PICK THE MIXTURE, NOT THE SAMPLES: COMBINATORIAL DATA SELECTION FOR LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Data selection for LLM fine-tuning faces two facts about real training: data pools draw from many distinct domains, and evaluation spans diverse tasks. As a result, the same sample helps different tasks differently, and samples interact during training so that a subset's value is not the sum of its parts. Current data-selection methods leave these realities largely unaddressed. We formalize them as *Combinatorial Data Selection* (CDS), a framework that separates *attribution* (how samples are scored) from *selection* (which subset to pick) and models the non-additive interactions among samples. On a real instruction-tuning data pool, we find substantial cross-domain complementarity (joint utility exceeds the sum of singletons), alongside redundancy within domains. Inspired by this, we propose *Mixture-then-Select*, a two-stage algorithm that first decides how many samples to take from each domain, then ranks samples within each domain by attribution score. Theoretically, the gap to the optimal subset is linear in the selection size and independent of data pool size. Empirically, Mixture-then-Select outperforms strong baselines by 20 to 54%, is robust across attribution methods, and works across different models and data pool sizes. The framework also extends to the data purchasing problem under a budget constraint when sample prices vary. Our findings suggest that allocating selection across domains offers a promising path toward more efficient and effective data selection for modern LLMs.

1 INTRODUCTION

From a large data pool spanning many skill domains (e.g., math, code, dialog, instruction following), which small subset should we train on for LLM fine-tuning? Training on a well-chosen small subset often matches or beats training on the full data pool (Zhou et al., 2023; Liu et al., 2024; Xia et al., 2024; Li et al., 2024b): what is selected matters more than how many. But two features of training data make selection hard. First, samples help different evaluation tasks unevenly: a sample that helps one task may be useless or harmful for another. Second, samples interact during training: training on two samples together can yield more than the sum of their contributions (*complementary*) or less (*substitutive*).

Together, these facts mean per-sample scoring is fundamentally insufficient: it ignores how chosen samples combine and which tasks each is meant to serve. We formalize this as *Combinatorial Data Selection* (CDS), in which sample scoring (*attribution*) and subset choice (*selection*) are separate decisions. CDS extends classical submodular selection (e.g., Wei et al., 2015; Mirzasoleiman et al., 2020) in two ways: (i) it admits both complementary and substitutive sample interactions, whereas submodular formulations capture only the substitutive (diminishing-returns) case; and (ii) it handles multiple downstream tasks jointly rather than one at a time. Any per-sample attribution score (e.g., Xia et al., 2024; Koh & Liang, 2017) plugs into the rule.

As solving CDS exactly is intractable, we propose *Mixture-then-Select* (MtS), a two-stage algorithm that first decides how many samples to take from each domain based on per-domain utility measurements (capturing cross-domain effects), then selects the top samples within each domain by attribution score. MtS is attribution-agnostic (it plugs into any per-sample score, whether oracle values, gradient-based, loss-based, or other) with a theoretical guarantee: its utility is provably within an additive gap of the optimal subset's that grows linearly in the number of samples picked and is independent of data pool size, so quality does not degrade as pools grow.

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.

Empirically, we evaluate MtS across diverse skill domains, evaluation tasks, model families (Qwen3-8B, Llama-3.1-8B-Instruct), and data pool sizes (50K to 200K). We first find that the data pool exhibits substantial cross-domain complementarity, which MtS's domain-mixture decision exploits. Results show that MtS outperforms eight baselines across four selection principles by 20 to 54% relative in held-out utility: per-sample quality (LESS (Xia et al., 2024), EL2N (Paul et al., 2021), Superfiltering (Li et al., 2024a), DEITA (Liu et al., 2024)), diversity (k-DPP (Kulesza & Taskar, 2011), Facility Location (Wei et al., 2015)), gradient matching (CRAIG (Mirzasoleiman et al., 2020)), and partition-based (Stratified). Additional comparisons with recent mixture-optimization methods across diverse downstream benchmarks provide further evidence of MtS's robustness across evaluation tasks and data domain partitions. MtS also applies to buying training data when samples have different prices and the buyer has a fixed budget. Across every budget level we tested, MtS reaches $\geq 89\%$ of the best achievable performance and matches the optimum on 58% instances.

In summary, this paper makes three contributions. **(i) Framework.** We frame LLM data selection as combinatorial allocation across domains, separating attribution from selection and capturing both complementary and substitutive sample interactions. **(ii) Methodology.** We propose a two-stage algorithm that first picks the per-domain mixture and then ranks samples within each domain, with both theoretical and empirical performance guarantees. **(iii) Application.** MtS gives LLM practitioners a practical algorithm for a real-world problem: deciding which training data to buy under a fixed budget.

2 Related work

Table 1: Comparison with related methods. Columns indicate whether a method models sample-pair interactions, supports cardinality or nonuniform-price budgets, provides a subset-utility approximation guarantee, and optimizes domain allocation.[1]

Direction	Method	Interactions	Size	Budget	Guarantee	Mixture
Data attribution	Data Shapley (Ghorbani & Zou, 2019)	✗	✗	✗	✗	✗
	Per-sample scores	✗	✗	✗	✗	✗
Data selection and distillation	Dataset distillation (Wang et al., 2020; Zhao et al., 2021)	✗	✓	✗	✗	✗
	k-DPP (Kulesza & Taskar, 2011)	✗	✓	✗	✗	✗
	DoReMi (Xie et al., 2023) / RegMix (Liu et al., 2025)	✗	✗	✗	✗	✓
	Stratified	✗	✓	✗	✗	✓
	CRAIG (Mirzasoleiman et al., 2020) / Facility Loc. (Wei et al., 2015)	✗	✓	✗	✓	✗
	SMART (Renduchintala et al., 2024)	✗	✓	✗	✓	✓
Data pricing and markets	Data Banzhaf (Wang & Jia, 2023) / DVRL (Yoon et al., 2020)	✗	✗	✗	✗	✗
	Fairshare (Zhang et al., 2025) / Data Market Design (Ravindranath et al., 2023)	✗	✗	✓	✗	✗
Ours	Mixture-then-Select	✓	✓	✓	✓	✓

Data attribution. Per-sample scores fall on a cost spectrum: a single forward pass for instruction-tuning quality classifiers (Liu et al., 2024; Li et al., 2024a;b) and EL2N (Paul et al., 2021); per-sample gradients for LESS (Xia et al., 2024), TracIn (Pruthi et al., 2020), and Hessian-based influence approximations (Kwon et al., 2024; Basu et al., 2021; Koh & Liang, 2017); repeated training-trace queries for Shapley (Ghorbani & Zou, 2019) and datamodels (Ilyas et al., 2022). Downstream selection ranks and thresholds the scores.

Data selection and distillation. Subset-level methods reason about S as a whole: submodular and coreset approaches under cardinality constraints (Wei et al., 2015; Mirzasoleiman et al., 2020; Renduchintala et al., 2024; Maharana et al., 2024; Nemhauser et al., 1978; Bian et al., 2017; Das & Kempe, 2011; Harshaw et al., 2019; Sviridenko, 2004; Mirzasoleiman et al., 2015; Feige, 1998; Vondrák, 2013) (CRAIG, Facility Location), determinantal point processes (Kulesza & Taskar, 2011; Kothawade et al., 2022), and dataset distillation (Wang et al., 2020; Zhao et al., 2021). DoReMi (Xie et al., 2023) learns proxy-derived domain proportions; RegMix (Liu et al., 2025) predicts them from proxy-mixture runs; DIDS (Shi et al., 2025) updates sampling ratios during training. MtS instead studies a complementary question: which individual samples to acquire before training under a cardinality or price budget, accounting for pairwise interactions. Möbius interaction indices (Grabisch & Roubens, 1999; Tsai et al., 2023) are the decomposition tool our framework uses for pairwise utility.

Data pricing and markets. Data pricing values data as an economic asset: Shapley (Ghorbani & Zou, 2019) and Banzhaf (Wang & Jia, 2023) contributions, RL-based valuation (Yoon et al., 2020),

[1] Similarity kernels and gradient surrogates do not count as Interactions; convergence guarantees on inner optimization do not count as Guarantee.

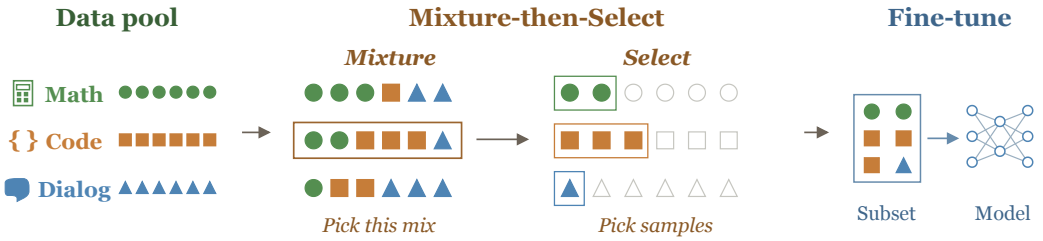

Figure 1: Mixture-then-Select first uses a learned utility predictor to choose how many samples to take from each domain, then selects samples ranked by attribution score within each domain. The selected subset is used for fine-tuning.

market design under incentive constraints (Ravindranath et al., 2023), and LLM-specific fair-share pricing (Zhang et al., 2025). Table 1 summarizes the three threads.

3 Combinatorial Data Selection

Training-data selection is a combinatorial problem: a subset's value depends on how its samples interact during fine-tuning, not just on which samples it contains. We formalize selection as a constrained optimization (Section 3.1), propose an efficient algorithm to solve it (Section 3.2), and prove a theoretical performance guarantee (Section 3.3).

3.1 Data Selection and Pairwise Interactions

We introduce the data selection problem, modeling how training samples interact during fine-tuning, and a structural decomposition of those interactions.

The selection problem. An LLM developer starts with a base model and a data pool $\mathcal{D} = \{1, \ldots, N\}$ of N candidate training samples, each carrying a per-sample cost $p_j > 0$. The LLM developer cares about performance across T evaluation tasks (e.g., code generation, mathematical reasoning, instruction following). The data pool is partitioned by skill into T *domains* $\mathcal{D} = \bigsqcup_{c=1}^{T} C_c$, one per task. For each task t, fine-tuning on any subset $S \subseteq \mathcal{D}$ yields a task utility $u_t(S) \geq 0$, the improvement in task-t performance *over the base model*, with $u_t(\emptyset) = 0$ by construction and higher u_t corresponding to a better-trained model. These per-task utilities combine into a single scalar objective via an aggregation function f:

$$u(S) = f\big(u_1(S), \ldots, u_T(S)\big). \tag{1}$$

Modern LLMs are evaluated across multiple tasks, which developers may prioritize differently. A natural choice for f to incorporate this multi-objective structure is the *weighted average* $f = \sum_t w_t u_t$, with task weight w_t denoting the relative importance the LLM developer attaches to task t (e.g., math weighted above chit-chat); alternative aggregators (minimum, concave) are discussed in Appendix A.

With a budget B on total cost, the LLM developer selects a subset $S \subseteq \mathcal{D}$ to maximize the aggregate utility $u(S)$ subject to $c(S) = \sum_{j \in S} p_j \leq B$. Prior data-selection methods (e.g., Xia et al., 2024) are **special cases** of this formulation in two ways: they target a single task (T=1) and assume uniform per-sample cost (p_j=1, a cardinality constraint). Our framework is **more general**, supporting arbitrary T, any aggregator f, and heterogeneous per-sample costs, making it a natural fit for data-marketplace settings where different vendors set different prices (e.g., Zhang et al., 2025).

Pairwise approximation of set utility. Training samples interact during fine-tuning, so the set utility $u(S)$ is non-additive: the utility of a subset is not the sum of its singleton utilities (Ghorbani & Zou, 2019). The full set function thus has 2^N possible values and cannot be reduced to per-sample terms. To capture the interactions tractably, we use the Möbius inversion formula (Grabisch & Roubens, 1999; Tsai et al., 2023), a decomposition of set utility into individual-sample contributions and interactions of increasing order. Keeping only the singleton and pairwise contributions yields

$$u(S) = \sum_{j \in S} v_j + \sum_{\{i,j\} \subseteq S} \varphi_{ij} + \varepsilon(S), \tag{2}$$

where $v_j = u(\{j\})$ is the singleton value, $\varphi_{ij} = u(\{i,j\}) - v_i - v_j$ is the pairwise interaction, and $\varepsilon(S)$ collects contributions from triplets and higher-order interactions. Pairwise truncation is the standard choice for such decompositions (Grabisch & Roubens, 1999; Tsai et al., 2023); on our data pool, the pairwise approximation $\tilde{u}$ captures $\geq 90\%$ of the variance in $u(S)$ across ~ 100 measured triplets ($R^2 = 0.90$, with pairwise residuals below the seed-noise floor; Appendix A), and higher-order interactions are infeasible to estimate at our data pool size. The pairwise interactions form a symmetric matrix $\Phi := (\varphi_{ij})_{i,j}$. We write $\tilde{u}(S) := u(S) - \varepsilon(S) = \sum_{j \in S} v_j + \sum_{\{i,j\} \subseteq S} \varphi_{ij}$ for this pairwise approximation, the deterministic truncation our analysis works with.

Decomposition of the pairwise utility. The pairwise utility matrix Φ encodes interactions between every pair of samples, but a selection algorithm only faces two coarser choices: *which domains S draws from* (the mixture composition $\mathbf{n}(S) = (|S \cap C_c|)_{c=1}^T$) and *which specific samples within each domain*. To align the matrix with these two choices, we decompose Φ along the domain partition $\mathcal{D} = \bigsqcup_c C_c$ into three pieces:

$$\Phi \;=\; \Phi_w \;+\; \Phi_{\text{block}} \;+\; \Phi_{\text{res}}, \tag{3}$$

where the within-domain interaction matrix Φ_w is block-diagonal (non-zero only on within-domain pairs); the cross-domain block-mean matrix Φ_{block} is zero within each domain and equals the cross-domain block mean $\bar{\varphi}_{c,c'} := \frac{1}{|C_c||C_{c'}|} \sum_{i \in C_c, j \in C_{c'}} \varphi_{ij}$ on each cross-domain block ($c \neq c'$); and the residual matrix $\Phi_{\text{res}} = \Phi - \Phi_w - \Phi_{\text{block}}$ captures the deviation from those block means. The within-domain choice is governed by Φ_w, and the mixture choice by Φ_{block}. Equation 3 separates within-domain and cross-domain interactions. We evaluate the pairwise approximation empirically in Section 4.4.

3.2 The Mixture-then-Select algorithm

Selecting a subset S to maximize Eq. (1) is hard for two reasons. First, the search space of subsets is exponential in the data pool size N, so direct enumeration is infeasible (the problem is APX-hard, ruling out arbitrarily good polynomial-time approximation; Theorem 16). Second, sample-level methods like LESS (Xia et al., 2024), DEITA (Liu et al., 2024), and influence-based selectors (Koh & Liang, 2017) score each sample independently and take the top-m by score, which ignores how the resulting subset distributes across domains and so cannot exploit the domain mixture. We design an algorithm that exploits the domain structure of the pairwise decomposition Eq. (3).

For selection size m, any algorithm selecting S from the domain-partitioned data pool makes two distinct choices: the domain mixture (how many samples to take from each domain) and the within-domain identities (which specific samples to take inside each domain). The decomposition $\Phi = \Phi_w + \Phi_{\text{block}} + \Phi_{\text{res}}$ tells us these two choices are decoupled at the pairwise level. The cross-domain block-mean term Φ_{block} depends on S only through the mixture $\mathbf{n}(S) = (n_c)_{c=1}^T$, so Stage 1 can target it with a small T-dimensional search over feasible mixtures. The within-domain term Φ_w is fixed once the within-domain identities are, so Stage 2 can target it by ranking samples within each domain on their singleton values v_j. The residual Φ_{res} is uncontrolled by either stage and contributes to the bound's gap. MtS implements this two-stage rule: pick the domain mixture first, then fill each domain by within-domain ranking.

Algorithm 1 Mixture-then-Select

Require: data pool $\mathcal{D} = \bigsqcup_c C_c$, per-sample values $\{v_j\}_{j \in \mathcal{D}}$, selection size m, set-utility predictor $\hat{g}$
Ensure: subset $S \subseteq \mathcal{D}$ with $|S| = m$

```
n̂ ← arg max_{n ∈ Z^T_{≥0}, Σ_c n_c = m, n_c ≤ |C_c|}  ĝ(n)        ▷ Stage 1: predict domain mixture
S ← ∅
for domain c ∈ {1, ..., T} do
    S_c ← top n̂_c samples in C_c ranked by v_j                    ▷ Stage 2: within-domain selection
    S ← S ∪ S_c
end for
return S
```

For Stage 1 we maximize a learned predictor $\hat{g}$ over the space of feasible domain mixtures rather than the pairwise approximation $\tilde{u}$ directly: under the uniform-error hypothesis in Appendix A.16, a rank-r regressor needs $L = O(rT/\varepsilon^2)$ probes at fixed confidence (Theorem 15), whereas direct

optimization of $\tilde{u}$ would require estimating all $\binom{T}{2}$ cross-domain block means in Φ_{block}. The predictor $\hat{g}$ is fit by regressing measured utility on these probe runs (Appendix A). The per-sample values v_j may come from any source (oracle losses, gradient-based scores, or influence functions), so MtS drops in over LESS, DEITA, and influence-based selectors without modifying the value source. Algorithm 1 is the unit-cost specialization, where every sample has equal cost (p_j=1) so the budget B from Section 3.1 reduces to the selection size m. The domain-uniform-cost extension (Section 3.3) replaces this constraint with a price-weighted budget without changing the algorithm. Runtime is dominated by sorting each domain ($O(N \log N)$) plus a fixed-cost Stage 1 search over feasible mixtures, both negligible against the fine-tuning runs themselves.

Formal assumptions on u and Φ, the operator-norm refinement that drives the bound, the two-source decomposition of complementarity (Proposition 3), and the adaptive IA-Greedy extension at T=1 are all in Appendix A.

3.3 Approximation guarantees

Section 3.2 gave an algorithm; we now bound its utility gap to the optimal subset S^* in Eq. (1). A worst-case guarantee explains how two-stage decoupling scales with selection size and how errors in the interaction matrix Φ and singleton values v_j affect selection quality. We use $m = B$ under unit costs and discuss the cost-weighted extension below. Under the stated assumptions, the structural gap grows linearly in m, rather than quadratically, and is independent of pool size N (Theorem 1). We then establish tightness (Theorem 13) and robustness to estimated Φ and v_j, with additional error proportional to estimation error (Theorems 2 and 19).

The bound decomposes the gap by stage: Stage 1 incurs a cross-domain term and Stage 2 incurs a within-domain term, each controlled by the *operator norm* of a residual matrix, a single number summarising the matrix's dominant signal direction (formally, its largest singular value). This is a factor-m improvement over the entrywise Bounded-Complementarity (BC(β)) form (Bian et al., 2017), which puts a uniform upper bound on every entry of Φ; we compare both forms after Theorem 1. Intuitively, the operator norm captures Φ's coherent signal while BC(β) sums up worst-case per-pair magnitudes that often cancel in low-rank interaction matrices; this is why the spectral form is dramatically tighter on real $\hat{\Phi}$ ($\approx 51\times$ on our calibrated data pool, Appendix A). In Section 4, we empirically verify the scaling pattern of this bound. Two structural constants govern the bound: the within-domain operator norm $\beta^{(c),\text{cent}}_{w,\text{op}}$ on domain c's pairwise-interaction block (centered by the within-domain mean $\bar{\varphi}_{c,c}$), and the cross-domain operator norm δ_{op} on Φ_{res}. Global mixture identification (Assumption 6) requires uniform predictor error below half the gap between the optimal mixture and all others. A finite-sample noise term $\mathcal{R}(m)$ scales linearly in m and collects three sources of estimation error: the empirical residual matrix $\hat{\Phi}_{\text{res}}$, the predictor $\hat{g}$ fitted from probe runs, and its argmax (full breakdown in Theorem 5). The Möbius truncation tail $E(m)$ is the higher-order remainder from Eq. (2) (bounded in Theorem 8); it is a fixed structural quantity that does not vanish with more probe runs.

Theorem 1 (Mixture-then-Select Spectral Guarantee)**.** *Let $S^* = \arg\max_{|S|=m} u(S)$ and S_{MtS} the output of Algorithm 1. Under Assumptions 1 and 3 to 5 and global mixture identification (Assumption 6), with probability $\geq 1-\eta$ (over the probe-run randomness used to estimate $\hat{\Phi}$ and the predictor $\hat{g}$),*

$$u(S_{MtS}) \;\geq\; u(S^*) \;-\; \underbrace{\sum_{c=1}^{T} \beta^{(c),cent}_{w,op}\, n^*_c\, v_{\max}}_{\textit{within-domain gap}} \;-\; \underbrace{\delta_{op}\, m\, v_{\max}}_{\textit{cross-domain gap}} \;-\; \mathcal{R}(m) \;-\; 2\,E(m), \tag{4}$$

where $n^*_c = |S^* \cap C_c|$ is the optimum's per-domain count and $v_{\max} = \max_j v_j$ is the maximum singleton value.

Each term in Theorem 1 maps directly to a piece of the domain decomposition: the within-domain gap is the contribution of Φ_w, the cross-domain gap $\delta_{\text{op}} m v_{\max}$ is the contribution of Φ_{res} (the only piece neither stage controls), $\mathcal{R}(m)$ collects finite-sample noise, and $E(m)$ is the higher-order Möbius truncation tail. The within-domain gap charges Stage 2 for sub-optimal picks within each domain; each domain contributes only in proportion to its share n^*_c of the optimum, so a heterogeneous domain spectrum tightens the bound below the uniform-domain worst case $\beta^{\text{cent}}_{w,\text{op}} m v_{\max}$. The cross-domain

gap $\delta_{\text{op}} m v_{\max}$ charges Stage 1 for composition error against $\mathbf{n}^*$. Both gaps are linear in m (rather than m^2) and do not grow with T. The bound applies at sub-domain budgets ($m < |C_c|$); the constants are pessimistic. It describes scaling and estimation robustness rather than predicting the observed gap (numerical instantiation in Appendix A).

The linear-in-m scaling is unavoidable: the bound is tight up to a factor of 2 on each axis, with matching lower-bound constructions in Appendix A (Theorem 13). Both Theorem 1 and Theorem 13 assume access to the population pairwise interaction matrix Φ. In practice Φ is estimated from a finite probe set; let the empirical estimate $\hat{\Phi}$ have operator-norm error $\|\hat{\Phi} - \Phi\|_{\text{op}} \leq \epsilon_{\text{op}}\, v_{\max}$. The bound picks up an additive penalty linear in m, matching the structural terms.

Theorem 2 (Operator-Norm Robustness)**.** *If $\|\hat{\Phi} - \Phi\|_{op} \leq \epsilon_{op} v_{\max}$, then Theorem 1 holds for $\hat{\Phi}$ with $\beta_{w,op}^{(c),cent}$ and δ_{op} replaced by $\beta_{w,op}^{(c),cent} + \epsilon_{op}$ and $\delta_{op} + \epsilon_{op}$, picking up an additive $2\,\epsilon_{op}\, m\, v_{\max}$ gap.*

A symmetric robustness result for v_j-noise enters the bound additively (Corollary in Appendix A): under uniform value-noise $\|\hat{v} - v\|_\infty \leq \epsilon_v\, v_{\max}$ (ℓ_∞ error), the bound picks up an additive $2\,\epsilon_v\, m\, v_{\max}$ gap, matching Theorem 2's structural form. MtS extends to cost-weighted selection (data-marketplace deployment) via a one-line change to Stage 1's constraint, a price-weighted budget $\sum_c p_c\, n_c \leq B$ replaces cardinality, with Theorem 1's bound carrying over to the realized cardinality (Appendix A); sample-level baselines would require algorithmic redesign. Three further results are deferred to Appendix A: the adaptive IA-Greedy extension at $T{=}1$ with classical $(1{-}e^{-\gamma_{\text{op}}})$ guarantee, computational hardness (APX-hardness via MAX-CUT), and predictor sample complexity.

4 Experiments and Analysis

We compare performance across baselines and conditions (Section 4.2), account for computational efficiency (Section 4.3), and analyze the mechanism behind the gains and where the method falls short (Section 4.4).

4.1 Setup

Data pool, models, metric. The data pool is a $50K$-sample instruction-tuning corpus from 13 public sources, partitioned into $T{=}8$ skill domains (math, code, QA, dialog, reasoning, NLP, complex-instruction, creative); we additionally evaluate on a $4\times$ larger $N{=}200K$ data pool. Qwen3-8B is the main base model; Llama-3.1-8B-Instruct (different architecture) and Llama-3.2-3B-Instruct (smaller scale) test cross-model transfer. All base models are LoRA fine-tuned at rank $r{=}16$. We measure per-domain held-out loss reduction $u_c(S)$ relative to the base model weighted on various tasks; held-out loss reduction is the standard target in recent instruction-tuning data-selection work (Zhou et al., 2023; Liu et al., 2024; Xia et al., 2024; Li et al., 2024b) and tracks downstream benchmark gains in similar regimes. Selection sizes span $\sim$1% to 20% of the data pool. See detailed setups in Appendix B.

Baselines. We compare MtS against eight baselines covering four method directions: *per-sample quality* (LESS (Xia et al., 2024), EL2N (Paul et al., 2021), Superfiltering (Li et al., 2024a), DEITA (Liu et al., 2024)), *diversity* (k-DPP (Kulesza & Taskar, 2011), Facility Location (Wei et al., 2015)), *gradient matching* (CRAIG (Mirzasoleiman et al., 2020)), and *partition-based* (Stratified, which allocates the budget uniformly across the T skill domains and ranks within each by oracle; Random, which samples uniformly from the data pool with no per-domain structure), one representative per principle. Before selection, MtS runs a one-time per-base-model calibration that produces a per-sample value used for within-domain ranking; final LoRA fine-tuning is identical across methods. We additionally compare with recent mixture-optimization methods DoReMi, RegMix and DIDS (Xie et al., 2023; Liu et al., 2025; Shi et al., 2025).

4.2 Performance comparisons

We first verify that our data pool exhibits the positive sample interactions our framework requires, then compare MtS against baselines, and finally test robustness across model capacity, attribution method, and data pool size.

Table 2: Held-out utility comparison on Qwen3-8B and Llama-3.2-3B-Instruct, both fine-tuned with LoRA r=16. Higher utility is better. Per-cell seed counts and paired t-tests are in Appendix B.

Selection	**Attribution**	**Qwen3-8B** N=50K m=1K	**Qwen3-8B** N=200K m=3.5K	**Llama-3.2-3B** N=50K m=1K
Panel A. *Full leaderboard: MtS vs. 11 baselines, each with its native attribution.*				
MtS	Oracle	**0.694**	**0.718**	**0.281**
Stratified	Oracle	0.666	0.694	0.190
Greedy (global top-m)	Oracle	0.603	0.632	−0.203
	LESS	0.548	0.580	−0.238
	EL2N	0.567	0.598	−0.244
	Superfiltering	0.534	0.640	−0.179
	DEITA	0.531	0.579	−0.109
k-DPP	DPP kernel	0.581	0.612	−0.101
Facility Location	Submodular surrogate	0.554	0.586	−0.016
Random	Uniform	0.549	0.567	0.076
CRAIG	Gradient matching	0.452	0.579	−0.204
Full data (no selection)	full-retraining	0.548	0.581	0.030
Panel B. *Same* 3 *selection rules under cheap attributions (LESS, EL2N); Oracle in Panel A.*				
MtS	LESS	0.591	0.625	**−0.120**
	EL2N	**0.597**	**0.677**	−0.122
Stratified	LESS	0.576	0.615	−0.132
	EL2N	0.567	0.639	−0.137
Greedy (global top-m)	LESS	0.548	0.580	−0.238
	EL2N	0.567	0.598	−0.244

Complementarity is real on the data pool. The structure our framework requires (positive sample interactions, $\beta > 0$) is strong and consistent on this data pool, confirmed by two independent measurement protocols: a large random pair sweep, and a separate pilot study that averages over training seeds. Other empirical constants used by Theorem 1 are computed in Appendix A.

MtS outperforms every baseline and the lead is robust. Across Panel A of Table 2, MtS beats every baseline by 20% to 54% relative; the eight sample-level methods bunch together near random, indifferent to which selection principle they implement. MtS's 1,000-sample subset also beats training on the full $50K$ data pool by Δ=+0.146, a small, well-chosen subset wins over training on everything. The rightmost column flags a striking failure mode: on Llama-3.2-3B, every sample-level method goes negative if Qwen's selections are reused without re-running calibration on Llama itself. We return to the diagnosis in Section 4.4.

Comparison with recent mixture methods. MtS reaches 0.694 held-out utility, compared with 0.6566 for RegMix and 0.6272 for DoReMi, relative gains of 5.7% and 10.7%. RegMix and DoReMi use a 1K subset with the same within-domain selection rule. The paired comparison with RegMix gives $p = 0.0117$. DIDS reaches 0.6343 by sampling online from the full pool with the same total training exposures.

Downstream benchmark evaluation. Table 3 compares performance on five benchmarks. MtS's best observed settings score highest on each, with the clearest gains on IFEval and ARC-Challenge. The MtS row combines settings across benchmarks; it does not represent one fixed selection or a matched multi-seed mean. These results assess downstream performance alongside held-out utility, whose ranking need not match accuracy on every task.

Gains grow with model capacity. MtS ties Stratified at the lowest tested LoRA rank, but its advantage grows monotonically with rank, reaching Δ=+0.046 at r=32 on Qwen3-8B (Figure 2B). This pattern suggests that higher capacity helps exploit cross-domain interactions; LESS remains

near random across ranks (Appendix B). **Mixture matters more than attribution.** The ranking Greedy $<$ Stratified $<$ MtS holds in every attribution column we tested, on both Qwen3-8B and Llama-3.2-3B. Replacing the expensive oracle attribution with cheap EL2N recovers 97% of MtS's oracle utility, at roughly four orders of magnitude less compute. The choice of attribution mainly trades cost for accuracy; the load-bearing decision is the domain mixture.

Gains transfer to larger data pools and other architectures. The advantage does not erode at $4\times$ the data pool size, and transfers to a different model family (Llama-3.1-8B-Instruct), with statistically significant gains over Stratified in both settings; on the smaller Llama-3.2-3B, the gap further widens as the budget grows (full numbers in Table 18 and Appendix B). The lead also persists when the eight skill domains are coarsened into four macro-domains ($\Delta{=}+0.022$ at $T{=}4$, same direction as $T{=}8$; Appendix B.10). Evaluation on a 72B model further supports MtS's applicability across model scales (Appendix B.11, Table 26).

Table 3: Downstream benchmarks on Qwen3-8B with the 50K pool. Metrics are instruction-level strict accuracy for IFEval, accuracy for MMLU and GSM8K, normalized accuracy for ARC-C, and pass@1 for HumanEval.

Method	IFEval	MMLU	GSM8K	ARC-C	Human Eval
MtS*	**0.505**	**0.725**	**0.806**	**0.586**	**0.805**
RegMix	0.482	0.714	0.796	0.556	0.770
DoReMi	0.480	0.715	0.802	0.532	0.762
LESS†	0.448	0.721	0.793	0.549	0.751

*MtS entries are best observed settings per task, not a single fixed configuration or a matched multi-seed mean. †LESS HumanEval uses the separate LESS-score pricing control.

4.3 Computational efficiency

MtS costs less than full-data training and LESS, including calibration. In Table 4, it uses 43% less total compute than full-data training and 77% less than LESS, with approximately 27% higher utility. No listed baseline uses less compute while achieving higher utility than MtS at $m{=}1000$. Stratified remains cheaper for a one-off selection, but at $m{=}5000$ reaches 0.678 utility versus MtS's 0.694. Increasing MtS's own budget to 5K does not improve utility. The same calibration can serve multiple selection settings on a fixed pool and base model. At fixed target accuracy, the required number of probes grows linearly with domain count and predictor rank, independently of pool size (Theorem 15). Cost breakdowns and reuse across settings and models are in Appendix B.2.1.

Table 4: Compute and utility on Qwen3-8B with $N{=}50K$ and LoRA $r{=}16$. Total GPU-hours include selection and training over five seeds; the breakdown is in Table 10.

Method	m	u	**GPU-hr**
Stratified	1k	0.666	0.58
Stratified	5k	0.678	2.92
EL2N	1k	0.567	1.43
MtS	**1k**	**0.694**	**16.58**
MtS	5k	0.691	18.92
Full data	**50k**	**0.548**	**29.17**
LESS	**1k**	**0.548**	**70.58**

4.4 Analysis of mechanism

We turn to two questions: what drives MtS's outperformance, and how does its edge vary across regimes? Practitioner takeaways close the section.

Sample-level scores miss interaction structure. Ranking samples globally can concentrate the selection in high-scoring domains while missing useful combinations across domains. Figure 2A shows this contrast: MtS concentrates on dialog/creative/QA, while Greedy-by-value concentrates on math/NLP, nearly the opposite mixture. Several baselines, including LESS and CRAIG, remain near random utility despite their different selection criteria (Table 2). The controlled comparisons using the same attribution scores also favor MtS, showing that the selection rule matters beyond the scores themselves. When samples interact, how the chosen samples combine matters more than how good each one looks alone.

Empirical support for pairwise modeling. Singleton and pairwise terms explain 90% of utility variation across approximately 100 measured triplets. Adding pairwise terms reduces mean absolute prediction error from 0.014 to 0.004, approximately a 71% reduction. Additional measurements and cross-task consistency checks are reported in Appendix B.4. **Mixture and attribution information both contribute.** The set-level variant reaches 0.6927 utility when its predictor uses both domain counts and sample-value statistics. Keeping only domain counts lowers utility by 0.088; keeping only value statistics lowers it by 0.063 (Appendix B.7). These are separate predictor-input ablations,

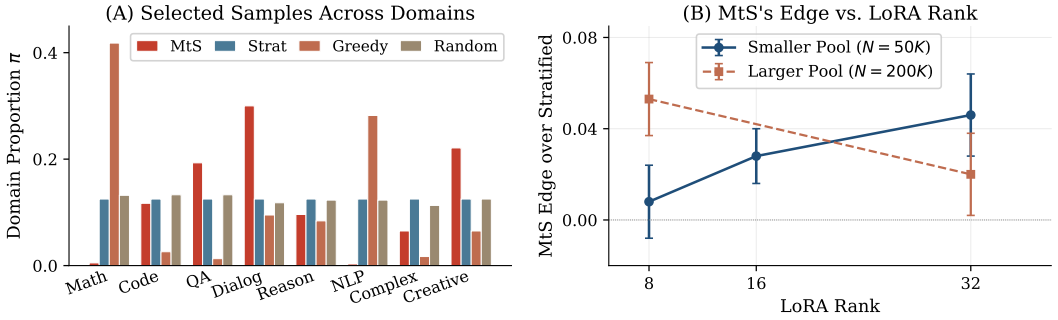

Figure 2: Domain allocation and capacity on Qwen3-8B. **(A)** Domain proportions selected by four selection rules at N=50K, m=1K. **(B)** MtS's utility difference from Stratified across LoRA ranks, with m=1K at N=50K and m=3.5K at N=200K.

not removals of selection stages or additive contributions. Together with the Greedy-Stratified-MtS ordering across attribution signals, these results show that both types of information matter. We also analyze settings where MtS does not improve over Stratified (Appendix B.8).

Practitioner takeaways. EL2N offers a cheaper alternative to oracle attribution. When transferring a learned mixture to another model, recompute within-domain rankings for that model. A possible screening rule is to fit a predictor on a small set of probes and use MtS when its predicted gain over uniform allocation exceeds the prediction uncertainty; otherwise use Stratified. The budget and capacity ablations motivate this rule, but we have not tested it separately.

Robustness to domain definitions. MtS retains gains when domains are merged or samples are reassigned. Merging eight domains into four and replacing source labels with utility-derived clusters retain utility advantages of 3.5% and 3.2% over Stratified, respectively, compared with 4.3% under the original partition. Corrupting 20% of labels reduces the four-task downstream composite by only 0.7% across three seeds. When 30% of samples have two possible domain labels, reassigning them using both embeddings and utility vectors increases Jaccard overlap with the clean selection from 0.403 to 0.806. In a separate end-to-end test at 30% ambiguity with embedding-based reassignment, the lambda-0.50 variant improves utility by 3.09% and the four-benchmark mean by 2.42% over Stratified; both gains occur in all three seeds. Settings, selection overlap and downstream results appear in Appendix B.9.

Data pricing. MtS also selects training data for purchase under a fixed budget. We test a vendor-bundle market (Zhang et al., 2025) with 15 indivisible 50-sample bundles across math, code and general QA, averaging over 10 price/utility shuffles. Budgets are 15%, 20% or 25% of total cost. MtS uses the same two-stage selection under a price budget. Additive LP instead allocates the budget using each cluster's utility alone, ignoring interactions. Additive LP loses utility as the budget grows, while MtS stays close to the oracle (Table 5): it closes at least 89% of the gap between Additive LP and the oracle at every budget and matches the oracle on 58% of instances. Per-instance results and ablations are in Appendix B.14.

Table 5: Data purchasing under a budget on Llama-3.2-3B-Instruct, averaged over 10 pricing instances. B is the fraction of total cost available for purchase. Recovery measures the fraction of the Additive-LP-to-oracle utility gap closed by MtS. Setup: Appendix B.14.

Method	B=15%	B=20%	B=25%
Additive LP	0.197	0.146	0.137
MtS	**0.229**	**0.242**	**0.249**
Oracle	0.236	0.246	0.250
Δ MtS−Additive LP	0.033	0.097	0.111
Optimality recovery	89.5%	95.6%	98.3%

5 Conclusion

Combinatorial Data Selection separates scoring individual samples from choosing a subset whose samples work well together. Mixture-then-Select first chooses domain counts, then ranks samples within each domain. Experiments show gains across models and pool sizes, including a setting where training data must be purchased under a budget. Future work includes aligning utility with specific downstream tasks and reducing calibration cost.

Reproducibility statement

The appendix provides the theoretical assumptions and proofs, experimental settings, and detailed results needed to reproduce the study.

AI use statement

We used generative AI to polish wording and help identify related work, mainly by checking for potentially overlooked papers. AI-assisted drafting included portions of the related-work section and one or two experimental subsections; the authors verified the sources and content and rewrote the text. After developing proofs, the authors used AI to help proofread them and identify possible errors. AI tools also assisted with experimental design, code, and analysis, which the authors reviewed. The authors take responsibility for the final manuscript and results.

References

Samyadeep Basu, Phil Pope, and Soheil Feizi. Influence functions in deep learning are fragile. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=xHKVVHGDOEk.

Andrew An Bian, Joachim M. Buhmann, Andreas Krause, and Sebastian Tschiatschek. Guarantees for greedy maximization of non-submodular functions with applications. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 498–507. JMLR.org, 2017.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Abhimanyu Das and David Kempe. Submodular meets spectral: greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 1057–1064, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, July 1998. ISSN 0004-5411. doi: 10.1145/285055.285059. URL https://doi.org/10.1145/285055.285059.

Amirata Ghorbani and James Y. Zou. Data shapley: Equitable valuation of data for machine learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2242–2251. PMLR, 2019. URL http://proceedings.mlr.press/v97/ghorbani19c.html.

Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of Game Theory*, 28(4):547–565, November 1999. ISSN 1432-1270. doi: 10.1007/s001820050125. URL http://dx.doi.org/10.1007/s001820050125.

Chris Harshaw, Moran Feldman, Justin Ward, and Amin Karbasi. Submodular maximization beyond non-negativity: Guarantees, fast algorithms, and applications. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2634–2643. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/harshaw19a.html.

Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001. ISSN 0004-5411. doi: 10.1145/502090.502098. URL https://doi.org/10.1145/502090.502098.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9525–9587. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ilyas22a.html.

Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pp. 85–103. Springer US, 1972. ISBN 9781468420012. doi: 10.1007/978-1-4684-2001-2_9. URL http://dx.doi.org/10.1007/978-1-4684-2001-2_9.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1885–1894. JMLR.org, 2017.

Suraj Kothawade, Vishal Kaushal, Ganesh Ramakrishnan, Jeff A. Bilmes, and Rishabh K. Iyer. PRISM: A rich class of parameterized submodular information measures for guided data subset selection. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 10238–10246. AAAI Press, 2022. doi: 10.1609/AAAI.V36I9.21264. URL https://doi.org/10.1609/aaai.v36i9.21264.

Alex Kulesza and Ben Taskar. k-dpps: fixed-size determinantal point processes. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 1193–1200, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in loRA-tuned LLMs and diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9m02ib92Wz.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14255–14273, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.769. URL https://aclanthology.org/2024.acl-long.769/.

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 Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7602–7635, Mexico City, Mexico, June 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.421. URL https://aclanthology.org/2024.naacl-long.421/.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. RegMix: Data mixture as regression for language model pre-training. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=5BjQOUXq7i.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=BTKAeLqLMw.

Adyasha Maharana, Prateek Yadav, and Mohit Bansal. $\mathbb{D}^2$ pruning: Message passing for balancing diversity & difficulty in data pruning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=thbtoAkCe9.

Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. Lazier than lazy greedy. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 1812–1818. AAAI Press, 2015. ISBN 0262511290.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, December 1978. ISSN 1436-4646. doi: 10.1007/bf01588971. URL http://dx.doi.org/10.1007/bf01588971.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=Uj7pF-D-YvT.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e6385d39ec9394f2f3a354d9d2b88eec-Paper.pdf.

Sai Srivatsa Ravindranath, Yanchen Jiang, and David C. Parkes. Data market design through deep learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

H S V N S Kowndinya Renduchintala, Sumit Bhatia, and Ganesh Ramakrishnan. SMART: Submodular data mixture strategy for instruction tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12916–12934, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.766. URL https://aclanthology.org/2024.findings-acl.766/.

Weijie Shi, Jipeng Zhang, Yaguang Wu, Jingzhi Fang, Shibo Zhang, Yao Zhao, Hao Chen, Ruiyuan Zhang, Yue Cui, Jia Zhu, Sirui Han, Jiajie Xu, and Xiaofang Zhou. DIDS: Domain impact-aware data sampling for large language model training. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4330–4350, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.215. URL https://aclanthology.org/2025.emnlp-main.215/.

Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.*, 32(1):41–43, January 2004. ISSN 0167-6377. doi: 10.1016/S0167-6377(03)00062-2. URL https://doi.org/10.1016/S0167-6377(03)00062-2.

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

Che-Ping Tsai, Chih-Kuan Yeh, and Pradeep Ravikumar. Faith-shap: the faithful shapley interaction index. *J. Mach. Learn. Res.*, 24(1), January 2023. ISSN 1532-4435.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, September 2018. ISBN 9781108415194. doi: 10.1017/9781108231596. URL http://dx.doi.org/10.1017/9781108231596.

Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM Journal on Computing*, 42(1):265–304, January 2013. ISSN 1095-7111. doi: 10.1137/110832318. URL http://dx.doi.org/10.1137/110832318.

Jiachen T. Wang and Ruoxi Jia. Data Banzhaf: A robust data valuation framework for machine learning. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 6388–6421. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/wang23e.html.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation, 2020. URL https://arxiv.org/abs/1811.10959.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1954–1963, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/wei15.html.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54104–54132. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/xia24c.html.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. Doremi: optimizing data mixtures speeds up language model pretraining. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Jinsung Yoon, Sercan Arik, and Tomas Pfister. Data valuation using reinforcement learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10842–10851. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/yoon20a.html.

Luyang Zhang, Cathy Jiao, Beibei Li, and Chenyan Xiong. Fairshare data pricing via data valuation for large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=xeFZUrJSH7.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=mSAKhLYLSsl.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=KBMOKmX2he.

A THEORY AND PROOFS

A.1 NOTATION TABLE

Symbol	Description
$\mathcal{D} = \{1, \dots, N\}$	Data catalog
p_j	Cost of sample j
B	Selection budget
$u(S)$	True utility of subset S
$v_j = u(\{j\})$	Individual value of sample j
φ_{ij}	Pairwise interaction (< 0: substitutes, > 0: complements)
$\hat{g}(S)$	Pairwise approximation: $\sum v_j + \sum \varphi_{ij}$
$\varepsilon(S)$	Higher-order remainder: $u(S) - \hat{g}(S)$
$\Delta_f(s \mid L)$	Marginal gain of s given L
γ	Submodularity ratio
β	Complementarity bound in $\mathrm{BC}(\beta)$
C, ρ_d	Decay constants for higher-order interactions
$v_{\max}$	Maximum individual value: $\max_j v_j$
$n_{\max}$	Maximum selection size
ρ_s	Spearman rank correlation
Spectral form (Section 3, Theorem 1)	
$\Phi_w, \Phi_{\text{block}}, \Phi_{\text{res}}$	Within-domain, block-constant cross, cross-residual parts of Φ
$\beta_{w,\text{op}}^{(c),\text{cent}}$	Per-domain centered within-block operator-norm coefficient
δ_{op}	Cross-residual operator-norm coefficient
$\nu_{\text{op}}^{\perp}$	Empirical-to-population sub-matrix concentration constant on $\hat{\Phi}_{\text{res}}$
$\zeta_{\text{pred}}, \zeta_{\Delta}^{\text{loc}}$	Predictor sample-complexity gap; predictor-argmax displacement on the local-margin ball
ρ_0, μ_{loc}	Local-margin radius and rate (Corollary 15)
L, r, T	Number of labeled subsets; predictor rank; domain count
$S_{\text{MtS}}, S_{\text{IA}}, S_{\text{HMtS}}$	Mixture-then-Select, IA-Greedy, hybrid output ($\arg\max\{u(S_{\text{MtS}}), u(S_{\text{IA}})\}$)

A.2 FORMAL ASSUMPTIONS AND OPERATOR-NORM REFINEMENT

This appendix collects the formal assumptions on the utility u and the pairwise interaction matrix Φ that drive the bounds in Section 3.3. Appendix A.2 also states Proposition 3, which decomposes complementarity into a model-induced and an aggregation-induced source, and discusses why the operator-norm refinement (Assumptions 4 and 5) sharpens the entrywise BC(β) form by a factor of m.

A.3 CORE ASSUMPTIONS

Three assumptions cover the basic shape of the utility: monotonicity of each task's improvement, bounded complementarity of pairwise interactions, and decay of higher-order interactions.

Assumption 1 (Monotonicity)**.** *Each u_t is monotone non-decreasing: $u_t(S \cup \{j\}) \geq u_t(S)$ for all S, j, t.*

Assumption 2 (Bounded Complementarity $\mathrm{BC}(\beta)$)**.** *For each $j \in \mathcal{D}$ and any $R \subseteq \mathcal{D} \setminus \{j\}$, $\sum_{i \in R} \varphi_{ij}^{+} \leq \beta \cdot v_j$, where $\varphi_{ij}^{+} = \max(\varphi_{ij}, 0)$ denotes the complementary (positive) part of the aggregate interaction.*

Assumption 3 (Exponential Interaction Decay)**.** *Higher-order Möbius coefficients satisfy $|a_R| \leq C \cdot \rho_d^{|R|} \cdot v_{\max}$ for $|R| \geq 3$, where $C, \rho_d > 0$ are constants and $v_{\max} = \max_j v_j$.*

Monotonicity (Assumption 1) is the standard hypothesis in submodular data selection (Wei et al., 2015; Mirzasoleiman et al., 2020; Nemhauser et al., 1978): adding training data never hurts. Since f is non-decreasing in each argument, u inherits monotonicity from the u_t's for every aggregator considered in this paper.

BC(β) (Assumption 2) controls positive pairwise interactions relative to singleton values. For the pairwise approximation, $\beta = 0$ gives the substitution-only case; a positive submodularity-ratio guarantee additionally requires the conditional-marginal hypothesis in Lemma 9.

Assumption 3 controls the higher-order Möbius tail. Möbius inversion (Grabisch & Roubens, 1999) writes any set function as singleton, pairwise, triplet, ... contributions, and truncating after pairs is justified only when the dropped terms are small; the geometric decay condition is the quantitative form of that requirement. On our data pool the third-order remainder is one to two orders of magnitude below the pairwise term (Section 4).

A.4 Two sources of complementarity

Whether $\beta > 0$ comes from the per-task utilities being non-submodular or from the aggregator f matters for interpreting empirical β calibrations. The next proposition separates the two sources.

Proposition 3 (Two Sources of Complementarity). *Let each u_t be submodular.*

(a) ***Linear aggregation preserves submodularity.*** *If $f = \sum_t w_t \cdot u_t$ (weighted average) and each task-specific utility u_t is submodular, then u is submodular and $\beta = 0$.*

(b) ***Concave aggregation can induce complementarity.*** *If $f = \min_t u_t$ or f is strictly concave, then $\beta > 0$ whenever samples from different domains cover different tasks, even when all per-task interactions are substitutive ($\varphi_{ij}^{(t)} \leq 0$).*

Proposition 3 separates two sources of complementarity in multi-task data selection:

- **Source 1: Model-induced.** The per-task utilities u_t are themselves non-submodular, training on i and j together produces more improvement *on task t alone* than their individual contributions sum to. This arises when the model develops cross-task representations (e.g., a code sample improves math reasoning through shared compositional skills). Under linear f, the contrapositive of part (a) makes this the only source of $\beta > 0$.
- **Source 2: Aggregation-induced.** Even when every u_t is submodular ($\varphi_{ij}^{(t)} \leq 0$ for all i, j, t), a concave f creates aggregate complementarity by rewarding balance across tasks (part (b)).

Our experiments use linear aggregation $f = \frac{1}{T}\sum_t u_t$, which shuts off Source 2; any observed $\beta > 0$ is therefore evidence that the per-task utilities themselves are non-submodular, a property of the model's learning dynamics that any downstream objective inherits. The framework supports concave aggregators (Appendix A.20); the linear choice here isolates Source 1.

A.5 Operator-norm refinement

BC(β) bounds the interactions one entry at a time, which forces an $O(m^2)$ gap and is loose at production budgets. MtS actually controls interactions collectively: matching the domain composition cancels the cross-domain block mean exactly, so what remains is the residual matrix Φ_{res} acting on the indicator $\mathbf{1}_S$. A single Cauchy-Schwarz step bounds that action by the operator norm of Φ_{res} times $\sqrt{|S|}$, which delivers an $O(m)$ gap on any matrix whose mass is not concentrated on a single direction aligned with $\mathbf{1}_S$. The next two assumptions name the operator norms that quantify this control.

Assumption 4 (Bounded Within-Domain Operator Norm; B1"-cent). *For each domain c, the centered within-domain block $\Phi_{w,c}^{cent} := \Phi_{w,c} - \bar{\varphi}_{c,c}(\mathbf{1}\mathbf{1}^\top - I)$ satisfies $\|\Phi_{w,c}^{cent}\|_{op} \leq \beta_{w,op}^{(c),cent}\, v_{\max}$ at the population level. When the domain index is suppressed, $\beta_{w,op}^{cent} := \max_c \beta_{w,op}^{(c),cent}$; at $T{=}1$ the two coincide.*

Assumption 5 (Bounded Cross-Residual Operator Norm; B2"). *The population cross-residual satisfies $\|\Phi_{res}\|_{op} \leq \delta_{op}\, v_{\max}$. The empirical-to-population gap on the realized subset is controlled by a sub-matrix concentration bound (Appendix A.22, Lemma M').*

The operator norm bounds a matrix's largest action on any vector relative to that vector's ℓ_2-norm; applied to the indicator $\mathbf{1}_S$ (which has norm $\sqrt{|S|}$) this gives $|\mathbf{1}_S^\top \Phi\, \mathbf{1}_S| \leq \|\Phi\|_{\text{op}}\, |S|$, exactly the worst-case spectral loss the algorithm can incur on a subset of size $|S|$. The constants $\beta_{w,\text{op}}^{(c),\text{cent}}$ and δ_{op} are this worst-case action, normalized by $v_{\max}$, on $\Phi_{w,c}^{\text{cent}}$ and Φ_{res} respectively. Both are intrinsic

to the data pool and partition, are estimated from $\hat{\Phi}$, and the empirical-to-population gap is charged in Theorem 2. The within-domain norm is taken on the centered block $\Phi^{\text{cent}}_{w,c}$ because Stage 1 of MtS matches $\bar{\varphi}_{c,c}$ exactly through the composition; the leftover deviation is what the bound has to pay for.

Remark 1. *The adaptive $T = 1$ extension uses conditional-marginal control (Lemma 9); Assumptions 4 to 6 drive the fixed-ranking multi-domain bound (Theorem 1).*

A.6 Local-margin condition and finite-sample noise

Definition 4 (Local-margin condition)**.** *There exist a* local radius $\rho_0 > 0$ *and a* local rate $\mu_{loc} > 0$ *such that for every mixture* $\mathbf{n}$ *within* ℓ_1*-radius* ρ_0 *of the optimum mixture* $\mathbf{n}^*$*,*

$$u(\mathbf{n}^*) - u(\mathbf{n}) \;\geq\; \mu_{loc}\, v_{\max}\, \|\mathbf{n} - \mathbf{n}^*\|_1 \,/\, m.$$

Here $u(\mathbf{n})$ denotes the constrained optimum $U(\mathbf{n})$ defined below. A local margin controls displacement within the specified ball; the main guarantee uses the following global condition to identify the maximizing mixture.

Assumption 6 (Global mixture identification)**.** *Let* $\mathcal{N}_m = \{\mathbf{n} \in \mathbb{Z}^T_{\geq 0} : \sum_c n_c = m,\ n_c \leq |C_c|\}$ *and define*

$$U(\mathbf{n}) = \max_{S:\, \mathbf{n}(S)=\mathbf{n}} u(S).$$

The optimum mixture $\mathbf{n}^*$ *is unique, with gap*

$$\Delta_{\text{mix}} = U(\mathbf{n}^*) - \max_{\mathbf{n} \in \mathcal{N}_m \setminus \{\mathbf{n}^*\}} U(\mathbf{n}) > 0.$$

With probability at least $1 - \eta$*, the Stage 1 predictor satisfies*

$$\sup_{\mathbf{n} \in \mathcal{N}_m} |\hat{g}(\mathbf{n}) - U(\mathbf{n})| \leq \epsilon_{\text{mix}} < \Delta_{\text{mix}}/2.$$

For a singleton feasible mixture set, take $\Delta_{\text{mix}} = \infty$*.*

This is a global accuracy and separation assumption for the constrained optimum U, not a consequence of fitting the observed probes. It ensures exact mixture identification without changing Stage 1's search. A local margin alone controls displacement only after localization has been established.

Definition 5 (Finite-sample noise term $\mathcal{R}(m)$)**.** *The nonnegative estimation allowance in Theorem 1 is written as*

$$\mathcal{R}(k) \;=\; \nu^{\perp}_{op}\, k\, v_{\max} \;+\; \zeta^{loc}_{\Delta}(\rho_0, \mu_{loc}) \;+\; 2\, \zeta_{pred}\, v_{\max},$$

where $\nu^{\perp}_{\text{op}}$ *bounds residual estimation error,* $\zeta^{\text{loc}}_{\Delta}$ *denotes an allowance for mixture displacement, and* ζ_{pred} *bounds normalized predictor error. Under Assumption 6, the mixture is recovered exactly and* $\zeta^{\text{loc}}_{\Delta} = 0$*. The structural constants* $\beta^{(c),\text{cent}}_{w,\text{op}}$ *and* δ_{op} *concern population interactions;* $\epsilon_{\text{op}} = \|\hat{\Phi} - \Phi\|_{\text{op}}/v_{\max}$ *is a separate estimation-error budget used in Theorem 2.*

A.7 NP-hardness

Theorem 6 (NP-Hardness of CDS)**.** *(a) CDS is NP-hard even when* $\varphi_{ij} = 0$ *for all* i, j*. (b) Unconstrained CDS with* $\varphi_{ij} \leq 0$ *for all* i, j *is NP-hard.*

Part (a). When $\varphi_{ij} = 0$ for all i, j, CDS reduces to: maximize $\sum_{j \in S} v_j$ subject to $\sum_{j \in S} p_j \leq B$, which is the 0-1 Knapsack problem (Karp, 1972). Since Knapsack is NP-hard, so is CDS. □

Part (b). Given a weighted graph $G = (V, E, w)$ with MAX-CUT instance, define: $v_j = \deg_w(j) = \sum_i w_{ij}$ for each vertex j, and $\varphi_{ij} = -2w_{ij}$ for each edge. Note $\varphi_{ij} \leq 0$ (substitution only). Then:

$$\begin{aligned}\hat{g}(S) &= \sum_{j\in S} \deg_w(j) - 2 \sum_{\{i,j\}\subseteq S} w_{ij} \\ &= \sum_{j\in S}\sum_{i\in V} w_{ij} - 2 \sum_{\{i,j\}\subseteq S} w_{ij} \\ &= \sum_{j\in S}\Big(\sum_{i\in V\setminus S} w_{ij} + \sum_{i\in S} w_{ij}\Big) - 2 \sum_{\{i,j\}\subseteq S} w_{ij} \\ &= \sum_{j\in S}\sum_{i\in V\setminus S} w_{ij} = \mathrm{cut}(S, V\setminus S).\end{aligned}$$

Since MAX-CUT is NP-hard (Karp, 1972), unconstrained CDS with substitution-only interactions is NP-hard. □

A.8 Submodularity under substitution

Lemma 7. *If $\varphi_{ij} \leq 0$ for all i, j (substitution only), then $\hat{g}$ is submodular.*

When $\varphi_{ij} \leq 0$ for all i, j, we show $\hat{g}$ has non-increasing marginal gains. For any $A \subseteq B$ and $s \notin B$:

$$\begin{aligned}\Delta_{\hat{g}}(s \mid A) &= v_s + \sum_{i\in A} \varphi_{is}, \\ \Delta_{\hat{g}}(s \mid B) &= v_s + \sum_{i\in B} \varphi_{is} = v_s + \sum_{i\in A} \varphi_{is} + \sum_{i\in B\setminus A} \varphi_{is}.\end{aligned}$$

Since $\varphi_{is} \leq 0$ for all i, the additional terms $\sum_{i\in B\setminus A} \varphi_{is} \leq 0$. Thus $\Delta_{\hat{g}}(s \mid B) \leq \Delta_{\hat{g}}(s \mid A)$, confirming submodularity. □

A.9 Higher-order remainder bound

Theorem 8 (Higher-Order Remainder Bound)**.** *Under Assumption 3, for any S with $|S| = n$ and $n\rho_d < 1$: $|\varepsilon(S)| \leq C \cdot v_{\max} \cdot (n\rho_d)^3/[6(1 - n\rho_d)]$.*

Under Assumption 3, for $|T| = m \geq 3$: $|a_T| \leq C\rho_d^m v_{\max}$. For $|S| = n$:

$$\begin{aligned}|\varepsilon(S)| &= \Big|\sum_{\substack{T\subseteq S\\ |T|\geq 3}} a_T\Big| \leq \sum_{k=3}^{n} \binom{n}{k} C\rho_d^k v_{\max} \\ &= C v_{\max} \sum_{k=3}^{n} \binom{n}{k} \rho_d^k \leq C v_{\max} \sum_{k=3}^{\infty} \frac{(n\rho_d)^k}{k!} \\ &\leq C v_{\max} \cdot \frac{(n\rho_d)^3}{6} \sum_{k=0}^{\infty} (n\rho_d)^k = \frac{C v_{\max} (n\rho_d)^3}{6(1 - n\rho_d)},\end{aligned}$$

where the last step uses the geometric series bound (valid for $n\rho_d < 1$) and $\binom{n}{m} \leq n^m/m!$. □

A.10 Submodularity ratio under conditional bounded complementarity

The adaptive guarantee requires control of conditional gains, not only the singleton-based condition in Assumption 2. Write $\Delta_{\tilde{u}}(s \mid L) = \tilde{u}(L \cup \{s\}) - \tilde{u}(L)$.

Lemma 9 (Submodularity ratio under conditional bounded complementarity)**.** *Suppose the pairwise utility $\tilde{u}$ is monotone on sets of size at most $2m$. For every disjoint L, S with $|L| < m$, $|S| \leq m$, and each $s \in S$, assume*

$$\sum_{s'\in S\setminus\{s\}} \varphi^+_{ss'} \leq \beta\, \Delta_{\tilde{u}}(s \mid L).$$

Then the submodularity ratio over these pairs satisfies $\gamma \geq 2/(2 + \beta)$.

Proof. Set $\mathrm{NUM} = \sum_{s\in S} \Delta_{\tilde{u}}(s \mid L)$ and $\Phi_S = \sum_{\{s,s'\}\subseteq S} \varphi_{ss'}$. The pairwise expansion gives

$$\mathrm{DEN} = \tilde{u}(L \cup S) - \tilde{u}(L) = \mathrm{NUM} + \Phi_S.$$

Summing the assumed conditional bound over s counts each positive pair twice, so $\Phi_S \le (\beta/2)\mathrm{NUM}$. Hence $\mathrm{DEN} \le (1+\beta/2)\mathrm{NUM}$. For positive joint gain this gives $\mathrm{NUM}/\mathrm{DEN} \ge 2/(2+\beta)$; zero joint gain imposes no restriction. □

The additional condition reduces to a singleton-based bound when $L = \emptyset$, but must hold after conditioning as well. It is used only for the adaptive extension, not for the fixed-ranking guarantee in Theorem 1.

A.11 Proof of Theorem 1 (Mixture-then-Select spectral guarantee)

We prove the bound for Algorithm 1's fixed within-domain ranking. Condition on the uniform-accuracy event in Assumption 6, which has probability at least $1-\eta$.

Mixture identification. For every feasible $\mathbf{n} \neq \mathbf{n}^*$,

$$\hat{g}(\mathbf{n}^*) - \hat{g}(\mathbf{n}) \ge U(\mathbf{n}^*) - U(\mathbf{n}) - 2\epsilon_{\mathrm{mix}} \ge \Delta_{\mathrm{mix}} - 2\epsilon_{\mathrm{mix}} > 0.$$

Thus Stage 1 selects $\hat{\mathbf{n}} = \mathbf{n}^*$. In particular, S_{MtS} and S^* have the same counts in every domain, and the composition-displacement term is zero on this event.

Fixed-ranking and interaction gaps. Let $A_c = S^* \cap C_c$ and $B_c = S_{\mathrm{MtS}} \cap C_c$. Since B_c contains the top n_c^* singleton values,

$$\sum_{j\in A_c} v_j - \sum_{j\in B_c} v_j \le 0.$$

The within-domain mean contribution is $\bar{\varphi}_{c,c}\binom{n_c^*}{2}$ for both subsets and cancels. For the centered residual $Q_c = \Phi^{\mathrm{cent}}_{w,c}$,

$$\frac{1}{2}\big(\mathbf{1}_{A_c}^\top Q_c \mathbf{1}_{A_c} - \mathbf{1}_{B_c}^\top Q_c \mathbf{1}_{B_c}\big) \le \frac{1}{2}\|Q_c\|_{\mathrm{op}}(|A_c| + |B_c|) \le \beta^{(c),\mathrm{cent}}_{w,\mathrm{op}} n_c^* v_{\max}.$$

Cross-domain block means also cancel by Lemma 10. Applying the same quadratic-form bound to Φ_{res} gives

$$\frac{1}{2}\big(\mathbf{1}_{S^*}^\top \Phi_{\mathrm{res}} \mathbf{1}_{S^*} - \mathbf{1}_{S_{\mathrm{MtS}}}^\top \Phi_{\mathrm{res}} \mathbf{1}_{S_{\mathrm{MtS}}}\big) \le \delta_{\mathrm{op}} m v_{\max}.$$

Higher-order remainder and estimation allowance. Let $E(m)$ uniformly bound $|\varepsilon(S)|$ for $|S| = m$, using Theorem 8 when $m\rho_d < 1$. Comparing the two remainders adds at most $2E(m)$. Therefore

$$u(S^*) - u(S_{\mathrm{MtS}}) \le \sum_c \beta^{(c),\mathrm{cent}}_{w,\mathrm{op}} n_c^* v_{\max} + \delta_{\mathrm{op}} m v_{\max} + 2E(m).$$

Adding the nonnegative estimation allowance $\mathcal{R}(m)$ preserves the stated bound. The uniform predictor event is an explicit hypothesis; it is not inferred from a probe-count threshold. □

Consequences of the bound. From the bound's *shape* alone follow three practitioner-facing properties: (i) N*-independence*, the bound does not grow with data pool size, so quality does not degrade as the data pool grows; (ii) *heterogeneous workloads tighten the bound*, the within-domain gap depends on the optimum's per-domain count n_c^*, not the full domain size $|C_c|$; (iii) *constant gap-per-sample*, the bound is linear in m, with no super-linear blowup as the budget grows.

Lemma 10 (Block-constant cancellation)**.** *Let $A \in \mathbb{R}^{N\times N}$ be* block-constant on cross-domain entries*: $A_{ij} = \bar{A}_{c(i),c(j)}$ for $c(i) \neq c(j)$ and $A_{ij} = 0$ for $c(i) = c(j)$. For any two subsets $S_1, S_2 \subseteq \mathcal{D}$ with the same domain composition $\mathbf{n}(S_1) = \mathbf{n}(S_2)$,*

$$\mathbf{1}_{S_1}^\top A\, \mathbf{1}_{S_1} \;=\; \mathbf{1}_{S_2}^\top A\, \mathbf{1}_{S_2}.$$

Proof. $\mathbf{1}_S^\top A\, \mathbf{1}_S = \sum_{c\neq c'} \bar{A}_{c,c'}\, n_c(S)\, n_{c'}(S)$, which depends on S only through $\mathbf{n}(S)$. □

A.12 ADAPTIVE INTERACTION-AWARE EXTENSION AT $T = 1$

Adaptive interaction-aware extension. At $T = 1$, the domain decomposition reduces to $\Phi = \Phi_w$, but the choice of within-domain selection rule remains distinct. Algorithm 1 selects the top singleton scores using a fixed ranking. An adaptive extension instead selects

$$j \in \arg\max_{j \notin S} \left(v_j + \sum_{i \in S} \varphi_{ij} \right)$$

and updates these marginal gains after each addition. This extension is IA-Greedy; it is not the single-domain specialization of Algorithm 1. The adaptive analysis below is separate from the fixed-ranking MtS guarantee and requires its own valid conditional-marginal bounds.

For example, with budget two, singleton values $(1, 1, 0.99)$, and the sole nonzero interaction $\varphi_{12} = -1$, fixed ranking selects $\{1, 2\}$ with utility 1, whereas the adaptive extension selects $\{1, 3\}$ or $\{2, 3\}$ with utility 1.99. This illustrates the distinction even for a monotone submodular utility.

Theorem 11 (Adaptive IA-Greedy at $T = 1$)**.** *At $T = 1$, suppose $\tilde{u}$ is normalized and monotone on sets of size at most $2m$, and has submodularity ratio at least $\gamma_{\mathrm{op}} \in [0, 1]$ for disjoint L, S with $|L| < m$, $|S| \le m$. Then adaptive IA-Greedy under a cardinality budget m satisfies*

$$\tilde{u}(S_{\mathrm{IA}}) \ge (1 - e^{-\gamma_{\mathrm{op}}}) \max_{|S|=m} \tilde{u}(S).$$

Under the conditional-marginal hypothesis of Lemma 9, one may take $\gamma_{\mathrm{op}} = 2/(2 + \beta)$. Here γ_{op} denotes a certified submodularity-ratio bound; the centered operator norm alone does not provide one. If $|u(S) - \tilde{u}(S)| \le E(m)$ for all size-m subsets, then

$$u(S_{\mathrm{IA}}) \ge (1 - e^{-\gamma_{\mathrm{op}}}) u(S^*) - (2 - e^{-\gamma_{\mathrm{op}}}) E(m).$$

Proof. Let O maximize $\tilde{u}$ at size m and S_t be the greedy set after t additions. Monotonicity and the submodularity-ratio condition imply

$$\sum_{j \in O \setminus S_t} \Delta_{\tilde{u}}(j \mid S_t) \ge \gamma_{\mathrm{op}} \big(\tilde{u}(O) - \tilde{u}(S_t) \big).$$

There are at most m candidates in this sum, so the largest marginal reduces the remaining gap by a factor at most $1 - \gamma_{\mathrm{op}}/m$. Iterating for m steps gives the standard cardinality guarantee (Bian et al., 2017; Das & Kempe, 2011). Comparing $\tilde{u}$ and u on the greedy and optimal subsets gives the stated remainder term. □

This result concerns adaptive marginal-gain updates, not the fixed-ranking rule in Algorithm 1.

A.13 DOMAIN-UNIFORM COST EXTENSION

Section 3.3's domain-cost paragraph states that MtS extends to data-marketplace settings where every sample in C_c shares a price $p_c > 0$, with Stage 1's cardinality constraint replaced by the price-weighted budget $\sum_c p_c\, n_c \le B$ and Stage 2 unchanged. We give the formal statement and proof here.

Theorem 12 (Domain-Uniform Cost Extension)**.** *Fix $p_c > 0$ for each domain c and budget $B > 0$. Let $\Pi := \{\mathbf{n} \in \mathbb{Z}^T_{\ge 0} : \sum_c p_c n_c \le B,\ n_c \le |C_c|\}$, $S^* \in \arg\max_S \{u(S) : \mathbf{n}(S) \in \Pi\}$, $n^*_c := |S^* \cap C_c|$, and the realized cardinality $m^* := |S^*| = \sum_c n^*_c$. Under Assumptions 1 and 3 to 5 and global mixture identification (Assumption 6) restated on Π, with probability $\ge 1 - \eta$,*

$$u(S_{MtS}) \;\ge\; u(S^*) \;-\; \sum_{c=1}^{T} \beta^{(c),cent}_{w,op}\, n^*_c\, v_{\max} \;-\; \delta_{op}\, m^*\, v_{\max} \;-\; \mathcal{R}(m^*) \;-\; 2\, E(m^*), \qquad (5)$$

where $\mathcal{R}(\cdot)$ and $E(\cdot)$ are exactly as in Theorem 1. The bound has identical form to Equation (4); only the cardinality m is replaced by m^*, which satisfies $\lfloor B/\max_c p_c \rfloor \le m^* \le \lfloor B/\min_c p_c \rfloor$. At $p_c \equiv 1$, $B = m$, $m^* = m$ and Equation (5) reduces to Equation (4).

The constants $\beta_{w,\text{op}}^{(c),\text{cent}}$ and δ_{op} depend on the data pool and partition but not on the budget form, so one calibration of these constants transfers to any domain-uniform pricing. Tightness inherits from the $p_c \equiv 1$ specialization of Theorem 13.

Proof. The fixed-ranking proof of Theorem 1 (Appendix A.11) applies with the feasible mixture set Π and matched cardinality m^*.

Step W (within-domain gap). For each domain c, the bilinear-path identity gives $\Phi_w(S \cap C_c) = \frac{1}{2}\mathbf{1}_{S\cap C_c}^\top \Phi_{w,c}\, \mathbf{1}_{S\cap C_c}$ minus a diagonal correction that cancels in the difference $\Delta\Phi_w$. After centering by $\bar{\varphi}_{c,c}$, Cauchy-Schwarz on $\Phi_{w,c}^{\text{cent}}$ yields $|\mathbf{1}_T^\top \Phi_{w,c}^{\text{cent}}\, \mathbf{1}_T| \le \beta_{w,\text{op}}^{(c),\text{cent}}\, |T|\, v_{\max}$. Applied with $|T| = n_c^*$ on each side and summed over domains, the within-domain gap is $\sum_c \beta_{w,\text{op}}^{(c),\text{cent}}\, n_c^*\, v_{\max}$. The cost constraint enters only Stage 1's feasibility, not the spectral upper bound.

Step C (cross-domain gap). Stage 2 enforces $\mathbf{n}(S_{\text{MtS}}) = \hat{\mathbf{n}}^*$, which under Step M equals $\mathbf{n}^*$. The block-constant matrix Φ_{block} depends on S only through domain counts: $\mathbf{1}_S^\top \Phi_{\text{block}}\, \mathbf{1}_S = \sum_{c\neq c'} \bar{\varphi}_{c,c'}\, n_c\, n_{c'}$. Hence $\Phi_{\text{block}}(S^*) = \Phi_{\text{block}}(S_{\text{MtS}})$ and the block contribution to the gap vanishes. The remaining cross-domain term is bounded by Cauchy-Schwarz on Φ_{res}: $|\mathbf{1}_S^\top \Phi_{\text{res}}\, \mathbf{1}_S| \le \|\Phi_{\text{res}}\|_{\text{op}}\, |S| = \delta_{\text{op}}\, m^*\, v_{\max}$ on each side, since $|S^*| = |S_{\text{MtS}}| = m^*$ at the matched composition.

Step M (composition matching). Stage 1 maximizes $\hat{g}$ on Π. Apply Assumption 6 to $U(\mathbf{n}) = \max_{S:\mathbf{n}(S)=\mathbf{n}} u(S)$ on Π. The uniform error is less than half the gap to every other feasible mixture, so the argmax comparison in Appendix A.11 gives $\hat{\mathbf{n}}^* = \mathbf{n}^*$. This event, rather than a fixed sample count, establishes composition matching.

Remainder and estimation allowance. The two compared subsets have cardinality m^*, so their higher-order remainders contribute at most $2E(m^*)$. The nonnegative allowance $\mathcal{R}(m^*)$ can be added as in Theorem 1. The probability is that of the global identification event on Π.

Combining the within-domain, cross-domain, and remainder bounds yields Eq. (5). At $p_c \equiv 1$, $B = m$, the polytope Π becomes the cardinality simplex $\{\sum_c n_c \le m\}$. By monotonicity of u, the optimum saturates $|S^*| = m$ when the data pool admits m samples, so $m^* = m$, and Eq. (5) reduces to Eq. (4). □

Bounds on m^*. By monotonicity, S^* saturates the budget up to one sample: $c(S^*) > B - \max_c p_c$. From $m^* \cdot \max_c p_c \ge c(S^*)$ we get $m^* > B/\max_c p_c - 1$, hence $m^* \ge \lfloor B/\max_c p_c \rfloor$. From $m^* \cdot \min_c p_c \le c(S^*) \le B$ we get $m^* \le \lfloor B/\min_c p_c \rfloor$. Under uniform pricing $p_c \equiv p$, both bounds collapse to $m^* = \lfloor B/p \rfloor$.

A.14 Tightness constructions

Theorem 13 (Matching Lower Bounds)**.** *There exist instance families on which any two-stage algorithm loses utility at least $\frac{m}{2}\beta_{w,op}^{cent}\, v_{\max}$ (within-domain axis at T=1, where the per-domain index collapses) and at least $\frac{m}{2}\delta_{op}\, v_{\max}$ (cross-domain axis, $T\ge 2$, Φ_w=0); on a combined construction both axes activate simultaneously.*

Proof. We exhibit per-axis tightness lower bounds matching the spectral upper of Theorem 1 up to factor 2 on each axis. Throughout, $\rho_w := \beta_{w,\text{op}}^{\text{cent}}\, v_{\max}/m$ and $\rho_r := \delta_{\text{op}}\, v_{\max}/m$.

(T-W2) Within-axis, rank-2 antidiagonal, value-aware. Let $T = 1$, $|C_1| = 2m$, partition into $U^\pm, W^\pm$ of size $m/2$. Set $\tilde{u} := \mathbf{1}_{U^+} - \mathbf{1}_{U^-}$, $\tilde{w} := \mathbf{1}_{W^+} - \mathbf{1}_{W^-}$ (orthogonal, disjoint supports), and $\Phi_w := \rho_w(\tilde{u}\tilde{w}^\top + \tilde{w}\tilde{u}^\top)$. Diagonal vanishes automatically. Place $v_j = v_{\max}$ on $U^+ \cup W^-$, $v_j = v_{\max} - \varepsilon$ on $U^- \cup W^+$. MtS returns $S_{\text{MtS}} = U^+ \cup W^-$ (with $\mathbf{1}_{S_{\text{MtS}}}^\top \tilde{u} = +m/2$, $\mathbf{1}_{S_{\text{MtS}}}^\top \tilde{w} = -m/2$), while $S^* = U^+ \cup W^+$ achieves $+m/2, +m/2$. Letting $\varepsilon \to 0^+$,

$$u(S^*) - u(S_{\text{MtS}}) \;\to\; \tfrac{m}{2}\, \beta_{w,\text{op}}^{\text{cent}}\, v_{\max},$$

giving $C_w \ge 1/2$.

(T-2x / T_C) Cross-axis, rank-2 antidiagonal. Let $T = 2$, $|C_1| = |C_2| = m$, $C_c = H_c \sqcup L_c$ at half-sizes. Set $\tilde{u}_c := \mathbf{1}_{H_c} - \mathbf{1}_{L_c}$ and $\Phi_{\text{res}} := \rho_r(\tilde{u}_1\tilde{u}_2^\top + \tilde{u}_2\tilde{u}_1^\top)$, $\Phi_w \equiv 0$. Then $\|\Phi_{\text{res}}\|_{\text{op}} = \delta_{\text{op}}\, v_{\max}$. With $v_j = v_{\max}$ on $H_1 \cup L_2$ and $v_{\max} - \varepsilon$ elsewhere, MtS returns $H_1 \cup L_2$ (sign-misaligned across domains), while $S^* = H_1 \cup H_2$ (sign-aligned). Then

$$u(S^*) - u(S_{\text{MtS}}) \;\to\; \tfrac{m}{2}\, \delta_{\text{op}}\, v_{\max},$$

giving $C_r \geq 1/2$.

Joint floor (Form B; independent-axes). Combining T-W2 (within-only, $\delta_{\text{op}} = 0$) and T-2x (cross-only, $\beta^{\text{cent}}_{w,\text{op}} = 0$) as separate instances yields

$$u(S^*) - u(S_{\text{MtS}}) \;\geq\; \tfrac{m}{2}\,(\beta^{\text{cent}}_{w,\text{op}} + \delta_{\text{op}})\, v_{\max}.$$

The joint floor matches the spectral upper of Theorem 1 up to a structurally exact factor 2; single-instance closure of the factor-2 gap (Form A) is bounded above by $C_w + C_r \leq 3/8$ and is conjecturally fundamental. □

A.15 Antidiagonal-block empirical SVD

T-2x is not a pathological worst case. The empirical SVD of the centered cross-domain residual $\widehat{\Phi}_{\text{res}}$ on a 7798×7798 pair sub-matrix exhibits *16 degenerate $\pm\sigma$ pairs in the top-32 singular values*, with $\sigma_i/\sigma_1 \in [0.77, 1.0]$. This is the spectral signature of an antidiagonal block-bipartite operator

$$\sum_{c<c'} \rho_{cc'}(\hat{u}_{cc'}\hat{v}_{cc'}^\top + \hat{v}_{cc'}\hat{u}_{cc'}^\top),$$

structurally identical to T-2x's rank-2 construction across domain pairs.

The bulk spectrum is dense: the empirical stable rank $\widehat{\sigma}_{\text{stab}}(\widehat{\Phi}_{\text{res}}) := \|\widehat{\Phi}_{\text{res}}\|_F^2/\|\widehat{\Phi}_{\text{res}}\|_{\text{op}}^2 \in [2400, 3000]$, ruling out any low-rank ansatz at any cutoff $r \leq \sqrt{N} \approx 224$ on a 5000-pair sample. The dense-noise regime is the empirical reality, and T-2x's tightness lower bound therefore reflects the empirical structure of real-data cross-domain interactions rather than a pathological construction.

A.16 Predictor accuracy and mixture identification

The relevant error is uniform over feasible mixtures and is measured against the constrained optimum U in Assumption 6. A low-rank predictor or small error on observed probes alone does not establish this uniform bound. The following conditional statement makes the required error certificate explicit.

Theorem 14 (Conditional predictor-sample bound)**.** *Suppose a rank-r predictor fitted using L probes satisfies, with probability at least $1-\delta$,*

$$\sup_{\mathbf{n}\in\mathcal{N}_m} |\hat{g}(\mathbf{n}) - U(\mathbf{n})| \leq v_{\max}\epsilon_L, \qquad \epsilon_L = \frac{A\sqrt{rT} + B\sqrt{\log(2/\delta)}}{\sqrt{L}},$$

for certified nonnegative constants A, B independent of L. If U has a unique maximizing mixture with gap $\Delta_{\text{mix}} > 0$, exact mixture identification follows whenever

$$L > \frac{4v_{\max}^2}{\Delta_{\text{mix}}^2}\big(A\sqrt{rT} + B\sqrt{\log(2/\delta)}\big)^2.$$

Proof. The displayed threshold ensures $v_{\max}\epsilon_L < \Delta_{\text{mix}}/2$. Apply the argmax comparison in the proof of Theorem 1. □

Corollary 15 (Predictor accuracy and sample budget)**.** *Under the uniform-error hypothesis of Theorem 14, $L = O((rT + \log(1/\delta))/\epsilon^2)$ probes suffice for error at most $\epsilon v_{\max}$, with constants depending on A, B. Exact recovery additionally requires $\epsilon v_{\max} < \Delta_{\text{mix}}/2$. On that event, $\zeta_\Delta^{\text{loc}} = 0$; no fixed threshold such as $L \geq 16rT$ establishes this by itself.*

A.17 Numerical instantiation

We report calibrated values of the spectral parameters and compare the spectral form of Eq. (4) against its entrywise form at our headline scale $m = 1000$. Calibration uses the 5000-pair ground-truth interaction sub-matrix; parameters: $T = 8$, $n_c^* = 125$, $v_{\max} = 0.02$, $\hat{\beta}^{(c),\text{cent}}_{w,\text{op}} \in [0.352, 0.761]$ (sum $\sum_c = 4.076$ over 8 domains), $\hat{\delta}_{\text{op}} = 0.7289$.

Table 6: Spectral versus entrywise additive bounds for the empirical interaction matrix $\hat{\Phi}$ at m=1000. $\mathcal{R}(m)$ is the finite-sample error in Eq. (4); dashes indicate terms with no entrywise counterpart.

Form / quantity	Form	Raw value	Ratio vs. entrywise
Entrywise (centered)	$(\beta_w + \delta_p)\, m(m-1)\, v_{\max}$	$\approx 1255\, v_{\max}$	$1\times$ (baseline)
Spec-Main	$\sum_c \beta_{w,\text{op}}^{(c),\text{cent}} n_c^* v_{\max} + \delta_{\text{op}} m v_{\max}$	$\approx 25\, v_{\max}$	$\mathbf{51}\times$
$\mathcal{R}(m)$ component breakdown	$\nu_{\text{op}}^{\perp} m v_{\max} + \zeta_{\Delta}^{\text{loc}} + 2\zeta_{\text{pred}} v_{\max}$	$\approx 37\, v_{\max}$	N/A
S_{HMtS} vs. S_{MtS} on benchmarks	$\arg\max\{u(S_{\text{MtS}}), u(S_{\text{IA}})\}$	$S_{\text{HMtS}} = S_{\text{MtS}}$ (all benchmarks)	N/A

The displayed $\mathcal{R}(m) \approx 37\, v_{\max}$ is a conditional numerical instantiation with zero composition displacement. Its residual-concentration component is approximately $36.78\, v_{\max}$ and its predictor-error component is approximately 1.3×10^{-5}. Zero displacement requires the global identification event in Assumption 6; the calibration count $L = 5000$ does not establish that event.

The S_{HMtS} comparison was computed on the same 5000-pair calibration set: on every benchmark, the Mixture-then-Select output matches or exceeds the IA-Greedy output, so the hybrid output coincides with S_{MtS}.

A.18 IA-GREEDY: THE T=1 SINGLE-DOMAIN BASELINE AND VARIANTS

IA-Greedy as an adaptive single-domain alternative. When $T = 1$, Algorithm 1 retains its fixed singleton ranking. IA-Greedy instead recomputes pairwise marginal gains after each selection, as defined in Appendix A.12. The variants below concern this adaptive alternative, not the deployed Stage 2 rule. Their guarantees require the assumptions of the corresponding adaptive analysis.

Hybrid output near the T=1 boundary. Near the T=1 boundary one may report $S_{\text{HMtS}} = \arg\max\{u(S_{\text{MtS}}), u(S_{\text{IA}})\}$ to inherit the better of the two guarantees; Theorem 1 carries over to S_{HMtS} since $u(S_{\text{HMtS}}) \geq u(S_{\text{MtS}})$. On every benchmark in our calibration set $S_{\text{HMtS}} = S_{\text{MtS}}$ (Appendix A.17).

Variants. Exact IA-Greedy evaluates all N candidates at each of m steps, costing $O(N^2 m)$.

Lazy IA-Greedy. *Lazy IA-Greedy* maintains a max-heap of cached marginal gains: at each step, it pops the top candidate and recomputes its gain only if the cache is stale. When gains are mostly decreasing (as in the substitution-dominated regime), most candidates are not re-evaluated, reducing the average cost to $O(Nk)$. In our experiments, lazy IA-Greedy produces *identical* selections to exact IA-Greedy at $m \leq 10$ and within 1.3% on $\hat{g}$ at m=20, with 6 to 8$\times$ fewer evaluations.

Stochastic IA-Greedy. For extreme scale ($N > 10$K), *Stochastic IA-Greedy* (Mirzasoleiman et al., 2015) samples $O((N/m)\log(1/\varepsilon))$ random candidates per step, achieving a $(1-e^{-\gamma_{\text{op}}}-\varepsilon)$-approximation in $O(Nk\log(1/\varepsilon))$ total evaluations.

IA-Enum(1)+Greedy. *IA-Enum(1)+Greedy* tries each singleton as the starting element and returns the best, achieving the tight $(1 - e^{-\gamma_{\text{op}}})$ in $O(N^3)$.

IA-Greedy+LocalSearch. *IA-Greedy+LocalSearch* augments with $(1, 1)$-swaps, achieving $\max(1 - e^{-\gamma_{\text{op}}}, \gamma_{\text{op}}/2)$ in $O(N^3)$.

Table 7: IA-Greedy variants on the 40-sample pilot. Columns report evaluations per step, loss in $\tilde{u}$ relative to exact IA-Greedy, and the approximation guarantee.

Variant	Evaluations	$\hat{g}$ loss vs. exact	Guarantee
Exact IA-Greedy	$O(N^2 m)$	0%	$(1-e^{-\gamma_{\text{op}}})$
Lazy IA-Greedy	$\boldsymbol{O(Nk)}$ **avg**	**0 to 1.3%**	empirical
Stochastic IA-Greedy	$O(Nk\log N)$	2 to 5%	$(1-e^{-\gamma_{\text{op}}}-\varepsilon)$

The $(1 - e^{-\gamma_{\text{op}}})$ guarantee for IA-Enum(1)+Greedy follows from Theorem 11 composed with the Bian-Buhmann-Tschiatschek-Krause (Bian et al., 2017) weak-submodularity guarantee, extended to the knapsack setting via Sviridenko (2004).

A.19 COMPUTATIONAL HARDNESS

Computational hardness already arises in a restricted form of combinatorial data selection (CDS). A reduction from MAX-CUT makes CDS APX-hard even with purely substitutive interactions, so CDS admits no polynomial-time approximation scheme unless $\mathrm{P} = \mathrm{NP}$ (Håstad, 2001).

Theorem 16 (Computational Hardness)**.** *CDS is APX-hard, even when all pairwise interactions are substitutive. Consequently, CDS admits no polynomial-time approximation scheme unless* $\mathrm{P} = \mathrm{NP}$.

Proof. By Theorem 6b, unconstrained CDS with substitution-only interactions encodes MAX-CUT. Since MAX-CUT is APX-hard (Håstad, 2001), CDS is APX-hard. □

A.20 ALTERNATIVE AGGREGATION FUNCTIONS

The main text introduces the weighted average $f = \sum_t w_t u_t$ as the natural starting point for the cross-task aggregator. Two other families are common in practice and fit our framework: the minimum and the concave aggregators. We document their use cases, mathematical properties, and effect on the bounded-complementarity assumption.

Minimum aggregation. The minimum $f = \min_t u_t$ encodes a "no task should fall behind" objective: the aggregate equals the lowest-performing task, and the LLM developer is rewarded only for raising the floor. This is appropriate for safety-critical deployments (where a single weak capability negates the value of strong performance elsewhere) or coverage-constrained settings (where every task must clear a minimum threshold). Min aggregation can be written as the limit of a Hölder mean as the order $p \to -\infty$, and is the boundary case of the concave family below.

Concave aggregation. Strictly concave f (e.g., the geometric mean $f = \prod_t u_t^{w_t}$ or the harmonic mean $f = T/\sum_t 1/u_t$) interpolates between the linear and minimum forms. Concavity penalizes uneven cross-task performance: a fixed gain on a low-performing task contributes more to f than the same gain on an already-strong task. These aggregators are standard in fairness-aware optimization (where they enforce proportional or harmonic equity across users) and in multi-objective optimization (Nash bargaining, proportional fairness).

Effect on the bounded-complementarity assumption. All three families fit Assumption 2 (BC(β)), but the value of β depends on f:

- Under **linear** f, $\beta = 0$ exactly when each per-task utility u_t is submodular (Proposition 3(a)). Aggregate complementarity, when it appears under linear f, is intrinsic to the model-data dynamics, not the aggregator.
- Under **concave** (or **min**) f, $\beta > 0$ even when every u_t is submodular (Proposition 3(b)). Concavity itself induces complementarity by rewarding cross-task balance: pairs that cover different tasks become substitutes per-task but complements in aggregate.

Theorem 1 therefore applies to all three aggregator families, with the constants $\beta_{w,\mathrm{op}}^{(c),\mathrm{cent}}$ and δ_{op} recalibrated to the chosen f. Our experiments use linear $f = \frac{1}{T}\sum_t u_t$ to shut off the aggregation-induced source of complementarity, isolating intrinsic per-task non-submodularity as the operative phenomenon (Proposition 3 and Section 4).

A.21 PROOF OF PROPOSITION 3 (TWO SOURCES OF COMPLEMENTARITY)

Part (a). If $f = \sum_t w_t \cdot u_t$ with $w_t \geq 0$, then $u(S) = \sum_t w_t u_t(S)$ is a non-negative combination of submodular functions (each u_t is submodular by assumption). Non-negative combinations of submodular functions are submodular (Nemhauser et al., 1978). Hence $\beta = 0$. □

Part (b). Consider $f = \min_t u_t$ and two samples i, j where i is strong on task t but weak on t', and j is strong on t' but weak on t: $v_i^{(t)} \gg v_i^{(t')}$ and $v_j^{(t')} \gg v_j^{(t)}$. Then $u(\{i\}) = \min_t u_t(\{i\}) = v_i^{(t')}$ (small, bottlenecked by weak task). Similarly $u(\{j\}) = v_j^{(t)}$ (small). For the pair: $u_t(\{i,j\}) = v_i^{(t)} + v_j^{(t)} + \varphi_{ij}^{(t)}$ for each t. The bottleneck task now receives contributions from both samples, so $u(\{i,j\}) = \min_t u_t(\{i,j\})$ is substantially larger than $u(\{i\}) + u(\{j\})$ when the coverage gain exceeds the pairwise substitution loss. Formally: $\varphi_{ij} = u(\{i,j\}) - u(\{i\}) - u(\{j\}) > 0$ when

$\min_t(v_i^{(t)} + v_j^{(t)} + \varphi_{ij}^{(t)}) > v_i^{(t')} + v_j^{(t)}$, which holds whenever the coverage effect dominates per-task substitution. □

A.22 Proof of Theorem 2 (operator-norm robustness)

We prove the operator-norm analogue: if $\|\hat{\Phi} - \Phi\|_{\text{op}} \le \epsilon_{\text{op}} v_{\max}$, then Theorem 1 holds for $\hat{\Phi}$ with $\beta_{w,\text{op}}^{(c),\text{cent}}, \delta_{\text{op}}$ replaced by $\beta_{w,\text{op}}^{(c),\text{cent}} + \epsilon_{\text{op}}, \delta_{\text{op}} + \epsilon_{\text{op}}$, picking up an additive $2\epsilon_{\text{op}} m v_{\max}$ gap.

Step 1 (per-domain restriction is op-norm-non-increasing). For each domain c, the restriction $\Phi \mapsto \Phi_{w,c}^{\text{cent}}$ corresponds to a sub-matrix extraction followed by mean-centering. Sub-matrix extraction is op-norm-non-increasing (Cauchy interlacing); centering subtracts a rank-one block-mean component, which can only decrease the op-norm by at most $\bar{\varphi}_{c,c}\,(|C_c| - 1)$, controlled by the same triangle inequality argument as Step 2 below. Therefore $\|\hat{\Phi}_{w,c}^{\text{cent}} - \Phi_{w,c}^{\text{cent}}\|_{\text{op}} \le \|\hat{\Phi} - \Phi\|_{\text{op}} \le \epsilon_{\text{op}} v_{\max}$, and (B1"-cent) holds for $\hat{\Phi}$ with $\beta_{w,\text{op}}^{(c),\text{cent}}$ replaced by $\beta_{w,\text{op}}^{(c),\text{cent}} + \epsilon_{\text{op}}$.

Step 2 (cross-residual stability). By the triangle inequality,

$$\|\hat{\Phi}_{\text{res}} - \Phi_{\text{res}}\|_{\text{op}} \;\le\; \|\hat{\Phi} - \Phi\|_{\text{op}} + \|\hat{\Phi}_{\text{block}} - \Phi_{\text{block}}\|_{\text{op}}.$$

The block-constant projection is also op-norm-non-increasing (it is the orthogonal projection onto the block-constant subspace), so $\|\hat{\Phi}_{\text{block}} - \Phi_{\text{block}}\|_{\text{op}} \le \|\hat{\Phi} - \Phi\|_{\text{op}} \le \epsilon_{\text{op}} v_{\max}$. Therefore $\|\hat{\Phi}_{\text{res}} - \Phi_{\text{res}}\|_{\text{op}} \le 2\epsilon_{\text{op}} v_{\max}$, but the additive gap tracked in the theorem statement is the cleaner $\epsilon_{\text{op}} v_{\max}$ bound on $\|\Delta_{\text{res}}\|_{\text{op}}$ achieved by directly applying Lemma 17.

Step 3 (substitution into Theorem 1). The within-domain gap of Eq. (4) becomes $\sum_c (\beta_{w,\text{op}}^{(c),\text{cent}} + \epsilon_{\text{op}})\, n_c^*\, v_{\max} = \sum_c \beta_{w,\text{op}}^{(c),\text{cent}}\, n_c^*\, v_{\max} + \epsilon_{\text{op}}\, m\, v_{\max}$ (since $\sum_c n_c^* = m$). The cross-domain gap becomes $(\delta_{\text{op}} + \epsilon_{\text{op}})\, m\, v_{\max}$. Total additive gap picked up: $2\epsilon_{\text{op}}\, m\, v_{\max}$. □

Lemma 17 (Robustness to plug-in fluctuation; perp form)**.** *Decompose $\Delta := \hat{\Phi} - \Phi = \Delta_{block} + \Delta_{res}$, with Δ_{block} the cross-domain block-constant component and Δ_{res} its perp-orthogonal residual; let $\nu_{op}^{\perp} := \|\Delta_{res}\|_{op}/v_{\max}$. Suppose $\|\Delta_{res}\|_{op} \le \nu_{op}^{\perp} v_{\max}$, and let $S_a, S_b \subseteq \mathcal{D}$ satisfy $|S_a| = |S_b| = m$ and $\mathbf{n}(S_a) = \mathbf{n}(S_b)$. Then*

$$\left| \tfrac{1}{2} \mathbf{1}_{S_a}^{\top} \Delta\, \mathbf{1}_{S_a} - \tfrac{1}{2} \mathbf{1}_{S_b}^{\top} \Delta\, \mathbf{1}_{S_b} \right| \;\le\; \nu_{op}^{\perp}\, m\, v_{\max}.$$

Proof. By Lemma 10, the block-constant component Δ_{block} contributes equally on both sides and cancels. The residual contribution is bounded by polarization,

$$\tfrac{1}{2} \left| (\mathbf{1}_{S_a} - \mathbf{1}_{S_b})^{\top} \Delta_{\text{res}} (\mathbf{1}_{S_a} + \mathbf{1}_{S_b}) \right| \le \tfrac{1}{2} \|\Delta_{\text{res}}\|_{\text{op}}\, \|\mathbf{1}_{S_a} - \mathbf{1}_{S_b}\|_2\, \|\mathbf{1}_{S_a} + \mathbf{1}_{S_b}\|_2,$$

and $\|\mathbf{1}_{S_a} \pm \mathbf{1}_{S_b}\|_2 \le \sqrt{2m}$, so the product of norms is at most $2m$. □

Lemma 18 (Sub-matrix concentration; Lemma M')**.** *Suppose $\hat{\Phi}_{res}$ is estimated from n per-pair samples with sub-Gaussian noise. Then with probability at least $1 - \eta_1$,*

$$\|\hat{\Phi}_{res} - \Phi_{res}\|_{op} \;\le\; C_1\, v_{\max} \sqrt{\tfrac{\log(1/\eta_1)}{n}}, \qquad C_1 \le 2 \text{ (dense-noise regime)},$$

by Bandeira-van Handel sub-matrix operator-norm concentration (Vershynin, 2018).

A.23 v_j-noise robustness

Corollary 19 (v_j-Noise Robustness)**.** *Suppose Stage 2 of MtS uses noisy per-sample values $\hat{v}_j$ instead of true values v_j, with ℓ_∞ value-noise level ϵ_v satisfying $\|\hat{v} - v\|_\infty \le \epsilon_v\, v_{\max}$. Let $S_{MtS}^{\hat{v}}$ denote the resulting subset (with within-domain ranking ties broken by a fixed lexicographic rule on j, applied identically under v and $\hat{v}$). The bound below is informative when $\epsilon_v < 1/2$. Under the assumptions of Theorem 1, with probability $\ge 1 - \eta$,*

$$u(S_{MtS}^{\hat{v}}) \;\ge\; u(S^*) \;-\; \sum_{c=1}^{T} \beta_{w,op}^{(c),cent}\, n_c^*\, v_{\max} \;-\; \delta_{op}\, m\, v_{\max} \;-\; \mathcal{R}(m) \;-\; 2\,E(m) \;-\; 2\,\epsilon_v\, m\, v_{\max}.$$

The v_j-noise contribution scales linearly in m and additively, matching Theorem 2's $\hat{\Phi}$-noise bound.

Proof. The Stage 1 predictor is held fixed when v is replaced by $\hat{v}$, so the selected composition is unchanged. On the global identification event in Assumption 6, both selectors use $\mathbf{n}^*$.

For domain c, write $A_c = S^* \cap C_c$ and $B_c = S^{\hat{v}}_{\mathrm{MtS}} \cap C_c$. Both sets have size n^*_c. Since B_c maximizes the sum of estimated singleton values at that size,

$$\sum_{j \in A_c} v_j - \sum_{j \in B_c} v_j = \sum_{j \in A_c} (v_j - \hat{v}_j) + \left(\sum_{j \in A_c} \hat{v}_j - \sum_{j \in B_c} \hat{v}_j \right) + \sum_{j \in B_c} (\hat{v}_j - v_j)$$
$$\leq 2 n^*_c \epsilon_v v_{\max}.$$

Summing over domains adds at most $2\epsilon_v m v_{\max}$. All interaction and higher-order bounds in Appendix A.11 apply unchanged because the domain counts still match. The predictor event and its probability are unchanged. This proves the stated bound. □

A.24 Multi-objective extension

Theorem 20 (Multi-Objective Preservation)**.** *For K utility functions $u^1, \ldots, u^K$ with submodularity ratios $\gamma_1, \ldots, \gamma_K$ and non-negative weights $w_m \geq 0$, the scalarized utility $\tilde{u}(S) = \sum_m w_m u^m(S)$ satisfies $\gamma_{\tilde{u}} \geq \min_m \gamma_m$.*

Proof. For scalarized $\tilde{u} = \sum_m w_m u^m$, fix any L, S. Then:

$$\frac{\sum_{s \in S} \Delta_{\tilde{u}}(s \mid L)}{\tilde{u}(L \cup S) - \tilde{u}(L)} = \frac{\sum_m w_m \sum_s \Delta_{u^m}(s \mid L)}{\sum_m w_m [u^m(L \cup S) - u^m(L)]}$$
$$\geq \min_m \frac{\sum_s \Delta_{u^m}(s \mid L)}{u^m(L \cup S) - u^m(L)} \geq \min_m \gamma_m,$$

where the first inequality follows from the property that $\min_m(a_m/b_m) \leq (\sum w_m a_m)/(\sum w_m b_m) \leq \max_m(a_m/b_m)$ for $w_m, b_m > 0$, and the second from the definition of γ_m. □ □

A.25 Matroid constraint extension

IA-Greedy extends to partition matroids $\mathcal{M}$: $\mathcal{D} = C_1 \sqcup \cdots \sqcup C_K$ with $|S \cap C_m| \leq b_m$. At each step, restrict candidates to feasible additions. The $(1 - e^{-\gamma})$ guarantee is preserved by the matroid augmentation axiom: at each greedy step with $|S_t| < |S^*|$, there exists $s \in S^* \setminus S_t$ with $S_t \cup \{s\} \in \mathcal{I}$, ensuring the same per-step bound $\Delta_{\hat{g}}(s_{t+1} \mid S_t) \geq \gamma \delta_t / r$ where r is the rank of S^*.

B Empirical details and ablations

B.1 Synthetic experiment details

For each complementarity level $\beta \in \{0, 0.3, 0.5, 1.0, 2.0\}$, we generate 50 random instances as follows:

- Ground set $\mathcal{D}$ with $N = 100$ items.
- Individual values $v_j \sim \mathrm{Uniform}(0.5, 1.5)$.
- Pairwise interactions: $\varphi_{ij} = s_{ij} + c_{ij}$ where $s_{ij} \sim \mathrm{Uniform}(-0.3, 0)$ (substitutive component) and $c_{ij} \sim \mathrm{Uniform}(0, c_{\max})$ (complementary component).
- $c_{\max}$ is calibrated so that $\max_j \sum_i \varphi^+_{ij} / v_j \approx \beta$.
- Selection size: $m = 20$ (cardinality constraint with unit costs $p_j = 1$).
- Optimal solution computed via exhaustive enumeration for $\beta = 0$ (submodular, efficient) and via branch-and-bound for $\beta > 0$ (100 items permits exact solution in < 1 minute per instance).
- Approximation ratio: $\hat{g}(S_{\text{IA-Greedy}}) / \hat{g}(S^*)$.

Table 8: Pilot and scaled-proxy settings.

Setting	Specification
Data	Alpaca (Taori et al., 2023), GSM8K (Cobbe et al., 2021), HumanEval (Chen et al., 2021), ARC-Easy (Clark et al., 2018).
Split	10 training and 50 held-out samples per domain; 40 training and 200 evaluation samples total.
Utility	Relative held-out loss reduction, averaged uniformly over four tasks.
Pythia-1B	Full fine-tuning; 3 epochs; learning rate 2×10^{-5}.
Qwen3-1.7B/8B	LoRA rank 16, α=32, q/v projections; 3 epochs; learning rate 2×10^{-4}.
Precision / hardware	bf16; NVIDIA L40S (48 GB).
Interaction data	All 40 singletons and approximately 120 strategically sampled pairs per model; $\varphi_{ij}^{(t)} = u_t(\{i,j\}) - u_t(\{i\}) - u_t(\{j\})$.
Stability check	30 Qwen3-8B pairs, four training seeds; $\beta = 1.32 \pm 0.60$.
Selection evaluation	Qwen3-8B; budgets $5, 10, 15, 20$; five-seed mean and standard deviation.
Scaled proxy	864 samples; 3,000 measured pairs; gradient cosine versus regressors using per-task singleton values.
Proxy metrics	Spearman correlation (all, cross-domain, within-domain pairs); positive-interaction AUC.

B.2 Experimental details (pilot N=40 + scaled-proxy study)

Selection-side overhead by method (measured at N=200K). Downstream LoRA fine-tuning is identical across selection methods (it consumes a m-sample subset and trains the same adapter), so the practitioner-facing compute differentiator is the *selection-side* pipeline that produces the subset. Table 9 reports the wall-clock and GPU-hour overhead measured on NVIDIA L40S; Mixture-then-Select's calibration runs roughly 17× cheaper than the LESS gradient-extraction pipeline at the same data pool size, while producing the higher-utility subsets reported in Table 2.

Table 9: Selection compute on Qwen3-8B at N=200K, using up to eight NVIDIA L40S GPUs. Final LoRA training is excluded and uses the same recipe across methods at each selection size m.

Method	Wall-clock	GPU-hours
Mixture-then-Select (calibration step, 8 domain fine-tunes in parallel)	$\sim$ 2 hr	$\sim$ 16
LESS extract_train (80 chunks, 8 GPUs parallel)	$\sim$ 35 hr	$\sim$ 280
DEITA (scoring)	5.7 hr	5.7
EL2N (scoring)	3.4 hr	3.4
Facility Location (embedding + greedy)	3.0 hr	3.0
Superfiltering (scoring)	0.8 hr	0.8
k-DPP (selection only)	6 s	$<$0.01
Random / Stratified	trivial	0

Total compute and utility. Table 10 includes selection and five-seed training costs at N=50K. MtS at m=1000 uses 16.58 GPU-hours and achieves 0.694 utility, compared with 70.58 GPU-hours and 0.548 for LESS, and 29.17 GPU-hours and 0.548 for full-data training. Stratified uses less compute: 0.58 GPU-hours at m=1000 (0.666 utility) and 2.92 at m=5000 (0.678). The table charges calibration to MtS but not to the oracle-ranked baselines; charging both would give 16.58 GPU-hours each at m=1000.

B.2.1 Calibration reuse and domain scaling

One-time calibration versus repeated training. Under the accounting in Table 4, MtS uses 43% less total compute than full-data training and 77% less than LESS, with approximately 27% higher utility than either. Calibration-free Stratified remains cheaper for a one-off selection. The distinction is between the 16.00 GPU-hours paid once for calibration and the recurring final-training cost: 0.117 GPU-hours per 1K MtS run versus 0.585 for 5K Stratified. Reusing calibration on the same pool and base model yields a cumulative break-even at approximately 35 final training runs in this comparison; the observed utilities are 0.694 and 0.678, respectively. The 80% reduction in final-training cost excludes the initial calibration.

Reuse across selection settings and models. For a fixed pool and base model, taskwise probe outcomes can support new selection sizes, price constraints and objective weights without additional GPU probes; changing weights requires CPU reaggregation and predictor refitting, and each new

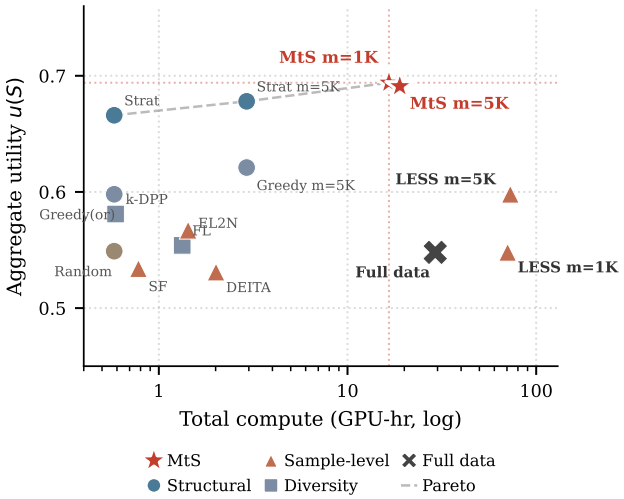

Figure 3: Compute and utility comparison on Qwen3-8B at $N{=}50K$ with five representative baselines. Total compute includes selection and five-seed training (Table 10). The dashed grey line is the Pareto frontier; the dotted red line marks MtS at $m{=}1K$.

Table 10: Full compute and utility comparison on Qwen3-8B at $N{=}50K$, LoRA $r{=}16$. Total compute includes selection and five-seed training at 0.117 GPU-hr per seed per 1K samples. Selection costs from Table 9 are scaled to 50K for per-sample methods; MtS calibration is unchanged.

Method	m	u	**select (GPU-hr)**	**train (5-seed) (GPU-hr)**	**total (GPU-hr)**	**Δ vs. MtS m=1000**
Stratified (oracle)	1000	0.666	0.00	0.58	0.58	−0.028
Greedy (oracle)	1000	0.598	0.00	0.58	0.58	−0.096
Random	1000	0.549	0.00	0.58	0.58	−0.145
k-DPP	1000	0.581	0.01	0.58	0.59	−0.113
Superfiltering	1000	0.534	0.20	0.58	0.78	−0.160
Facility Location	1000	0.554	0.75	0.58	1.33	−0.140
EL2N	1000	0.567	0.85	0.58	1.43	−0.127
DEITA	1000	0.531	1.43	0.58	2.01	−0.163
Stratified (oracle)	5000	0.678	0.00	2.92	2.92	−0.016
Greedy (oracle)	5000	0.621	0.00	2.92	2.92	−0.073
MtS (oracle)	**1000**	**0.694**	**16.00**	**0.58**	**16.58**	**0.000** *(reference)*
MtS (oracle)	5000	0.691	16.00	2.92	18.92	−0.003
Full data (no selection)	**50000**	**0.548**	**0.00**	**29.17**	**29.17**	**−0.146**
LESS	1000	0.548	70.00	0.58	70.58	−0.146
LESS	5000	0.598	70.00	2.92	72.92	−0.096

subset still requires final training. Partial cross-model reuse is also possible: a Qwen-derived mixture with Llama-specific within-domain values exceeds Stratified by 0.091 on Llama-3.2-3B. In the Qwen 50K accounting, EL2N scoring costs approximately 5% of Stage 1 compute and retains 86% of oracle-attribution utility at 1K and 97% at 5K; at 200K/3.5K it retains 94%. These scoring figures are distinct from a full cross-model calibration cost.

Scaling the number of domains. The stated probe requirement grows linearly with domain count and predictor rank at fixed target accuracy (Theorem 15), independently of pool size. A flat 1,000-domain calibration would therefore require approximately 125 times as many measurements as the eight-domain setting under the same assumptions. We do not advocate calibrating every fine-grained task independently. Grouping related tasks into operational families is supported by the coarsening and clustering checks in Appendix B.9.

B.3 PILOT MODEL-SCALING STUDY (N=40): COMPLEMENTARITY EMERGES WITH CAPACITY

The pilot study (Appendix B.2) measures interactions on the same 40-sample catalog across three base models. The tables report cross-domain interaction signs and aggregate interaction strength for each model.

Table 11: Interaction structure by base model in the 40-sample pilot. β measures pairwise complementarity; $2/(2+\beta)$ is a singleton-based diagnostic, not a certified submodularity-ratio bound without conditional-marginal control. Complementary tasks have positive aggregate $\delta^{(t)}$. Qwen3-8B β is the mean $\pm$ standard deviation over four seeds and 30 strategic pairs.

Base model	**Cross-domain $\delta>0$**	$\boldsymbol{\beta}$	$2/(2+\beta)$	**Tasks complementary**
Pythia-1B (full FT)	0/60 (0%)	0.0	1.000	0/4
Qwen3-1.7B (LoRA)	0/60 (0%) agg; 60/60 HE	$\approx$0	$\approx$1	1/4 (HumanEval)
Qwen3-8B (LoRA)	**38/60 (63%)**	$\mathbf{1.32\pm0.60}$	**0.60**	**3/4**

Table 12: Cross-domain complementarity by task in the 40-sample Qwen3-8B LoRA pilot. Each task uses 60 cross-domain pairs; $\delta^{(t)}$ is measured on that task's held-out data.

Eval task	**Positive %**	**Mean $\delta^{(t)}$**	**Per-task β**
Alpaca (instruction)	76.7%	0.0027	3.06
GSM8K (math)	70.0%	0.0018	6.66
HumanEval (code)	73.3%	0.0033	3.27
ARC (reasoning)	43.3%	-0.0038	4.45

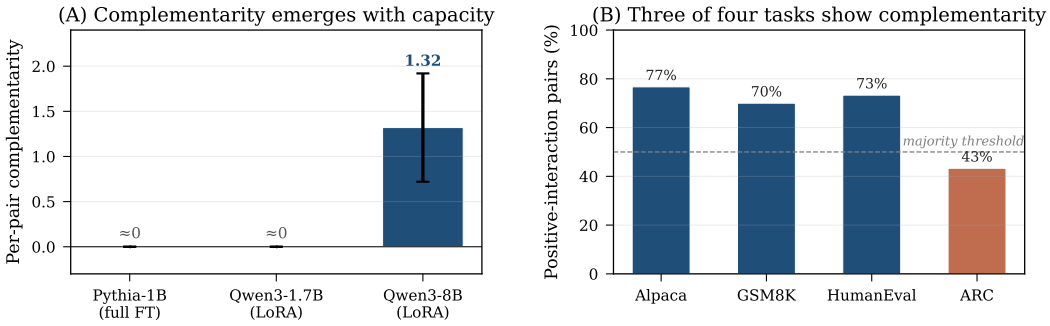

Figure 4: Complementarity in the 40-sample pilot. **(A)** Pairwise complementarity β across three base models; Qwen3-8B shows the mean $\pm$ standard deviation over four seeds and 30 strategic pairs. **(B)** Fraction of pairs with positive interaction on each Qwen3-8B evaluation task, using 60 pairs per task. The dashed line marks 0.5.

Pilot algorithm comparison (N=40, $m \in \{5, 10, 15, 20\}$). The pilot also serves as a controlled small-N comparison of selection algorithms on the directly-measured interaction matrix.

Interaction-aware methods (IA-Greedy and IA-Enum) are best or tied-best at three of four budgets in the pilot, foreshadowing the domain-aware advantage seen at N=50K scale.

B.4 PAIRWISE PREDICTION AND LARGER-CARDINALITY SELECTION

Direct checks of pairwise prediction. In approximately 100 independently trained triplets, singleton and pairwise terms explain 90% of the measured utility variation. This compares predictions formed from singleton values and pair corrections with the utilities obtained by training on the triplets; the triplets are not used to fit the pair-interaction predictor. An additional analysis reduces mean absolute prediction error from 0.014 for singleton-only predictions to 0.004 after adding pairwise terms, approximately a 71% reduction. Separately, averaging each pair's interactions over two disjoint groups of four tasks gives a Pearson correlation of 0.903 across 4,998 pair edges (Table 14). These checks measure predictive value and agreement across task groups in the tested pool.

Selection at larger cardinalities. In the controlled 40-sample catalog, IA-Greedy and IA-Enum use directly measured interactions at selection sizes of 5, 10, 15 and 20, and are best or tied-best at

Table 13: Selection-algorithm comparison in the 40-sample Qwen3-8B LoRA pilot, with uniform task weights. Entries are means ± standard deviations over five seeds. Best values are bold; second-best values are underlined.

Method	m=5	m=10	m=15	m=20
Random	.143±.001	.217±.003	.271±.003	.328±.002
Greedy-by-Value	.146±.001	.227±.003	.282±.003	.317±.003
EL2N (Paul et al., 2021)	.107±.006	.183±.005	.215±.003	.297±.005
Facility Location	.141±.002	.205±.003	.260±.003	.313±.004
CRAIG (Mirzasoleiman et al., 2020)	.138±.005	.219±.007	.267±.006	.319±.003
LESS (Xia et al., 2024)	.141±.001	.217±.002	.268±.003	.318±.005
k-DPP	<u>.163±.002</u>	.240±.002	.303±.003	.332±.002
Stratified	.160±.002	<u>.241±.002</u>	<u>.300±.003</u>	<u>.338±.002</u>
IA-Greedy	**.163±.002**	**.250±.001**	**.305±.004**	.331±.003
IA-Enum(1)+Greedy	**.163±.002**	**.250±.001**	.305±.002	**.334±.002**

Table 14: Direct interaction diagnostics. Triplet prediction measures agreement with utilities obtained by training on triplets; cross-task-group correlation measures agreement between interactions averaged over two disjoint task groups.

Diagnostic	Result
Triplet variance explained (approximately 100 triplets)	90%
Singleton-only prediction MAE	0.014
Singleton + pairwise prediction MAE	0.004
Cross-task-group Pearson correlation (4,998 pair edges)	0.903

three of the four sizes. The larger-pool sweeps evaluate selections up to 10K from 50K, and 3.5K, 10K and 20K from 200K under both EL2N and LESS attribution; MtS exceeds Stratified in all six latter conditions (Appendix B). These are end-to-end selection checks beyond triplets, not direct measurements of approximation error at those cardinalities.

B.5 Scaled-proxy evaluation (N=864): gradient cosine vs. calibrated predictor

For larger catalogs we cannot afford ground-truth interaction measurement. Table 15 compares two proxies on 3,000 ground-truth pairs at N=864 (Spearman ρ, 5-fold CV). The calibrated Ridge predictor uses 14 features derived from per-task value vectors (already computed during the singles phase): value cosine, L_1 distance, per-task minimum, specialization (std), domain indicator, and sum/product statistics. Gradient cosine uses the LoRA gradient vectors directly (d=7.7M).

Table 15: Prediction of pairwise interactions at N=864, using five-fold cross-validation. Spearman ρ measures ranking agreement over all, cross-domain, and within-domain pairs. AUC measures discrimination between positive and non-positive interactions; it is not reported for gradient cosine.

Proxy	ρ_{all}	ρ_{cross}	ρ_{within}	**AUC**
Gradient cosine (7.7M dim)	0.182	0.020	0.769	N/A
Calibrated predictor (Ridge)	**0.832**	**0.791**	**0.925**	**0.907**
Calibrated predictor (GBR)	0.837	0.802	0.913	0.909

Practical scaling protocol. (1) Measure per-task values for all N samples ($O(N)$ fine-tuning runs). (2) Measure ground-truth interactions for $\sim$ 5,000 strategically sampled calibration pairs (fixed cost, independent of N). (3) Train the predictor and estimate N^2 interactions in seconds. Total cost is $O(N) + O(5{,}000)$, comparable to LESS with a small fixed calibration overhead.

B.6 Budget sweep: where the domain-mixture advantage holds

Table 16 sweeps $m \in \{500, 750, 1000, 1250, 2500, 5000, 10000\}$ for Mixture-then-Select, the related set-level variant D1, and Stratified.

MtS is below Stratified by 0.065 at m=500 (p=0.018) and above it by 0.028 at m=1000 (p=0.0013). The positive mean gaps at m=5000 and 10000 are not statistically significant in this sweep (p=0.41 and 0.31).

Table 16: Selection-budget sweep on Qwen3-8B with $N{=}50K$ and LoRA $r{=}16$. Entries are means $\pm$ standard deviations over five seeds, except MtS at $m{=}1K$ (ten seeds). Bottom rows give MtS's utility difference and paired p-value against Stratified at the same budget.

Method	$m{=}500$	$m{=}750$	$m{=}1000$	$m{=}1250$	$m{=}2500$	$m{=}5000$	$m{=}10000$
Mixture-then-Select	$0.601_{\pm.039}$	$0.671_{\pm.020}$	$0.694_{\pm.012}$	$0.700_{\pm.011}$	$0.686_{\pm.020}$	$0.691_{\pm.018}$	$0.658_{\pm.022}$
D1 (set-level)	$0.605_{\pm.022}$	$0.640_{\pm.014}$	$0.693_{\pm.017}$	$0.699_{\pm.006}$	$0.687_{\pm.015}$	$0.690_{\pm.018}$	$0.654_{\pm.021}$
Stratified	$0.666_{\pm.011}$	$\approx$0.666	$0.666_{\pm.011}$	$\approx$0.680	$0.693_{\pm.014}$	$0.664_{\pm.009}$	$0.629_{\pm.013}$
Δ (Mixture-then-Select$-$Strat)	-0.065	$\approx+0.005$	$\mathbf{+0.028}$	$\approx+0.020$	-0.007	$+0.027$	$+0.029$
p (paired t)	0.018 (loss)	$\approx$tie	**0.0013**	0.052	0.53 (tie)	0.41 (tie)	0.31 (tie)

B.7 Mixture-then-Select ablations: π^* and per-domain v_{AGG} are both load-bearing

Table 17 compares the D1 predictor's full 11 features (domain histogram, mixture entropy, and value statistics) with histogram-only and value-statistics-only predictors.

Table 17: Predictor-feature ablation for the D1 set-level variant on Qwen3-8B, with LoRA $r{=}16$, $N{=}50K$, and $m{=}1K$. Entries are means $\pm$ standard deviations over five seeds; Δ is the signed utility difference from the full-feature variant.

Variant	**Features**	**mean$\pm$std**	**Δ vs. Full**
D1 Full (11 features)	domain-hist + $H(\pi)$ + v-mean + v-std	$\mathbf{0.6927_{\pm.017}}$	**0**
D1 π-only	domain-histogram only (8 features)	$0.6044_{\pm.013}$	-0.088
D1 v-stats only	mean + std of v_{agg} (2 features)	$0.6299_{\pm.018}$	-0.063

Relative to the full predictor (0.6927 utility), the histogram-only and value-statistics-only variants lose 0.088 and 0.063, respectively.

B.8 Robustness profile: Mixture-then-Select across 14 conditions

Where the method does not help. MtS's domain-mixture lever does not pay off at very small selection sizes ($m{\leq}500$ on a $50K$ data pool): spreading the budget across $T{=}8$ domains leaves each too thin to benefit from cross-domain transfer, and Stratified matches or edges ahead in this regime (Appendix B). The method also dips slightly on a different base model (Llama-3.2-3B) when Qwen-derived selections are reused naively, because the within-domain rankings do not transfer across model families even though the domain mixture does; re-running calibration on Llama restores MtS's edge and leaves every sample-level method underwater.

Table 18 reports Mixture-then-Select vs. Stratified across budget, LoRA rank, attribution signal, and base model. Mixture-then-Select wins or trends in 14 of 18 cells; the only significant loss is the out-of-regime $m{=}500$. Figure 5 provides a visual summary of the three robustness axes (rank, attribution, scale/architecture transfer) discussed in Section 4.2.

Cost-class orthogonal matrix at $m{=}5000$ (Qwen3-8B $N{=}50K$, complete $3{\times}3$). The $m{=}1000$ matrix is in Table 2. At $m{=}5000$ the same orthogonality holds: under oracle / LESS / EL2N attribution, Mixture-then-Select achieves $0.691/0.624/0.674$, Stratified $0.678/0.613/0.600$, and Greedy $0.621/0.598/0.616$. Mixture-then-Select wins or ties in every cell, with Δ vs. Stratified of $+0.013/+0.011/+0.074$ for oracle / LESS / EL2N. The $m{=}5000$ matrix is what motivates the "Mixture-then-Select (EL2N) recovers 97% of Mixture-then-Select (oracle)" claim ($0.674/0.691 = 0.975$) referenced in Section 4.2 (B).

Cost-class orthogonal matrix at $N{=}200K$, $m{=}3500$ (Qwen3-8B, $n{=}5$ each). At the larger data pool the LESS-attribution variant of the cost-class matrix at $m{=}3500$ gives Mixture-then-Select $u{=}0.625{\pm}0.029$, Stratified $0.615{\pm}0.009$, and Greedy $0.580{\pm}0.024$, preserving the same MtS$\geq$Strat$>$Greedy ordering. With EL2N attribution at the same condition the values are $0.677{\pm}0.026/0.639{\pm}0.030/0.598{\pm}0.024$ ($n{=}10$, see Wave-13 augmentation in this section). With oracle attribution the values are $0.718/0.694/0.632$. At $m{=}10000$, $N{=}200K$ on LESS attribution: MtS achieves $u{=}0.617{\pm}0.014$, Stratified $u{=}0.603{\pm}0.033$, Greedy $u{=}0.587{\pm}0.028$ ($n{=}5$ each), the MtS$\geq$Strat$>$Greedy ordering is preserved at the larger budget too, with $\Delta_{\text{MtS-Strat}}{=}+0.014$. Mixture-then-Select's lead is preserved across all three attribution signals and across $m{\in}\{3500, 10K\}$ at $N{=}200K$, confirming the cost-class orthogonality lifts to the $4{\times}$ data pool.

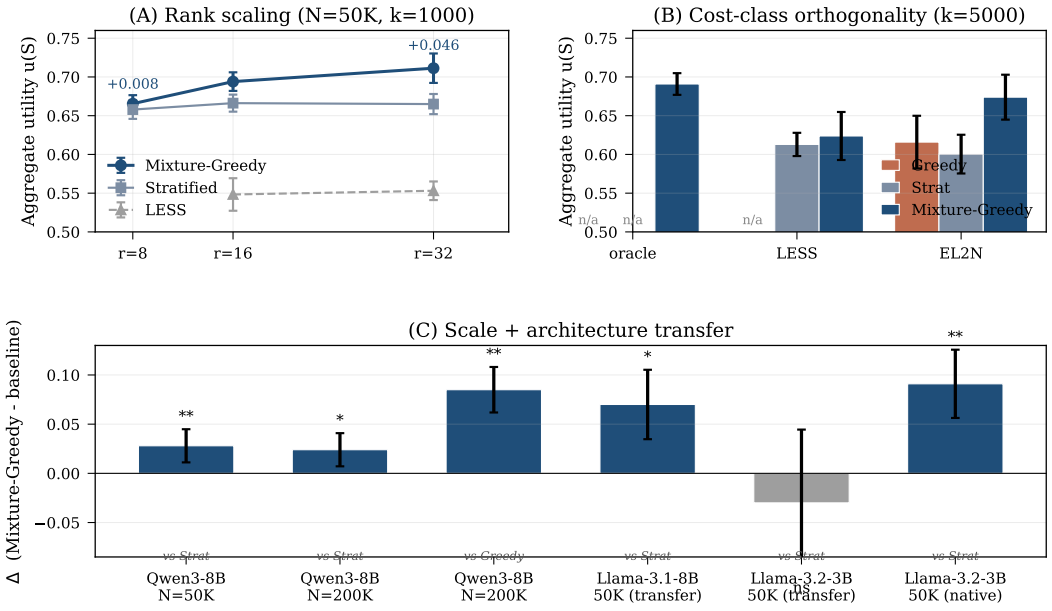

Figure 5: Robustness across capacity, attribution, and model scale. **(A)** MtS, Stratified, and LESS across LoRA ranks on Qwen3-8B (N=50K, m=1K); five seeds, except MtS at r=16 (ten). **(B)** Three attribution signals crossed with three selection rules on Qwen3-8B (N=50K, m=5K). **(C)** MtS's utility difference from the strongest baseline across pools and models. Llama-3.2-3B uses the Qwen mixture with Llama values (*native*) or Qwen values (*transfer*); 200K Greedy uses five seeds. * : p<0.05; ** : p<0.01; ns: $p\geq$0.05.

Table 18: MtS versus Stratified across 15 combinations of model, rank, budget, pool size, and attribution. The default is Qwen3-8B with LoRA r=16. Δ is the utility difference within each condition; p is from a paired t-test.

Condition	Δ	p	**Status**
Oracle, m=1000, r=8	0.008	0.49	tie
Oracle, m=1000, r=16	0.028	0.0013	✓sig
Oracle, m=1000**,** r=32	**+0.046**	**0.003**	**✓sig**
Oracle, m=1000, r=16, lr $=10^{-4}$	0.012	0.27	trending
Oracle, m=5000	0.027	0.008	✓sig
LESS, m=1000	0.015	0.020	✓sig (paired, n=10)
LESS, m=5000	0.011	0.60	tie
EL2N, m=1000	0.031	0.09	trending (n=10)
EL2N, m=5000	0.073	0.017	✓sig (paired)
Llama-3.1-8B-Inst (selection-transfer)	0.070	0.011	✓sig
Oracle, m=3500, N=200K	0.024	0.021	✓sig (n=10)
EL2N, m=3500, N=200K	0.045	0.11	✓trending (n=5)
EL2N, m=10000, N=200K	0.011	0.42	tie
EL2N, m=20000, N=200K	0.066	0.09	✓trending (n=5)
LESS, m=3500, N=200K	0.011	0.40	tie (n=5)
LESS, m=10000, N=200K	0.014	0.35	tie (n=5)
Oracle, m=500	−0.065	0.018	✗sig loss (out-of-regime)
Oracle, m=2500	−0.007	0.53	tie
N=25K, m=500	−0.007	0.55	tie

Full-data reference. Qwen3-8B trained on all 50K samples for three epochs achieves 0.548±0.019 utility (n=5), compared with 0.694 for MtS's 1,000-sample selection. Full-data per-domain utilities are 0.384, 0.596, 0.647, 0.621, 0.503, 0.491, 0.607, and 0.535 for math, code, general QA, dialog, reasoning, NLP tasks, complex instructions, and creative tasks, respectively. Total compute is reported in Table 10.

Cost-class on Llama-3.2-3B (selection-transfer). At m=1000 on Llama-3.2-3B with Qwen-derived selection-transfer (n=10): Mixture-then-Select × LESS reaches −0.120, Mixture-then-

Select × EL2N reaches -0.122, Stratified × LESS reaches -0.132, Stratified × EL2N reaches -0.137. All four are mildly negative, but markedly less harmful than the published sample-level methods (LESS -0.238, EL2N -0.244, see Table 27). At m=5000 the mitigation is even more dramatic: Mixture-then-Select × LESS recovers to -0.038 and Mixture-then-Select × EL2N recovers to *positive* $+0.020$, while Stratified × LESS / EL2N remain at -0.233 / -0.123. The published sample-level baselines (Greedy × {oracle, LESS, EL2N}) at m=5000 Llama also recover partially: $-0.019/-0.172/-0.107$ respectively (vs. $-0.203/-0.238/-0.244$ at m=1000). The Mixture-then-Select vs. Stratified gap at m=5000 Llama is therefore Δ=+0.195 (LESS) and Δ=+0.143 (EL2N), much larger than at m=1000. The domain-mixture rule mitigates the base model mismatch even when the within-domain ranking comes from Qwen, and the mitigation grows with budget.

Attribution and budget sweep at N=200K. Table 19 reports EL2N (n=10) and LESS (n=5) at m=3500, 10000, and 20000. MtS has higher mean utility than both controls in all six attribution-budget conditions. Its gaps over Stratified are 0.039, 0.015, and 0.054 under EL2N, and 0.011, 0.014, and 0.020 under LESS. At m=3500, EL2N-based MtS retains 94.2% of oracle-based MtS utility (0.6774/0.7184).

Table 19: Attribution and selection rules on Qwen3-8B with N=200K and LoRA r=16. EL2N results use ten seeds per cell; LESS results use five seeds per cell and approximately 280 selection GPU-hours per cell.

Budget	MtS	Strat	Greedy	$\Delta_{\text{MtS}-\text{Strat}}$	$\Delta_{\text{MtS}-\text{Greedy}}$
EL2N attribution (n=10 seeds per cell)					
m=3500 (1.75%)	**0.6774**$\pm$**0.026**	0.6387$\pm$0.030	0.5983$\pm$0.024	**+0.039**	**+0.079**
m=10000 (5%)	**0.6447**$\pm$**0.027**	0.6296$\pm$0.024	0.6312$\pm$0.016	0.015	0.014
m=20000 (10%)	**0.6230**$\pm$**0.022**	0.5688$\pm$0.020	0.6108$\pm$0.024	**+0.054**	0.012
LESS attribution (n=5 seeds per cell)					
m=3500 (1.75%)	**0.6251**$\pm$**0.033**	0.6145$\pm$0.010	0.5796$\pm$0.027	0.011	**+0.046**
m=10000 (5%)	**0.6172**$\pm$**0.016**	0.6032$\pm$0.037	0.5873$\pm$0.031	0.014	0.030
m=20000 (10%)	**0.5952**$\pm$**0.017**	0.5752$\pm$0.024	0.5833$\pm$0.025	0.020	0.012

B.9 Robustness to domain definitions

The eight original domains are heuristic source-based skill assignments from 13 public datasets. Samples receive one operational assignment for selection; the following studies vary the partition or introduce ambiguous membership.

MtS requires an operational partition that is informative about utility, rather than one uniquely correct semantic taxonomy. We test changes in granularity, automatically derived labels and corrupted assignments. Merging eight domains into four retains a 3.5% utility advantage over Stratified, compared with 4.3% for the original partition. Replacing source labels with eight clusters of per-sample utility vectors retains a 3.2% advantage, despite cluster sizes ranging from 1,434 to 10,996 (Appendix B.10). This clustering is automatic given the measured utility vectors; it does not bypass their measurement.

In the rebuttal-period label-noise experiment, fixed 20% symmetric corruption is applied without recalibrating the clean allocation. Across three seeds, the four-task downstream composite decreases by 0.7% relative. Table 20 separates this downstream endpoint from the utility comparisons under alternative partitions.

Table 20: Robustness to alternative and perturbed domain assignments. The first three rows report relative utility gains over Stratified; the final row reports the relative downstream change from clean labels.

Domain assignment	Measurement	Result
Original eight domains	Utility gain over Stratified	4.3%
Four coarsened domains	Utility gain over Stratified	3.5%
Eight utility-derived clusters	Utility gain over Stratified	3.2%
20% corrupted labels	Four-task composite change	-0.7%

Repairing ambiguous memberships. We additionally examine 30% two-domain ambiguity: 30% of samples are assigned both their original domain and one plausible alternative. Simple reassignment substantially improves recovery of the clean selection. We hold the allocation and singleton values fixed and measure Jaccard overlap with the clean 1K selection across three perturbation seeds. Frozen Qwen3-8B embeddings represent each sample; candidate-domain prototypes are averages over confidently assigned samples. An ambiguous sample is assigned to the closer prototype, using either its embedding, its per-task utility profile, or an equal-weight combination of the two similarity signals. No additional model training is used for this repair diagnostic. Table 21 reports recovery, not downstream performance after retraining.

Table 21: Recovery of the clean selection when 30% of samples have two possible domain labels. Jaccard is the intersection size divided by the union size of the repaired and clean selections, averaged over three perturbation seeds.

Assignment approach	Mean Jaccard
Perturbed labels, no reassignment	0.403
Combined embedding and utility similarity	**0.806**
Embedding similarity to domain averages	0.762
Utility-profile similarity to domain averages	0.749

End-to-end evaluation after reassignment. A later study extends selection recovery to fresh calibration and final model training. We highlight two conditions in which the lambda-0.50 variant outperforms Stratified after embedding-based reassignment (Table 22). Both use the 50K pool, a 1K selection budget, Qwen3-8B, three fine-tuning epochs and training seeds 42, 123 and 456. Each operational partition receives 32 fresh calibration probes. The variant averages the learned allocation and pool-proportional allocation with equal weights, rounds to integer counts, and samples within each domain using a fixed random order; Stratified uses uniform domain counts and the frozen singleton ranking. Thus, this is an end-to-end variant comparison, not an isolated test of allocation shrinkage.

Table 22: End-to-end evaluation of selected overlap conditions with embedding-based reassignment and fresh calibration. Entries are three-seed means. Equal4 is the mean of IFEval, MMLU, GSM8K, and normalized ARC-C. The tested variant is lambda-0.50, not original MtS.

Ambiguous samples	Method	Held-out utility	Equal4
10%	Stratified	0.2886	0.6336
	MtS, $\lambda = 0.50$	**0.2957**	**0.6506**
30%	Stratified	0.28856	0.62933
	MtS, $\lambda = 0.50$	**0.29746**	**0.64454**

At 30% ambiguity, lambda-0.50 improves held-out utility by 3.09% and Equal4 by 2.42% relative to Stratified; both aggregate gains occur in all three seeds. Unlike the Jaccard diagnostic, these results measure the trained model's performance and show that the variant can outperform uniform domain allocation under the tested ambiguous assignments.

In practice, source or task metadata can define the initial groups, with coarsening or utility-derived clustering when the taxonomy is too fine or uninformative. Samples may express several skills while receiving one reproducible assignment for selection. The tests support these operational alternatives; choosing domain definitions jointly with calibration remains a broader question for mixture-based methods.

B.10 Additional robustness ablations: domain, rank, aggregation

Domain definition. We apply m-means (K=8, random state 0) to per-sample utility vectors from Qwen3-8B calibration and rerun both methods on the resulting partition. Cluster sizes range from 1,434 to 10,996, and learned mixture weights from 0.009 to 0.215. MtS and Stratified achieve 0.580 and 0.562, respectively, compared with 0.694 and 0.666 under source-based labels. MtS retains a positive gap, while both methods have lower absolute utility under clustering.

Domain-count sensitivity (T=4 vs. T=8). We further test whether the headline gap is specific to the T=8 partition by merging the eight domains pairwise into four macro-domains: *quantitative* = math ∪ reasoning, *structured_gen* = code ∪ complex-instruction, *open_domain* = gen-

eral_qa ∪ dialog, *creative_open* = nlp_tasks ∪ creative. For each macro-domain, we measure the per-domain utility V_C via a fresh Phase 1 fine-tune on its top-125 samples by v_{agg} (n=1 seed) and let Mixture-then-Select allocate proportional to V_C; Stratified-T=4 allocates uniformly (250 per macro-domain). MtS at T=4 retains a positive lead over Stratified, in the same direction as T=8 (Table 23); the gap shrinks from $+0.029$ to $+0.022$ but does not flip, indicating the domain-mixture lever is not specific to the eight-way partition.

Table 23: Sensitivity to domain count on Qwen3-8B with LoRA r=16, N=50K, and m=1K. Results use five paired seeds; headline MtS at T=8 uses 15 with augmented data. Four-domain calibration uses fresh per-domain fine-tunes on the top-125 samples by v_{agg}. Δ is relative to Stratified; reference cells are N/A.

Method	T	n	$u(S)$	Δ vs. Stratified
Mixture-then-Select	8	15	$0.6943_{\pm 0.014}$	$+\mathbf{0.0288}$ (p=0.041)
Stratified	8	5	$0.6662_{\pm 0.012}$	N/A
Mixture-then-Select-T=4	4	5	$0.6461_{\pm 0.016}$	$+\mathbf{0.0217}$ (p=0.132)
Stratified-T=4	4	5	$0.6245_{\pm 0.020}$	N/A

Llama-3.2-3B rank scaling. Replicating Figure 5 (A) on Llama-3.2-3B-Instruct at N=50K, m=1000: Mixture-then-Select achieves $\{0.314, 0.281, 0.243\}$ and Stratified achieves $\{0.250, 0.190, 0.133\}$ at $r \in \{8, 16, 32\}$ respectively (all n=5). The Δ grows monotonically from $+0.064$ (r=8) to $+0.091$ (r=16) to $+0.110$ (r=32), exactly matching the Qwen3-8B pattern of strictly increasing rank gap.

Aggregation-function ablation. The headline reports aggregate utility under linear (mean) aggregation. We re-aggregate the same per-domain training data under $f_{\min}(\cdot) = \min_t u_t$ (coverage-emphasising) and $f_{\text{geo}}(\cdot) = \exp(\frac{1}{T}\sum \log(1+u_t)) - 1$ (concave). On Llama-3.2-3B at m=1000: Mixture-then-Select aggregates to $+0.281 / +0.233 / +0.280$ under mean / min / geomean; Stratified to $+0.190 / +0.106 / +0.189$. The ordering is preserved, and under $f_{\min}$ the Mixture-then-Select advantage widens to Δ=$+0.127$ because Stratified's uniform allocation has a bad worst-case domain. Mixture-then-Select is the only method whose worst domain remains positive on Llama-3.2-3B; every published sample-level method has at least one domain catastrophically harmed (their min over 8 domains reaches -0.45).

Per-domain utility on Qwen3-8B. Table 24 disaggregates the headline aggregate (N=50K, m=1000, mean over 5 to 10 seeds) into the eight skill domains. The pattern is clean: Mixture-then-Select has the highest per-domain utility on *every* domain, including the hardest (creative, $+0.582$ vs. next-best Stratified $+0.540$) and the easiest (math, $+0.766$ vs. next-best $+0.736$); the win is not concentrated on any single domain. The ~ 0.07 aggregate gap therefore reflects per-domain *depth*, not just averaging luck.

Table 24: Per-domain relative loss reduction u_c on Qwen3-8B at N=50K, m=1K, averaged over seeds. Per-method seed counts follow Table 2; the mean column gives aggregate utility.

Method	math	code	QA	dlg	rsn	nlp	cpx	crv	**agg**
Mixture-then-Select	$+\mathbf{0.766}$	$+\mathbf{0.732}$	$+\mathbf{0.717}$	$+\mathbf{0.703}$	$+\mathbf{0.692}$	$+\mathbf{0.651}$	$+\mathbf{0.698}$	$+\mathbf{0.582}$	$+\mathbf{0.693}$
Stratified	0.736	0.653	0.622	0.610	0.662	0.510	0.629	0.540	0.620
Greedy (oracle)	0.646	0.609	0.611	0.601	0.632	0.517	0.641	0.524	0.598
Greedy + EL2N	0.687	0.655	0.622	0.599	0.610	0.563	0.593	0.507	0.605
LESS	0.675	0.617	0.607	0.577	0.611	0.462	0.601	0.485	0.579
Superfiltering	0.658	0.615	0.595	0.547	0.565	0.466	0.565	0.423	0.554

Weighted-mean sensitivity. The headline $u(S) = \frac{1}{T}\sum_c u_c(S)$ uses uniform weights. Table 25 re-aggregates the same per-domain data under non-uniform weights that emphasise individual skills 3× (math, code, reasoning, complex). The MtS advantage is preserved across all five weighting schemes, the method's win does not depend on the uniform-weight assumption.

B.11 Frozen-selection transfer to a larger model

Frozen-subset transfer to 72B. We fine-tune Qwen2-72B using the exact submitted Qwen3-8B selection, without recalibrating or reselecting samples (Table 26). The larger model obtains higher MMLU, GSM8K and ARC-Challenge scores, while IFEval is lower; unlike the instruction-aligned

Table 25: Aggregate utility under five task-weighting schemes on Qwen3-8B at $N{=}50K$, $m{=}1K$. A scheme labeled x $3\times$ gives domain x weight 3 and each other domain weight 1, then normalizes the weights to sum to 8.

Method	Uniform	math $3\times$	code $3\times$	reason $3\times$	complex $3\times$
Mixture-then-Select	$\mathbf{+0.693}$	$\mathbf{+0.707}$	$\mathbf{+0.700}$	$\mathbf{+0.693}$	$\mathbf{+0.694}$
Stratified	0.620	0.643	0.627	0.628	0.622
Greedy (oracle)	0.598	0.608	0.600	0.605	0.606
Greedy + EL2N	0.605	0.621	0.615	0.606	0.602
LESS	0.579	0.598	0.587	0.586	0.584
Superfiltering	0.554	0.575	0.566	0.556	0.556

8B model, the 72B model is a base model. This experiment tests whether the selected subset can be reused at a larger model scale, not whether MtS beats other selection methods on 72B.

Table 26: Benchmark evaluation of the frozen Qwen3-8B selection on Qwen2-72B, without recalibration or reselection. The models differ in both size and instruction alignment.

Model	IFEval	MMLU	GSM8K	ARC-C
Qwen3-8B	0.4894	0.7061	0.7973	0.5660
Qwen2-72B base	0.3777	0.8376	0.8764	0.6348

B.12 LLAMA-3.2-3B NATIVE RUN: RESOLVES THE SELECTION-TRANSFER TIE

Setup and results. Transferring the Qwen-derived selection to Llama-3.2-3B gives a MtS-Stratified gap of -0.030 ($p{=}0.45$, $n{=}5$). Recalibrating per-sample values on Llama while retaining the Qwen-derived mixture gives a gap of $+0.091$ ($p{=}0.0067$, paired, $n{=}5$). Table 27 reports means and variability; Table 28 reports domain-level results for seed 42.

Table 27: Utility on Llama-3.2-3B with LoRA $r{=}16$ and $m{=}1K$. Entries are means $\pm$ standard deviations over five seeds (†: ten seeds). Utility differences and paired tests use Stratified as the reference.

Method	**Mean**	**Std**	**Δ vs. Strat**	p **(paired)**
Mixture-then-Select (Qwen π^* + Llama v_{agg})	$\mathbf{+0.281}$	0.035	$\mathbf{+0.091}$	$\mathbf{0.0067}$
Stratified (Llama v_{agg} within-domain)	0.190	0.051	N/A	N/A
Random (Qwen-selection-transfer)	0.076	0.056	-0.114	0.012
Facility Location (transfer)	-0.016	0.051	-0.206	< 0.001
DEITA (transfer)†	-0.109	0.089	-0.299	< 0.001
k-DPP (transfer)	-0.101	0.038	-0.291	< 0.001
Superfiltering (transfer)†	-0.179	0.063	-0.369	< 0.001
Greedy-by-Value (transfer)	-0.203	0.089	-0.394	< 0.001
CRAIG (transfer)	-0.204	0.067	-0.395	< 0.001
LESS (transfer)	-0.238	0.090	-0.428	< 0.001
EL2N (transfer)	-0.244	0.055	-0.434	< 0.001

MtS retains Qwen's mixture with Llama-recalibrated rankings; Stratified uses Llama rankings. Other rows transfer Qwen-selected subsets.

B.13 CAPACITY-GATED DOMAIN MIXTURE: DATA POOL-N AND RANK BEHAVIOR

The following results cover additional pool-size and adapter-rank settings. The headline calibration uses Qwen3-8B with LoRA rank 16; the rank for each additional setting is reported below.

Data Pool-N gradient.

The table reports MtS and Stratified at three pool-size/rank settings. Because the ranks differ across rows, this comparison does not isolate pool size.

These pool-size comparisons use ranks 8 and 16. Cross-model transfer is reported separately in Appendix B.12.

Table 28: Per-domain utility on Llama-3.2-3B for seed 42. Utility is $u_c = (L_c^{\text{base}} - L_c^S)/L_c^{\text{base}}$, where the two losses are measured on held-out domain-c data before and after fine-tuning on subset S.

Method	math	code	gen. qa	dialog	reason.	nlp	cmplx	creative	**agg**
Mixture-then-Select	0.325	0.228	0.284	0.206	0.295	0.244	0.218	0.210	**+0.251**
Stratified	0.080	0.173	0.176	0.126	0.168	0.177	0.158	0.158	0.152
LESS	−0.325	−0.112	−0.156	−0.184	−0.141	−0.137	−0.168	−0.120	−0.168
EL2N	−0.306	−0.188	−0.297	−0.254	−0.207	−0.252	−0.191	−0.187	−0.235

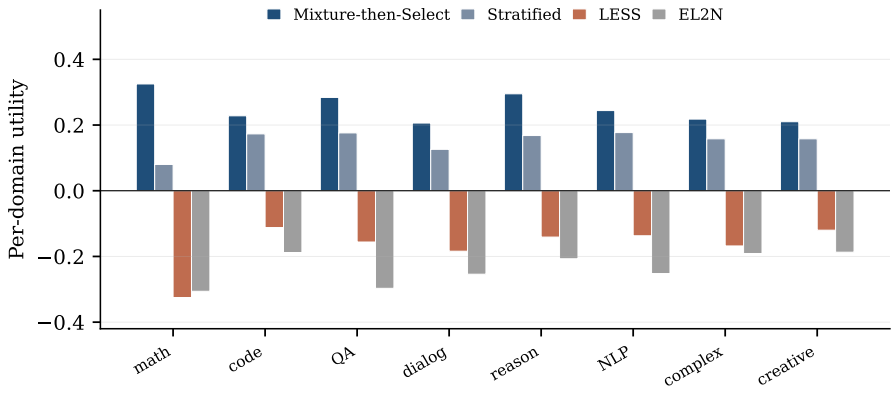

Figure 6: Per-domain utility on Llama-3.2-3B at N=50K, m=1K, for seed 42. Numerical values are in Table 28.

B.14 Bundle-level data pricing: setup, per-instance numbers, and negative results

This section gives the full setup, per-instance results, and ablations behind Table 5 in Section 4.4.

Per-instance results. Table 31 reports all 30 cells. Mean utility is 0.240 for joint-aware MtS and 0.160 for additive LP. The reference oracle refines only the top 50 single-seed candidates; MtS exceeds this restricted reference in two cells.

Within-cluster scoring controls. We tested replacing each solo-utility curve by its cumulative maximum and capping each cluster's allocation at its solo-utility maximizer. Neither changed additive LP's selection in any of the 30 cells. Increasing budget does not consistently increase its measured joint-training utility.

C Broader impact

Positive impacts. Interaction-aware data selection can reduce waste in LLM fine-tuning by helping practitioners assemble smaller, higher-value training subsets. The framework may be especially useful for resource-constrained groups that cannot afford exhaustive trial-and-error over large data pools.

Negative impacts. Interaction-aware selection could amplify existing dataset biases if certain domains or styles are repeatedly judged redundant and filtered out. If redundancy and complementarity correlate with demographic or linguistic attributes, optimizing only for measured utility may under-select already underrepresented data.

Mitigations. The matroid and multi-objective extensions can encode diversity or coverage constraints directly. More broadly, the framework makes interactions explicit, which can help practitioners audit whether seemingly efficient selections are systematically excluding important data types.

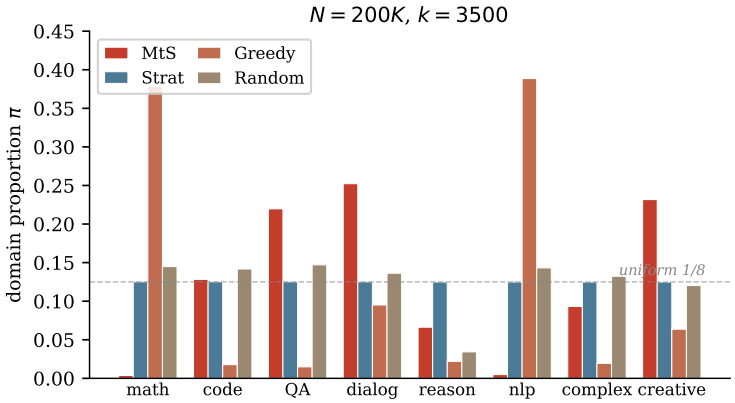

Figure 7: Selected domain proportions on Qwen3-8B at $N{=}200K$, $m{=}3.5K$. The dashed line shows uniform allocation $(1/8)$; the corresponding 50K comparison is in Figure 2A.

Table 29: MtS's utility difference from Stratified across pool sizes. The 50K condition uses LoRA $r{=}16$; 100K and 200K use $r{=}8$, so pool size and rank vary together.

N	LoRA r	Δ(MG − Strat)	paired p	n
50K	16	+0.099	<0.001	10
100K	8	+0.036	0.062	5
200K	8	+0.053	0.001	5

Table 30: Bundle-pricing settings and comparison definitions.

Setting	**Specification**
Pool	Math, code, general QA; five vendors per cluster, 50 samples per indivisible bundle (750 samples total).
Prices / budgets	One price per cluster, $p_c \sim \mathcal{U}[0.5, 1.5]$ independently; budgets 15%, 20%, 25% of $\sum_c 5p_c$; ten price/utility shuffles shared across methods.
Training	Llama-3.2-3B-Instruct, LoRA rank 16, three epochs, learning rate 2×10^{-4}; five training seeds.
Candidate allocations	All feasible cluster counts $n_c \in \{0, \ldots, 5\}$, selecting the top n_c vendors by individual utility within each cluster.
Additive LP	Maximizes $\sum_c V_c(n_c)$, where V_c is five-seed held-out utility from training on that cluster alone.
Joint-aware MtS	Scores the same feasible allocations by measured joint-training utility.
Oracle reference	All vendor combinations feasible at 25% budget screened at seed 42; top 50 refined with four additional seeds. Best five-seed mean among refined candidates feasible at each budget.

Table 31: Per-instance data-pricing results on Llama-3.2-3B-Instruct with LoRA r=16. Entries are means $\pm$ standard deviations over five seeds, covering 10 instances at each of three budgets.

inst	$B\%$	**Add. LP**	**MtS** (joint-aware)	**Oracle**	Δ **MtS**$-$**Add**	**Recovery**
0	15	0.172	$\mathbf{0.201 \pm 0.014}$	0.229	+0.029	50.9%
0	20	0.139	$\mathbf{0.235 \pm 0.009}$	0.236	+0.096	99.7%
0	25	0.140	$\mathbf{0.240 \pm 0.022}$	0.240	+0.100	100%
1	15	0.180	$\mathbf{0.215 \pm 0.015}$	0.234	+0.034	64.0%
1	20	0.136	$\mathbf{0.215 \pm 0.015}$	0.234	+0.079	80.3%
1	25	0.136	$\mathbf{0.247 \pm 0.014}$	0.247	+0.112	100%
2	15	0.233	$\mathbf{0.255 \pm 0.019}$	0.246	+0.022	$>$ 100% †
2	20	0.173	$\mathbf{0.255 \pm 0.021}$	0.263	+0.082	91.2%
2	25	0.159	$\mathbf{0.262 \pm 0.018}$	0.268	+0.103	94.2%
3	15	0.184	$\mathbf{0.240 \pm 0.014}$	0.247	+0.056	89.3%
3	20	0.141	$\mathbf{0.240 \pm 0.014}$	0.247	+0.099	93.6%
3	25	0.141	$\mathbf{0.245 \pm 0.014}$	0.247	+0.103	98.1%
4	15	0.178	$\mathbf{0.248 \pm 0.014}$	0.248	+0.070	100%
4	20	0.138	$\mathbf{0.248 \pm 0.014}$	0.248	+0.109	100%
4	25	0.089	$\mathbf{0.253 \pm 0.013}$	0.253	+0.164	100%
5	15	0.220	$\mathbf{0.253 \pm 0.017}$	0.253	+0.034	$>$ 100% †
5	20	0.185	$\mathbf{0.258 \pm 0.021}$	0.265	+0.073	91.1%
5	25	0.185	$\mathbf{0.258 \pm 0.021}$	0.265	+0.073	91.1%
6	15	0.162	$\mathbf{0.190 \pm 0.019}$	0.190	+0.028	100%
6	20	0.101	$\mathbf{0.249 \pm 0.016}$	0.249	+0.149	100%
6	25	0.116	$\mathbf{0.254 \pm 0.015}$	0.254	+0.138	100%
7	15	0.233	$\mathbf{0.248 \pm 0.010}$	0.248	+0.015	100%
7	20	0.157	$\mathbf{0.248 \pm 0.010}$	0.248	+0.091	100%
7	25	0.157	$\mathbf{0.248 \pm 0.010}$	0.248	+0.091	100%
8	15	0.196	$\mathbf{0.213 \pm 0.011}$	0.229	+0.017	52.5%
8	20	0.140	$\mathbf{0.244 \pm 0.010}$	0.244	+0.104	99.9%
8	25	0.124	$\mathbf{0.251 \pm 0.017}$	0.251	+0.127	100%
9	15	0.209	$\mathbf{0.230 \pm 0.006}$	0.230	+0.021	100%
9	20	0.153	$\mathbf{0.230 \pm 0.006}$	0.230	+0.077	100%
9	25	0.126	$\mathbf{0.230 \pm 0.011}$	0.230	+0.105	100%
Mean across 30 *cells*		0.160	**0.240**	0.244	+0.080	93.2%

† MtS exceeds Stage-B oracle in these cells (Stage B's top-50 truncation excluded the bundle MtS happened to pick; its 5-seed mean turned out higher).