixing rate α, temperatures τ, β, k , target model parameters $\boldsymbol{\theta}$

Initialization:
  Obtain embeddings $\{\mathbf{e}(\mathbf{z})\}_{\mathbf{z}\in\mathcal{D}_{\text{tr}}}$
  Fit GP kernel hyperparameters on warm-up data
  Sample strategies $\mathcal{S} = \{f_1, -f_1, \ldots, f_{m/2}, -f_{m/2}\}$ with $f_j \sim \text{GP}(0, \kappa)$
  Initialize $p(s) \leftarrow 1/m$ for all $s \in \mathcal{S}$
for $t = 1, 2, \ldots, T$ **do**
  Strategy Selection:
    Sample strategy $s \sim p(\cdot|\mathcal{D}_{t-1})$
  Candidate Batch Sampling (see Remark 2)**:**
    Sample $\tilde{\mathcal{B}}_t \subset \mathcal{D}_{\text{tr}}$ where each $\mathbf{z} \in \tilde{\mathcal{B}}_t$ is drawn with $p(\mathbf{z}|s) \propto \exp(s(\mathbf{z})/\tau)$
  Batch Refinement (optional, see Remark 1)**:**
    Compute $\{q(\mathbf{z})\}_{\mathbf{z}\in\tilde{\mathcal{B}}_t}$ using $\mathcal{D}_{\text{v}}$
    Refine $\tilde{\mathcal{B}}_t$ to $\mathcal{B}_t \subseteq \tilde{\mathcal{B}}_t$
  Training:
    Update $\boldsymbol{\theta}_t \to \boldsymbol{\theta}_{t+1}$ using $\mathcal{B}_t$
  Strategy Posterior Update (fixed-share Hedge):
    $\forall s \in \mathcal{S}$: $\hat{p}(s|\mathcal{D}_t) \propto p(s|\mathcal{D}_{t-1}) \times p_{\text{top-}k}(q(\mathcal{B}_t)|s)^{\lambda}$
    $\forall s \in \mathcal{S}$: $p(s|\mathcal{D}_t) \leftarrow \alpha/m + (1-\alpha)\,\hat{p}(s|\mathcal{D}_t)$
end for
return $\boldsymbol{\theta}_T$

update of Herbster and Warmuth [26], which tempers each batch's evidence by a learning rate λ and re-mixes the posterior with a uniform distribution at every step. The uniform mixing ensures a strictly positive floor on every strategy's probability, so no strategy is permanently ruled out. This update admits an exact Bayesian interpretation as the filtering distribution under a Markovian switching prior over strategies.

Given observations $\mathcal{D}_t$, the resulting update reduces to an $\mathcal{O}(|\mathcal{S}|)$ incremental two-step procedure:

$$\forall s \in \mathcal{S},\ \hat{p}(s|\mathcal{D}_t) \propto p(s|\mathcal{D}_{t-1})\, p\big(q(\mathcal{B}_t)|s\big)^{\lambda}\,, \quad p(s|\mathcal{D}_t) = \frac{\alpha}{m} + (1-\alpha)\,\hat{p}(s|\mathcal{D}_t)\,, \tag{2}$$

where $q(\mathcal{B}_t) \triangleq \{q(\mathbf{z})\}_{\mathbf{z}\in\mathcal{B}_t}$, $\lambda > 0$ is a learning rate controlling how strongly each batch's evidence shifts the posterior, and $\alpha \in [0, 1)$ is the *mixing rate*. The uniform floor α/m propagates into the next iteration's Hedge step, allowing a previously down-weighted strategy to regain mass once it starts producing high-likelihood batches. Setting $\alpha = 0$ recovers pure Hedge, which corresponds to a stationary Bayesian posterior. Section 4.4 states the dynamic-regret guarantee that this update inherits from fixed-share Hedge. Under the Gaussian likelihood in Equation (6), the likelihood factorizes as

$$p_{\text{Gauss}}\big(q(\mathcal{B}_t)|s\big) \triangleq \prod_{\mathbf{z}\in\mathcal{B}_t} \mathcal{N}\big(q(\mathbf{z}); s(\mathbf{z}), \sigma_n^2\big)\,, \tag{3}$$

where p_{Gauss} highlights the use of the Gaussian likelihood.

4.3 Top-k Likelihood

While the Gaussian likelihood in Equation (6) is commonly employed in GP modeling, it is not well-suited to our setting: the range of observed quality scores can vary drastically across iterations as the model parameters evolve. To address this, we propose a top-k likelihood that depends only on the relative ordering of quality scores within each batch.

Let $\mathbf{z}_{(1)}, \ldots, \mathbf{z}_{(|\mathcal{B}_t|)}$ denote the examples in $\mathcal{B}_t$ sorted by observed quality score in descending order, i.e., $q(\mathbf{z}_{(1)}) \geq q(\mathbf{z}_{(2)}) \geq \cdots \geq q(\mathbf{z}_{(|\mathcal{B}_t|)})$. We define the top-$k$ likelihood as the probability that strategy s ranks the same k examples highest:

$$p_{\text{top-}k}\big(q(\mathcal{B}_t)|s\big) \triangleq \prod_{j=1}^{k} \left[\exp(s(\mathbf{z}_{(j)})/\beta) \Big/ \sum_{i=1}^{|\mathcal{B}_t|} \exp(s(\mathbf{z}_{(i)})/\beta)\right]\,, \tag{4}$$

where $\beta > 0$ is a temperature parameter. Algorithm 1 summarizes the framework for **GAIA**.

4.4 Dynamic-Regret Guarantee

The posterior update in Section 4.1 is designed to be robust to non-stationarity, and fixed-share Hedge is the canonical online-learning algorithm with this property. This subsection makes the robustness precise. For a sequence of loss functions $\ell_t(s) \triangleq -\log p_{\text{top-}k}(q(\mathcal{B}_t)|s)$, the natural non-stationary baseline is a comparator *sequence* $\mathbf{s}^* = (s_1^*, \ldots, s_T^*)$ that may use a different strategy at each iteration, subject to switching between strategies at most K times. Fixed-share Hedge is known to track such a comparator with sublinear regret.

Theorem 1 (Dynamic regret; cf. 26, 27)**.** *As shown in Appendix A.3, the update in Section 4.1 is exactly fixed-share Hedge applied to ℓ_t with learning rate λ and mixing rate α. Let L be a uniform bound on ℓ_t over $\mathcal{S}$. For any comparator $\mathbf{s}^* \in \mathcal{S}^T$ with at most K switches, there exist choices of λ and α such that*

$$\sum_{t=1}^{T} \mathbb{E}_{s\sim p_t}[\ell_t(s)] - \sum_{t=1}^{T} \ell_t(s_t^*) \le \mathcal{O}\Big(L\sqrt{T\big(K\log m + K\log(T/K) + \log m\big)}\Big)\,. \tag{5}$$

The bound is sublinear in T for any K sublinear in T (i.e., $K \le \sqrt{T}$), so the algorithm remains *no-regret* as long as the optimal strategy does not switch too frequently. This justifies the algorithmic form of our posterior update as a response to non-stationarity: it is the mechanism by which the algorithm avoids committing irreversibly to a strategy that may become suboptimal later.

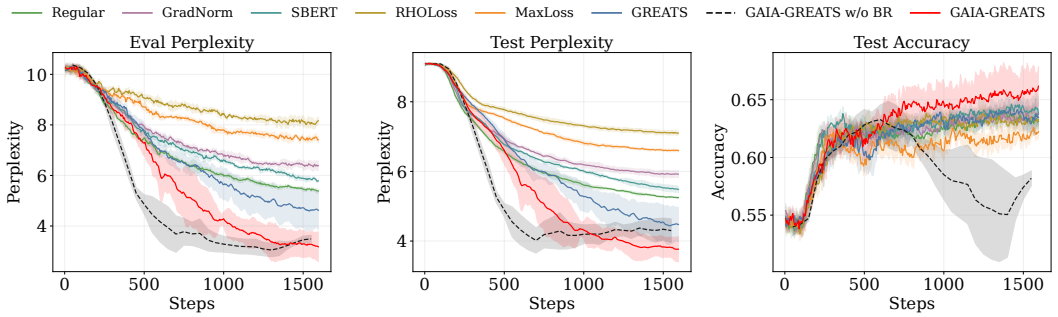
Interpretation. The bound is $\mathcal{O}\big(L\sqrt{T(K\log m + K\log(T/K) + \log m)}\big)$ in T, sublinear whenever K is sublinear in T, and it decomposes into three interpretable cost terms. The factor $\sqrt{T\log m}$ captures the cost of discretizing the GP into m strategies, which is the classical Hedge penalty. The factor $\sqrt{TK\log(T/K)}$ is the additional cost of tracking a comparator that switches up to K times. The loss bound $L = k(\log b + 2S/\beta)$ ties the bound to **GAIA**'s design parameters: the top-k truncation k enters linearly, the likelihood temperature β appears via $1/\beta$ (trading the regret bound against the sharpness of the likelihood signal), and the scale S is the maximum absolute value of strategy scores on the training set, a finite constant determined by the sampled strategies. Setting $K = 0$ recovers the standard static-regret bound $\mathcal{O}\big(L\sqrt{T\log m}\big)$ with $\alpha^* = 0$, which corresponds to pure Hedge, and the regret grows with K only through the slowly growing factor $\sqrt{K\log(T/K)}$, so the algorithm remains no-regret as long as the number of switches is $o(T/\log m)$.

Scope of the guarantee. Theorem 1 bounds regret against comparators in $\mathcal{S}^T$: it guarantees that **GAIA** tracks the best strategies *within* the fixed set $\mathcal{S}$, not that these strategies remain good as $\boldsymbol{\theta}_t$ drifts. Whether $\mathcal{S}$ remains adequate is an empirical property, which we measure in Section 5.2: the best in-set loss $\min_{s\in\mathcal{S}} \ell_t(s)$ stays nearly flat throughout training, while a typical strategy degrades.

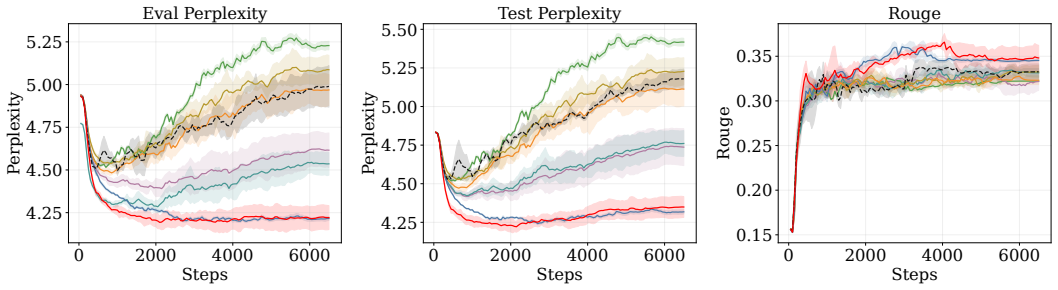
Practical remarks on tuning λ and α. The optimal λ^* and α^* in Theorem 1 depend on the horizon T, the loss bound L, and the number of switches K. Since K is not observable and characterizes how much the identity of the optimal strategy changes as training progresses, setting λ and α from the theoretical expressions requires either an estimate of K or an adaptive procedure such as the doubling trick over candidate values of K, which incurs an additional logarithmic factor. In our experiments, we bypass this estimation problem and tune λ and α empirically on a small grid, using the theoretical bound only as qualitative guidance (small α for sharp exploitation, larger α for broader tracking). A Bayesian-optimization search over α lands in a region statistically indistinguishable from the grid's choice (Appendix A.5).

5 Experiments

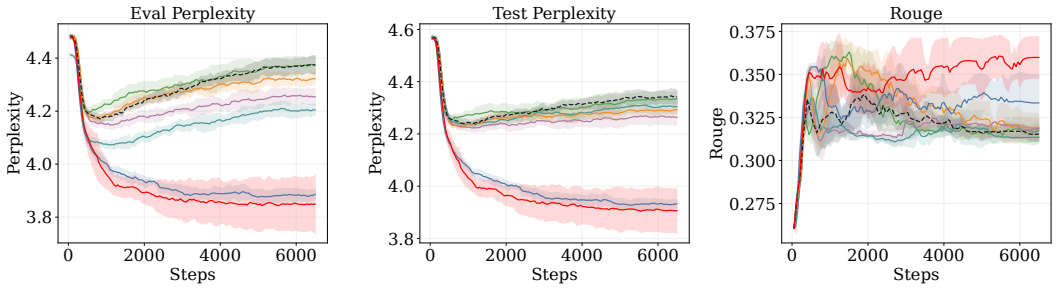
We follow the settings of [7]. We evaluated our method on three instruction tuning tasks, MMLU [28], SAMSUM [29] and TYDIQA [30], and on four backbone models, LLAMA-2-7B [31], LLAMA-3.1-8B [32], QWEN-3-4B [33] and MISTRAL-7B [34]. Training details are presented in Appendix A.4. We evaluate **GAIA** against a diverse set of representative methods that also serve as our score functions, including gradient-based (`GradNorm`), loss-prioritizing (`MaxLoss`, `RHOLoss`), semantic-aware (`SBERT`), and greedy optimization (`GREATS`) methods, also two recent diversity-aware online batch selection methods, `DivBS` [24] and `UDS` [25], as refinement stages. Detailed descriptions are provided in Appendix A.6. Main results are averaged over three independent runs; the integration, kernel, and refinement analyses use two to four seeds, as stated in each table.



(a) MMLU-Sociology, Llama-2-7B



(b) SAMSUM, Llama-3.1-8B



(c) SAMSUM, Qwen3-4B (no RHOLoss)

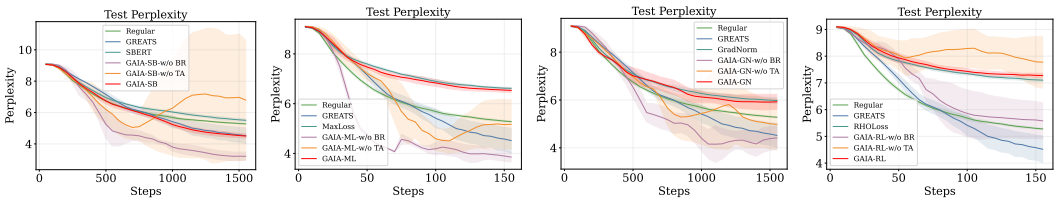
Figure 2: Training dynamics on MMLU-Sociology with Llama-2-7B (a) and SAMSUM with Llama-3.1-8B (b) and Qwen3-4B (c). We benchmark **GAIA**-GREATS against full-data training (Regular), various selection baselines, and **GAIA**-GREATS without batch refinement (w/o BR). The left and middle columns report validation and test perplexity (lower is better), while the right column presents downstream task performance (Accuracy for MMLU, ROUGE for SAMSUM). **GAIA**-GREATS demonstrates superior convergence speed and robust generalization, achieving the lowest or near-lowest perplexity levels and the highest or comparable task scores across all settings. All results are reported as the average of three independent runs.

5.1 Efficient Gaussian Process Construction

To make GP inference computationally feasible for LLMs, we introduce a decoupled pipeline.

Proxy Semantic Space. Rather than using evolving hidden states, we extract static features $\mathbf{e}(\mathbf{x})$ via a sentence transformer [35] **(robustness evaluated in Appendix A.17)**. To further accelerate inference, SVD projects these embeddings into a lower-dimensional manifold **(dimensionality analysis in Appendix A.18)**.

Warm-up Initialization. To address label scarcity, we evaluate true utility scores on a small subset ($N = 500$) during early training **(size sensitivity analyzed in Appendix A.19)**. The GP is then fitted to these observations to generalize predictions across the remaining data pool. The warm-up requires no data or compute beyond ordinary training, since quality scores are computed at every step anyway; a larger N gives the GP more observations but delays the start of GP-guided sampling, and we use $N = 500$ as a compute-frugal default. Implementation details are in Appendix A.5.



(a) **GAIA**-SBERT (b) **GAIA**-MaxLoss (c) **GAIA**-GradNorm (d) **GAIA**-RHOLoss

Figure 3: Performance comparison of four different scoring functions on MMLU-Sociology. All results are reported as the average of three independent runs. The annotation **w/o BR** denotes the exclusion of the *Batch Refinement* step, while **w/o TA** indicates the removal of the *Temperature Adaptation* mechanism.

5.2 Results and Analysis

Figure 2 depicts the comparative training dynamics of **GAIA**-GREATS (where our **GAIA** framework leverages the GREATS scoring function) against baselines on MMLU-Sociology with Llama-2-7B and SAMSUM with Llama-3.1-8B and Qwen3-4B. Additional MMLU subjects and TYDIQA results are reported in Appendix A.9 and Appendix A.8, respectively. Notably, these performance gains come with negligible computational overhead: for instance, on MMLU ($\approx$13k samples), the one-time embedding and GP fitting take $\approx$23 seconds, and the recurring per-step cost of strategy sampling and posterior updates is about 2.6 ms, roughly 0.2% of a training step. A detailed efficiency analysis is provided in Appendix A.13.

Training Dynamics. As shown in the perplexity plots, **GAIA**-GREATS (red line) exhibits a significantly faster convergence rate than the baselines on both evaluation and test sets. Among the competitors, GREATS proves to be the strongest baseline; however, our method demonstrates distinct advantages depending on the task. On MMLU-Sociology, **GAIA**-GREATS consistently maintains a lower perplexity throughout training, achieving a minimum of **3.75**$\pm$**0.33** compared to 4.37$\pm$0.64 for GREATS, representing a substantial **14.13% reduction**. This indicates a more precise estimation of data utility. To illustrate this behavior, we visualize the specific GP data sampling dynamics for the MMLU-Sociology task in Appendix A.14. On the SAMSUM task, while both methods eventually reach a comparable performance floor, **GAIA**-GREATS distinguishes itself with a much sharper descent in the initial phase. This rapid drop indicates that our Gaussian Process model effectively identifies high-value data points immediately at the start, significantly accelerating the early learning process compared to the gradient-based selection in GREATS. With Qwen3-4B on SAMSUM, the other baselines level off at a test perplexity between 4.20 and 4.23 and then rise, whereas **GAIA**-GREATS and GREATS keep decreasing, to minima of 3.85$\pm$0.07 and 3.91$\pm$0.02, respectively.

Downstream Performance. Our method consistently outperforms or matches the best competitors. For MMLU-Sociology, **GAIA**-GREATS achieves a peak accuracy of **66.50%**$\pm$**1.92**, surpassing the second-best method GREATS (64.34% $\pm$1.02) with a clear gain of **+3.36%**. On the SAMSUM summarization task, we evaluate performance using the ROUGE score. While standard methods suffer from overfitting, evidenced by increasing test perplexity after 2000 steps, both GREATS and our method maintain robustness across both backbones. Notably, **GAIA**-GREATS achieves higher ROUGE scores early in training and, when using Qwen3-4B, ultimately reaches the highest final ROUGE, albeit with a larger spread across seeds. This further validates that our strategy successfully prioritizes the most informative samples early on, leading to a more efficient optimization path without sacrificing generalization capability.

Robustness across Scoring Functions. To assess the universality of our framework, we evaluate **GAIA** across a diverse set of scoring functions. We observe that our GP-based mechanism acts as a robust performance amplifier, though its effectiveness is heavily influenced by the nature and fidelity of the underlying signal. Fundamentally, **GAIA** excels with scoring functions that measure an example's intrinsic relevance to the target task, such as GREATS and SBERT. Because these scores represent inherent properties of individual examples, the GP can effectively map them as smooth functions over the embedding space. Within this category, the actual performance gain depends on the metric's signal fidelity. For high-quality metrics like SBERT, our framework significantly boosts convergence and achieves results comparable to the strong GREATS baseline. Conversely, for weaker metrics like GradNorm, MaxLoss, and RHOLoss, the method yields limited additional gains, although

it consistently maintains performance parity or achieves slight improvements over the baselines. In contrast, when we compose **GAIA** with diversity-aware selectors like `DivBS` [24] and `UDS` [25], the benefits become less pronounced. Because these scores depend on the surrounding candidate batch rather than individual examples, they are more difficult for the GP to generalize as smooth spatial functions. As a result, on MMLU-Sociology, **GAIA** improves `UDS` accuracy for both Llama-2-7B and Qwen3-4B but does not improve `DivBS`, while on SAMSUM, the differences largely fall within seed variance (Appendix A.11).

Selection Quality over Training. Compared with uniform random sampling, the data selected by **GAIA** accumulates a substantially higher quality score throughout training (Figure 4). A related question is whether the strategy set $\mathcal{S}$ itself remains adequate: Theorem 1 guarantees tracking only within $\mathcal{S}$, which is sampled once after warm-up and never re-shaped. We replay the logged batches and quality scores of each run against strategy sets drawn independently following Section 4.2, and compute the per-step loss $\ell_t(s)$ of every strategy. On MMLU-Sociology with `GREATS`, the best in-set loss $\min_{s\in\mathcal{S}} \ell_t(s)$ stays nearly flat from the first to the last training quartile (2.91 $\rightarrow$ 3.06), whereas the in-set median degrades (7.27 $\rightarrow$ 9.68): individual strategies go stale, but the set retains coverage. Refitting the GP on all accumulated observations and redrawing $\mathcal{S}$ at steps 800, 1600 and 2400 never improves on the frozen set, and the best in-set strategy changes about 27 times per run at 100-step granularity, which motivates fixed-share over pure Hedge. The same pattern holds for all five scoring functions and on MMLU-US-Foreign-Policy and TYDIQA (Appendix A.12).

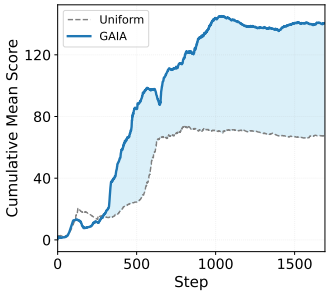
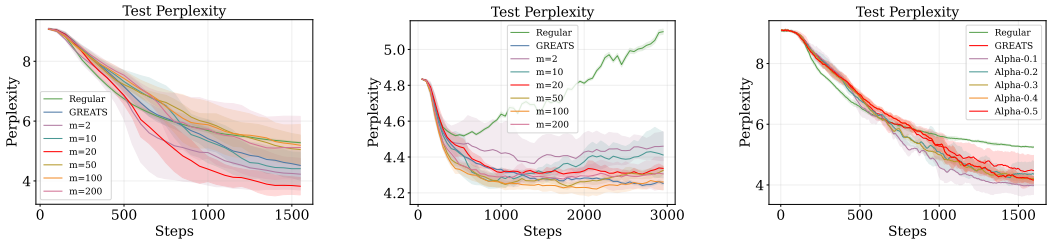


Figure 4: Cumulative data value scores (LLAMA-2-7B, MMLU-Sociology).

Role of Batch Refinement. To isolate the contribution of global selection, we disable the batch refinement (BR) step and train directly on the batch sampled via the GP (*w/o BR*), shown in Figure 2 and Figure 3. On MMLU-Sociology, **GAIA** without refinement lowers perplexity quickly, but its accuracy is lower and much less stable across seeds (Figure 2a); this holds for all five scorers, where w/o BR is less accurate than **GAIA**-X. On SAMSUM, where perplexity is computed over the full summary, refinement improves both metrics with both backbones: without it, test perplexity rises as in Regular training and ROUGE-L is lower. Refinement thus complements global sampling: **GAIA** decides where to sample, and the selector removes the less useful examples within each candidate batch, which matters most for the downstream metric.



(a) m on MMLU-Sociology (b) m on SAMSUM (c) α on MMLU-Sociology

Figure 5: Impact of strategy number on test perplexity for MMLU-Sociology and SAMSUM and α for MMLU-Sociology. We benchmark **GAIA**-`GREATS` against *Regular* training and the `GREATS` baseline. All results are averaged over three independent runs.

Effect of Strategy Pool Size. Results in Figure 5 show that the optimal strategy pool size m varies with the size of the dataset. For the smaller MMLU-Sociology, a compact pool ($m = 20$) is optimal, as excessive strategies tend to dilute the utility signal toward uniform sampling (*Regular* baseline). Conversely, the larger SAMSUM dataset demands greater strategic diversity; while $m \geq 50$ yields strong convergence, a constrained pool ($m < 50$) results in a noticeably higher perplexity floor due to insufficient utility coverage. Therefore, m must be adaptively scaled: we should prioritize focused selection for small datasets and expand the ensemble for larger datasets, and more complex tasks.

Impact of Sampling Temperature. The sampling temperature τ serves as a critical control for the exploration-exploitation trade-off within our GP sampling framework. Our analysis reveals a distinct tension: lower temperatures accelerate early-stage convergence via aggressive exploitation of high-utility samples, whereas higher temperatures ensure batch diversity but slow down the identification of informative data (See Appendix A.16 for details experiments). To reconcile these objectives, **GAIA**-GREATS employs an **adaptive temperature** mechanism that dynamically transitions from focused exploitation to broader exploration. The necessity of this schedule is empirically confirmed by the **w/o TA** ablation results (Figure 3), where using a fixed temperature leads to significant training instability and mid-stage performance corruption. We provide a detailed discussion of the adaptive mechanism in Appendix A.15.

Effect of Mixing Rate α. The fixed-share mixing rate α enforces a uniform floor on every strategy's probability, allowing the posterior to recover when a previously down-weighted strategy becomes more informative. Sweeping $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ on MMLU-Sociology with LLAMA-2-7B, we find that $\alpha = 0.1$ achieves the lowest validation and test perplexity, and is among the highest test accuracies. This finding supports two complementary design choices. First, non-zero mixing is essential: it prevents the posterior from irreversibly committing to an early-favored strategy, empirically validating the fixed-share update over pure Hedge. Second, the optimal mixing is small, indicating that the GP-informed posterior signal should be largely preserved rather than diluted toward the uniform prior. A sharp posterior anchored to the Gaussian process does the heavy lifting of identifying high-utility regions, while a modest mixing floor maintains robustness under the non-stationarity of quality scores.

6 Conclusion

Limitations. Despite its effectiveness, our method has several limitations. First, its performance is tied to the fidelity of the base scoring function, acting as an amplifier rather than a corrective mechanism for suboptimal data utility signals. Second, the method exhibits sensitivity to hyperparameters such as the strategy pool size m and sampling temperature τ, which require calibration based on dataset size to effectively balance exploration and exploitation. Third, Theorem 1 only guarantees tracking a comparator within the fixed strategy set; the adequacy of that set is established empirically, not theoretically. Finally, our largest candidate pool has about 108k examples; validation at very large scale, including pre-training corpora, remains limited and is left to future work.

Conclusion. We have presented an adaptive data selection framework for LLM fine-tuning that moves beyond conventional batch-constrained approaches. By modeling global utility distributions with Gaussian Process regression and dynamically fusing multiple valuation strategies, our method steers candidate batches toward high-utility regions of the training pool. Casting the strategy-posterior update as an instance of the classical fixed-share Hedge framework for tracking the best expert, **GAIA** inherits a dynamic-regret guarantee for tracking the best strategy within its strategy set, and our replay analysis shows that this set remains adequate throughout training. Empirical results across MMLU, TYDIQA, and SAMSUM with four backbone models show that our approach accelerates convergence at negligible overhead, reaching the lowest or near-lowest perplexity while matching or exceeding the strongest baseline on downstream metrics. These results highlight the value of global, rather than batch-local, data selection for efficient instruction tuning.

Acknowledgments and Disclosure of Funding

The authors would like to thank the anonymous reviewers and the Area Chair for their insightful comments and constructive feedback, which helped improve this paper. We also thank Dr Abhirup Mondal for his careful reading of the manuscript. This work was conducted as part of the authors' employment at Amazon and received no third-party funding or support. The authors declare no competing interests.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[3] Juncheng Wu, Wenlong Deng, Xingxuan Li, Sheng Liu, Taomian Mi, Yifan Peng, Ziyang Xu, Yi Liu, Hyunjin Cho, Chang-In Choi, et al. Medreason: Eliciting factual medical reasoning steps in llms via knowledge graphs. *arXiv preprint arXiv:2504.00993*, 2025.

[4] Emilio Ferrara. Should chatgpt be biased? challenges and risks of bias in large language models. *Challenges and Risks of Bias in Large Language Models (October 26, 2023)*, 2023.

[5] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *ICML*, 2019.

[6] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017.

[7] 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. *Advances in Neural Information Processing Systems*, 37:131197–131223, 2024.

[8] Hoang Anh Just, Feiyang Kang, Tianhao Wang, Yi Zeng, Myeongseob Ko, Ming Jin, and Ruoxi Jia. Lava: Data valuation without pre-specified learning algorithms. In *The Eleventh International Conference on Learning Representations*. OpenReview, 2023.

[9] Samuel Kessler, Tam Le, and Vu Nguyen. Sava: Scalable learning-agnostic data valuation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[10] Jiongli Zhu, Parjanya Prajakta Prashant, Alex Cloninger, and Babak Salimi. Kairos: Scalable model-agnostic data valuation. *Advances in Neural Information Processing Systems*, 2025.

[11] Wenqin Liu, Weizhi Quan, Aoqi Zuo, Erdun Gao, Vu Nguyen, Dino Sejdinovic, Howard Bondell, and Mingming Gong. Timelava: Learning-agnostic valuation for time series data. In *International Conference on Machine Learning*, pages 78066–78099. PMLR, 2026.

[12] Guangyi Zhang, Qiyu Liu, and Aristides Gionis. Shapley-based data valuation for weighted k-nearest neighbors. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[13] Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Minsoo Kang, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, et al. What is your data worth to gpt? llm-scale data valuation with influence functions. 2025.

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

[15] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020.

[16] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930, 2020.

[17] Jiachen T Wang, Tianji Yang, James Zou, Yongchan Kwon, and Ruoxi Jia. Rethinking data shapley for data selection tasks: Misleads and merits. In *International Conference on Machine Learning*, pages 52033–52063. PMLR, 2024.

[18] 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. *Advances in Neural Information Processing Systems*, 36: 69798–69818, 2023.

[19] Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, pages 54104–54132, 2024.

[20] Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models. In *Forty-first International Conference on Machine Learning*.

[21] 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. In *International Conference on Learning Representations*, 2019.

[22] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in neural information processing systems*, 34:20596–20607, 2021.

[23] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

[24] Feng Hong, Yueming Lyu, Jiangchao Yao, Ya Zhang, Ivor W. Tsang, and Yanfeng Wang. Diversified batch selection for training acceleration. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

[25] Heming Zou, Yixiu Mao, Yun Qu, Qi Wang, and Xiangyang Ji. Utility-diversity aware online batch selection for LLM supervised fine-tuning. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026.

[26] Mark Herbster and Manfred K Warmuth. Tracking the best expert. *Machine Learning*, 32(2): 151–178, 1998.

[27] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[28] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[29] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79. Association for Computational Linguistics, November 2019.

[30] Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 2020.

[31] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[32] 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.

[33] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[34] 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. *CoRR*, abs/2310.06825, 2023. doi: 10.48550/ARXIV.2310.06825.

[35] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[36] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[37] Artem Artemev, David R Burt, and Mark van der Wilk. Tighter bounds on the log marginal likelihood of gaussian process regression using conjugate gradients. In *International Conference on Machine Learning*, pages 362–372. PMLR, 2021.

[38] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[39] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR, 2023.

[40] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[41] Conover Mike, Hayes Matt, Mathur Ankit, Xie Jianwei, Wan Jun, Shah Sam, Ghodsi Ali, Wendell Patrick, Zaharia Matei, and Xin Reynold. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023.

[42] Andreas Köpf, Yannic Kilcher, Dimitri Von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations-democratizing large language model alignment. *Advances in neural information processing systems*, 36:47669–47681, 2023.

[43] Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31, 2018.

[44] Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80, pages 2530–2539. PMLR, 2018.

[45] Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. *CoRR*, abs/1511.06343, 2015.

[46] Sören Mindermann, Jan Markus Brauner, Muhammed Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning, ICML 2022*, volume 162, pages 15630–15649. PMLR, 2022.

A Appendix

A.1 Notation

Table 1: Summary of notation.

Symbol	Description
$\mathcal{D}_{\text{tr}}$, $\mathcal{D}_{\text{v}}$	Training and validation sets
$\mathbf{z} \triangleq (\mathbf{x}, y)$	Training example (input, target)
$\boldsymbol{\theta}_t$	Model parameters at iteration t
$l(\mathbf{z}; \boldsymbol{\theta}_t)$	Loss for example $\mathbf{z}$ at iteration t
$\mathbf{g}_t(\mathbf{z})$	Gradient $\nabla l(\mathbf{z}; \boldsymbol{\theta}_t)$
$\tilde{\mathcal{B}}_t \subset \mathcal{D}_{\text{tr}}$	Candidate batch
$\mathcal{B}_t \subset \tilde{\mathcal{B}}_t$	Refined training batch
$q(\mathbf{z})$	Observed quality score
$\mathbf{e}(\mathbf{z}) = \mathbf{e}(\mathbf{x})$	Embedding of example $\mathbf{z}$
$s(\mathbf{z})$	Latent quality (GP sample)
$\mathcal{S}$	Set of m strategies
$T \in \mathbb{N}$	Total number of training iterations
κ	GP kernel
$\lambda \in \mathbb{R}_+^*$	Hedge learning rate
$\alpha \in [0, 1)$	Mixing rate (fixed-share)
$\tau \in \mathbb{R}_+^*$	Sampling temperature
$\beta \in \mathbb{R}_+^*$	Likelihood temperature
$k \in \mathbb{R}_+^*$	Number of top positions in likelihood
$\eta_t \in \mathbb{R}_+^*$	SGD learning rate

A.2 Gaussian Process Background

We model the latent quality function using a *Gaussian process* (GP) [36]. Let $s : \mathcal{D}_{\text{tr}} \to \mathbb{R}$ denote the GP prior over quality scores, where $s(\mathbf{z})$ represents the intrinsic, long-term quality of example $\mathbf{z}$. As a simple modeling choice, we assume the observed quality score $q(\mathbf{z})$ is a noisy realization of $s(\mathbf{z})$:

$$q(\mathbf{z}) = s(\mathbf{z}) + \epsilon \,, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2) \,, \tag{6}$$

where σ_n^2 denotes the observation noise variance. The noise term captures iteration-specific fluctuations: since $q(\mathbf{z})$ depends on the current model parameters $\boldsymbol{\theta}_t$, which evolve during training, the same example may yield different quality scores at different iterations. We revisit this likelihood model in Section 4 to better capture the relationship between observed scores and the latent quality function.

Under the GP prior, $\{s(\mathbf{z})\}_{\mathbf{z} \in \mathcal{D}_{\text{tr}}}$ follows a multivariate Gaussian distribution fully specified by a prior mean and a covariance kernel $\kappa_{\mathbf{z},\mathbf{z}'} \triangleq \text{cov}(q(\mathbf{z}), q(\mathbf{z}'))$. The kernel encodes the assumption that similar training examples have similar quality scores. To evaluate $\kappa_{\mathbf{z},\mathbf{z}'}$, we embed the input component $\mathbf{x}$ of each example $\mathbf{z} = (\mathbf{x}, y)$ using a pre-trained model and apply random projection to obtain a low-dimensional representation $\mathbf{e}(\mathbf{z}) \triangleq \mathbf{e}(\mathbf{x})$. Note that the embedding depends only on the input, not the label.

Following standard practice in GP modeling, we employ the squared exponential (SE) kernel; other kernels may also be used, and the choice is problem-dependent (Appendix A.10 compares SE with Matérn and rational quadratic kernels):

$$\kappa_{\mathbf{z},\mathbf{z}'} \triangleq \sigma_s^2 \exp\left(-\frac{1}{2}\left\|\frac{\mathbf{e}(\mathbf{z}) - \mathbf{e}(\mathbf{z}')}{\mathbf{l}^2}\right\|^2\right) \,, \tag{7}$$

where σ_s and $\mathbf{l}$ denote the signal variance and lengthscales, respectively. These hyperparameters can be estimated directly from the data by maximizing the GP log marginal likelihood [36, 37]. Let $\mathcal{D}_t \triangleq \{(\mathbf{z}, q(\mathbf{z}))\}_{\mathbf{z} \in \bigcup_{t'=1}^{t-1} \mathcal{B}_{t'}}$ denote the set of observed quality scores from previous iterations. Given $\mathcal{D}_t$, the posterior distribution of $s(\mathbf{z}_*)$ for any $\mathbf{z}_* \in \mathcal{D}_{\text{tr}}$ is Gaussian with closed-form mean $\mu_t(\mathbf{z}_*)$ and variance $\sigma_t^2(\mathbf{z}_*)$ [36]. However, exact GP inference requires $\mathcal{O}(n^3)$ time, where n is the number of observations, rendering it impractical for per-iteration evaluation. Even sparse GP

approximations with n_{ind} inducing points require $\mathcal{O}(nn_{\text{ind}}^2)$ time. Our approach (Section 4) addresses this computational bottleneck while preserving the expressiveness of the GP framework for online data selection.

A.3 Dynamic-Regret Guarantee via Fixed-Share Hedge

This appendix supports Theorem 1 of the main text. We state the posterior update of Section 4 as an instance of the *fixed-share Hedge* algorithm of Herbster and Warmuth [26] and instantiate the standard fixed-share Hedge regret bound for our setting. We condition on the set of strategies $\mathcal{S}$ constructed at initialization by paired sampling from $\text{GP}(0, \kappa)$ (see Section 4) and analyze the regret of the posterior update over these fixed strategies, which is the component of the algorithm that is updated online. The GP sampling step is a one-time offline operation whose analysis is orthogonal to the regret argument.

Setup. Let $\mathcal{S}$ be the finite strategy set of size m constructed in Section 4 by paired sampling from $\text{GP}(0, \kappa)$, and let T denote the total number of training iterations. At iteration $t \in \{1, \dots, T\}$, the algorithm holds a distribution p_t over $\mathcal{S}$ (defined as $p_t(s) \triangleq p(s \mid \mathcal{D}_{t-1})$, the posterior from Section 4), samples a strategy, uses it to construct the candidate batch, and observes the per-strategy loss

$$\ell_t(s) \triangleq -\log p_{\text{top-}k}\big(q(\mathcal{B}_t) \mid s\big) \,. \tag{8}$$

Note that $\ell_t(s)$ can be computed for every $s \in \mathcal{S}$ from the pre-computed values $\{s(\mathbf{z})\}_{\mathbf{z}\in\mathcal{B}_t}$, so the algorithm operates in the *full-information* setting rather than the bandit setting. Because $\mathcal{B}_t$ is itself generated using the sampled strategy $s_t \sim p_t$, the loss function ℓ_t is a random object determined by the algorithm's own trajectory; Theorem 1 therefore bounds regret along this realized trajectory rather than against a counterfactual trajectory driven by the comparator.

Loss boundedness. Let $b \triangleq |\mathcal{B}_t|$. For a fixed set of strategies $\mathcal{S}$ sampled at initialization, define

$$S \triangleq \sup_{s\in\mathcal{S},\, \mathbf{z}\in\mathcal{D}_{\text{tr}}} |s(\mathbf{z})| \,, \tag{9}$$

which is a finite deterministic quantity that can be computed once after the strategies are drawn. We bound $\ell_t(s)$ by direct calculation. Let $a_i \triangleq s(\mathbf{z}_{(i)})/\beta$. Then

$$\ell_t(s) = \sum_{j=1}^{k} \log \sum_{i=1}^{b} e^{a_i - a_j} \,. \tag{10}$$

The lower bound $\ell_t(s) \geq 0$ follows because the inner sum includes the term $i = j$, which equals 1, so each log term is non-negative. The upper bound uses $a_i - a_j \in [-2S/\beta, 2S/\beta]$, giving $\sum_i e^{a_i - a_j} \leq b \cdot e^{2S/\beta}$ and therefore $\ell_t(s) \leq L$ with

$$L \triangleq k\big(\log b + 2S/\beta\big) \,. \tag{11}$$

Connection to fixed-share Hedge. We show that the posterior update of Section 4 is a particular instance of fixed-share Hedge.

Proposition 1 (Equivalence to fixed-share Hedge)**.** *The two-step posterior update of Section 4,*

$$\hat{p}(s|\mathcal{D}_t) \propto p(s|\mathcal{D}_{t-1})\, p_{\textit{top-}k}\big(q(\mathcal{B}_t) \mid s\big)^{\lambda} \,, \tag{12}$$

$$p(s|\mathcal{D}_t) = \alpha/m + (1-\alpha)\, \hat{p}(s|\mathcal{D}_t) \,, \tag{13}$$

where $\hat{p}(s|\mathcal{D}_t)$ denotes the intermediate post-Hedge distribution and $p(s|\mathcal{D}_t)$ is the final posterior carried into the next iteration, is equivalent to the fixed-share Hedge algorithm of Herbster and Warmuth [26] with learning rate λ and mixing rate α, applied to the loss sequence $\ell_t(s) = -\log p_{\textit{top-}k}(q(\mathcal{B}_t) \mid s)$.

Proof. Substituting $p_{\text{top-}k}(q(\mathcal{B}_t) \mid s) = \exp(-\ell_t(s))$ into the first step gives $\hat{p}(s|\mathcal{D}_t) \propto p(s|\mathcal{D}_{t-1}) \exp(-\lambda \ell_t(s))$, which is the exponential-weights (Hedge) update with learning rate λ on the loss ℓ_t. The second step is the uniform mixing step of fixed-share Hedge with mixing rate α and uniform prior $1/m$. □

Dynamic-regret comparator. To capture non-stationarity of the optimal strategy, we compare the algorithm's cumulative loss to that of an arbitrary comparator sequence $\mathbf{s}^* = (s_1^*, \ldots, s_T^*) \in \mathcal{S}^T$ that is allowed to switch up to K times, i.e. $|\{t : s_{t+1}^* \neq s_t^*\}| \leq K$. The dynamic regret against $\mathbf{s}^*$ is

$$\mathrm{Reg}_T(\mathbf{s}^*) \triangleq \sum_{t=1}^{T} \mathbb{E}_{s\sim p_t}[\ell_t(s)] - \sum_{t=1}^{T} \ell_t(s_t^*) \,. \tag{14}$$

Statement of the bound.

Theorem 2 (Dynamic regret bound, instantiation of 26 for **GAIA**). *Fix the set of strategies $\mathcal{S}$ sampled at initialization, which determines S and hence L. Let $\mathbf{s}^* \in \mathcal{S}^T$ be any comparator sequence with at most K switches. Running the posterior update of Section 4 with learning rate*

$$\lambda^* = \sqrt{\frac{8\big(K \log m + K \log(T/K) + \log m\big)}{L^2 T}} \tag{15}$$

and mixing rate $\alpha^* = K/(T-1)$ yields

$$\mathrm{Reg}_T(\mathbf{s}^*) \leq L\sqrt{\frac{T}{2}\big(K \log m + K \log(T/K) + \log m\big)} \,, \tag{16}$$

where $L = k(\log b + 2S/\beta)$ is the loss bound from Equation (11).

Proof. By Proposition 1, the posterior update of Section 4 is fixed-share Hedge applied to the loss sequence $(\ell_t)_{t=1}^T$. The losses satisfy $\ell_t(s) \in [0, L]$ per Equation (11), and the algorithm operates in the full-information setting because $\ell_t(s)$ is computable for every $s \in \mathcal{S}$ from $\{s(\mathbf{z})\}_{\mathbf{z}\in\mathcal{B}_t}$. The bound then follows from the standard dynamic-regret analysis for fixed-share Hedge with bounded losses (see Herbster and Warmuth [26] for the algorithm and Cesa-Bianchi and Lugosi [27] for the $[0, L]$-loss analysis via Hoeffding-type exponential-weights bounds), instantiated with the stated choices of λ^* and α^*. □

A.4 Training Tasks

We follow the experimental setting in `GREATS`, reported our method on three instruction finetuning tasks:

- **MMLU** Finetuning LLAMA-2-7B and QWEN-3-4B on 5% LESS data, and evaluate on MMLU, we target the query, key, value, and output projections, setting the LoRA rank to 128, LoRA α to 1.0, and dropout to 0.1. The learning rate is set to 2e-5.
- **SAMSUM** Finetuning LLAMA-3.1-8B and QWEN-3-4B on ALPACA dataset, and evaluate on SAMSUM, we target the query, value, and output projections within the self-attention layers, as well as the gate, up, and down projections of the feed-forward networks, setting the LoRA rank to 8, LoRA α to 16.0, and dropout to 0.1. The learning rate is set to 2e-5.
- **TYDIQA** Finetuning MISTRAL-7B on 40% LESS data, and evaluate on TYDIQA, we target the query and key attention projections, setting the LoRA rank to 128, LoRA α to 1.0, and dropout to 0.1. The learning rate is set to 1e-5.

Training Dataset There are two primary instruction tuning datasets are involved in our experiments: LESS [19] and ALPACA [38].

- **LESS:** A composite dataset combining four instruction tuning sources: FLAN V2 [39], COT [40], DOLLY [41], and OPENASSISTANT [42].
- **ALPACA:** An instruction-following dataset comprising 52k examples, covering a diverse range of user-oriented tasks.

Evaluation Dataset To ensure a comprehensive evaluation, we assess performance on three distinct benchmarks: MMLU, TYDIQA, and SAMSUM. We report perplexity for all tasks

- **MMLU** consists of multiple-choice questions across 57 subjects. While our main experiments focus on the Sociology subject, we report accuracy across 5 different subjects to demonstrate robustness.

- **TyDiQA** is a multilingual question-answering benchmark covering 11 diverse languages, where the objective is to extract an answer from a provided passage. For evaluation, we randomly select a held-out set of 500 test samples.
- **SamSum** involves summarizing dialogues. For this task, we report Perplexity, and ROUGE scores to measure generation quality.

A.5 Implementation Details

GP Training and Hyperparameters Our Gaussian Process model is implemented using the `GPyTorch` library [43]. We optimize the kernel hyperparameters—specifically the signal variance σ_s and lengthscales $\mathbf{l}$—by maximizing the marginal log-likelihood (MLL) of the warm-up observations.

The GP training involves 200 iterations with an initial noise standard deviation of 0.1 and a Hedge learning rate of $\lambda = 0.8$. For the adaptive temperature schedule, we update the temperature every 100 training steps with an increment of 0.01 per interval. The semantic embeddings are projected into a 10-dimensional subspace using SVD to ensure efficient posterior sampling.

Task-Specific Configurations We provide the detailed strategy pool sizes and temperature settings for each benchmark in Table 2.

Table 2: Hyperparameter settings for **GAIA** across different tasks.

Task	**Strategy Pool Size** (m)	**Sampling Temp** (τ)	**Likelihood Temp** (β)
MMLU	20	0.2	0.1
Samsum	100	0.1	0.01
TyDiQA	150	0.1	0.01

Hyperparameter Tuning via Bayesian Optimization We apply Bayesian optimization (BO) to the mixing rate α, the hyperparameter that is not set by an explicit rule (m follows the dataset-size rule and τ the adaptive schedule). We use a log-uniform prior over $[0.01, 0.9]$, a GP surrogate with expected-improvement acquisition, and five evaluations in total, each a short single-seed training run of 800 steps on MMLU-Sociology. BO converges after two rounds to $\alpha \in [0.03, 0.04]$ (Table 3). The spread across the evaluated range is within single-seed noise, so BO and the grid select statistically indistinguishable regions. We report this experiment on one benchmark only. For the sensitivity analyses in the paper we keep small, pre-specified grids (3–5 values per hyperparameter), because a fixed grid applies the same candidate values and protocol across seeds and keeps sensitivity and run-to-run variability directly comparable.

Configuration Guide. **GAIA** has two groups of hyperparameters (Table 2). (i) *Dataset-scaled*: the strategy pool size m and the temperatures τ, β. We use $m = 20$ for MMLU (≈13.5k examples), $m = 100$ for SAMSum (≈52k), and $m = 150$ for TyDiQA (≈108k): focused pools for small datasets and larger ensembles for larger ones. Smaller datasets benefit from slightly higher temperatures to preserve diversity, larger ones from sharper distributions. (ii) *Global defaults*, used unchanged across all tasks and backbones: mixing rate $\alpha = 0.1$; confirmed by Bayesian optimization shown above, Hedge learning rate $\lambda = 0.8$, top-k with $k = 2$, and warm-up size $N = 500$.

Table 3: BO over α on MMLU-Sociology (single seed, short runs).

α (source)	Best test PPL within 800 steps
0.030 (BO, round 2)	6.39
0.100 (grid selection)	6.71
0.034 (BO, initialization)	6.74
0.176 (BO, initialization)	6.75
0.040 (BO, round 2)	6.78

A.6 Baselines and Scoring Functions

We compare **GAIA** against several representative data selection and valuation methods:

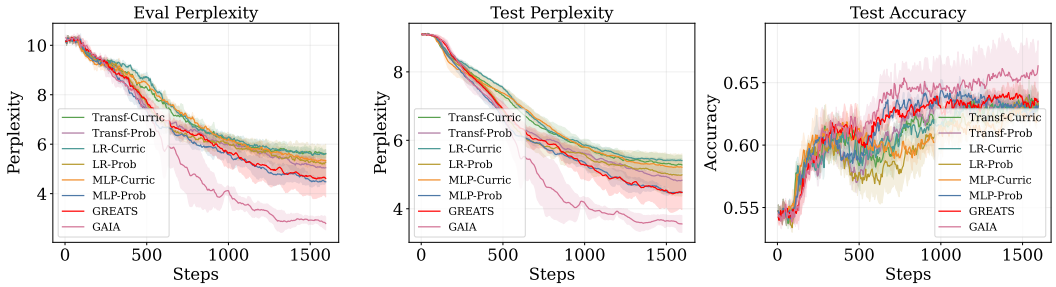


Figure 6: Training dynamics on MMLU-Sociology using the LLAMA-2-7B backbone, comparing downstream performance when using alternative surrogate models for data sampling.

Table 4: Comparison of Test Perplexity on MMLU subsets. **AVG** denotes the mean performance across all five subjects. Bold values indicate the best performance (lower is better).

Method	Anat.	Astro.	B. Ethics	Clin. Know.	Abs. Alg.	AVG
Regular	7.4797	7.7748	8.0511	7.5611	8.5603	7.8854
GREATS	7.2278	**6.1785**	6.5014	6.1407	7.4420	6.6981
GAIA	**5.8763**	6.2896	**5.5351**	**5.2791**	**6.6205**	**5.9201**

- `GREATS`: [7] An efficient online batch selection method that employs a greedy algorithm to optimize data batch quality, approximated via Taylor expansion.
- `GradNorm`: [44] A selection strategy that prioritizes training data points exhibiting the highest gradient norms.
- `MaxLoss`: [45] A strategy that selects training data points with the highest loss values, focusing on "hard" examples.
- `SBERT`: [7] A semantic selection method that curates training batches based on their embedding similarity to the validation set, utilizing Sentence-BERT [35].
- `RHOLoss`: [46] A selection strategy that selects data points by maximizing the difference between the target model's loss and a reference model's loss, thereby filtering out noise while prioritizing learnable examples.
- `DivBS` [24]: A reference-model-free online batch selection method that selects the subset maximizing group-wise orthogonalized representativeness, reducing the redundancy of sample-wise criteria.
- `UDS` [25]: An online batch selection method for LLM supervised fine-tuning that scores utility and intra-sample diversity via the nuclear norm of the logits matrix, and inter-sample diversity against a memory buffer of historical embeddings.

A.7 The Other Surrogate Models

To demonstrate the superiority of Gaussian Processes for guiding data sampling, Figure 6 evaluates the downstream performance of alternative sampling pipelines driven by standard parametric models (Linear Regression, MLP, and Transformer). We trained these alternative surrogates on the 500 warm-up samples to predict global utility scores, followed by data selection using either a deterministic curriculum (Curric) or probabilistic (Prob) sampling strategy. The results clearly show that these alternative sampling methods fall short. **GAIA** significantly and consistently outperforms all baseline sampling configurations across all downstream metrics, achieving substantially lower validation and test perplexity, as well as higher test accuracy. This confirms that our GP-based framework provides a much more robust and effective utility signal for data selection, translating directly to superior downstream task performance

A.8 Performance on TYDIQA

Figure 7 illustrates the convergence and downstream performance on the TYDIQA dataset. Although **GAIA**-`GREATS` exhibits a slightly more gradual decline in evaluation and test perplexity during the initial training phase, it eventually achieves a superior convergence floor compared to `GREATS`.

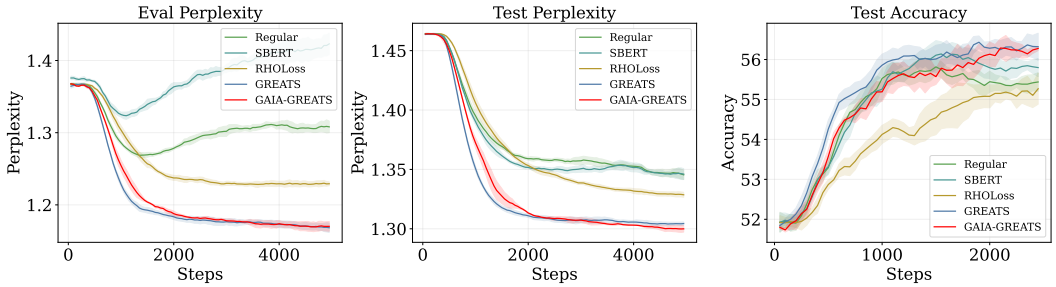
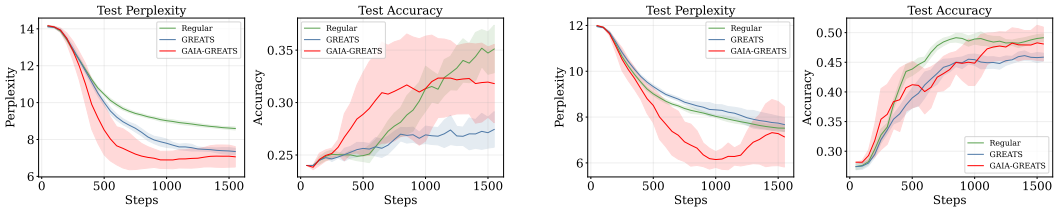
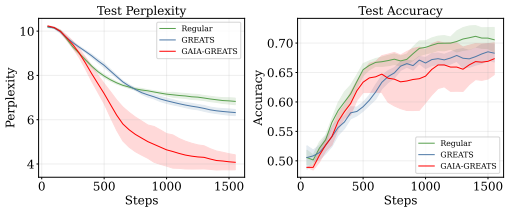


Figure 7: Training dynamics on the TyDiQA benchmark. Our method achieves better perplexity on the test set. All results are averaged over three independent runs.

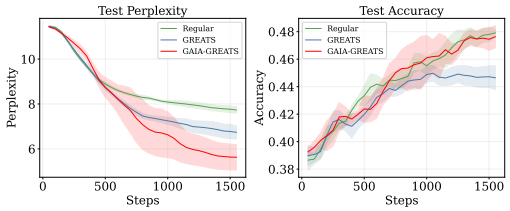


(a) Abstract Algebra

(b) Anatomy



(c) US Foreign Policy



(d) Astronomy

Figure 8: Performance comparison on additional MMLU subjects (*Abstract Algebra, Anatomy, US Foreign Policy, Astronomy*). We evaluate **GAIA**-GREATS against *Regular* training and the GREATS baseline, reporting both test perplexity (PPL) and accuracy. Notably, our method achieves a significant and consistent reduction in PPL across all tasks. While the accuracy metrics exhibit some variance, typical for the inherent stochasticity of multiple-choice tasks, **GAIA**-GREATS maintains competitive performance.

Specifically, the final perplexity levels (both eval and test) of our method are marginally lower than those of the strongest baseline. Furthermore, in terms of downstream task performance, our method maintains an F1 score that is on par with GREATS. This suggests that while our global selection strategy may take longer to identify optimal data patterns on certain datasets, it ultimately reaches a more refined state of model alignment, demonstrating competitive generalization capabilities across diverse linguistic tasks.

A.9 Performance on Additional MMLU Subjects

As shown in Table 4, **GAIA** achieves the lowest average perplexity (5.9201), significantly outperforming the baseline (7.8854) and GREATS (6.6981). These results confirm that **GAIA**'s data selection strategy more effectively captures specialized knowledge than existing approaches. Figure 8 presents a detailed performance comparison across various MMLU subjects, including *Abstract Algebra, Anatomy, US Foreign Policy, and Astronomy*. Our method **substantially outpaces** both the Regular training and GREATS baselines, establishing significantly faster convergence and achieving a **noticeably lower convergence floor** in terms of test perplexity (PPL). This decisive advantage in PPL underscores our method's superior efficiency in identifying and learning from high-utility data points. Regarding downstream accuracy, we observe a higher degree of variance compared to the baselines. This phenomenon stems from our framework's inherent mechanism: by utilizing GREATS as the scoring function—which is fundamentally designed for loss maximization—our approach

amplifies the optimization of PPL. While this leads to a superior state of language modeling, it simultaneously introduces instability in accuracy, a metric further sensitized by the discrete nature of multiple-choice tasks. Nevertheless, the overall results demonstrate that our framework possesses **formidable convergence capabilities** and robust generalization, maintaining highly competitive performance across all benchmarks.

A.10 Kernel Ablation

We evaluate the sensitivity of **GAIA** to the GP kernel at two levels (Table 5). At the surrogate level, we replay the logged training batches against strategy sets generated with each kernel: the best in-set loss differs by less than 0.06 early and 0.12 late in training, well below seed variance. End-to-end, Matérn-5/2 is statistically indistinguishable from SE, rational quadratic is slightly worse, and the rougher Matérn-3/2 lags, which suggests that a smoothness prior suits our embedding-based utility landscape. SE is thus a reasonable default rather than an essential component.

Table 5: Kernel ablation on MMLU-Sociology with `GREATS`. Surrogate level: best in-set top-k loss $\min_{s\in\mathcal{S}} \ell_t(s)$ in the first (Q1) and last (Q4) training quartile, replaying logged batches (4 seeds $\times$ 2 strategy-set draws). End-to-end: new training runs.

	Surrogate: $\min_s \ell_t(s)$		End-to-end	
Kernel	Q1	Q4	Best test PPL	Best acc.
SE (default)	2.91 ± 0.08	3.06 ± 0.20	3.992 ± 0.293	0.6368
Matérn-5/2	2.88 ± 0.05	3.02 ± 0.21	4.233 ± 0.233	0.6368
Rational quadratic	2.90 ± 0.08	3.07 ± 0.24	4.545 ± 0.314	0.6517
Matérn-3/2	2.86 ± 0.08	2.95 ± 0.13	5.526 ± 0.140	0.6418

A.11 Composing GAIA with Diversity-Aware Selectors

`DivBS` [24] and `UDS` [25] select diverse examples from a uniformly sampled candidate batch. **GAIA** can sample that batch instead, using the selector as its refinement step and the selector's scores as GP observations. We compare each selector with and without **GAIA** on MMLU-Sociology and SAMSUM, keeping all other settings fixed.

GAIA composes with both selectors without extra forward or backward passes, and its effect depends on the selector and the task (Table 6). On MMLU-Sociology, it improves `UDS` accuracy on both backbones but not `DivBS` accuracy; on SAMSUM, it improves `DivBS` on both metrics and both backbones but not `UDS` ROUGE-L.

A.12 Adequacy of the Frozen Strategy Set

Replay protocol. The training logs record, at every step, the sampled batch indices, the observed quality scores $q(\mathcal{B}_t)$, and the strategy posterior. We reconstruct the training subset, refit the warm-up GP on the logged $N = 500$ observations with the same configuration as in training (sparse GP with 500 inducing points), and draw strategy sets following Section 4.2. Because the original strategy sets were not stored, these are independent draws from the same procedure; agreement across draws indicates that the results do not depend on a particular draw. For every step and every $s \in \mathcal{S}$ we compute $\ell_t(s) = -\log p_{\text{top-}k}(q(\mathcal{B}_t) \mid s)$; no model forward or backward passes are needed.

Results. Table 7 reports the best in-set loss and the in-set median in the first and last training quartiles, and the number of best-strategy switches K. In every setting the best in-set loss changes little while the median degrades. Table 8 reports the resampling ablation: refitting the GP on all accumulated observations and redrawing $\mathcal{S}$ never improves on the frozen set. We attribute this to the scale non-stationarity of mid-training scores, which pollutes the GP regression targets and motivates the rank-based top-k likelihood.

A.13 Computational Efficiency

The overhead of our framework consists of a one-time cost and a small recurring cost. The one-time cost is embedding acquisition and GP training (Table 9): even on our largest dataset (TYDIQA,

Table 6: Composing **GAIA** with UDS and DivBS (final third of training, mean ± s.d. over seeds 40–43). Alone: the selector on uniformly sampled candidate batches (8, keeping 4); **+GAIA**: on batches sampled by **GAIA**. Bold: the better of each pair. UDS weight $\alpha = 10^{-4}$ for Qwen3-4B and 2×10^{-4} for Llama-2-7B.

		MMLU-Sociology				SAMSUM			
		Accuracy (%) ↑		Test PPL ↓		ROUGE-L (%) ↑		Test PPL ↓	
Backbone	Selector	Alone	**+GAIA**	Alone	**+GAIA**	Alone	**+GAIA**	Alone	**+GAIA**
Llama-2-7B	UDS	$62.94_{\pm 0.75}$	$\mathbf{64.01}_{\pm 1.36}$	$6.47_{\pm 0.04}$	$\mathbf{6.07}_{\pm 0.52}$	$\mathbf{30.18}_{\pm 3.13}$	$27.58_{\pm 2.35}$	$\mathbf{3.80}_{\pm 0.06}$	$3.92_{\pm 0.11}$
	DivBS	$\mathbf{63.34}_{\pm 0.32}$	$61.40_{\pm 3.28}$	$\mathbf{6.02}_{\pm 0.06}$	$6.06_{\pm 0.55}$	$28.17_{\pm 1.10}$	$\mathbf{29.74}_{\pm 1.50}$	$3.62_{\pm 0.03}$	$\mathbf{3.60}_{\pm 0.06}$
Qwen3-4B	UDS	$82.64_{\pm 0.03}$	$\mathbf{82.90}_{\pm 0.34}$	$2.12_{\pm 0.00}$	$2.12_{\pm 0.04}$	$\mathbf{32.91}_{\pm 1.01}$	$32.89_{\pm 1.48}$	$4.38_{\pm 0.04}$	$\mathbf{4.36}_{\pm 0.04}$
	DivBS	$\mathbf{82.98}_{\pm 0.10}$	$82.96_{\pm 0.30}$	$2.15_{\pm 0.01}$	$\mathbf{2.12}_{\pm 0.05}$	$31.64_{\pm 0.29}$	$\mathbf{32.18}_{\pm 1.03}$	$4.27_{\pm 0.02}$	$\mathbf{4.22}_{\pm 0.02}$

Table 7: Replay analysis of the frozen strategy set. Best in-set loss $\min_{s \in \mathcal{S}} \ell_t(s)$ and in-set median $\ell_t(s)$ in the first (Q1) and last (Q4) training quartile (mean ± s.d. across 8 curves: 4 seeds × 2 strategy-set draws). K: number of switches of the best in-set strategy at 100-step granularity; T: training steps.

		$\min_s \ell_t(s)$		median $\ell_t(s)$			
Task	Scorer	Q1	Q4	Q1	Q4	K	T
MMLU-Sociology	GREATS	2.91 ± 0.08	3.06 ± 0.20	7.27 ± 0.75	9.68 ± 1.79	27.1	3264
MMLU-Sociology	GradNorm	2.94 ± 0.11	3.18 ± 0.24	7.43 ± 0.70	11.10 ± 2.66	38.4	4937
MMLU-Sociology	MaxLoss	2.67 ± 0.21	2.89 ± 0.06	6.62 ± 0.39	9.16 ± 0.32	37.8	4937
MMLU-Sociology	RHOLoss	2.83 ± 0.13	2.68 ± 0.09	5.50 ± 0.12	6.88 ± 0.14	10.8	1629
MMLU-Sociology	SBERT	2.86 ± 0.18	2.74 ± 0.07	6.58 ± 0.52	8.15 ± 0.41	32.5	4937
MMLU-US-Foreign-Policy	GREATS	2.92 ± 0.07	2.99 ± 0.15	6.55 ± 0.51	9.26 ± 1.44	30.2	3612
TYDIQA	GREATS	2.94 ± 0.10	3.19 ± 0.14	8.17 ± 0.90	11.06 ± 1.08	41.9	4938

$N = 108{,}271$) these steps take less than 80 seconds, and on MMLU about 23 seconds. The recurring cost per training step consists of strategy and candidate-batch sampling and the Hedge posterior update, which is linear in the number of strategies m. Table 10 breaks down a training step: **GAIA**'s own recurring cost is about 2.6 ms, roughly 0.2% of a step, while the dominant cost is the scoring function itself, which all online selection baselines share.

A.14 Visualization of Sampling Dynamics.

Figure 9 visualizes the indices of sampled data points and their corresponding source strategies to illustrate the sampling behavior of our framework. We observe that the sampling distribution evolves significantly throughout the training process. In the initial phase, the framework exhibits a strong concentration on specific data regions, most notably the dense clusters sampled from strategies 2 and 10, facilitating **rapid targeted learning** on high-utility samples. As training progresses, however, the sampling pattern naturally shifts towards a more dispersed distribution, covering a broader range of the dataset. This temporal evolution demonstrates that our framework effectively prioritizes efficient convergence through focused exploitation in the early stages, before seamlessly transitioning to broader exploration to ensure sufficient data diversity for robust model generalization.

A.15 Effectiveness of Adaptive Sampling Temperature.

To balance exploration and exploitation during the sampling process, we introduce an **adaptive temperature** mechanism (See Section 4.1). In the early stages of training, a lower temperature is employed to prioritize high-utility samples, facilitating rapid model alignment. As training progresses, the temperature is gradually increased to incorporate greater data diversity and prevent overfitting.

To verify its importance, we conduct an ablation study by disabling the adaptive schedule and using a fixed temperature instead. As shown in Figure 3, **w/o TA** results indicate the absence of adaptive temperature leads to significant training instability, with the model often experiencing performance

Table 8: Resampling ablation (MMLU-Sociology, `GREATS`): the GP is refit on all accumulated observations and the strategy set redrawn at the given step; entries are the best in-set loss over the following 500 steps.

Refresh at step	Refreshed set	Frozen set
800	3.07 ± 0.57	2.95 ± 0.34
1600	3.03 ± 0.55	2.99 ± 0.35
2400	3.24 ± 0.65	3.04 ± 0.37

Table 9: Analysis of the additional computational overhead introduced by our method across different benchmarks. We report the training set size and the specific incremental costs: embedding acquisition and Gaussian Process training. All time measurements are reported in seconds.

TASK	SIZE	EMB. TIME (S)	GP TIME (S)
MMLU	13,533	11.68	11.20
SAMSUM	52,002	17.73	10.54
TYDIQA	108,271	67.33	11.55

corruption during the middle stages of training. This empirical evidence confirms that the adaptive temperature mechanism is vital for maintaining a stable optimization trajectory. It effectively prevents the model from over-focusing on a narrow subset of data early on, while ensuring sufficient diversity in later stages to mitigate overfitting and ensure robust convergence.

A.16 Effect of sampling temperature.

The sampling temperature, τ, plays a critical role in governing the sharpness of the sampling distribution derived from the Gaussian Process. As illustrated in Figure 10, results on MMLU-Sociology and MMLU-Anatomy, we observe a consistent pattern across different subjects. A lower temperature (e.g., $\tau = 0.1$) leads to a significantly steeper and more rapid decline in test perplexity during the initial training phase. This is because a lower temperature amplifies the differences between data valuation scores, encouraging the model to aggressively exploit high-utility samples.

In contrast, a higher temperature results in a more gradual and flatter convergence curve. By smoothing the probability distribution, a larger τ increases the diversity of the selected batches but slows down the identification of the most informative data points. While lower temperatures accelerate early-stage convergence, they require a reliable valuation signal to avoid over-focusing on a narrow data subset. These results demonstrate that the sampling temperature serves as a vital trade-off parameter between aggressive exploitation of perceived data utility and the maintenance of batch diversity.

A.17 Embedding methods

We ablate the proxy semantic space using three different sentence transformers in Figure 11a. While all models successfully guide optimization, richer extractors like `all-mpnet-base-v2` and `multi-qa` yield a lower test perplexity floor than the lightweight `all-MiniLM-L6`. This confirms **GAIA**'s robustness to embedding choices, though higher-quality semantic representations explicitly improve the final optimization bounds.

A.18 Embedding Dimension

To accelerate GP inference, **GAIA** uses SVD to project high-dimensional embeddings into a compact space. As shown in Figure 11b, we observe that the model's performance consistently improves as the projection dimension increases. Higher dimensions preserve more comprehensive semantic information from the raw embeddings, leading to faster convergence and lower test perplexity.

Table 10: Per-step cost breakdown on MMLU-Sociology (Llama-2-7B, LoRA, one H100), averaged over ∼3,300 steps per seed across 4 seeds. The step time is the median wall-clock per step, including amortized periodic evaluation.

Component	When	Wall-clock	Share of a step
Embedding + GP fit + strategy sampling	once, after warm-up	≈23 s	one-time
Strategy + candidate-batch sampling	per step	∼2.1 ms	0.14%
Hedge posterior update	per step	∼0.5 ms	0.03%
`GREATS` scoring (ghost inner product)	per step	∼759 ms	53%
Forward/backward + optimizer	per step	∼683 ms	∼47%
Total per step (median)		1.44 s	100%

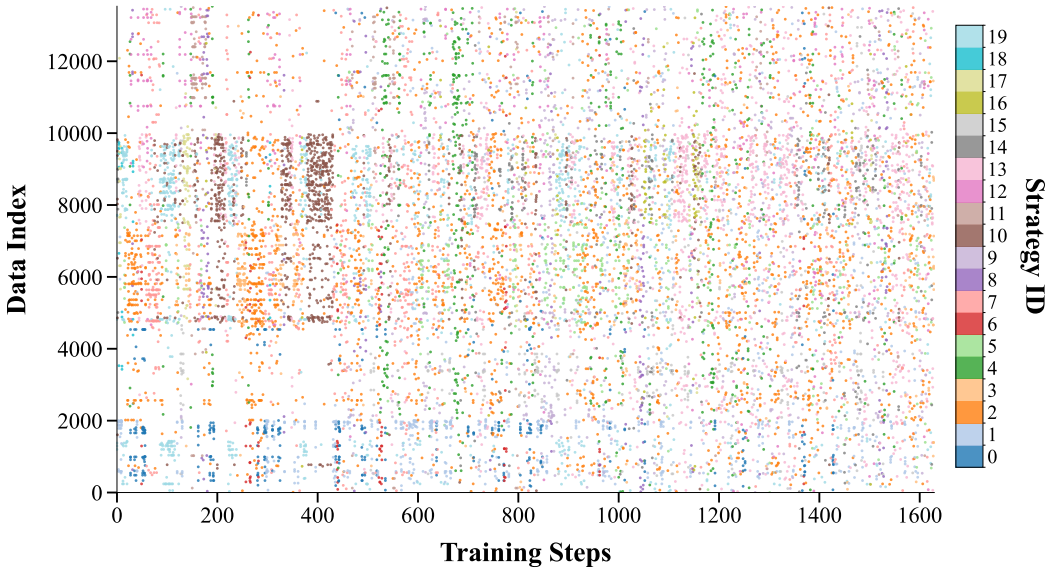
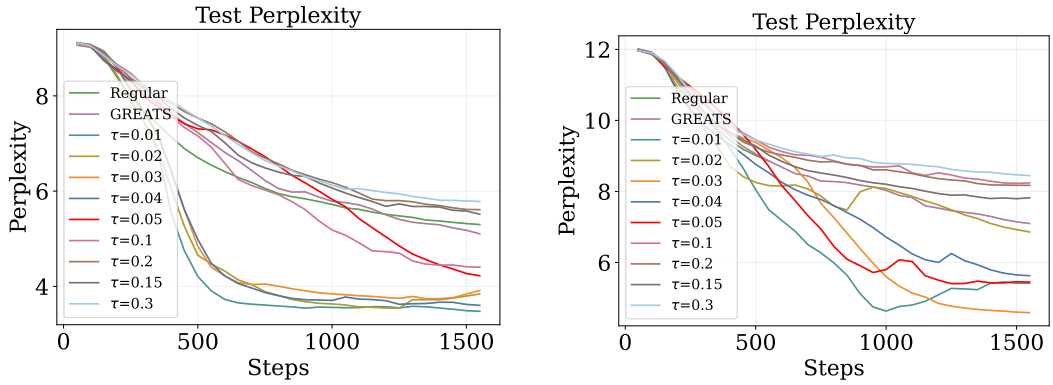


Figure 9: Visualization of sampling dynamics on MMLU-Sociology over the course of training. The scatter plot maps sampled data indices (y-axis) against training steps (x-axis), with colors representing the distinct strategies from which samples were drawn. The distinct clustering observed in the initial phase demonstrates that our method prioritizes high-value data first. As training progresses, the selection becomes increasingly broad, thereby achieving greater diversity in the later stages.

A.19 Warm-up: Impact of Initial Observation Size

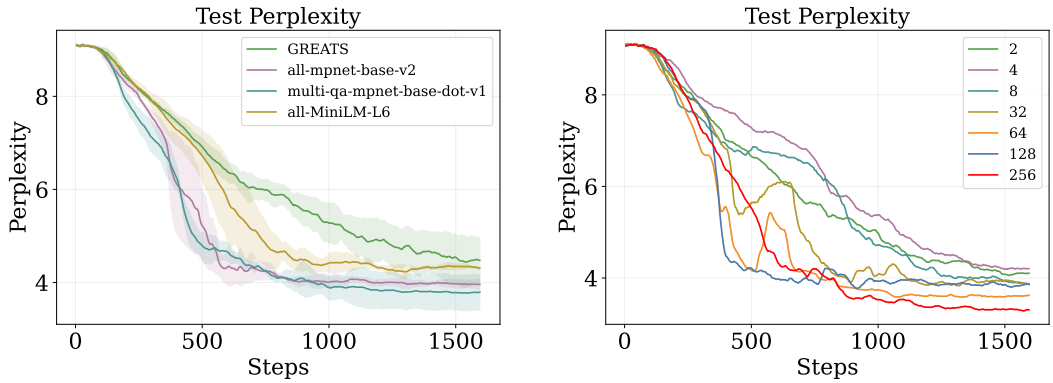
Figure 12 illustrates warm-up dataset size impact on our framework. Extreme undersampling (e.g., $N = 100$) fails to accurately model the utility manifold, while N=200 brings a noticeable improvement. Performance plateaus across $N \in \{300, 500, 1000\}$, indicating that our default N=500 achieves an optimal balance between sample efficiency and computational overhead. Further scaling to $N = 2000$ yields another performance leap, suggesting that a larger initial budget enables higher-resolution manifold modeling for better downstream optimization.



(a) MMLU-Sociology

(b) MMLU-Anatomy

Figure 10: Impact of sampling temperature on test perplexity for MMLU-Sociology and MMLU-Anatomy.



(a) Effect of different pre-trained embedding models on test perplexity (MMLU-Sociology)

(b) Effect of SVD projection dimensionality (d) on test perplexity (MMLU-Sociology)

Figure 11: Embedding ablations on MMLU-Sociology

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our paper defines the GAIA framework in the Abstract and Introduction , detailing three key contributions: global quality modeling via GPs , efficient strategy discretization , and seamless integration with existing methods. These claims are supported by empirical evaluations across three datasets.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

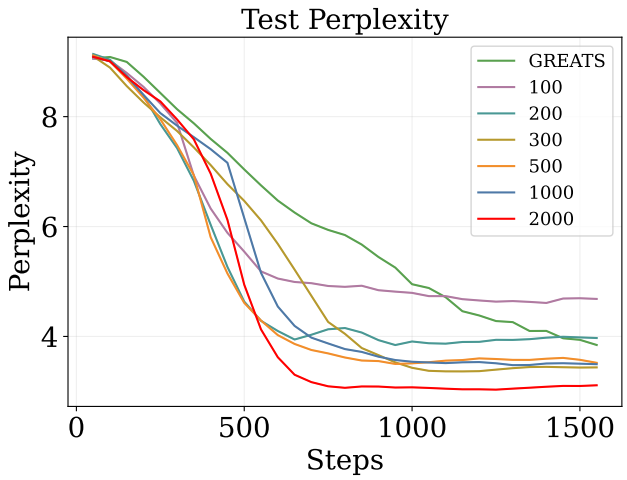


Figure 12: Impact of warm-up dataset size on test perplexity (MMLU-Sociology). We evaluate the effect of varying the number of initial samples used to fit the Gaussian Process prior.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have limitation in Conclusion section

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper presents Theorem 1 regarding dynamic regret in the main text and provides the full set of assumptions, loss boundedness calculations, and a complete proof in Appendix A.3.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We disclose all necessary details for reproduction, including specific LoRA hyperparameters for each task in Appendix A.4 , GP training iterations and noise levels in Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We attached code in supplemental materails

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Comprehensive experimental settings, including learning rates, model backbones, and data splits, are provided in Appendix A.4 and Appendix A.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Most of the main results are reported as the average of three independent runs , and performance metrics include standard deviation to indicate statistical variance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide a detailed efficiency analysis in Table 9, reporting specific incremental costs in seconds for embedding acquisition and GP training across different benchmarks.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research focuses on algorithmic improvements for data selection and adheres to standard ethical practices in machine learning research.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: We are focusing on LLM training algorithm, and aiming to improve the efficiency of training. We do not foresee direct negative societal impacts specific to our contribution.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper doesn't release any data and models

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets, including datasets (MMLU, SAMSUM, TYDIQA) and models (Llama, Qwen, Mistral), are properly cited and credited to their original authors.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: This paper introduces GAIA, a new algorithmic framework for online data selection. While it primarily utilizes existing datasets and model backbones , the documentation for the new method—including training procedures , hyperparameter configurations , and its theoretical foundations, is provided in the main text and the Appendix. And the code will be released after accepted

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This research does not involve crowdsourcing or experiments with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper focuses on the development of a data selection framework for machine learning and did not involve human participants.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: While this research focuses on the fine-tuning of Large Language Models , LLMs were not used as a functional or original component of the core methodology. An LLM was utilized solely for language editing and polishing to improve the clarity of the manuscript.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.