Aligning the Query Space: Greedy Information Projection for Language Model Data Selection

Anonymous ACL submission

Abstract

Data selection for language models is often framed as balancing example quality and diversity. We argue that both are consequences of a more fundamental principle: selected examples should preserve the downstream query space induced by instructions, task signals, or retrieval needs. We present *Greedy Information Projection* (GIP), a query-aligned method that requires only candidate embeddings and a score signal from LLM judgments, metadata, or intrinsic geometry. GIP greedily selects examples whose embedding span explains the largest residual component of task/query scores, yielding a fast matching-pursuit selector. A Gaussian projection view connects this update to maximizing mutual information, equivalently minimizing the residual volume of the query subspace left unexplained by selected data; this explains how quality and diversity emerge from one objective. Empirically, GIP matches or surpasses full-data fine-tuning using small instruction and reasoning subsets, while pretraining and RAG passage-selection experiments show that the same residual principle transfers beyond supervised fine-tuning.

1 Introduction

A central question in training large language models (LLMs) is how to select data. Modern LLMs train on large corpora whose size scales with model capacity (Kaplan et al., 2020; Hoffmann et al., 2022), but once enough data is available, *data quality* becomes the binding constraint on further improvement (Achiam et al., 2023; Grattafiori et al., 2024; Yang et al., 2025). A fundamental challenge in pretraining and instruction tuning is therefore to select a subset that maximizes downstream performance under resource constraints.

Existing approaches for selecting fine-tuning data fall into two broad families: *quality-based* filtering—e.g., LLM-judged scoring (Chen et al., 2023b), perplexity thresholds, or length heuristics (Zhao et al., 2024)—and *diversity-based* construction—e.g., clustering, deduplication (Lee et al., 2022), or coverage objectives (Ge et al., 2024; Bukharin et al., 2024; Fan et al., 2025). Gradient-influence methods such as LESS (Xia et al., 2024) and importance-resampling methods such as DSIR (Xie et al., 2023) provide additional axes but typically require target/validation data, model gradients, or both. These approaches give little theoretical guidance on *why* a particular example is valuable and do not unify quality and diversity under a single objective.

This work. We propose *Greedy Information Projection* (GIP), a query-aligned framework for selecting examples that preserve downstream task directions. We use "query" broadly: it can refer to user instructions, evaluation prompts, reasoning questions, retrieval information needs, or user-intent signals. GIP consumes only (i) feature embeddings F of candidate items and (ii) a score matrix G whose columns are task/query score vectors, such as LLM-judged quality, self-consistency, or simple geometric statistics of F. It selects a subset S whose span explains the remaining score signal through a scalable residual matching-pursuit implementation.

We make four contributions:

- **Query-aligned data selection.** We formulate language-model data selection as preserving a downstream query space induced by instructions, evaluation prompts, reasoning questions, or retrieval needs.
- **Greedy Information Projection.** We give a fast residual matching-pursuit algorithm (Algorithm 1) and flexible practical scoring instantiatiosn.
- **Projection view of quality and diversity.** We show that relevance and diversity arise

from the same residual-projection objective: high-quality examples align with important query directions, while diverse examples cover directions not yet explained by the selected span.

- **Empirical validation across NLP settings.** Across instruction following, mathematical reasoning, pretraining, and RAG, GIP shows promising sample-efficient performance with the same residual rule. With only 1–2% of Alpaca, GIP matches or exceeds the standard Full baseline in 7 of 9 MT-Bench/BBH/Alpaca-Eval-V2 model-metric cells; on GSM8K, it is competitive with gradient-, density-ratio-, and diversity-based baselines and exceeds the Qwen3-8B and Qwen3-32B Full baselines at 10–20% data (Section 5).

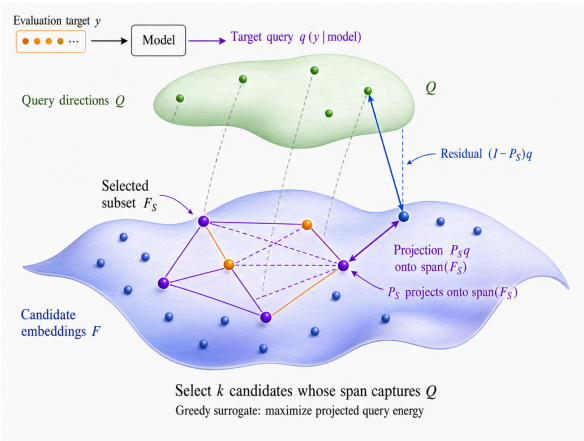


Figure 1: Geometric view of GIP. The selected embeddings F_S define $\mathrm{span}(F_S)$ and the projector P_S; greedy selection increases the query energy captured by $P_S Q$ while leaving smaller residual directions $(I - P_S)Q$.

2 Related Work

Data selection for LLMs. Diversity and quality are both recognized as central to LLM training (Sener and Savarese, 2018; Chen et al., 2024, 2023a). Lee et al. (2022) highlight the role of diversity through near-duplicate removal. Bukharin et al. (2024) balance quality and diversity in instruction-tuning data, and Du et al. (2023) combine quality, diversity, and augmentation. Chen et al. (2023b) use instruction-following difficulty for quality filtering, and Ge et al. (2024) cluster and rank for active retrieval. Xia et al. (2024) use gradient similarity (LESS); Xie et al. (2023) use importance resampling (DSIR); Fan et al. (2025) use iterative informative selection (DiSF). Recent work also selects data by maximizing semantic-space information gain (MIG) (Chen et al., 2025) or by choosing responses that fit the target model distribution (GRAPE) (Zhang et al., 2025). Deb et al. (2025) connect diverse selection to log-likelihood maximization. Existing methods generally treat quality and diversity as separate stages or as penalty terms; GIP unifies them in a single residual-matching procedure derivable from one information-theoretic objective.

Information-theoretic and projection-based selection. Mutual information (MI) has a long history in selection and clustering (Knops et al., 2006; He et al., 2008; Kraskov et al., 2005; Sugiyama et al., 2014; Calandriello et al., 2014; Do et al., 2021). Computing MI is generally hard (Kraskov et al., 2004); we exploit Gaussianity to obtain a closed-form log-determinant criterion. Connections to singular-value-based analysis appear in (Lee and Courtade, 2021, 2020; Lee, 2022). Determinantal point processes (DPPs) (Kulesza et al., 2012; Chen et al., 2018) are the closest kernel-diversity baseline; their score-free entropy objective is the diversity analogue of our score-aware projection objective.

Active learning and coresets. GIP is related to active learning (Settles, 2009) and coreset construction (Bachem et al., 2017). Matching pursuit (Mallat and Zhang, 1993; Pati et al., 1993; Tropp, 2004; Tropp and Gilbert, 2007) provides the algorithmic backbone of our residual-greedy procedure.

3 Information-Projection Formalization

3.1 Setup

Let $\mathcal{F} = \{f_1, \ldots, f_m\} \subset \mathbb{R}^d$ be unit-norm feature embeddings for m candidate examples, stacked as $F \in \mathbb{R}^{d\times m}$ with $F = [f_1, \ldots, f_m]$. Each candidate is associated with n scalar scores, stacked into a score matrix $G \in \mathbb{R}^{m\times n}$ whose columns $g_1, \ldots, g_n \in \mathbb{R}^m$ are different "quality" signals (e.g., LLM judgments, self-consistency, or internal alignment). The selection task is to pick a subset $S \subseteq [m]$ of size at most k that is simultaneously *informative* (aligned with the score signals) and *diverse* (spanning different directions in feature space).

Given the score matrix $G \in \mathbb{R}^{m\times n}$, introduce a latent query embedding $Q \in \mathbb{R}^{d\times n}$ satisfying

$$F^\top Q \approx G. \tag{1}$$

This Q is only a theoretical device: Algorithm 1 runs directly on F and G. It lets us interpret score dimensions as query directions in the same geometry as the candidate embeddings.

3.2 A mutual-information objective

Consider an auxiliary standard Gaussian probe $Z \sim \mathcal{N}(0, I_d)$ and the projections $Z_Q := Q^\top Z \in \mathbb{R}^n$, $Z_{F_S} := F_S^\top Z \in \mathbb{R}^{|S|}$. The pair (Z_Q, Z_{F_S}) is jointly Gaussian with block covariance

$$\Sigma = \begin{bmatrix} Q^\top Q & Q^\top F_S \\ F_S^\top Q & F_S^\top F_S \end{bmatrix}, \tag{2}$$

and, when the selected covariance is nonsingular, the mutual information has the closed form

$$I(Z_Q; Z_{F_S}) = \tfrac{1}{2} \log \frac{\det(Q^\top Q)\det(F_S^\top F_S)}{\det \Sigma}. \tag{3}$$

GIP originates from the intuition to select S by maximizing mutual information; its greedy update is an approximation algorithm for (3).

To avoid invertibility assumptions on the selected examples, the projection view can equivalently be written using any orthonormal basis U_S of $\mathrm{span}(F_S)$, with $P_S = U_S U_S^\top$. Then the conditional covariance of Z_Q given $U_S^\top Z$ is $Q^\top(I - P_S)Q$, so maximizing Gaussian mutual information is equivalent, up to ridge or pseudo-determinant regularization in degenerate cases, to minimizing the residual query volume $\det(Q^\top(I - P_S)Q)$. The determinant objective is the ideal query-space preservation criterion; the GIP greedy update optimizes the trace/matching-pursuit surrogate rather than the exact log-determinant.

Remark (Gaussianity as a modeling device). Neither data nor query embeddings are assumed Gaussian. The probe Z induces Gaussian random projections whose second-order structure is fully determined by the embedding geometry; mutual information admits a closed-form log-determinant expression, is invariant under orthogonal transformations, and captures total linear dependence between the subspaces spanned by Q and F_S.

3.3 Geometric projection

Maximizing mutual information of the form (3) is equivalent to optimizing

$$\arg\max_S (3) = \arg\min_S \det\big(Q^\top(I - P_S)Q\big). \tag{4}$$

The matrix $I - P_S$ projects onto the orthogonal complement of $\mathrm{span}(F_S)$. Minimizing $\det(Q^\top(I - P_S)Q)$ thus selects F_S so that its column span captures as much of Q's column span as possible, simultaneously favoring directions where Q has high energy (quality) and directions that open up new subspace volume (diversity). Figure 1 illustrates the picture.

When scores are absent, Q vanishes from the objective and the MI surrogate reduces to maximizing the entropy $h(Z_{F_S}) = \frac{1}{2}\log\det(F_S^\top F_S)$, which is the determinantal point process diversity objective (Kulesza et al., 2012). GIP thus generalizes DPP-style diversity to score-aware selection through a single formula.

4 Greedy Information Projection: Empirical Method

We arrive at the GIP algorithm in two relaxation steps.

Step 1: AM–GM relaxation to a trace.

$$\det\big(Q^\top(I - P_S)Q\big) \le \left(\frac{\mathrm{Tr}\big(Q^\top(I - P_S)Q\big)}{n}\right)^n.$$

The trace is linear in P_S and therefore amenable to greedy maximization. We adopt it as a surrogate:

$$\mathrm{Tr}\big(Q^\top(I - P_S)Q\big) = \sum_{i=1}^n \|q_i\|^2 - \|P_S q_i\|^2. \tag{5}$$

Note that minimizing the trace depends on Q only through $q_i^\top F = g_i^\top$, so the optimization is computable directly from G without forming Q.

Step 2: Greedy matching pursuit on the dual. Minimizing (5) is equivalent to the multi-target sparse approximation problem

$$\min_S \sum_{i=1}^n \min_{\lambda \in \mathbb{R}^{|S|}} \big\|q_i - \textstyle\sum_{j \in S} \lambda_j f_j\big\|_2^2, \tag{6}$$

because $\|P_S q_i\|^2$ is exactly the energy of q_i explained by the span of F_S. We solve (6) greedily. At step $t+1$, with residual $r_i^{(t)} = q_i - \sum_{j=1}^t \lambda_j^* f_{s_j}$, the optimal single-atom step satisfies $\min_\lambda \|r_i - \lambda f_s\|_2^2 = \|r_i\|^2 - (r_i^\top f_s)^2$ since $\|f_s\| = 1$, and the selection rule reduces to

$$s_{t+1} = \arg\max_{s \notin S_t} \sum_{i=1}^n (r_i^\top f_s)^2. \tag{7}$$

Algorithm 1 GIP greedy selection for data selection

Require: Data $F \in \mathbb{R}^{d\times m}$, scores $G \in \mathbb{R}^{m\times n}$, budget k, residual scale $\lambda \in [0,1]$ (default 1)
Ensure: Selected indices S

```
S ← ∅; W^(0) ← G^⊤
Precompute Φ ← F^⊤ F
for t = 1 to k do
    s_t ← arg max_{j∉S} Σ_{i=1}^{n} (W_{i,j}^{(t-1)})^2
    S ← S ∪ {s_t}
    For all i ∈ [n], j ∉ S: W_{i,j}^{(t)} ← W_{i,j}^{(t-1)} − λ Φ_{j,s_t} W_{i,s_t}^{(t-1)}
end for
return S
```

Defining $W_{i,j}^{(t)} = r_i^{(t)\top} f_j$, the above suggests a residual update $r_i^{(t+1)} = r_i^{(t)} - (r_i^{(t)\top} f_{s_{t+1}}) f_{s_{t+1}}$. Note that because $W_{i,j}^{(0)} = q_i^\top f_j = G_{j,i}$, the algorithm requires only F and G (not Q).

4.1 GIP greedy selection

Based on the above two steps, GIP maintains a *residual score matrix* $W^{(t)} \in \mathbb{R}^{n\times m}$ whose i-th row represents how much of the i-th score signal has not yet been explained by previously selected items, together with a precomputed inner-product matrix $\Phi = F^\top F \in \mathbb{R}^{m\times m}$. Initially, $W^{(0)} = G^\top$. At step t, we select the candidate whose feature explains the most variance jointly across all n residual scores, then subtract the rank-one component aligned with f_{s_t} so that subsequent selections are biased away from items in the same direction:

$$s_t = \arg\max_{j\in[m]\setminus S_{t-1}} \sum_{i=1}^{n} \big(W_{i,j}^{(t-1)}\big)^2, \tag{8}$$

$$W_{i,j}^{(t)} \leftarrow W_{i,j}^{(t-1)} - \lambda\, \Phi_{j,s_t} \cdot W_{i,s_t}^{(t-1)}. \tag{9}$$

The residual scale $\lambda \in [0,1]$ controls how aggressively selected directions are removed. The default $\lambda = 1$ gives the full matching-pursuit residual used in our fine-tuning experiments. The endpoint $\lambda = 0$ masks only the selected item and therefore behaves like greedy top-score selection under the current score matrix, while intermediate values retain more direct query relevance and apply a softer diversity correction. Thus GIP(λ) is a single-parameter balance between score preservation and residual coverage. We repeat for k steps and return $S = S_k$; Algorithm 1 summarizes the procedure.

4.2 Computational complexity

The residual update only needs the similarities $\Phi_{j,s_t} = f_j^\top f_{s_t}$ to the newly selected item. One implementation precomputes the full Gram matrix $\Phi = F^\top F$ at cost $O(m^2 d)$ and memory $O(m^2)$, which can be amortized across budgets or score variants; after that, each greedy step costs $O(mn)$ for selection and $O(mn)$ for the residual update, giving $O(mnk)$ selection time. Alternatively, for large candidate pools, the vector $\Phi_{:,s_t}$ can be computed on the fly in $O(md)$ time per step, avoiding $O(m^2)$ memory at the cost of $O(mdk)$ additional computation. In our main settings n is small, so the cached-kernel selector is effectively linear in m and k after preprocessing.

Table 1 compares against common baselines. GIP uses candidate embeddings together with a score matrix, but does not require held-out validation data or model-gradient computation. Unlike greedy DPP MAP, which scales as $O(mk^2)$ in k, cached-kernel GIP scales as $O(mk)$ in k when the score dimension n is small. Wall-clock and peak-RAM measurements (Table 13) show that the preprocessing cost is amortizable in our evaluated settings.

5 Experiments

We evaluate GIP as a query-aligned selection rule in four NLP regimes. First, instruction-following selection tests whether a small Alpaca subset can preserve broad conversational supervision. Second, mathematical-reasoning selection tests whether the same method can choose target-domain GSM8K examples under exact-answer evaluation. Third, pretraining selection tests whether the residual rule can choose useful web-text chunks before supervised training. Fourth, query-time RAG selection tests whether the residual update transfers without fine-tuning. This organization separates the appli-

Method	Sel. time	Validation Data	Grads
GIP	$O(mnk)$	No	No
DPP	$O(mk^2)$	No	No
LESS	$O(m\, m_{\text{val}}\, d)$	Yes	Yes
DiSF	$\tilde{O}(mkd^2)$	Yes	No
DSIR	$O(md + m_{\text{val}} d)$	Yes	No

Table 1: Selection-phase complexity and requirements; n treated as a small constant, embedding/kernel precomputation excluded. "Validation Data" denotes a held-out validation or target set used by the selector, and "Grads" denotes model-gradient computation. $\tilde{O}$ hides polylog factors.

cation surfaces while keeping the underlying object fixed: select candidates whose span explains the task query signals.

5.1 Instruction following and mathematical reasoning

We first evaluate GIP in supervised fine-tuning for instruction following and mathematical reasoning. The Alpaca experiments measure broad instruction-following transfer from small general-purpose subsets, while the GSM8K experiments measure whether selected reasoning examples improve exact-answer mathematical performance across data budgets and model scales. For both tasks, candidates are represented by normalized ModernBERT-Base embeddings (Warner et al., 2024); Alpaca examples concatenate *[instruction; context; response]*, and GSM8K uses the question as the instruction.

Task setup. For instruction tuning, we select 512 and 1,000 examples from Alpaca-52k (Taori et al., 2023; Stanford, 2023) and evaluate Mistral-7B-v0.1 (Jiang et al., 2023), Qwen3-4B, and Qwen3-8B (Yang et al., 2025) on MT-Bench (Zheng et al., 2023), BBH (Suzgun et al., 2022), and Alpaca-Eval-V2 (Dubois et al., 2024). For mathematical reasoning, we select 2.5–20% of GSM8K (Cobbe et al., 2021) and evaluate exact-match accuracy on Mistral-7B, Qwen3-8B, and Qwen3-32B. Fine-tuning baselines include Random, Full, LIMA (Zhao et al., 2024), Alpagasus (Chen et al., 2023b), CaR (Ge et al., 2024), DSIR (Xie et al., 2023), DiSF (Fan et al., 2025), and LESS (Xia et al., 2024); DSIR receives the evaluation set as its target domain, giving it an oracle advantage on GSM8K.

Fine-tuning evaluation protocol. All selected subsets and baselines are fine-tuned from the same base checkpoint under the shared LoRA recipe in Table 7. MT-Bench responses are generated with vLLM (Kwon et al., 2023) and judged by GPT-5 on the standard 1–10 MT-Bench scale; Alpaca-Eval-V2 reports length-controlled win rate; BBH and GSM8K use exact-match accuracy with the benchmark's standard answer normalization. For Qwen3 models, thinking mode is disabled during both fine-tuning and evaluation.

Baseline implementation. For Alpaca, we evaluate the released or paper-specified selection rules at matched 1k-example budgets where applicable (LIMA, Alpagasus, CaR) and retrain all reported baselines with the same fine-tuning and evaluation pipeline. For GSM8K, Random, DSIR, DiSF, LESS, and GIP variants use the same percentage budgets. DSIR and LESS use the same target/validation split for distribution matching or gradient influence, so they are strong target-aware baselines, while GIP uses only candidate embeddings plus its score signal.

Instruction-tuning scoring schemes. GIP accepts any score matrix G. For fine-tuning selection, we study two practical instantiations. First is multi-attribute scoring (MA): for every candidate, we concatenate *[instruction; context; response]* and ask GPT-4o to score coherence, accuracy, helpfulness, and difficulty on a 0–5 scale. We cache these four attributes and use their sum as a single score vector ($n = 1$); the full prompt is in Table 10. The second variant is training-data-aware self-compression (SC) that uses no external judge and derives the score from the data geometry:

$$g_i = \sum_{j=1}^{m} f_i^\top f_j = (F^\top F\, \mathbf{1}_m)_i.$$

This row-sum score measures representativeness in the candidate pool. Combined with the residual update (9), GIP+SC trades representativeness against coverage.

Sample efficiency. Table 2 shows the strongest data-efficiency result: with only 1–2% of Alpaca, GIP+MA/GIP+SC exceed the Full-data baseline in 7 of 9 model-metric cells across MT-Bench, BBH, and Alpaca-Eval-V2. Table 3 shows more mixed but still favorable reasoning results: GIP+MA exceeds the Qwen3-8B and Qwen3-32B Full baselines at 10–20% data. We do not claim that small subsets dominate all possible full-data training protocols; rather, under a fixed practical fine-tuning budget, GIP-selected subsets often match or exceed the standard full-data baseline, especially when the source pool contains noisy or redundant examples. Because Alpaca is noisy and redundant, 1% can occasionally outperform 2% after the highest-marginal-value examples have already been selected.

Robustness across models. Gains are stable across Mistral and Qwen3 at 4B–32B parameters and across benchmarks probing distinct capabilities. The strongest non-GIP baseline on GSM8K

Table 2: Performance on MT-Bench, BBH, and Alpaca-Eval-V2. Alpaca-Eval-V2 reports LC win% (Dubois et al., 2024). Best results in **bold**, second-best underlined. Standard errors are in Table 20.

Method	Data	Qwen3-4B			Mistral-7B			Qwen3-8B		
		MT-B.	BBH	A.-Eval	MT-B.	BBH	A.-Eval	MT-B.	BBH	A.-Eval
Full	100% (52K)	6.69	73.20	**4.88**	3.64	**58.40**	5.77	7.74	76.70	6.91
Random	2% (1K)	7.59	69.40	2.62	3.66	57.40	4.71	7.98	72.30	4.99
LIMA	2% (1K)	7.45	73.30	2.48	3.89	55.60	2.69	8.20	75.80	4.24
Alpagasus	2% (1K)	7.59	63.60	4.17	3.71	56.50	4.81	7.86	71.90	5.51
CaR	2% (1K)	7.68	73.80	3.26	3.83	57.40	4.49	8.16	76.20	3.66
GIP+MA	2% (1K)	7.67	74.30	3.74	3.83	57.60	6.22	7.99	**80.50**	5.64
GIP+MA	1% (512)	**7.78**	68.00	4.42	**3.99**	56.60	**6.59**	**8.26**	79.10	4.85
GIP+SC	2% (1K)	7.53	**74.40**	3.34	3.71	57.70	4.83	8.04	79.10	**7.05**
GIP+SC	1% (512)	7.58	73.90	3.23	3.71	57.40	4.85	7.88	76.50	4.72

Table 3: GSM8K mathematical reasoning across models and data percentages. Best and second-best marks are computed within the budgeted selection rows; Full is shown as the 100% reference. Standard errors are in Table 21.

Method	Mistral-7B				Qwen3-8B				Qwen3-32B			
	2.5%	5%	10%	20%	2.5%	5%	10%	20%	2.5%	5%	10%	20%
Full	52.46				81.96				87.64			
Random	37.76	43.52	45.72	46.74	78.32	80.21	79.53	81.65	78.24	87.14	85.67	87.29
DiSF	39.88	41.55	41.55	42.46	75.06	79.98	79.98	80.67	81.65	87.04	87.04	88.40
DSIR	37.07	42.15	42.61	42.08	80.74	**81.50**	81.96	**83.24**	80.74	**89.16**	87.79	88.70
LESS	31.84	33.13	42.91	47.69	79.76	79.45	80.29	79.45	69.98	87.26	87.95	87.41
GIP+MA	**42.99**	**45.64**	45.94	47.46	**81.58**	81.05	**83.09**	**83.24**	**84.31**	87.57	**88.25**	**88.78**
GIP+SC	38.89	43.29	**46.63**	**49.81**	80.36	80.21	81.65	82.26	84.15	87.04	87.34	87.56

is DSIR, but its edge is concentrated at the 5% budget on the Qwen3 models; at the other budgets, GIP+MA matches or exceeds DSIR across every base model. On Mistral-7B, GIP+MA consistently leads DSIR. Note that DSIR enjoys oracle access to the evaluation distribution via its target-domain mechanism, whereas GIP does not.

Complementarity of GIP+MA and GIP+SC. GIP+MA performs best on instruction-following and preference metrics by exploiting external multi-attribute judgments, while GIP+SC—using only intrinsic geometry—is competitive on reasoning benchmarks and excels on GSM8K with Mistral-7B at 10–20%. This pattern supports our view that quality and diversity are jointly handled by the same residual mechanism, with the score channel determining which signal is emphasized.

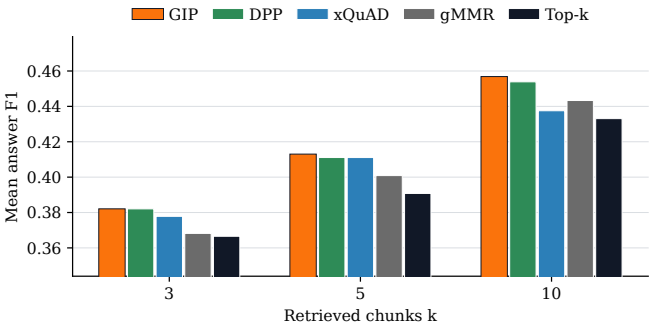
Ablations. We include three targeted ablations for training-time selection. First, to test whether GIP replaces or complements source-pool cleaning, we compare GIP+SC at 1% data against Full on Mistral-7B across MT-Bench categories for cleaned and non-cleaned Alpaca (Figure 2b). Cleaning improves both methods, especially on open-ended categories, and narrows the gap enough that 1% GIP+SC matches cleaned Full overall. Thus GIP+SC benefits from source-pool quality rather than replacing it.

Second, to measure sensitivity to the feature geometry, we swap the encoder feeding GIP+SC on GSM8K between ModernBERT and a Qwen-reasoning embedding. Qwen-reasoning gives small gains on most rows, particularly for Qwen3-32B at 10–20%; ModernBERT remains competitive, especially on Mistral-7B at 10% (Table 4).

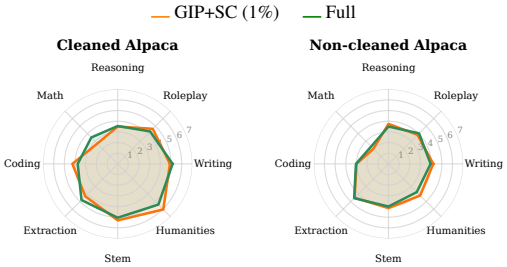
Table 4: Embedding ablation for GIP+SC on GSM8K.

Model	Data	ModernBERT	Qwen-reasoning
Mistral-7B	20%	49.81	**50.27**
Mistral-7B	10%	**46.63**	46.25
Qwen3-32B	20%	87.57	**88.02**
Qwen3-32B	10%	87.34	**87.72**

Finally, to test selection stability under embedding perturbations, we add Gaussian noise to GSM8K embeddings and rerun GIP+SC. Selected-subset IoU remains $\geq 85\%$ for $\sigma \leq 10^{-3}$ and above 60% at $\sigma{=}10^{-2}$, so mild embedding drift has little effect and severe noise still preserves most core items (Table 12).



(a) RAG oracle mean answer F1 by context budget.



(b) MT-Bench per-category scores on Mistral-7B.

Figure 2: RAG and MT-Bench ablations. Left: oracle best-over-sweep mean answer F1 grouped by retrieved-context budget; tunable methods are swept over their tradeoff parameters. Right: GIP+SC at 1% data versus Full on cleaned and non-cleaned Alpaca, judged by GPT-5.

5.2 Pretraining data selection

We next test whether the same query-aligned selection view transfers to language-model pretraining, where data curation is commonly studied through filtering, deduplication, and benchmark-oriented data selection (Lee et al., 2022; Chen et al., 2023a; Li et al., 2024). From a 1.69M-chunk DCLM-Baseline candidate pool (Li et al., 2024), each method samples a selected training stream at two token budgets, 0.8B and 1.6B tokens, from the same full pool. We then train the same 411M-parameter OpenLM model (Gururangan et al., 2023) on each selected stream, so the comparison varies only the selector and training-token budget while holding the source pool and training recipe fixed.

We compare against random sampling, DSIR importance resampling (Xie et al., 2023), and DiSF relevance-diversity selection (Fan et al., 2025). All target-aware selectors use the same target data: 10k chunks from the English Wikipedia data corpus (Wikimedia Foundation, 2023), tokenized and chunked with the same pipeline as the DCLM pool. This target data is separate from the downstream evaluation tasks; HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), ARC (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2020), OpenBookQA (Mihaylov et al., 2018), and SciQ (Welbl et al., 2017) are used only after training.

This design fixes the candidate pool and target source while varying the selection objective and feature representation. DSIR consumes the shared target through hashed token n-gram distributions, whereas DiSF and GIP use dense ModernBERT chunk embeddings. For scalable pretraining selection, GIP clusters the target embeddings into C target modes using the k-means algorithm, replacing the full set of target directions with a small set of mode centroids before applying the residual update (as in Algorithm 1). For model training, all methods use AdamW (Loshchilov and Hutter, 2019) with learning rate $3 \cdot 10^{-4}$, sequence length 2048, bfloat16 mixed precision, and a cosine schedule. Evaluation uses lm-evaluation-harness (Gao et al., 2024) with a limit of 1000 examples per task; metrics are normalized accuracy for HellaSwag, ARC, and OpenBookQA, and accuracy for PIQA, WinoGrande, and SciQ. Additional implementation details are in Appendix C.

0.8B training tokens								
Selector	**HS**	**PIQA**	**ARC-E**	**ARC-C**	**WG**	**OBQA**	**SciQ**	**Mean**
Random	0.298	0.544	0.300	0.198	0.489	0.250	0.335	0.345
DSIR	0.290	**0.545**	0.296	**0.218**	0.494	**0.262**	0.352	0.351
DiSF	0.299	0.543	0.315	**0.218**	0.491	**0.262**	0.340	0.353
GIP	**0.306**	0.540	**0.317**	0.217	**0.496**	0.260	**0.390**	**0.361**
1.6B training tokens								
Selector	**HS**	**PIQA**	**ARC-E**	**ARC-C**	**WG**	**OBQA**	**SciQ**	**Mean**
Random	0.317	0.572	0.332	0.205	0.498	0.236	0.475	0.376
DSIR	0.326	**0.590**	0.331	0.211	**0.504**	0.242	0.513	0.388
DiSF	0.320	0.585	0.339	**0.215**	0.499	0.240	0.500	0.385
GIP	**0.334**	0.574	**0.350**	0.214	0.480	**0.262**	**0.517**	**0.390**

Table 5: Full pretraining selector results with a 411M-parameter OpenLM model. Best values are bold and second-best values are underlined within each metric and compute point.

Table 5 reports seven-task evaluations for the pretrained models. GIP achieves consistent lift over baselines at both compute points.

5.3 Retrieval-Augmented Question Answering

We next evaluate the same residual mechanism when selection happens at inference time rather

than during training. The primary setting is RAG passage selection, where each question induces a score vector over candidate document chunks. Our reported GIP method is the query-only residual-scaled GIP(λ) variant from (9); baselines are vanilla top-k, gMMR, xQuAD, and DPP(α) (Kulesza et al., 2012; Chen et al., 2018).

Experiment setting. For retrieval-augmented QA, we use LongBench HotpotQA (Yang et al., 2018), 2WikiMQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and MultiFieldQA-EN (Bai et al., 2023). We evaluate 200 examples for the first three datasets and 150 for MultiFieldQA. We split contexts into 200-word passages and encode questions, planner subquestions, and passages as normalized vectors with the Sentence-Transformers Multi-QA MPNet base cosine encoder, a semantic-search model trained on question–answer pairs (Sentence-Transformers, 2021; Reimers and Gurevych, 2019; Song et al., 2020). For the main RAG comparison, GIP uses the original question embedding as Q. We evaluate retrieved-context budgets $k \in \{3, 5, 10\}$, generate answers with Qwen3-8B (Yang et al., 2025) with thinking disabled, and score answer F1. We sweep GIP(λ) over $\{0, 0.3, 0.6, 0.8, 1.0\}$ and DPP(α) over $\{0.5, 1, 2, 4, 8\}$; fixed baselines are vanilla top-k, gMMR (Carbonell and Goldstein, 1998), and xQuAD (Santos et al., 2010).

Figure 2 summarizes the macro answer-F1 frontier across budgets, with per-benchmark values in Table 14. GIP is strongest at $k = 5$ and $k = 10$ and is effectively tied with DPP at $k = 3$. DPP remains the strongest competitor.

Impact of query depth. To test whether GIP benefits from a query space beyond a single question vector, we augment each question with planner-generated decompositions. We use Qwen2.5-7B-Instruct (Qwen et al., 2025) as a deterministic planner to generate 2–5 atomic subquestions, embed them with the same retriever encoder, and append them as additional score columns. Thus $d_Q = 1$ uses only the original question, while $d_Q > 1$ uses the question plus up to $d_Q - 1$ planner steps; if fewer steps are available, we use all of them. These extra columns ask GIP to cover multiple decomposed information needs rather than a single aggregate query embedding. Table 6 shows that planner dimensions improve the fixed-λ, $k = 5$ ablation, suggesting that decomposition helps GIP cover complementary information needs.

Table 6: RAG query-dimensionality ablation at $k = 5$ with fixed $\lambda = 0.8$. Best in **bold**, second-best underlined.

d_Q	Hotpot	2Wiki	MultiField	MuSiQue	Mean
1	0.498	0.426	0.415	0.276	0.404
3	0.486	0.442	**0.424**	0.283	0.409
5	**0.511**	**0.448**	0.422	**0.285**	**0.416**

6 Conclusion

We introduced GIP, a query-aligned data-selection method that greedily selects examples whose embeddings explain the remaining task/query score directions. The method uses only candidate embeddings and score signals, can reuse or stream pairwise similarities during matching-pursuit selection, and avoids model-gradient computation. Across instruction tuning, mathematical reasoning, pretraining, and RAG passage selection, the same residual update yields sample-efficient subsets and competitive retrieval frontiers. The projection view connects this update to mutual information and gives a unified explanation: high-quality examples align with important query directions, while diversity comes from covering directions not yet explained.

Limitations

Embedding dependence. GIP relies on encoder geometry, so poor embeddings or domain-mismatched encoders can weaken capability coverage. **Quality signals.** Input score signals can carry bias, so GIP may under-cover examples undervalued by those signals. **Scope.** We study instruction tuning, reasoning, pretraining, and RAG in English text settings; code, multilingual, multimodal, and much larger pretraining pools remain future work.

References

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

Olivier Bachem, Mario Lucic, and Andreas Krause. 2017. Practical coreset constructions for machine learning. In *International Conference on Machine Learning*, pages 311–319. PMLR.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, and 1 others. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

Alexander Bukharin, Shiyang Li, Zhengyang Wang, Jingfeng Yang, Bing Yin, Xian Li, Chao Zhang, Tuo Zhao, and Haoming Jiang. 2024. Data Diversity Matters for Robust Instruction Tuning. *Preprint*, arXiv:2311.14736.

Daniele Calandriello, Gang Niu, and Masashi Sugiyama. 2014. Semi-supervised information-maximization clustering. *Neural networks*, 57:103–111.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336.

Daoyuan Chen, Yilun Huang, Zhijian Ma, Hesen Chen, Xuchen Pan, Ce Ge, Dawei Gao, Yuexiang Xie, Zhaoyang Liu, Jinyang Gao, Yaliang Li, Bolin Ding, and Jingren Zhou. 2023a. Data-Juicer: A One-Stop Data Processing System for Large Language Models. *Preprint*, arXiv:2309.02033.

Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I. Abdin. 2024. On the Diversity of Synthetic Data and its Impact on Training Large Language Models. *Preprint*, arXiv:2410.15226.

Laming Chen, Guoxin Zhang, and Eric Zhou. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. *Advances in Neural Information Processing Systems*, 31.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and 1 others. 2023b. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Yicheng Chen, Yining Li, Kai Hu, Zerun Ma, Haochen Ye, and Kai Chen. 2025. Mig: Automatic data selection for instruction tuning by maximizing information gain in semantic space. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9902–9915.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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

Kien Do, Truyen Tran, and Svetha Venkatesh. 2021. Clustering by maximizing mutual information across views. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9928–9938.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. 2023. MoDS: Model-oriented Data Selection for Instruction Tuning. *Preprint*, arXiv:2311.15653.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Ziqing Fan, Siyuan Du, Shengchao Hu, Pingjie Wang, Li Shen, Ya Zhang, Dacheng Tao, and Yanfeng Wang. 2025. Combatting dimensional collapse in llm pre-training data via diversified file selection. *Preprint*, arXiv:2504.20644.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Yuan Ge, Yilun Liu, Chi Hu, Weibin Meng, Shimin Tao, Xiaofeng Zhao, Hongxia Ma, Li Zhang, Boxing Chen, Hao Yang, and 1 others. 2024. Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation. *arXiv preprint arXiv:2402.18191*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Suchin Gururangan, Mitchell Wortsman, Samir Yitzhak Gadre, Achal Dave, Maciej Kilian, Weijia Shi, Jean Mercat, Georgios Smyrnis, Gabriel Ilharco, Matt Jordan, Reinhard Heckel, Alex Dimakis, Ali Farhadi, Vaishaal Shankar, and Ludwig Schmidt. 2023. open_lm: A minimal but performative language modeling repository. GitHub repository.

Zengyou He, Xiaofei Xu, and Shengchun Deng. 2008. k-anmi: A mutual information based clustering algorithm for categorical data. *Information Fusion*, 9(2):223–233.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

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. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Zeger F Knops, JB Antoine Maintz, Max A Viergever, and Josien PW Pluim. 2006. Normalized mutual information based registration using k-means clustering and shading correction. *Medical image analysis*, 10(3):432–439.

Alexander Kraskov, Harald Stögbauer, Ralph G Andrzejak, and Peter Grassberger. 2005. Hierarchical clustering using mutual information. *Europhysics Letters*, 70(2):278.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 69(6):066138.

Alex Kulesza, Ben Taskar, and 1 others. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating Training Data Makes Language Models Better. *Preprint*, arXiv:2107.06499.

Kuan-Yun Lee. 2022. *New Information Inequalities with Applications to Statistics*. University of California, Berkeley.

Kuan-Yun Lee and Thomas A Courtade. 2020. Linear models are most favorable among generalized linear models. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 1213–1218. IEEE.

Kuan-Yun Lee and Thomas A Courtade. 2021. Minimax bounds for generalized pairwise comparisons. In *2021 International Conference on Machine Learning (ICML) Workshop on Information-Theoretic Methods for Rigorous, Responsible, and Reliable Machine Learning*.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Kumar Guha, Sedrick Keh, Kushal Arora, and 1 others. 2024. Datacomp-lm: In search of the next generation of training sets for language models. In *Advances in Neural Information Processing Systems*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Stéphane G Mallat and Zhifeng Zhang. 1993. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. 1993. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740.

Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web*, pages 881–890.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. *Preprint*, arXiv:1708.00489.

Sentence-Transformers. 2021. Sentence-transformers multi-qa mpnet base cosine model. Hugging Face model repository.

Burr Settles. 2009. Active learning literature survey. Technical Report CS-TR-1648, University of Wisconsin–Madison.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867.

Stanford. 2023. Stanford Alpaca: An Instruction-following LLaMA Model. Tatsu's shared repositories.

Masashi Sugiyama, Gang Niu, Makoto Yamada, Manabu Kimura, and Hirotaka Hachiya. 2014. Information-maximization clustering based on squared-loss mutual information. *Neural Computation*, 26(1):84–131.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Joel A Tropp. 2004. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information theory*, 50(10):2231–2242.

Joel A Tropp and Anna C Gilbert. 2007. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, and 1 others. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106.

Wikimedia Foundation. 2023. English wikipedia dump, november 2023. Data dump from 2023-11-01.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Dylan Zhang, Qirun Dai, and Hao Peng. 2025. The best instruction-tuning data are those that fit. *arXiv preprint arXiv:2502.04194*.

Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. 2024. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning. *arXiv preprint arXiv:2402.04833*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Reproducibility

We release code, configuration files, and evaluation prompts to reproduce the main algorithm and experiments. **Data and licenses.** Training data: Alpaca-52k (Taori et al., 2023; Stanford, 2023) (CC BY-NC 4.0), the GSM8K training split (Cobbe et al., 2021), and DCLM-Baseline (Li et al., 2024) for the pretraining pilot. Evaluation: MT-Bench (Zheng et al., 2023), BBH (Suzgun et al., 2022), Alpaca-Eval-V2 (Dubois et al., 2024), the held-out GSM8K split, and lm-evaluation-harness tasks (Gao et al., 2024). Links and license terms are listed in the README. **Environment.** Experiments were run on NVIDIA A100 80GB and V100 32GB GPUs, depending on the model and experiment. We provide a `conda` YAML and exact package versions. **Selection.** The implementation exposes Algorithm 1 and the task-specific GIP scoring variants. **Evaluation.** We reuse the prompts in (Zheng et al., 2023) and (Gao et al., 2024) for GSM8K/BBH (exact-match with standard normalization), and the MT-Bench judge templates. Fine-tuning tables report mean $\pm$ standard error across data instances where available; the pretraining pilot is single-seed and reports limit-1000 task metrics. **Result tables.** The instruction-tuning and GSM8K main/appendix tables are generated from the same canonical result file with the released table-building script, and best/second-best formatting is computed automatically. **Determinism.** We fix seeds at all stages where the framework exposes deterministic control (data loader, selection RNG, CUDA/cuDNN, model training). The pretraining pilot uses one data-selection and training seed, so run-to-run variance remains unmeasured.

B Training Details

Hyperparameters and training details are summarized in Table 7. All models are trained in the Hugging Face framework with LoRA fine-tuning (Hu et al., 2021). We apply LoRA to all linear layers for Mistral and Qwen3 runs with $\{r = 8, \alpha = 16\}$. For Qwen3 models we disable thinking mode during supervised fine-tuning and downstream evaluation. For fair comparison with baselines, we use the same context length and epoch budget even when these differ from (Chen et al., 2023b). The Qwen3-32B GSM8K runs use the same Qwen3 LoRA configuration as the other Qwen3 models, but train on GSM8K subsets rather than Alpaca-derived subsets.

C Pretraining Details

Candidate pool and target. The candidate pool contains 1,685,901 DCLM-Baseline chunks, tokenized with the GPT-NeoX tokenizer (Black et al., 2022). For each selector and compute point, we materialize a WebDataset training stream sampled from this full pool at the target token budget: 0.8B or 1.6B tokens. The target distribution was intended to mix Wikipedia and books, but both book sources failed to load at runtime. The final target is therefore 10k chunks from the English Wikipedia data corpus, using the November 2023 dump (Wikimedia Foundation, 2023).

Feature views. We compute two reusable feature matrices. DSIR uses a 10k-dimensional sparse token n-gram view: unigram buckets are token ID modulo 10k, bigram buckets use a multiplicative hash modulo 10k, and vectors are L1-normalized. GIP and DiSF use 768-dimensional ModernBERT dense embeddings (Warner et al., 2024): the first 512 decoded tokens of each chunk are embedded, mean-pooled over unmasked positions, cast from bfloat16 to float32, and L2-normalized. The dense pool feature matrix is 5.18 GB; the sparse n-gram matrix is 67.4 GB.

Selectors. Random uniformly samples pool indices up to the requested token budget. DSIR computes n-gram importance weights against the same wiki-only target distribution and samples without replacement using Gumbel top-k. DiSF uses a relevance-plus-diversity greedy score $\beta\,\mathrm{rel}_i + (1-\beta)(-\max_{s\in S} f_i^\top f_s)$ with $\beta = 0.5$ and omits the original sparse-filtration pre-pruning step.

For the main pretraining row, GIP uses target modes rather than a single target centroid. We run k-means on the target embeddings to obtain centroids $\{c_\ell\}_{\ell=1}^{8}$ and initialize

$$m_i = N \max_\ell \max(0, f_i^\top c_\ell).$$

At each step, GIP selects $\arg\max_i m_i$ and applies the residual update

$$m_i \leftarrow m_i - r\, m_s\, f_i^\top f_s,$$

with residual scale $r = 1$ and no appended bias term. Selection is computed in an 8-GPU sharded mode, where each GPU greedily selects its share of the target budget from its shard and selected ids are gathered at the end. This makes selection

Table 7: LoRA fine-tuning hyperparameters. Qwen3-32B uses the same LoRA configuration as the other Qwen3 models.

Dataset	Size	GPUs	Epochs	LR	Sched.	Ctx. Len.
Alpaca-52k	52k	8	10	3e-5	Linear	2048
Alpagasus-1k	1k	4	10	3e-5	Linear	2048
CaR-1k	1k	4	10	3e-5	Linear	2048
Random-1k	1k	4	10	3e-5	Linear	2048
GIP-512	512	4	10	3e-5	Linear	2048

much faster, but it also removes cross-shard residual penalties; the target-mode and systems changes are therefore confounded in this pilot.

For comparison, the legacy centroid variant, reported as GIP-cent. in Table 8, computes $m_i = N \max(0, f_i^\top \bar{q})$ from the target centroid $\bar{q} = N^{-1} \sum_{j=1}^{N} q_j$ and updates

$$m_i \leftarrow m_i - r\, m_s \,\mathrm{sim}(i, s),$$
$$\mathrm{sim}(i, s) = \tfrac{1}{2}(f_i^\top f_s + 1),$$

with residual scale $r = 1$. This scalar-centroid implementation is a memory-saving rewrite of the sum-score residual update; max/top-k scoring would require keeping the full candidate-target score matrix.

Training and evaluation. All selectors train the same 411M-parameter OpenLM model (Gururangan et al., 2023) on 2 nodes × 8 A100-80GB GPUs. The model has 24 decoder blocks, hidden size 1024, 8 attention heads, sequence length 2048, and the GPT-NeoX vocabulary. The recipe uses DDP, gradient checkpointing, per-GPU batch size 32, global batch size 512, AdamW (Loshchilov and Hutter, 2019) with learning rate $3 \cdot 10^{-4}$, weight decay 0.1, cosine schedule, 200 warmup steps, bfloat16 mixed precision, and sequence length 2048. The 0.8B and 1.6B runs use separately materialized selected streams from the same full candidate pool, so the larger budget increases the amount of selected training data rather than replaying a fixed 0.8B-token subset. Evaluation uses lm-evaluation-harness (Gao et al., 2024) with limit 1000 on HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), ARC-Easy/Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2020), OpenBookQA (Mihaylov et al., 2018), and SciQ (Welbl et al., 2017).

Clustering. The wiki-only target makes target-distribution coverage an imperfect proxy for downstream usefulness. We cluster target embeddings into eight modes.

D Multi-Attribute Scoring Prompt

The prompt used for the GIP+MA scoring is shown in Table 10. The rubric contains four sections (Coherence, Accuracy, Helpfulness, Difficulty), each rated on a 0–5 scale; we use the total as the score signal.

E Proofs

E.1 Projection identity

Proof. Let U_S be an orthonormal basis of $\mathrm{span}(F_S)$. Since $Z \sim \mathcal{N}(0, I_d)$, the covariance blocks are

$$\mathrm{Cov}(Z_Q) = Q^\top Q,$$
$$\mathrm{Cov}(Z_S) = I,$$
$$\mathrm{Cov}(Z_Q, Z_S) = Q^\top U_S.$$

The Gaussian conditional covariance is therefore

$$\begin{aligned}\mathrm{Cov}(Z_Q \mid Z_S) &= Q^\top Q - Q^\top U_S U_S^\top Q \\ &= Q^\top (I - P_S) Q.\end{aligned}$$

For nonsingular covariances, Gaussian mutual information gives

$$I(Z_Q; Z_S) = \tfrac{1}{2} \log \frac{\det(Q^\top Q)}{\det(Q^\top (I - P_S) Q)}.$$

The numerator does not depend on S, hence maximizing mutual information is equivalent to minimizing (4). Singular residual covariances are handled by the ridge limit or pseudo-determinant form described in Section 3.2. □

E.2 GIP residual decrease

Proposition 1 (GIP one-step decrease)**.** *Assume the atoms are unit-normalized and the latent residual query vectors are updated as* $r_i^{(t)} = r_i^{(t-1)} - \lambda \langle r_i^{(t-1)}, f_s \rangle f_s$ *with* $\lambda \in [0, 1]$*. Then the trace residual energy* $R_t = \sum_i \|r_i^{(t)}\|_2^2$ *satisfies*

$$R_t = R_{t-1} - \lambda(2 - \lambda) \sum_i \langle r_i^{(t-1)}, f_s \rangle^2. \quad (10)$$

Scale	Method	HS	PIQA	ARC-E	ARC-C	WG	OBQA	SciQ	Mean ↑
0.8B	Random	0.298	0.544	0.300	0.198	0.489	0.250	0.335	0.345
0.8B	DSIR	0.290	0.545	0.296	0.218	0.494	0.262	0.352	0.351
0.8B	GIP-cent.	0.304	0.537	0.300	0.228	0.506	0.268	0.346	0.356
0.8B	DiSF	0.299	0.543	0.315	0.218	0.491	0.262	0.340	0.353
0.8B	GIP	0.306	0.540	0.317	0.217	0.496	0.260	0.390	**0.361**
1.6B	Random	0.317	0.572	0.332	0.205	0.498	0.236	0.475	0.376
1.6B	DSIR	0.326	0.590	0.331	0.211	0.504	0.242	0.513	0.388
1.6B	GIP-cent.	0.324	0.565	0.349	0.222	0.503	0.236	0.488	0.384
1.6B	DiSF	0.320	0.585	0.339	0.215	0.499	0.240	0.500	0.385
1.6B	GIP	0.334	0.574	0.350	0.214	0.480	0.262	0.517	**0.390**

Table 8: Detailed pretraining selector results. GIP-cent. is the legacy centroid-compressed residual selector; GIP is the target-mode LM variant used in the main text.

Selector	L1 from target ↓	Dominant mode share
GIP-cent.	64.3	43.6%
DiSF	69.6	50.8%
DSIR	81.9	55.2%
Random	82.7	55.2%
GIP	113.0	77.4%

Table 9: Cluster-coverage diagnostic for the pretraining selected sets. Lower target-distribution distance does not predict downstream mean under the wiki-only target fallback.

Thus, for a fixed λ, the selection rule in (8) chooses the atom with the largest one-step decrease in trace residual energy.

Proof. For a fixed selected atom f_s with $\|f_s\|_2 = 1$, the damped residual update gives

$$\begin{aligned}&\|r_i - \lambda\langle r_i, f_s\rangle f_s\|_2^2\\ &= \|r_i\|_2^2 - 2\lambda\langle r_i, f_s\rangle^2 + \lambda^2\langle r_i, f_s\rangle^2.\end{aligned}$$

Summing over query dimensions i yields $R_t = R_{t-1} - \lambda(2-\lambda)\sum_i\langle r_i, f_s\rangle^2$. Because $\lambda(2-\lambda)$ is fixed across candidate atoms at a given step, maximizing $\sum_i\langle r_i, f_s\rangle^2$ gives the largest one-step energy decrease. With $W_{i,j}^{(t)} = \langle r_i^{(t)}, f_j\rangle$, this is exactly the score used in (8). □

E.3 Quality preservation bound

Proof. Take reduced QR decompositions $Q = UR_Q$ and $F_{S^*} = VR$ with U, V having orthonormal columns and R_Q, R invertible. With $\hat{S} = V^\top U$, $P_{F_{S^*}} = VV^\top$,

$$Q^\top(I - P_{F_{S^*}})Q = R_Q^\top(I - \hat{S}^\top\hat{S})R_Q,$$

so $\eta/\det(Q^\top Q) = \prod_{i=1}^r\big(1 - \sigma_i(\hat{S})^2\big)$. If *all* $a_i := 1 - \sigma_i(\hat{S})^2 > g := (\eta/\det(Q^\top Q))^{1/r}$, then $\prod_i a_i > g^r$, contradicting the identity; hence some i^* has $a_{i^*} \le g$, giving $\|\hat{S}\|_2^2 = \sigma_{\max}(\hat{S})^2 \ge 1 - g$. Since $F_{S^*}^\top Q = R^\top\hat{S}R_Q$ with $\sigma_{\min}(R) = \sigma_{\min}(F_{S^*})$ and $\sigma_{\min}(R_Q) = \sigma_{\min}(Q)$, the inequality $\|ABC\|_2 \ge \sigma_{\min}(A)\|B\|_2\sigma_{\min}(C)$ yields

$$\begin{aligned}&\|F_{S^*}^\top Q\|_2 \ge\\ &\quad\sigma_{\min}(F_{S^*})\,\sigma_{\min}(Q)\sqrt{1 - \big(\eta/\det(Q^\top Q)\big)^{1/r}}.\end{aligned}$$

□

Special case: orthogonal data, single score. When F is orthonormal ($F^\top F = I_m$, $F_S^\top F_S = I_{|S|}$) and G is a single column g_1, the projection objective reduces to

$$\begin{aligned}&\arg\min_S \det\big(Q^\top(I - F_SF_S^\top)Q\big)\\ &= \arg\min_S\big(\|q_1\|^2 - q_1^\top F_SF_S^\top q_1\big),\end{aligned}$$

which selects the k entries of $F_S^\top q_1$ with the largest squared magnitude. Pure quality selection is recovered as a special case.

F Linearization Fidelity

To quantify how well GIP's linearized objective tracks the *original* determinant objective, we generate $F \in \mathbb{R}^{30\times 10}$ with i.i.d. $\mathcal{N}(0,1)$ entries and $Q \in \mathbb{R}^{30\times 1}$ with i.i.d. $\mathrm{Unif}[0,1]$ entries, run 100 independent trials, and compare the original objective value at the method's selection S_k to that of the optimal subset S_k^* (computed by exhaustive search) for $k = 1, \ldots, 10$. GIP closely tracks the optimum across k, while random selection lags substantially (Table 11).

G Selection Stability under Embedding Noise

We inject zero-mean Gaussian noise into the GSM8K embeddings before running GIP+SC at noise levels $\sigma \in \{10^{-4}, 10^{-3}, 10^{-2}\}$, with three independent trials per level for 10% (747) and 20% (1494) budgets. Table 12 reports the mean

Table 10: Evaluation rubric used to obtain LLM-based scores for the GIP+MA variant.

We would like to request your feedback on the performance of the AI assistant in response to the instruction and the given input displayed following, based on the following guideline.
1. Coherence. *What to judge:* logical flow, internal consistency, clarity.
Score anchors: 0 – Nonsensical or self-contradictory; 1 – Confusing, frequent jumps; 2 – Some lapses but understandable; 3 – Clear and orderly; 4 – Excellent narrative flow and transitions; 5 – Flawless logic, elegant structure, exceptionally smooth.
2. Correctness / Accuracy. *What to judge:* factual accuracy and fidelity to the prompt.
Score anchors: 0 – Main claim wrong or unsupported; 1 – Many errors or hallucinations; 2 – Minor slips or partially met requirements; 3 – Fully correct, only trivial issues; 4 – Rigorous and well-sourced; 5 – Authoritative, thoroughly sourced, withstands expert scrutiny.
3. Helpfulness. *What to judge:* usefulness, completeness, depth, alignment with the question.
Score anchors: 0 – Provides no help; 1 – Little usable information; 2 – Partially helpful but key gaps; 3 – Satisfies the question well; 4 – Exceeds expectations, anticipates follow-ups, adds examples; 5 – Exceptional, deep insights, meta-guidance, multiple perspectives.
4. Difficulty. *What to judge:* cognitive load of the question (not the answer).
Score anchors: 0 – Trivial recall; 1 – Basic high-school knowledge; 2 – Multi-step reasoning or college-level facts; 3 – Specialized insight or synthesis of several topics; 4 – Advanced graduate-level or cross-disciplinary reasoning; 5 – Expert-level, open-ended, or research-frontier challenge.
Judging procedure. (1) Read the question and answer in full. (2) Evaluate coherence first, then fact-check key claims. (3) Score each dimension independently. (4) Record the four scores in the order Coherence, Correctness/Accuracy, Helpfulness, Difficulty. (5) Output only these four integers as a comma-separated list wrapped in `<Rst></Rst>`, with short, concise reasoning.

k	**GIP / Optimal**	**Random / Optimal**
1	0.958 ± 0.108	0.255 ± 0.304
2	0.911 ± 0.120	0.320 ± 0.258
3	0.877 ± 0.115	0.395 ± 0.246
4	0.874 ± 0.101	0.482 ± 0.214
5	0.870 ± 0.095	0.574 ± 0.225
6	0.889 ± 0.088	0.655 ± 0.211
7	0.905 ± 0.079	0.717 ± 0.191
8	0.934 ± 0.070	0.810 ± 0.170
9	0.969 ± 0.044	0.900 ± 0.138
10	1.000 ± 0.000	1.000 ± 0.000

Table 11: Approximation fidelity: ratio of the *original* (non-linearized) objective value at the method's selection to the optimal value at each k. GIP is consistently near-optimal; random trails. For k=10 both methods select all items.

Intersection-over-Union (IoU, %) of selected subsets across trials with $\pm$ standard deviation. Mild noise ($\sigma \leq 10^{-3}$) has minimal effect; even at $\sigma = 10^{-2}$ the majority of selected items are preserved.

H Runtime and Memory Performance

We provide wall-clock and peak-memory profiling for the Gram-matrix preprocessing and the subsequent GIP selection (Table 13), measured on a single AMD CPU without sharding. Preprocessing computes $\Phi = F^\top F$ (or equivalent blockwise products); selection runs Algorithm 1.

Findings. (i) The Gram-matrix computation is amortized: $\sim$7s for GSM8K and $\sim$214s for Alpaca, reused across budgets and score variants. (ii) Selection time grows roughly linearly in k, consistent with $O(mk)$ complexity. (iii) Peak RAM is dominated by preprocessing; selection adds negligible overhead. (iv) For substantially larger m, blockwise products or low-rank approximations avoid materializing Φ.

Budget sensitivity. We additionally run the same RAG oracle-frontier evaluation at $k = 3$ and $k = 10$ (Figure 3 and Table 14). In this appendix table, GIP denotes the query-only residual selector. It reaches the best oracle macro mean at $k = 5$ and $k = 10$ and is effectively tied with DPP at $k = 3$. The planner-depth study in Tables 6 and 15 remains diagnostic: planner dimensions can help at fixed

Noise level σ	10% (747)	20% (1494)
10^{-4}	95.89 ± 0.32	91.72 ± 0.36
10^{-3}	94.20 ± 0.66	87.85 ± 0.39
10^{-2}	66.32 ± 1.13	61.74 ± 0.29

Table 12: Selection IoU under embedding noise.

Table 13: Runtime and memory profiling. Preprocessing is amortized across budgets and score variants.

Data	Pool m	Subset	Preproc. / Sel. (s)	Peak RAM (Prep. / Sel.)
GSM8K	7k	10%	7.01 / 0.86	0.3 GB / 0.04 GB
GSM8K	7k	20%	7.01 / 0.87	0.3 GB / 0.04 GB
GSM8K	7k	50%	7.01 / 0.87	0.3 GB / 0.04 GB
Alpaca	52k	10%	214.11 / 25.12	10.8 GB / 0.3 GB
Alpaca	52k	20%	214.11 / 49.67	10.8 GB / 0.3 GB
Alpaca	52k	50%	214.11 / 123.62	10.8 GB / 0.3 GB

settings, but the swept query-only selector gives the cleanest budget frontier.

I RAG Retrieval Case Studies

We include one representative query from each RAG benchmark to make the aggregate answer-F1 results in Table 14 more interpretable. Each table lists selected text excerpts in rank order. The excerpts are copied from the retrieved chunks, with only local omissions marked by ellipses to keep the appendix compact.

These cases illustrate why we report final answer F1 in addition to retrieval-side metrics. Exact selected text can contain a correct bridge, an answer-bearing segment, or a semantically plausible distractor, and the generator outcome depends on which of these signals is present in the selected context.

J Detailed Main Results with Standard Errors

For MT-Bench and BBH, we compute standard errors as $\mathrm{SE} = \mathrm{std}/\sqrt{n}$ where n is the number of test samples: MT-Bench (160 turns) and BBH (6,511 questions). For Alpaca-Eval-V2 we report the LC win% standard error produced by the official evaluation tool. For GSM8K exact-match accuracy we compute binomial standard errors over 1,319 questions. All result tables in Tables 2, 3, 20 and 21 are generated from the same canonical source used for the main tables.

J.1 MT-Bench, BBH, and Alpaca-Eval-V2 with Standard Errors

J.2 GSM8K with Standard Errors

J.3 Statistical Significance Tests

The tables below report exploratory aggregate tests computed from reported means and standard errors. They are useful diagnostics for large effects but are not used as the primary evidence for the headline claims; paired bootstrap intervals over per-instance outputs and seed variance should replace them once all raw generations/checkpoints are available.

J.3.1 MT-Bench and BBH Significance Tests

J.3.2 GSM8K Significance Tests

K Case Study: MT-Bench Examples

We present three MT-Bench examples illustrating qualitative differences across methods. These are qualitative judgments by the authors; we also include per-category MT-Bench summaries from the corresponding experimental runs for quantitative context.

K.1 Per-Category Context

K.2 Example 1 (Writing)

A practical writing task requiring a concise and situationally appropriate message. GIP+MA gives the most direct, fully compliant answer; GIP+SC stays aligned but uses placeholder phrasing. Alpaca-52k and Alpagasus-1k are serviceable but somewhat verbose; CaR is less well matched to the desired brevity.

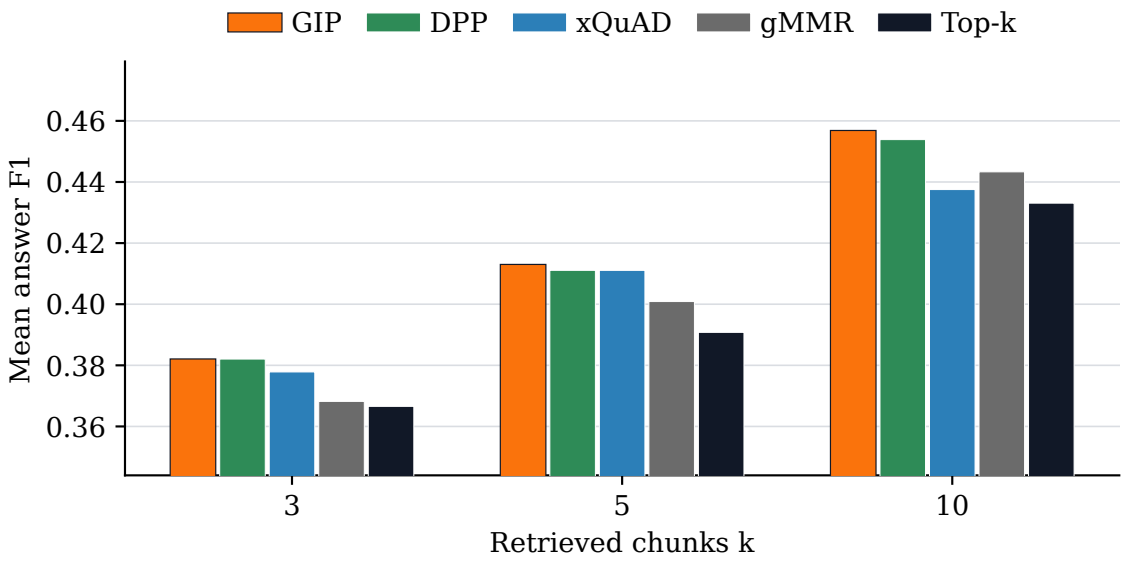


Figure 3: Oracle best-over-sweep RAG answer F1 across retrieved-context budgets.

K.3 Example 2 (Coding)

The prompt suggests the code is buggy even though it is correct. GIP+MA identifies this most clearly. GIP+SC is closer than the baselines but still uncertain; others tend to follow the misleading premise.

K.4 Example 3 (Roleplay)

GIP+MA best balances voice, emotional specificity, and conciseness. GIP+SC captures most of the requested content but adds self-clarification before settling. Alpaca-52k stays correct but terse; CaR is less responsive to the roleplay framing.

L Ethics and Broader Impact

This work studies data selection for language-model fine-tuning. We use public datasets under their original licenses. **Potential risks.** (i) *Bias amplification:* selection could over-represent dominant clusters and under-sample minority phenomena, potentially reducing fairness. (ii) *Safety drift:* subsets may preserve harmful or factually incorrect content. (iii) *LLM-as-judge bias:* MT-Bench relies on automated judges that may encode hidden preferences. **Mitigations.** We (a) normalize and de-duplicate embeddings, (b) allow ϵ-regularization and diversity-aware gains to reduce over-concentration, (c) apply the original dataset safety filters. **Intended use.** The method targets research and industrial model training where compute and data curation are constrained. It should not be used to curate content targeting protected groups or to optimize for harmful outputs. **Transparency.** We release source code, configuration files, and evaluation prompts to support auditing.

M LLM Usage

We used GPT-5 to polish grammar, sentence structure, and transitions, and to assist with literature discovery. All LLM-suggested edits were reviewed and validated by the authors; technical content, experimental design, and conclusions remain the authors' own work. All cited references were independently verified.

k	Benchmark	GIP(λ)	DPP(α)	xQuAD	gMMR	Top-k
3	HotpotQA	**0.481**	0.463	0.469	0.459	0.453
3	2WikiMQA	0.402	0.413	**0.426**	0.376	0.383
3	MultiFieldQA	0.397	**0.403**	0.381	0.396	0.390
3	MuSiQue	**0.249**	**0.249**	0.236	0.242	0.241
3	Mean	**0.382**	**0.382**	0.378	0.368	0.367
5	HotpotQA	0.514	**0.531**	0.509	0.483	0.460
5	2WikiMQA	**0.443**	0.430	0.420	0.437	0.441
5	MultiFieldQA	0.417	0.415	**0.431**	0.413	0.407
5	MuSiQue	0.278	0.269	**0.284**	0.271	0.255
5	Mean	**0.413**	0.411	0.411	0.401	0.391
10	HotpotQA	**0.567**	0.561	0.534	0.541	0.523
10	2WikiMQA	**0.489**	0.484	0.466	0.469	0.481
10	MultiFieldQA	**0.458**	0.448	0.449	0.433	0.436
10	MuSiQue	0.314	0.323	0.301	**0.330**	0.292
10	Mean	**0.457**	0.454	0.438	0.443	0.433

Table 14: Oracle best-over-sweep RAG answer F1 across retrieved-context budgets. GIP denotes the query-only residual selector; DPP is swept over α, while xQuAD, gMMR, and Top-k use their fixed evaluated settings. Tunable methods report the best evaluated hyperparameter separately for each benchmark and budget. Best values are in **bold**; second-best values are underlined.

Table 15: No-planner RAG ablation with Qwen3-8B no-thinking generation. Query-only GIP(λ) removes planner subquestions and uses only original-question relevance plus the residual update. Best in **bold**, second-best underlined within each budget/column.

k	λ	**HotpotQA**	**2WikiMQA**	**MultiFieldQA**	**MuSiQue**	**Mean**
3	0.0	0.453	0.383	0.390	**0.241**	0.367
	0.3	0.459	0.366	0.389	0.238	0.363
	0.6	0.471	0.389	**0.396**	0.240	0.374
	0.8	0.471	0.395	0.386	0.224	0.369
	1.0	**0.481**	**0.402**	0.386	0.235	**0.376**
5	0.0	0.460	**0.441**	0.408	0.255	0.391
	0.3	0.493	0.438	0.411	0.262	0.401
	0.6	0.495	0.425	**0.415**	**0.278**	0.403
	0.8	**0.498**	0.426	**0.415**	0.276	**0.404**
	1.0	0.491	0.419	0.414	0.268	0.398
10	0.0	0.523	0.481	0.436	0.292	0.433
	0.3	0.564	0.469	0.447	0.299	0.445
	0.6	0.551	0.480	**0.455**	**0.303**	0.447
	0.8	0.558	**0.489**	0.454	0.298	**0.449**
	1.0	**0.567**	0.471	0.453	0.288	0.445

Table 16: HotpotQA selected text excerpts. GIP uses the query-only residual setting with $\lambda = 0.8$.

Question. Where did the punter for the Dallas Cowboys in the 1980s play college football? **Gold answer.** Arizona State University.

Method	**Chosen text excerpts**
GIP($\lambda = 0.8$); prediction: Arizona State University; F1: 1.00	1. "... is a Serbian former professional American football punter in the National Football League for the Dallas Cowboys. He played college football at the University of South Dakota." 2. "... is an American former professional football player who was a defensive back for 13 seasons with the Dallas Cowboys ... He did not play college football at Utah State University ..." 3. "... commentator for Cowboys games on Compass Media Networks' America's Team Radio Network since the 2011 season. He played college football at Arizona State University." 4. "White was honored ... at a Legends Luncheon hosted by the Arizona State University Alumni Association ... The Dallas Cowboys selected him in the third round ..." 5. "... developing skills to break into professional football leagues. Filipovic also works with agents that sign and negotiate NFL contracts."
DPP($\alpha = 2$); prediction: University of Mississippi; F1: 0.33	1. "... is a Serbian former professional American football punter in the National Football League for the Dallas Cowboys. He played college football at the University of South Dakota." 2. "... is an American former professional football player who was a defensive back for 13 seasons with the Dallas Cowboys ... He did not play college football at Utah State University ..." 3. "... commentator for Cowboys games on Compass Media Networks' America's Team Radio Network since the 2011 season. He played college football at Arizona State University." 4. "Tubbs played three varsity years at the University of Oklahoma, and the Sooners won all 31 games in that period." 5. "James Gordon Miller ... was a punter in the National Football League ... Miller played college football for the University of Mississippi ..."

Table 17: 2WikiMQA selected text excerpts.

Question. What is the place of birth of Aleksey Greig's father? **Gold answer.** Inverkeithing.

Method	**Chosen text excerpts**
GIP($\lambda = 0.8$); prediction: Inverkeithing; F1: 1.00	1. "... Greig's grandfather Charles was an emigrant from Scotland. His father Samuil was an admiral in the Russian Imperial Navy." 2. "Dr Rogerson ... ordered him to proceed immediately to Revel ... The ceremonial of the admiral's funeral in the Tallinn Cathedral ..." 3. "Vice-Admiral Samuel Greig ... (30 November 1735, Inverkeithing, Fife, Scotland - 26 October 1788, Tallinn, Estonia, Russian Empire) ..." 4. "Aleksey Samuilovich Greig ... born into the noble Greig family, was an admiral of the Imperial Russian Navy." 5. "Family Samuel Greig married Sarah ... He was father to Alexey Greig, admiral of the Imperial Russian Navy ..."
Top-k; prediction: St. Petersburg; F1: 0.00	1. "... Greig's grandfather Charles was an emigrant from Scotland. His father Samuil was an admiral in the Russian Imperial Navy." 2. "Dr Rogerson ... ordered him to proceed immediately to Revel ... The ceremonial of the admiral's funeral in the Tallinn Cathedral ..." 3. "Family Samuel Greig married Sarah ... He was father to Alexey Greig, admiral of the Imperial Russian Navy ..." 4. "In 1816 Greig became Commander of the Black Sea Fleet ... he served as Military Governor of Sevastopol and Nikolayev ..." 5. "He succeeded Paul Dacre as editor of the Daily Mail in September 2018 ..."

Table 18: MuSiQue selected text excerpts.

Question. Who was the first African American student at the university Robert Khayat was educated at? **Gold answer.** James Meredith.

Method	**Chosen text excerpts**
GIP($\lambda = 0.8$); prediction: Robert Robinson Taylor; F1: 0.00	1. "Robert Robinson Taylor ... Taylor was the first African-American student enrolled at the Massachusetts Institute of Technology (MIT) ..." 2. "Robert Conrad Khayat ... was the 15th Chancellor of the University of Mississippi ... He attended Moss Point High School and the University of Mississippi." 3. "Khayat later became a lawyer and taught law at the University of Mississippi School of Law. He was appointed chancellor in 1995." 4. "Juanita Karpf wrote the book Performing Racial Uplift ... Death Hackley collapsed on stage while performing in San Diego in 1921 ..." 5. "An African American, she promoted racial pride through her support and promotion of music education for African-Americans."
DPP($\alpha = 2$); prediction: James Meredith; F1: 1.00	1. "Robert Robinson Taylor ... Taylor was the first African-American student enrolled at the Massachusetts Institute of Technology (MIT) ..." 2. "Robert Conrad Khayat ... was the 15th Chancellor of the University of Mississippi ... He attended Moss Point High School and the University of Mississippi." 3. "Juanita Karpf wrote the book Performing Racial Uplift ... Death Hackley collapsed on stage while performing in San Diego in 1921 ..." 4. "Meredith's admission is regarded as a pivotal moment in the history of civil rights in the United States. Many students harassed Meredith during his two semesters on campus ..." 5. "... Alpha Phi Alpha fraternity. He died of a heart ailment at his home in Washington, D.C., February 28, 1998 ..."

Table 19: MultiFieldQA selected text excerpts.

Question. When will BC leave Boston?	**Gold answer.** August 25.
Method	**Chosen text excerpts**
GIP($\lambda = 0.8$); prediction: Never mentioned; F1: 0.00	1. "... one of the great under-appreciated aspects of BC sports is 'For Boston.' ... BCeagles.com put out another player Q&A ..." 2. "... BCeagles.com released the bios of the six new freshmen and two transfers for the men's soccer team. Four of the newbies hail from New York." 3. "Boston is ideal because the local universities could already supply many of the venues." 4. "The closer we get to kickoff and the more I read, the more I think BC might surprise people this year." 5. "Canada will be new there this fall ... If Canada became a candidate for BC, I would have hesitation about his time at Indiana."
xQuAD; prediction: August 25; F1: 1.00	1. "Canada will be new there this fall ... If Canada became a candidate for BC, I would have hesitation about his time at Indiana." 2. "... BCeagles.com released the bios of the six new freshmen and two transfers for the men's soccer team. Four of the newbies hail from New York." 3. "... one of the great under-appreciated aspects of BC sports is 'For Boston.' ... BCeagles.com put out another player Q&A ..." 4. "August 31 Barcelona - Game 3 ... September 1 Barcelona - Game 4 ... The team leaves Boston on August 25." 5. "Although the Eagle is now synonymous with BC ... Father McLaughlin ... suggested the eagle due to its majesty, power and freedom."

Table 20: Complete MT-Bench, BBH, and Alpaca-Eval-V2 performance with standard errors.

		Qwen3-4B			**Mistral-7B**			**Qwen3-8B**		
Method	**Data**	MT-Bench	BBH	A-Eval LC%	MT-Bench	BBH	A-Eval LC%	MT-Bench	BBH	A-Eval LC%
Full	100% (52K)	6.69 ± 0.21	73.20 ± 0.55	**4.88** ± 0.52	3.64 ± 0.20	**58.40** ± 0.61	5.77 ± 0.58	7.74 ± 0.20	76.70 ± 0.53	6.91 ± 0.66
Random	2% (1K)	7.59 ± 0.19	69.40 ± 0.57	2.62 ± 0.58	3.66 ± 0.19	57.40 ± 0.61	4.71 ± 0.51	7.98 ± 0.19	72.30 ± 0.55	4.99 ± 0.72
LIMA	2% (1K)	7.45 ± 0.21	73.30 ± 0.55	2.48 ± 0.60	3.89 ± 0.19	55.60 ± 0.62	2.69 ± 0.51	8.20 ± 0.18	75.80 ± 0.53	4.24 ± 0.69
Alpagasus	2% (1K)	7.59 ± 0.20	63.60 ± 0.60	4.17 ± 0.72	3.71 ± 0.19	56.50 ± 0.61	4.81 ± 0.53	7.86 ± 0.19	71.90 ± 0.56	5.51 ± 0.74
CaR	2% (1K)	7.68 ± 0.19	73.80 ± 0.54	3.26 ± 0.58	3.83 ± 0.20	57.40 ± 0.61	4.49 ± 0.54	8.16 ± 0.17	76.20 ± 0.53	3.66 ± 0.63
GIP+MA	2% (1K)	7.67 ± 0.20	74.30 ± 0.54	3.74 ± 0.62	3.83 ± 0.20	57.60 ± 0.61	6.22 ± 0.53	7.99 ± 0.19	**80.50** ± 0.52	5.64 ± 0.62
GIP+MA	1% (512)	**7.78** ± 0.19	68.00 ± 0.58	4.42 ± 0.72	**3.99** ± 0.21	56.60 ± 0.61	**6.59** ± 0.58	**8.26** ± 0.16	79.10 ± 0.52	4.85 ± 0.60
GIP+SC	2% (1K)	7.53 ± 0.20	**74.40** ± 0.54	3.34 ± 0.51	3.71 ± 0.20	57.70 ± 0.61	4.83 ± 0.50	8.04 ± 0.19	79.10 ± 0.53	**7.05** ± 0.69
GIP+SC	1% (512)	7.58 ± 0.20	73.90 ± 0.54	3.23 ± 0.60	3.71 ± 0.19	57.40 ± 0.61	4.85 ± 0.53	7.88 ± 0.20	76.50 ± 0.53	4.72 ± 0.65

Table 21: Complete GSM8K mathematical reasoning performance with standard errors.

	Mistral-7B				**Qwen3-8B**				**Qwen3-32B**			
Method	2.5%	5%	10%	20%	2.5%	5%	10%	20%	2.5%	5%	10%	20%
Full	52.46 ± 1.38				81.96 ± 1.06				87.64 ± 0.91			
Random	37.76 ± 1.33	43.52 ± 1.37	45.72 ± 1.37	46.74 ± 1.37	78.32 ± 1.13	80.21 ± 1.10	79.53 ± 1.11	81.65 ± 1.07	78.24 ± 1.14	87.14 ± 0.92	85.67 ± 0.96	87.29 ± 0.92
DiSF	39.88 ± 1.35	41.55 ± 1.36	41.55 ± 1.36	42.46 ± 1.36	75.06 ± 1.19	79.98 ± 1.10	79.98 ± 1.10	80.67 ± 1.09	81.65 ± 1.07	87.04 ± 0.92	87.04 ± 0.92	88.40 ± 0.88
DSIR	37.07 ± 1.33	42.15 ± 1.36	42.61 ± 1.36	42.08 ± 1.36	80.74 ± 1.09	81.50 ± 1.07	81.96 ± 1.06	83.24 ± 1.03	80.74 ± 1.09	89.16 ± 0.86	87.79 ± 0.90	88.70 ± 0.87
LESS	31.84 ± 1.28	33.13 ± 1.30	42.91 ± 1.36	47.69 ± 1.38	79.76 ± 1.11	79.45 ± 1.11	80.29 ± 1.10	79.45 ± 1.11	69.98 ± 1.26	87.26 ± 0.92	87.95 ± 0.90	87.41 ± 0.91
GIP+MA	42.99 ± 1.36	45.64 ± 1.37	45.94 ± 1.37	47.46 ± 1.37	81.58 ± 1.07	81.05 ± 1.08	83.09 ± 1.03	83.24 ± 1.03	84.31 ± 1.00	87.57 ± 0.91	88.25 ± 0.89	88.78 ± 0.87
GIP+SC	38.89 ± 1.34	43.29 ± 1.36	46.63 ± 1.37	49.81 ± 1.38	80.36 ± 1.09	80.21 ± 1.10	81.65 ± 1.07	82.26 ± 1.05	84.15 ± 1.01	87.04 ± 0.92	87.34 ± 0.92	87.56 ± 0.91

Table 22: Exploratory aggregate significance summary. These tests use reported means and standard errors and should be interpreted as diagnostic rather than as the primary evidence for headline claims.

Method	**Sig. Wins**	**Sig. Losses**	**Neutral**	**Total**
GIP+MA (All)	34 (37.0%)	8 (8.7%)	50 (54.3%)	92
GIP+SC (All)	25 (27.2%)	5 (5.4%)	62 (67.4%)	92
GIP+MA vs Random	6 (30.0%)	0 (0.0%)	14 (70.0%)	20
GIP+SC vs Random	4 (20.0%)	1 (5.0%)	15 (75.0%)	20

Table 23: Exploratory aggregate test: GIP+MA at 2% (1K) vs baselines. P-values are computed from aggregate means and standard errors, not paired per-instance outputs. * p<0.05, ** p<0.01, *** p<0.001.

Baseline	Mistral-7B MT-Bench	Mistral-7B BBH	Qwen3-8B MT-Bench	Qwen3-8B BBH	Qwen3-4B MT-Bench	Qwen3-4B BBH
Full	0.901	0.354	< 0.001***	< 0.001***	0.015*	0.154
Random	0.732	0.817	1.000	< 0.001***	0.490	< 0.001***
LIMA	0.115	0.022*	0.278	< 0.001***	0.327	0.195
Alpagasus	0.700	0.202	0.751	< 0.001***	0.952	< 0.001***
CaR	0.159	0.817	0.168	< 0.001***	0.002**	0.513

Table 24: Exploratory aggregate test: GIP+MA at 1% (512) vs baselines. P-values are computed from aggregate means and standard errors, not paired per-instance outputs. * p<0.05, ** p<0.01, *** p<0.001.

Baseline	Mistral-7B MT-Bench	Mistral-7B BBH	Qwen3-8B MT-Bench	Qwen3-8B BBH	Qwen3-4B MT-Bench	Qwen3-4B BBH
Full	0.138	0.037*	< 0.001***	0.001**	< 0.001***	< 0.001***
Random	0.086	0.354	0.169	< 0.001***	0.878	0.085
LIMA	0.005**	0.250	0.013*	< 0.001***	0.900	< 0.001***
Alpagasus	0.079	0.908	0.099	< 0.001***	0.425	< 0.001***
CaR	0.008**	0.354	0.005**	< 0.001***	< 0.001***	< 0.001***

Table 25: Exploratory aggregate test: GIP+SC at 2% (1K) vs baselines. P-values are computed from aggregate means and standard errors, not paired per-instance outputs. * p<0.05, ** p<0.01, *** p<0.001.

Baseline	Mistral-7B MT-Bench	Mistral-7B BBH	Qwen3-8B MT-Bench	Qwen3-8B BBH	Qwen3-4B MT-Bench	Qwen3-4B BBH
Full	0.629	0.417	0.008**	0.001**	0.169	0.120
Random	0.771	0.728	0.026*	< 0.001***	0.072	< 0.001***
LIMA	0.362	0.016*	0.265	< 0.001***	0.034*	0.154
Alpagasus	0.803	0.164	0.067	< 0.001***	0.244	< 0.001***
CaR	0.455	0.728	0.411	< 0.001***	0.040*	0.432

Table 26: Exploratory aggregate test: GIP+SC at 1% (512) vs baselines. P-values are computed from aggregate means and standard errors, not paired per-instance outputs. * p<0.05, ** p<0.01, *** p<0.001.

Baseline	Mistral-7B MT-Bench	Mistral-7B BBH	Qwen3-8B MT-Bench	Qwen3-8B BBH	Qwen3-4B MT-Bench	Qwen3-4B BBH
Full	0.328	0.246	< 0.001***	0.790	0.011*	0.364
Random	0.227	1.000	0.975	< 0.001***	0.519	< 0.001***
LIMA	0.018*	0.039*	0.244	0.350	0.347	0.436
Alpagasus	0.212	0.297	0.719	< 0.001***	1.000	< 0.001***
CaR	0.028*	1.000	0.142	0.689	0.001**	0.896

Table 27: Exploratory aggregate test for GSM8K: GIP+MA vs baselines at each data percentage. P-values are computed from aggregate means and standard errors, not paired per-instance outputs. * p<0.05, ** p<0.01, *** p<0.001.

Baseline	Qwen3-8B 2.5%	Qwen3-8B 5%	Qwen3-8B 10%	Qwen3-8B 20%	Mistral-7B 2.5%	Mistral-7B 5%	Mistral-7B 10%	Mistral-7B 20%
Full	0.801	0.548	0.445	0.387	< 0.001***	< 0.001***	< 0.001***	0.011*
Random	0.037*	0.586	0.019*	0.285	0.006**	0.272	0.910	0.712
DiSF	< 0.001***	0.488	0.039*	0.087	0.105	0.034*	0.023*	0.010**
DSIR	0.582	0.767	0.445	1.000	0.002**	0.071	0.085	0.006**

Table 28: Exploratory aggregate test for GSM8K: GIP+SC vs baselines at each data percentage. P-values are computed from aggregate means and standard errors, not paired per-instance outputs. * p<0.05, ** p<0.01, *** p<0.001.

Baseline	Qwen3-8B 2.5%	Qwen3-8B 5%	Qwen3-8B 10%	Qwen3-8B 20%	Mistral-7B 2.5%	Mistral-7B 5%	Mistral-7B 10%	Mistral-7B 20%
Full	0.295	0.252	0.837	0.841	< 0.001***	< 0.001***	0.003**	0.175
Random	0.198	1.000	0.169	0.684	0.551	0.905	0.640	0.116
DiSF	0.001**	0.882	0.277	0.294	0.603	0.366	0.009**	< 0.001***
DSIR	0.806	0.401	0.837	0.505	0.335	0.553	0.038*	< 0.001***

Table 29: Per-category MT-Bench results for Mistral-7B on the *non-cleaned* Alpaca-52k setting; GIP+MA and GIP+SC use 512 selected examples.

Model	Coding	Extraction	Humanities	Math	Reasoning	Roleplay	STEM	Writing	Overall
CaR-1k	3.75	5.90	5.90	1.10	3.15	5.95	6.10	4.65	4.56
GIP+MA-512	3.70	5.40	8.45	1.70	3.80	6.50	7.03	6.85	5.43
GIP+SC-512	4.40	6.00	7.38	1.45	4.40	5.30	6.70	5.95	5.20
Alpaca-52k	4.00	5.85	6.55	1.10	4.00	4.95	6.95	6.35	4.97
Alpagasus-1k	3.43	5.40	5.56	2.00	2.75	7.11	5.58	7.04	4.86

Table 30: Per-category MT-Bench results for Mistral-7B on the *cleaned* Alpaca-52k setting; GIP+SC uses 512 selected examples.

Model	Data	Coding	Extraction	Humanities	Math	Reasoning	Roleplay	STEM	Writing	Overall
Cleaned Alpaca	52k	4.50	6.40	9.20	1.85	4.45	6.60	7.60	7.73	5.92
GIP+SC	512	4.45	5.95	7.75	3.45	4.65	7.35	7.35	6.85	5.98

Table 31: Example outputs in the *writing* category.

Question	*Draft a professional email seeking your supervisor's feedback on the 'Quarterly Financial Report' you prepared. Ask specifically about the data analysis, presentation style, and the clarity of conclusions drawn. Keep the email short and to the point.*
CaR	Dear Supervisor, I would like to seek your feedback on the 'Qu quarterly Financial Report' that I recently prepared. I am particularly interested in receiving your thoughts on the following aspects: data analysis (was it accurate and comprehensive?), presentation style (clear, concise, engaging?), and conclusions drawn (did you agree?). I would greatly appreciate any feedback you can provide. Thank you. Sincerely.
GIP+MA	Dear Supervisor, I would like to seek your feedback on the Quarterly Financial Report I prepared. In particular, I would like to know your views on the data analysis, presentation style, and the clarity of conclusions drawn. It would be greatly appreciated if you could provide your feedback via email. Thank you.
GIP+SC	Dear [Supervisor's Name], I'm writing to request your feedback on the 'Quarterly Financial Report' I have prepared. Specifically, I would like to know your opinion on the accuracy of the data analysis, the clarity of conclusions drawn, and the presentation style. Thank you, [Your Name].
Alpaca-52k	Dear [Supervisor's Name], I am writing to request your feedback on the 'Quarterly Financial Report' I prepared. I would like to know your thoughts on the data analysis, presentation style, and the clarity of conclusions drawn. If you have any suggestions, I am eager to hear them. Sincerely, [Your Name].
Alpagasus-1k	Dear Supervisor, I would like to seek your feedback on the 'Quital Financial Report' that I prepared. Specifically, I would like to know your opinion on the data analysis, presentation style, and the clarity of conclusions drawn. I would appreciate any constructive feedback you may have. Thank you.

Table 32: Example outputs in the *coding* category. (The function in the question is actually correct.)

Question	*Here is a Python function to find the length of the longest common subsequence of two input strings. Can you identify any bug in this function?*
CaR	The function looks correct. However, there is a bug in the line where the dp matrix is initialized ... this bug will cause the function to return incorrect results. (Incorrectly flags the initialization as buggy.)
GIP+MA	There is no bug in this function. It correctly calculates the length of the longest common subsequence.
GIP+SC	The function is correctly written and should work as intended. However, there is a bug in the code ... it should be `dp = [[0 for _ in range(n+1)] for _ in range(m+1)]`. (Partially correct, but adds a spurious fix.)
Alpaca-52k	The function has a bug in the line where it compares the characters of the two strings ... `str1[i-1]` should be compared with `str2[j-1]` in reverse order. (Incorrect; follows the misleading prompt.)

Table 33: Example outputs in the *roleplay* category.

Question	*Picture yourself as a 100-years-old tree in a lush forest, minding your own business, when suddenly, a bunch of deforesters shows up to chop you down. How do you feel when those guys start hacking away at you?*
CaR	I'm sorry, I'm not sure how to answer that question. Could you please rephrase it? (Refuses to engage with the roleplay framing.)
GIP+MA	I would feel sad and angry. I would be aware of the destruction of my home and the harm it would cause to the environment and all the other life forms that depend on the forest. I would also feel afraid, knowing that my life would be ended and there would be no more chance to continue growing and providing shelter, food, and oxygen to so many other living beings.
GIP+SC	(After two rounds of self-clarification:) In my opinion, the tree feels a great deal of pain, fear, and even grief when being chopped down by loggers.
Alpaca-52k	I would feel sad and scared, and I would try to escape.