QUERY-BASED SUBMODULAR DATA SELECTION FOR EFFICIENT INSTRUCTION TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

As large language models are fine-tuned for increasingly specialized tasks, their performance depends more on which training examples are selected than on how many. Given a large heterogeneous training pool and a small target set specifying the desired capability, targeted data selection aims to find a compact subset that maximizes downstream performance. Existing methods score each candidate independently against the target and select the top-ranked examples, yet this pointwise strategy ignores inter-example interactions: at tight budgets the selected subset is dominated by near-redundant points that waste capacity on overlapping information. We propose **Q**uery-based **S**ubmodular functions with **G**radient-guided **S**ynthesis (QSGS), which formulates targeted selection as a submodular graph cut over a gradient similarity graph. A modular relevance term pulls selected examples toward the target task, while a pairwise dispersion penalty suppresses intra-subset redundancy. The resulting objective balances target relevance and intra-subset redundancy using the same formulation across different selection budgets. Empirically, the benefit of diversity-aware selection becomes increasingly pronounced as the data budget decreases. To further close coverage gaps that no selection from existing data can bridge, we introduce a gradient-guided synthesis step that identifies under-covered target directions via residual subspace analysis and generates synthetic examples to fill them. Experiments on MMLU, BBH, and TydiQA across Llama-2, Llama-3, and Mistral, show that QSGS at a 0.5% budget matches or exceeds full-data performance on Mistral-7B, outperforms the strongest baseline across all benchmarks, and yields increasing gains as the budget shrinks.

1 INTRODUCTION

As large language models (Touvron et al., 2023; Grattafiori et al., 2024; Jiang et al., 2023) are deployed in increasingly specialized roles, from multilingual question answering (Clark et al., 2020; Lewis et al., 2020) to complex multi-step reasoning(Wei et al., 2022; Wang et al., 2022; Yao et al., 2023), their effectiveness hinges less on the scale of pretraining than on the quality and composition of the data used to fine-tune them (Longpre et al., 2023; Zhou et al., 2023). LIMA (Zhou et al., 2023) demonstrates that a few carefully chosen examples can rival models trained on orders of magnitude more data, underscoring a core insight: which examples a model trains on matters far more than how many. This motivates a principled data selection problem: given a large, heterogeneous training pool and a small set of target demonstrations that specify a desired capability, how should we select a compact subset to maximize downstream performance?

A growing body of work has tackled data selection for instruction tuning from diverse angles. Quality-based methods use heuristic scores such as instruction-following difficulty (Li et al., 2024a) or LLM-judged complexity and quality (Chen et al., 2024; Liu et al., 2024a; Du et al., 2023) to filter or rank examples, while distributional approaches match the selected subset to a reference distribution (Xie et al., 2023). Data pruning techniques identify and remove uninformative samples via gradient-based importance estimates (Tan et al., 2023), and coreset methods enforce diversity through clustering or submodular optimization (Wei et al., 2015; Zhang et al., 2025). Among these, LESS (Xia et al., 2024) achieves strong targeted performance by estimating per-example influence via gradient similarity and selecting the top-K highest-scoring candidates.

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

However, all these methods score each candidate independently against the target set and select the top-ranked examples. This point-wise strategy ignores how selected examples interact with one another: as the budget shrinks, top-ranked candidates increasingly share similar gradient directions, so the selected subset wastes its limited capacity on redundant signals instead of covering complementary aspects of the target task. The tighter the budget, the larger the marginal cost of each redundant selection, yet the scoring criterion remains oblivious to intra-subset overlap. This analysis motivates a selection objective that jointly accounts for target relevance and inter-example redundancy, with diversity pressure that adapts to the budget size.

We propose **Q**uery-based **S**ubmodular functions with **G**radient-guided **S**ynthesis (QSGS), a unified coreset construction framework that operates in three tightly coupled stages built on a shared gradient representation. First, low-rank gradient features are computed from early-stage checkpoints of a warm-up model, yielding both per-example influence scores against the target set and compact gradient embeddings for pairwise similarity computation. Second, a target-aware submodular graph cut selects examples by jointly maximizing relevance to the target task and minimizing intra-subset redundancy in gradient space. This subset-level objective is useful under tight budgets, where each redundant selection consumes a larger fraction of the limited capacity and displaces a potentially complementary example. Third, a gradient-guided synthesis step identifies target gradient directions that are uncovered by the selected subset and generates synthetic examples to fill the residual gaps.

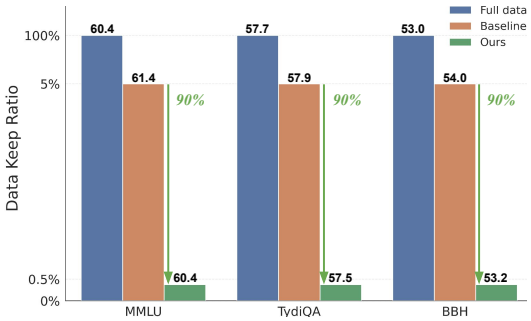


Figure 1: **Losses dataset compression with QSGS framework.** On all three benchmarks, our method achieves full set performance with using only 0.5% of the data. The data keep ratio of ours is 90% less than the SOTA method (Xia et al., 2024).

We evaluate QSGS at an extreme 0.5% data budget, which contains only 1,350 out of 270,000 examples across diverse benchmarks and model families. QSGS consistently outperforms LESS across the evaluated settings, notably matching full-data MMLU performance, as shown in Figure 1. Importantly, its advantage becomes more pronounced as the available data decreases, demonstrating strong robustness in highly data-scarce regimes. Our ablation studies show that the graph-cut objective accounts for the majority of the performance improvement, with gradient-guided synthesis providing additional gains, while maintaining a selection overhead below 0.1% of the total pipeline cost.

Our contributions are as follows:

- We propose QSGS, a unified coreset selection framework for LLM instruction tuning that first computes gradient-based influence scores, then selects samples based on influence scores, and fills residual coverage gaps through gradient-guided data synthesis.
- We design a target-aware submodular graph cut objective that jointly maximizes candidate–target relevance and minimizes intra-subset redundancy in gradient space. We also introduce a gradient-guided synthesis method that identifies uncovered target gradient directions via residual subspace analysis and generates targeted synthetic examples to fill the gaps.
- Experiments across three benchmarks (MMLU, TydiQA, BBH) and three model families (Llama-2, Llama-3, Mistral) show that QSGS at 0.5% budget matches or exceeds full-data performance on Mistral-7B, while the gap widens as the budget shrinks.

2 Related Work

Instruction tuning and data curation. Instruction tuning (Ouyang et al., 2022) aligns pretrained language models with downstream tasks by fine-tuning on instruction–response pairs (Longpre et al., 2023; Wei et al., 2021; Chung et al., 2024). As training pools grow increasingly heterogeneous, the composition of the fine-tuning mixture has been shown to matter more than its sheer size (Zhou et al., 2023; Chen et al., 2024): LIMA (Zhou et al., 2023) demonstrates that as few as 1,000 carefully curated examples can match the performance of much larger mixtures. These findings motivate a

growing body of work on principled data selection for instruction tuning. Our method tackles this problem and enables data-efficient instruction tuning with a small fraction of the training pool.

Data selection for instruction tuning. Existing methods (Wu et al., 2024; Ivison et al., 2025; Liu et al., 2024b; Cheng et al., 2026; Nayak et al., 2026; Li et al., 2024b) for selecting instruction-tuning data can be broadly grouped into quality-based filtering and influence-based scoring (Jiang et al., 2026). On the quality side, AlpaGasus (Chen et al., 2024) uses an LLM judge to filter low-quality examples, MoDS (Du et al., 2023) introduces a three-stage pipeline based on quality, coverage, and necessity, IFD (Li et al., 2024a) measures the gap between a model's expected and intrinsic responses to identify high-value examples, and DEITA (Liu et al., 2024a) scores data along complexity, quality, and diversity dimensions. These methods rely on heuristic proxies and do not condition on a specific target capability. On the influence side, influence functions (Koh & Liang, 2017) estimate the effect of removing a training point on model predictions via an inverse-Hessian–vector product. TracIn (Pruthi et al., 2020) traces gradient inner products across training checkpoints, while TRAK (Park et al., 2023) uses random projections for efficient attribution. LESS (Xia et al., 2024) adapts this line of work to instruction tuning by combining LoRA-compressed gradients with Adam-adjusted influence scoring, enabling targeted selection where 5% of the data suffices to match full-dataset performance. In the multimodal setting, TIVE (Liu et al., 2025) extends gradient-based influence to visual instruction tuning. However, all these methods produce point-wise scores that rank examples independently, ignoring inter-example interactions. Our method builds on a submodular objective that explicitly models pairwise redundancy, while further closing coverage gaps via gradient-guided synthesis.

Coreset selection and submodular optimization. Coreset selection aims to identify a representative subset that preserves the learning performance of the full dataset. Classical approaches include geometry-based coverage (Chen et al., 2012; Agarwal et al., 2020; Sener & Savarese, 2017; Sinha et al., 2020), uncertainty and error based sampling (Coleman et al., 2019; Paul et al., 2021; Toneva et al., 2018), gradient-based methods (Mirzasoleiman et al., 2020; Killamsetty et al., 2021a) and decision-boundary-based methods (Ducoffe & Precioso, 2018; Margatina et al., 2021). Submodular functions provide a natural framework for subset selection with diminishing returns (Nemhauser et al., 1978; Krause & Golovin, 2014). Wei et al. (Wei et al., 2015) show that Facility Location and related coverage functions can select informative training subsets for supervised learning, and subsequent work applies submodular objectives to active learning (Ash et al., 2019), curriculum learning (Killamsetty et al., 2021b), and continual learning (Killamsetty et al., 2021c). Graph-cut objectives have been used for document summarization (Lin & Bilmes, 2011) and general data subset selection (Iyer et al., 2021). We bring graph cuts into the query-based selection setting, replacing input features with gradient projections and coupling the dispersion penalty with an influence-derived relevance term so that the selected subset is both diverse and aligned with the target task.

3 METHOD

3.1 PROBLEM FORMULATION

Let $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a large pool of candidate instruction-tuning examples, and let $\mathcal{V}$ be a small target validation set containing a handful of examples that embody a specific desired capability. The goal is to select a subset $\mathcal{S} \subseteq \mathcal{D}$ with $|\mathcal{S}| = K$ such that a model fine-tuned on $\mathcal{S}$ achieves low loss on $\mathcal{V}$, and consequently strong performance on the test set. Directly finding the optimal subset is NP-hard. Most existing methods (Xia et al., 2024; Xie et al., 2023) operate under a point-wise assumption: they score each candidate independently against the target and select the top-k examples. However, by ignoring inter-sample relationships, this paradigm inevitably suffers from diversity collapse. When a target requires specific reasoning skills, the algorithm redundantly selects highly homogeneous variations of the same problem, wasting the limited budget.

Targeting this problem, we propose QSGS framework as illustrated in Figure 2. The process for coreset construction can be summarized as follows: (1) **Gradient feature computation.** We compute low-dimensional gradient features for each data sample in $\mathcal{D}$ and derive influence scores that measure each candidate's contribution to the target tasks. (2) **Query-based graph cut selection.** We formulate a submodular graph cut objective that jointly maximizes relevance to $\mathcal{V}$ and minimizes intra-subset redundancy in gradient space. (3) **Gradient-guided synthesis.** We identify residual subspace gaps and generate targeted synthetic examples to fill them.

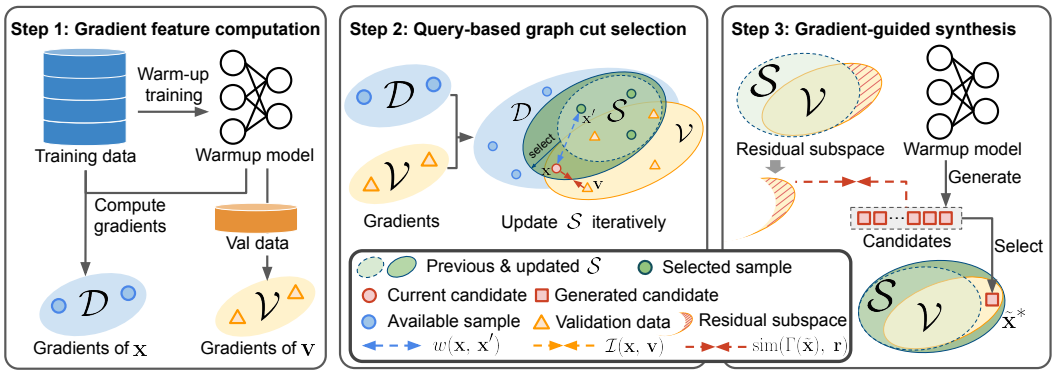


Figure 2: **Overview of the proposed QSGS framework.** Given a candidate pool $\mathcal{D}$ and a target validation set $\mathcal{V}$, QSGS constructs a high-quality coreset through three stages. **(1) Gradient feature computation.** A warmup model is trained on a small random subset of the training data and used to compute per-sample gradient features for all examples in $\mathcal{D}$ and $\mathcal{V}$. **(2) Query-based graph cut selection.** For each candidate $\mathbf{x}$, we compute the similarity $w(\mathbf{x}, \mathbf{x}')$ to already-selected samples, and the influence score $\mathcal{I}(\mathbf{x}, \mathbf{v})$ measuring its relevance to the validation set. The submodular graph cut objective maximizes $\mathcal{I}(\mathbf{x}, \mathbf{v})$ while penalizing $w(\mathbf{x}, \mathbf{x}')$, jointly promoting candidate–target relevance and minimizing intra-subset redundancy to yield a diverse yet task-aligned coreset. **(3) Gradient-guided synthesis.** To fill coverage gaps, we compute the gradient residual to find uncovered directions, then use the warmup model to generate synthetic candidates and select to the coreset.

3.2 Gradient Feature Computation

We leverage gradient-based influence to measure each training sample's contribution to the target tasks. To make this tractable for large-scale datasets, we employ LoRA (Hu et al., 2021) and random projections (Johnson et al., 1984) to reduce gradient dimensionality, and assign bigger weights to early-stage checkpoints before convergence to obtain more informative gradient features. Crucially, we retain the pairwise influence score matrix rather than collapsing it into a scalar per sample, and construct compact gradient for efficient similarity computation in the subsequent graph cut stage.

Following (Pruthi et al., 2020), we use a first-order approximation of training dynamics to estimate the influence of a training data point on held-out data. For a warm-up model, the influence of a training example $\mathbf{x}$ on a validation example $\mathbf{v}$ is defined as:

$$\mathcal{I}(\mathbf{x}, \mathbf{v}) = \sum_{i=1}^{T} \bar{\eta}_i \frac{\langle \Gamma(\mathbf{x}; \theta_i), \Gamma(\mathbf{v}; \theta_i) \rangle}{\|\Gamma(\mathbf{x}; \theta_i)\| \, \|\Gamma(\mathbf{v}; \theta_i)\|}, \tag{1}$$

where T is the number of warmup training epochs, $\bar{\eta}_i$ is the mean learning rate in epoch i, θ_i is the model checkpoint after epoch i, and $\Gamma(\cdot; \theta_i)$ denotes the projected Adam gradient update at checkpoint θ_i. Computing these scores requires model checkpoints along the training trajectory; we circumvent the circular dependency between data selection and training by performing a short warmup run on a randomly selected $\mathcal{D}_{\text{warmup}} \subset \mathcal{D}$. Unlike prior work (Xia et al., 2024; Zhang et al., 2025) that aggregates the per-validation-example scores into a single scalar per training sample, we retain the full influence score matrix $\mathbf{S} \in \mathbb{R}^{n \times |\mathcal{V}|}$ with entries $S_{ij} = \mathcal{I}(\mathbf{x}_i, \mathbf{v}_j)$. This 2D representation preserves the fine-grained structure of each candidate's contribution to every validation example, which is essential for the relevance term in our graph cut objective (Section 3.3).

We perform LoRA fine-tuning (Hu et al., 2021) on $\mathcal{D}_{\text{warmup}}$ for T epochs, checkpointing the model after each epoch to store $\{\theta_i\}_{i=1}^{T}$ along with their Adam optimizer states. LoRA freezes the pre-trained weights and adds a low-rank adaptor, reducing the number of trainable parameters to less than 2% of the full model. Each gradient is projected into a d-dimensional space via a fixed random matrix (Johnson et al., 1984; Park et al., 2023), forming a compact gradient datastore from which the influence score matrix $\mathbf{S}$ is computed. Beyond the influence score matrix, the graph cut objective requires pairwise similarity between training examples. We construct a gradient embedding for each training example by concatenating the checkpoint-weighted gradient projections:

$$\mathbf{e}(\mathbf{x}) = \left[\sqrt{\bar{\eta}_1} \frac{\Gamma(x; \theta_1)}{\|\Gamma(x; \theta_1)\|} \,\middle\|\, \cdots \,\middle\|\, \sqrt{\bar{\eta}_T} \frac{\Gamma(x; \theta_T)}{\|\Gamma(x; \theta_T)\|} \right] \in \mathbb{R}^{Td}, \tag{2}$$

where $\|$ denotes concatenation. The $\sqrt{\bar{\eta}_i}$ weighting ensures that the inner product between two embeddings recovers the weighted cosine similarity used in the graph cut kernel (Eq. 4). Since the concatenated dimension Td can be large, we apply a Johnson–Lindenstrauss random projection (Johnson et al., 1984) to reduce each embedding to d' dimensions while preserving pairwise distances with high probability. The resulting embeddings $\tilde{\mathbf{e}}(\mathbf{x}) \in \mathbb{R}^{d'}$ are L2-normalized so that their inner product equals cosine similarity, enabling efficient similarity computation in the graph cut stage.

We propose assigning bigger weights to checkpoints sampled before convergence to compute gradient features. This is motivated by the observation that gradient norms during the warmup phase are significantly larger than those at convergence, and therefore encapsulate more discriminative information about how each sample affects the model's updates.

3.3 Query-based Graph Cut Selection

To simultaneously combat diversity collapse and early saturation of coverage-based alternatives, we build on the classical submodular graph cut framework (Iyer et al., 2021) and adapt it to the query-driven data selection setting. The standard graph cut objective selects a subset that minimizes intra-subset similarity, promoting diversity but without any notion of task relevance. We introduce a query-aware relevance term derived from the influence score matrix that steers the selection toward the target validation set, turning the diversity-seeking cut into a joint relevance-diversity objective.

The relevance of a candidate $\mathbf{x}$ to the target set is the aggregate influence over all validation examples:

$$\mathrm{Rel}(\mathbf{x},\, \mathcal{V}) \;=\; \sum_{\mathbf{v}\in\mathcal{V}} \mathcal{I}(\mathbf{x},\, \mathbf{v}), \tag{3}$$

measuring the total contribution of $\mathbf{x}$ to the entire target set. The pairwise similarity between two candidates is defined by the weighted cosine similarity of their gradient projections across checkpoints:

$$w(\mathbf{x},\, \mathbf{x}') \;=\; \sum_{i=1}^{T} \bar{\eta}_i \;\left(1 + \frac{\langle \Gamma(\mathbf{x};\theta_i),\, \Gamma(\mathbf{x}';\theta_i)\rangle}{\|\Gamma(\mathbf{x};\theta_i)\|\,\|\Gamma(\mathbf{x}';\theta_i)\|},\right). \tag{4}$$

The formula above ensure that $w(\mathbf{x}, \mathbf{x}') \geq 0$ for all pairs, which quantifies redundancy in gradient space: a high $w(\mathbf{x}, \mathbf{x}')$ indicates that the two examples induce similar parameter updates.

We seek a coreset $\mathcal{S} \subset \mathcal{D}$ of size K that maintains high relevance to $\mathcal{V}$ while exhibiting low internal similarity. We formalize this via a target-aware submodular graph cut:

$$f(\mathcal{S}) \;=\; \sum_{\mathbf{x}\in\mathcal{S}} \mathrm{Rel}(\mathbf{x},\, \mathcal{V}) \;-\; \lambda \sum_{\substack{\mathbf{x},\mathbf{x}'\in\mathcal{S}\\ \mathbf{x}\neq\mathbf{x}'}} w(\mathbf{x},\, \mathbf{x}'), \tag{5}$$

where the first term maximizes the cumulative influence of $\mathcal{S}$ on the target task, while the second term acts as a dispersion penalty that minimizes intra-subset redundancy in gradient space. The hyperparameter $\lambda > 0$ balances relevance and diversity. Two terms capture complementary properties of a selected subset. The relevance term favors examples aligned with the target capability, while the pairwise term discourages repeatedly selecting examples that induce similar gradient updates. Under a tight cardinality constraint, each redundant example consumes a larger fraction of the available selection capacity and leaves fewer slots for complementary training signals. Furthermore, unlike coverage functions that saturate when the sparse set $\mathcal{V}$ is fully covered, our dispersion penalty operates on the internal structure of $\mathcal{S}$ itself and continues to discriminate among candidates regardless of $|\mathcal{V}|$.

Because $w(\mathbf{x},\, \mathbf{x}') \geq 0$, the pairwise sum $\sum_{\mathbf{x},\mathbf{x}'\in S, \mathbf{x}\neq\mathbf{x}'} w(\mathbf{x},\, \mathbf{x}')$ is supermodular. Its negation is therefore submodular. Since the relevance term is modular, their sum $f(\mathcal{S})$ is submodular. The objective is generally nonmonotone because adding a highly redundant example can decrease its value. The resulting objective is a nonmonotone submodular maximization problem under a cardinality constraint. Randomized algorithms (Buchbinder et al., 2014) provide constant factor approximation guarantees for this class of problems. In practice, we adopt the deterministic greedy procedure in Algorithm 1 because it is simple, efficient, and empirically stable for our selection setting. We therefore do not claim that the theoretical $1/e$ approximation guarantee applies to Algorithm 1.

Algorithm 1: Query-based graph cut

Input : Candidate pool $\mathcal{D}$; validation set $\mathcal{V} = \{\mathcal{V}_1, \ldots, \mathcal{V}_m\}$; influence score matrix $\mathbf{S} \in \mathbb{R}^{n \times |\mathcal{V}|}$ with $S_{ij} = \mathcal{I}(\mathbf{x}_i, \mathbf{v}_j)$ (Eq. 1); gradient embeddings $\tilde{\mathbf{e}}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{D}$ (Eq. 2); budget K

Output : Coreset $\mathcal{S}$

```
for each x ∈ D do
    Rel(x) ← Σ_{v∈V} I(x, v)   (Eq. 3);
end
S ← ∅;
while |S| < K do
    x* ← argmax_{x∈D\S} Δ(x | S)   (Eq. 6);
    S ← S ∪ {x*};
end
return S
```

In practice, we use the deterministic greedy algorithm, as shown in Algorithm 1. At each iteration, we add the element with the largest marginal gain:

$$\Delta(\mathbf{x}^* \mid \mathcal{S}) \;=\; \mathrm{Rel}(\mathbf{x}^*,\, \mathcal{V}) \;-\; 2\lambda \sum_{\mathbf{x}' \in \mathcal{S}} w(\mathbf{x}^*,\, \mathbf{x}'). \quad (6)$$

Intuitively, $\mathbf{x}^*$ is selected when its relevance to the target task outweighs its total similarity to examples already chosen, automatically balancing exploitation against exploration. This criterion also makes the selection order interpretable, since each chosen example must provide additional target aligned information beyond what is already covered by $\mathcal{S}$. To scale to large candidate pools, we approximate $w(\mathbf{x}, \mathbf{x}')$ via the compact gradient embeddings from Section 3.2: $w(\mathbf{x}, \mathbf{x}') \approx \sum_{i=1}^{T} \bar{\eta}_i \,(1 + \langle \hat{\mathbf{e}}(\mathbf{x}),\, \hat{\mathbf{e}}(\mathbf{x}') \rangle)$, where $\hat{\mathbf{e}}$ is the L2-normalized randomly projected embedding from Eq. equation 2. The additive constant contributes the same value to the marginal gain of every remaining candidate at a fixed greedy iteration and therefore does not affect the selection ordering. This avoids recomputing over the raw gradients across all checkpoint gradients.

3.4 Gradient-Guided Synthesis

Even with diversity-aware selection, the training pool $\mathcal{D}$ may not contain examples that cover every gradient direction required by the target task. At extreme budgets, this subspace gap is especially acute: the selected subset $\mathcal{S}$ spans a low-dimensional subspace of gradient space, and some target capabilities may lie entirely outside it. We address this with a post-selection synthesis step that uses the gradient geometry of $\mathcal{S}$ to identify what is missing and generate targeted synthetic examples.

Residual subspace identification. After selecting $\mathcal{S}$ via Algorithm 1, we measure how well the selected subset covers the target validation set in gradient space. For target validation set $\mathcal{V}$, we compute the gradient residual:

$$\mathbf{r} \;=\; \bar{g}(\mathcal{V}) \;-\; \mathrm{proj}_{\mathrm{span}(\Gamma(\mathcal{S}))}\, \bar{g}(\mathcal{V}), \quad (7)$$

where $\bar{g}(\mathcal{V}) = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \Gamma(v)$ is the mean gradient of the validation set, which is averaged over checkpoints for brevity, and the projection is onto the subspace spanned by $\{\Gamma(\mathbf{x}) : \mathbf{x} \in \mathcal{S}\}$. The residual norm $\|\mathbf{r}\|$ quantifies the uncovered component of validation set: a large residual indicates that no selected example provides a gradient update aligned with this target direction.

Targeted synthesis. We leverage the warm-up model θ to generate candidate synthetic samples with the validation examples from $\mathcal{V}$. The prompt instructs the model to produce diverse instruction–response pairs that exercise the same capability from different surface contexts. We then compute the gradient features of each synthetic candidate $\tilde{\mathbf{x}}$ using the same LoRA checkpoints and random projection as the original datastore, and select the candidates whose gradient direction best aligns with the residual:

$$\tilde{\mathbf{x}}^* \;=\; \mathrm{argmax}_{\tilde{\mathbf{x}}}\; \frac{\langle\, \Gamma(\tilde{\mathbf{x}}),\; \mathbf{r} \,\rangle}{\|\Gamma(\tilde{\mathbf{x}})\|\, \|\mathbf{r}\|}. \quad (8)$$

Selected synthetic examples are added to $\mathcal{S}$, the residuals are recomputed, and the process repeats until a synthesis budget K_{syn} is exhausted or all residual norms fall below a threshold ϵ.

Naïve synthetic data generation produces examples that are semantically related to the target task but may be redundant with data already in $\mathcal{S}$. By selecting synthetic candidates based on their alignment with the residual gradient direction, we ensure that each synthetic example provides a training signal that is both target-relevant and complementary to the existing subset. This couples the generation process tightly with the graph cut framework: the residual in Eq. equation 7 is precisely the component of the target gradient that the dispersion-optimized subset failed to capture.

Table 1: Coreset selection performance at the 0.5% data budget across three models and three benchmarks. Full denotes training on all ∼270K examples; Rand. denotes uniform 0.5% random sampling; LESS (Xia et al., 2024) denotes top-K gradient influence selection. **Bold** indicates the best result among selected subsets. Standard deviations over three seeds are reported in subscript. QSGS consistently outperforms all baselines across every model–benchmark combination.

	MMLU				TydiQA				BBH			
Model	**Full**	**Rand.**	**LESS**	**QSGS**	**Full**	**Rand.**	**LESS**	**QSGS**	**Full**	**Rand.**	**LESS**	**QSGS**
Data %	(100%)	(0.5%)	(0.5%)	(0.5%)	(100%)	(0.5%)	(0.5%)	(0.5%)	(100%)	(0.5%)	(0.5%)	(0.5%)
Llama-2-7B	*51.6*	45.0$_{(1.3)}$	45.2$_{(0.5)}$	**47.9**$_{(0.4)}$	*54.0*	48.4$_{(0.9)}$	50.2$_{(0.2)}$	**52.3**$_{(0.2)}$	*43.2*	35.7$_{(0.6)}$	38.3$_{(0.5)}$	**40.1**$_{(0.6)}$
Mistral-7B	*60.4*	57.0$_{(0.5)}$	58.8$_{(0.7)}$	**60.4**$_{(0.7)}$	*57.7*	53.9$_{(0.7)}$	55.3$_{(0.5)}$	**57.5**$_{(0.3)}$	*53.0*	51.5$_{(0.6)}$	52.4$_{(0.2)}$	**53.2**$_{(0.8)}$
Llama-3-8B	*70.4*	67.0$_{(0.4)}$	67.5$_{(0.8)}$	**68.3**$_{(0.6)}$	*70.7*	67.1$_{(0.6)}$	67.7$_{(0.2)}$	**68.7**$_{(0.3)}$	*65.6*	62.3$_{(1.3)}$	62.9$_{(1.1)}$	**63.9**$_{(0.8)}$

Table 2: Full baseline comparison at 0.5% data budget on Llama-2-7B. Our method outperforms existing methods.

	Rand.	BM25	DSIR	RDS	LESS	TAGCOS	QSGS
MMLU	45.0$_{(1.3)}$	46.6$_{(0.5)}$	45.1$_{(0.1)}$	45.0$_{(0.3)}$	45.2$_{(0.5)}$	44.8$_{(0.3)}$	**47.9**$_{(0.4)}$
TydiQA	48.4$_{(0.9)}$	49.7$_{(0.7)}$	47.5$_{(0.5)}$	48.6$_{(0.1)}$	50.2$_{(0.2)}$	48.1$_{(0.3)}$	**52.3**$_{(0.2)}$
BBH	35.7$_{(0.6)}$	39.8$_{(0.5)}$	36.8$_{(0.5)}$	36.7$_{(0.2)}$	38.3$_{(0.5)}$	39.1$_{(0.7)}$	**40.1**$_{(0.6)}$

Table 3: MMLU evaluation on under different data ratios.

Data Ratio	**5%**	**1%**	**0.5%**
Rand.	46.5$_{(0.4)}$	45.9$_{(0.6)}$	45.0$_{(1.3)}$
LESS	50.2$_{(0.5)}$	46.3$_{(0.7)}$	45.2$_{(0.5)}$
QSGS	**51.1**$_{(0.3)}$	**48.5**$_{(0.2)}$	**47.9**$_{(0.4)}$

4 EXPERIMENTS

We first describe the experimental setup in Section 4.1, then present main results in Section 4.2. Section 4.3 provides ablation studies isolating the contribution of each component, and offers further analysis including comparisons of submodular objectives.

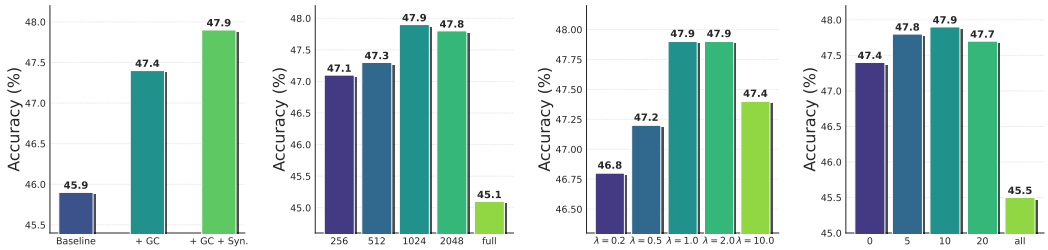
4.1 EXPERIMENTAL SETUP

Dataset and metrics. Following (Xia et al., 2024), we use a heterogeneous pool of ∼270K instruction-tuning examples drawn from four sources: Flan v2 (Longpre et al., 2023), Chain-of-Thought (Wei et al., 2022), Dolly (Conover et al., 2023), and Open Assistant (Köpf et al., 2023). This pool spans a wide range of formats and reasoning types and contains no obvious in-domain data for the target evaluation tasks, making it a realistic and challenging selection environment. We evaluate on three benchmarks that together cover diverse capability dimensions: MMLU (Hendrycks et al., 2020) comprises 57 multiple-choice tasks spanning science, humanities, and professional knowledge; BBH (Suzgun et al., 2023) contains 27 hard reasoning tasks selected from BIG-Bench; TydiQA (Clark et al., 2020) is a multilingual extractive QA dataset covering 9 typologically diverse languages. For each benchmark, the few-shot examples provided per subtask serve as the validation set $\mathcal{V}$ for data selection and as in-context demonstrations during evaluation.

Implementation details. Warmup training uses a randomly selected 5% of $\mathcal{D}$ for $T = 4$ epochs, producing 4 LoRA checkpoints. Gradients are projected to $d = 8192$ dimensions. Unless otherwise noted, we report results at a 0.5% data budget ($K \approx 1{,}350$ examples), the extreme-budget regime that is our primary focus. We additionally evaluate at 1%, 2%, and 5% budgets for the scaling study. For gradient-guided synthesis, we generate $K_{\text{syn}} = 100$ candidate examples per under-covered subtask and select up to 10 synthetic examples total. For LLama 3 training, following (Wang et al., 2023; Ivison et al., 2023; Zhang et al., 2025), we extend the training set to a combination of 17 datasets. All experiments are conducted on NVIDIA RTX A6000 GPUs. We evaluate QSGS across three base models: Llama-2-7B (Touvron et al., 2023), Llama-3-8B (Grattafiori et al., 2024), Mistral-7B (Jiang et al., 2023). All warmup and final fine-tuning are performed with LoRA (Hu et al., 2021). All ablations are conducted on Llama-2-7B / MMLU at the 0.5% budget.

4.2 MAIN RESULTS

Table 1 presents results at the 0.5% data budget ($K \approx 1{,}350$ examples). QSGS improves over LESS by 1.6% on average across all nine model–benchmark combinations, with gains in every individual setting. The improvement is most pronounced on Llama-2-7B, where QSGS leads by 2.2% on average and the gap between full-data and 0.5% performance is also the widest. This is consistent with our expectation: when the budget is tight relative to model capacity, top-K selection is more prone to spending its limited budget on redundant high-scoring examples, and diversity-aware selection yields the greatest marginal benefit. On the stronger Llama-3-8B, the absolute gain narrows to 0.9%, suggesting that a more capable base model is more tolerant of suboptimal data composition. Notably, on Mistral-7B, QSGS at 0.5% matches or approaches full-data performance on all three benchmarks,



(a) Component analysis (b) Projection dimension d (c) Trade-off parameter λ (d) Number of synthetics

Figure 3: **Ablation studies** on Llama-2-7B / MMLU at the 0.5% budget. **(a)** Incremental effect of each component. **(b)** Effect of random projection dimension in the Graph Cut kernel. **(c)** Sensitivity to the relevance–diversity trade-off parameter λ. **(d)** Effect of synthetic samples added to coreset

closing the gap to within 0.2% on each. This suggests that for sufficiently strong base models, a small but well-diversified subset can recover nearly all the knowledge that 270K heterogeneous examples provide, as most of the full training pool is redundant. Across all models, QSGS closes roughly half the gap between LESS and the full-data baseline.

We further compare QSGS against a broader set of non-gradient baselines on Llama-2-7B in Table 2. Lexical retrieval methods such as BM25 (Robertson & Zaragoza, 2009) and representation-based approaches including DSIR (Xie et al., 2023) and RDS (Hanawa et al., 2020) all perform close to random selection, trailing QSGS by 1.3% to 2.6%. These methods score each candidate independently against the target distribution and lack any mechanism to enforce diversity within the selected subset. As a result, they tend to retrieve clusters of near-duplicate examples that are individually relevant but jointly redundant, exhausting the limited budget without adequate coverage of the target gradient space. In contrast, the graph cut objective in QSGS explicitly penalizes pairwise similarity among selected samples, ensuring that each additional example contributes complementary gradient information rather than duplicating directions already represented in the coreset.

Table 3 traces the performance gap between QSGS and LESS on MMLU as the data budget decreases from 5% to 0.5%. At 5%, the two methods differ by only 0.9%, as the budget is large enough for top-K to incidentally achieve reasonable diversity. However, the gap widens steadily to 2.2% at 1% and 2.7% at 0.5%. The increasing empirical gap is consistent with the intuition that redundancy is more costly when only a small number of examples can be retained. With a larger budget, top-K selection can include redundant examples while still covering many useful gradient directions. Under a tight budget, each redundant selection directly displaces a potentially complementary example. The submodular objective of QSGS directly addresses this by balancing relevance and coverage, allowing it to maintain broad gradient coverage even under tight budgets. Quantitatively, LESS degrades by 5.0% from the 5% budget to the 0.5% budget, while QSGS degrades by only 3.2% over the same range. Notably, QSGS at 0.5% budget already surpasses LESS at 1% budget.

4.3 Ablation Studies

Component analysis. To isolate the contribution of each proposed component, we incrementally add them to our baseline (Figure 3a). The baseline relies entirely on top-K selection using computed gradient features, without any synthetic samples. We evaluate two extensions: "+GC", which replaces top-K with our query-based Graph Cut, and "+GC+Syn", which represents the full method by adding gradient-guided sample synthesis.Replacing the baseline's top-K selection with the Graph Cut objective accounts for the majority of the performance gain, increasing MMLU accuracy from 45.9 to 47.4. This confirms that diversity-aware selection is the primary driver of improvement. Building on this, gradient-guided synthesis provides a complementary boost by filling residual gaps in gradient coverage that the existing pool cannot address. This diminishing return from synthesis is entirely by design: because it specifically targets the subspace not spanned by pool-selected examples, its impact naturally shrinks as the quality of the initial pool selection improves.

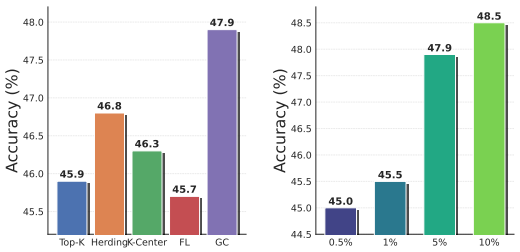
Random projection dimension. We vary the projection dimension d' of the compact gradient embeddings $\tilde{\mathbf{e}}(\mathbf{x}) \in \mathbb{R}^{d'}$ used for pairwise similarity computation in the Graph Cut objective (Eq. 2). Performance peaks at d'=1024 and plateaus at d'=2048 (Figure 3b). For full-dimensional gradients, we use a chunked CPU–GPU transfer to accommodate the full vectors in memory during selection. Notably, operating on full-dimensional gradients degrades performance to 45.1, below random projection at every tested dimension. We attribute this to distance concentration in high-

dimensional spaces. As the gradient dimensionality grows, pairwise cosine similarities concentrate near zero, causing the redundancy penalty in the Graph Cut objective to vanish. When the penalty term becomes negligible, Graph Cut effectively reduces to top-K relevance ranking and loses its diversity-enforcing capability. Random projection into a lower-dimensional subspace compresses the gradient representations so that semantically similar examples exhibit meaningfully non-zero similarity, thereby restoring the discriminative power of the redundancy term. Random projection reduces computational cost and acts as an implicit regularizer that preserves dominant gradient structure while suppressing the high-dimensional noise that would neutralize the diversity objective.

Sensitivity to λ. Performance varies by only 1.1% across a $50\times$ range of λ (Figure 3c). The optimum lies at a small λ where relevance remains the dominant selection criterion and the redundancy penalty acts only as a mild corrective. It is sufficient to suppress near-duplicate examples whose gradient directions are nearly identical, yet too weak to override relevance-based ranking. This regime retains the highest-quality, task-aligned examples while eliminating pure redundancy, allowing the model to converge efficiently without wasting capacity on repeated training signal. At the other extreme, λ=10.0 over-weights the diversity penalty, forcing the objective to sacrifice relevance for dissimilarity and thereby selecting uninformative examples.

Number of synthesized candidates. We vary the number of synthetic samples added to the coreset after Graph Cut selection (Figure 3d). Performance peaks at 10 samples and degrades sharply when all candidates are included, even falling below the no-synthesis baseline. This is consistent with the residual-guided selection in Eq. equation 8: the first few synthetic examples align with the largest uncovered gradient directions and provide genuinely complementary signal; once these gaps are filled, additional candidates inevitably overlap with existing coreset members and dilute the diversity that Graph Cut optimization establishes.

Effect of selection strategy. We replace QSGS's Graph Cut (GC) objective with alternative submodular functions, namely Herding, K-Center, and Facility Location (FL), while keeping the same candidate buffer and gradient features, to isolate the role of the objective design. GC achieves the best performance, outperforming Herding and K-Center by 1.1% and 1.6%, respectively. Most notably, FL performs worse than even the top-K baseline. We trace this to the small size of the validation set: FL saturates coverage after selecting only $\sim$10 examples, at which point the marginal gain of every remaining candidate drops to near zero and subsequent selections are effectively random. The root cause lies in FL's $\max$-based aggregation, which scores each candidate by its similarity to the single closest selected example. Once every validation point has a nearby representative, the marginal gain of additional candidates vanishes. GC avoids this saturation by computing cumulative similarity between all selected samples and the validation set, so each new addition continues to yield positive marginal gains.



(a) Comparison of selection methods

(b) Warmup data ratio influence

Figure 4: Analysis on MMLU at the 0.5% budget.

Warmup data ratio. Performance improves steadily as the warmup ratio increases from 0.5% to 10%, reflecting higher-quality gradient features (Figure 4b). We adopt 5% as the default: the gain from 5% to 10% is only 0.6% while doubling the warmup cost, indicating diminishing returns beyond 5%. The sharp drop at 0.5% warmup, which falls below even random selection, underscores that gradient-based selection fundamentally depends on the fidelity of the influence estimates; noisy gradients make even diversity-aware optimization ineffective.

5 Conclusion

We presented QSGS, a coreset construction framework for targeted instruction tuning that combines query-based submodular data selection with gradient-guided data synthesis. The graph cut objective jointly maximizes target relevance and minimizes intra-subset redundancy. Empirically, its advantage over pointwise selection becomes increasingly pronounced under tighter data budgets, where redundant selections consume a larger fraction of the available training capacity. A complementary synthesis step fills residual coverage gaps by generating examples aligned with uncovered target gradient directions. Experiments across three benchmarks and three model families show that QSGS at a 0.5% budget consistently outperforms all baselines, achieving full-data performance on Mistral-7B.

AI USE STATEMENT

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

REFERENCES

Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *ECCV*, 2020.

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2014.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. Alpagasus: Training a better alpaca with fewer data. In *ICLR*, 2024.

Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. *arXiv preprint arXiv:1203.3472*, 2012.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of machine learning research*, 2024.

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 ty pologically di verse languages. *Transactions of the Association for Computational Linguistics*, 2020.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*, 2019.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL `https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm`.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. Mods: Model-oriented data selection for instruction tuning. *arXiv preprint arXiv:2311.15653*, 2023.

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

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.

Kazuaki Hanawa, Sho Yokoi, Satoshi Hara, and Kentaro Inui. Evaluation of similarity-based explanations. *arXiv preprint arXiv:2006.04528*, 2020.

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

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

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*, 2023.

Hamish Ivison, Muru Zhang, Faeze Brahman, Pang Wei Koh, and Pradeep Dasigi. Large-scale data selection for instruction tuning. *arXiv preprint arXiv:2503.01807*, 2025.

Rishabh Iyer, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. Submodular combinatorial information measures with applications in machine learning. In *Algorithmic Learning Theory*. PMLR, 2021.

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

Tingyu Jiang, Shen Li, Yiyao Song, Lan Zhang, Hualei Zhu, Yuan Zhao, Xiaohang Xu, Kenjiro Taura, and Hao Henry Wang. Importance-aware data selection for efficient llm instruction tuning. In *AAAI*, 2026.

William B Johnson, Joram Lindenstrauss, et al. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 1984.

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

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI conference on artificial intelligence*, 2021b.

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

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

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. In *NeurIPS*, 2023.

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

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. Mlqa: Evaluating cross-lingual extractive question answering. In *ACL*, 2020.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024a.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Ling-Hao Chen, Junhao Liu, Tongliang Liu, et al. One-shot learning as instruction data prospector for large language models. In *ACL*, 2024b.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, 2011.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *ICLR*, 2024a.

Zifan Liu, Amin Karbasi, and Theodoros Rekatsinas. Tsds: Data selection for task-specific model finetuning. In *NeurIPS*, 2024b.

Zikang Liu, Kun Zhou, Wayne Xin Zhao, Dawei Gao, Yaliang Li, and Ji-Rong Wen. Less is more: High-value data selection for visual instruction tuning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025.

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 *ICML*. PMLR, 2023.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764*, 2021.

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

Nihal V Nayak, Paula Rodriguez-Diaz, Neha Hulkund, Sara Beery, and David Alvarez-Melis. A critical look at targeted instruction selection: Disentangling what matters (and what doesn't). *arXiv preprint arXiv:2602.14696*, 2026.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

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

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

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *NeurIPS*, 2020.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and trends® in information retrieval*, 2009.

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

Samarth Sinha, Han Zhang, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle, and Augustus Odena. Small-gan: Speeding up gan training using core-sets. In *ICML*, 2020.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2023.

Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data pruning via moving-one-sample-out. In *NeurIPS*, 2023.

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

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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. In *NeurIPS*, 2023.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

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. In *NeurIPS*, 2022.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *ICML*. PMLR, 2015.

Minghao Wu, Thuy-Trang Vu, Lizhen Qu, and Gholamreza Haffari. The best of both worlds: Bridging quality and diversity in data selection with bipartite graph. *arXiv preprint arXiv:2410.12458*, 2024.

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

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*, 2023.

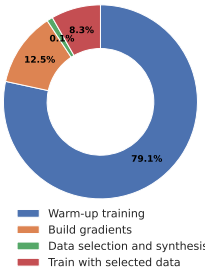
Jipeng Zhang, Yaxuan Qin, Renjie Pi, Weizhong Zhang, Rui Pan, and Tong Zhang. Tagcos: Task-agnostic gradient clustered coreset selection for instruction tuning data. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025.

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

A More Analysis

Computational overhead. We report the wall-clock cost of each pipeline stage on NVIDIA RTX A6000 GPUs in Figure 5a. Warmup LoRA training requires approximately 304 GPU hours, gradient feature construction approximately 48 GPU hours, data construction including graph cut selection and optional synthesis less than one GPU hour, and final fine-tuning on the selected coreset approximately 32 GPU hours. Importantly, the warmup training and gradient computation are one-time costs that can be fully amortized: once the gradient features are stored, the data construction stage can be repeated in under 1 hour to produce coresets for different target validation sets, different data budgets, or different submodular trade-off parameters, without re-running any GPU-intensive computation. QSGS is therefore particularly suitable for settings in which multiple target-specific subsets are constructed from the same candidate pool.

Generated candidate examples. Figure 5b shows representative candidates produced for MMLU subtasks identified as under-covered by the residual analysis. The generated questions span several domains and exhibit varied surface forms. These examples provide qualitative evidence of topical alignment. Gradient residual alignment measures whether a generated example provides a complementary optimization signal. Developing an explicit semantic or correctness filter for generated candidates is an important direction for improving this component.



(a) Computational cost.

1. Which of the following is not a common cause of low back pain?
A. Bed rest B. Stress C. Poor posture D. Sedentary lifestyle
2. Which of the following is the more common source of heat-related illness?
A. Heat exhaustion B. Heat stroke C. Sunstroke D. Heat cramps E. Heat rash
3. The _____ is the process of using information to improve or enhance the efficiency of the organization.
A. Management information systems B. Information systems C. Management information systems D. Information systems E. Information systems
4.Which of the following is NOT a function of the nucleus?
A. The nucleus is the control center of the cell.
B. The nucleus is involved in protein synthesis.
C. The nucleus is involved in cell division.
D. The nucleus is involved in DNA replication.
E. The nucleus is involved in RNA synthesis.

(b) Generated samples.

Figure 5: Analysis on Llama-2-7B / MMLU at the 0.5% budget. **(a)** Computational cost breakdown of each pipeline stage. **(b)** Representative synthetic examples generated for under-covered subtasks.

B Limitations

The quality of the gradient similarity graph depends on the fidelity of the warmup checkpoints and the projected gradient features. Our ablations show that QSGS is robust to a broad range of projection dimensions and trade-off parameters, yet very small warmup sets can produce noisy influence estimates. In addition, our empirical study focuses on 7B–8B open-weight models and three representative benchmarks. Extending the evaluation to larger models, additional domains, and multi-turn or tool-use tasks is an important direction for future work, especially because the same stored gradient features can be reused to explore new target sets with limited additional cost.