HARP: EFFICIENT DATA SELECTION FOR FINETUNING LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Finetuning data selection requires balancing two competing goals: selecting examples that improve the downstream objective, and doing so without repeatedly finetuning models. Train-free selectors are scalable but rely on proxies such as embedding similarity or clustering, which may not match the target objective. Train-based selectors better reflect downstream utility through gradient signals, subset evaluation, or Shapley attribution, but require many costly train–evaluate iterations. We propose **H**ierarchical **A**ctive **R**egion **P**runing (**HARP**), an efficient train-based selector that preserves downstream alignment while reducing selection cost. HARP organizes the training pool into a node–leaf hierarchy, evaluates only representative leaves, and infers unmeasured utilities with empirical Bayes posteriors. It then selects data using two complementary envelopes: **HARP-C**, which conservatively controls redundancy, and **HARP-E**, which additively rewards complementary regions. We theoretically show that, under local smoothness and bounded estimation error, HARP controls selection error while reducing train–evaluate cost. Across 3 base models, 3 finetuning datasets and 4 reasoning benchmarks (36 settings), HARP variants take the per-cell best among ranked methods on 23 cells and achieve 36-setting means of 64.2 / 63.5, compared with 59.3 for the strongest baseline, NICE. HARP-C trains on roughly $9\times$ fewer examples than the budgeted baselines and $75\times$ fewer than full finetuning, combining higher mean accuracy with a smaller mean training-set size than every ranked baseline.

1 INTRODUCTION

Finetuning (Zhang et al., 2024; Chung et al., 2024; Longpre et al., 2023; Wang et al., 2023; Dettmers et al., 2023; Hu et al., 2022) has become an effective approach for specializing foundation large language models (LLMs) to follow task specifications, improve downstream performance, and adapt to domain-specific needs. Recent studies (Zhou et al., 2023a) show that a relatively small subset of high-quality data can match or even surpass the performance achieved using the full dataset. This suggests that current finetuning datasets are often noisy and inefficient, containing redundancy, near-duplicate instances, low-signal examples, and distributional mismatches with the target evaluation setting. Nevertheless, building high-quality finetuning datasets requires extensive expert labor, making it impractical at scale. Recent works are primarily based on train-free data selection and train-heavy evaluation-based data selection. Train-free selection methods (Ge et al., 2024; Liu et al., 2024; Yang et al., 2025b; Zhou et al., 2023b; Xie et al., 2023) typically use heuristics or representations (e.g., embeddings, clustering, diversity/coverage objectives, or quality proxies) to choose subsets without repeatedly finetuning candidate subsets. These methods are cheap and can be applied before training. However, their proxies can be misaligned with the actual downstream objective. For example, some methods construct a training subset by ranking training examples by embedding similarity to target validation examples. When the downstream objective is task accuracy, high embedding similarity fails to guarantee high marginal utility for improving the downstream objective. Train-based data evaluation explicitly estimates the contribution of training examples (or groups of examples) to a downstream objective via influence/gradient-based approximations Xia et al. (2024); Wang et al. (2025), or Shapley-style marginal contribution estimates He et al. (2024). While objective-faithful, these approaches require substantial computational cost, driven by the finetuning and full-set evaluation costs per cycle and the total number of train–evaluate iterations. This motivates a middle ground: a selector that retains the downstream alignment of train-based evaluation, but

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.

avoids exhaustively evaluating many candidates. In this paper, we propose **H**ierarchical **A**ctive **R**egion **P**runing (**HARP**), an efficient training-based data selection framework for LLM finetuning. HARP organizes the training set into a node–leaf hierarchy, where each leaf contains a group of examples, and each node contains a group of related leaves. Instead of evaluating every leaf, HARP estimates the downstream utility of a small set of representative leaves and infers the utility of unmeasured leaves through empirical Bayes posteriors. Building on these estimates, we introduce two complementary selection envelopes: **HARP-C** and **HARP-E**. HARP-C uses a **C**onservative coverage envelope to avoid over-counting utility under redundancy, while HARP-E uses an **E**xpansive additive envelope to credit multiple complementary leaves within the same domain. We theoretically show that, under local smoothness and bounded estimation error, HARP provides stable utility estimates and controls selection error while substantially reducing the train-evaluate iterations.

We evaluate HARP across 3 base models, 3 fine-tuning datasets, 8 baselines and 4 reasoning benchmarks (36 settings). HARP variants achieve the best result among ranked methods in $\mathbf{23}$ settings (64%). HARP-C and HARP-E reach mean accuracies of 64.2 and 63.5, outperforming the strongest baseline, NICE, by $+4.9$ and $+4.2$ points, respectively. HARP-C uses roughly $9\times$ fewer training examples than the 10k-budget baselines and $75\times$ fewer than full finetuning, highlighting the effectiveness of efficient data selection for finetuning large language models.

2 Related Work

Train-free data selection methods for finetuning. In contrast, *train-free* methods select subsets using inexpensive *proxy signals* that avoid repeatedly finetuning candidate subsets. A common strategy is *geometry-driven selection*, where methods use embeddings together with clustering, submodular objectives, or diversity sampling to retain representative and non-redundant examples (e.g., Ge et al. (2024); Liu et al. (2024); Bukharin et al. (2024); Yang et al. (2025b)). Another strategy uses *quality proxies*, such as heuristic filters, lightweight model-based scoring, or proxy difficulty/utility estimates, to remove noisy or low-value examples before training; recent LLM-specific instances include self-guided difficulty-based selection, one-shot utility scoring, instruction mining, automatic quality/complexity scoring, and weak-to-strong filtering (e.g., Li et al. (2024b;c); Cao et al. (2023); Liu et al. (2023); Li et al. (2024a)). In the adjacent setting of retrieval-augmented generation, Deng et al. (2025) select inference-time contexts using predicted contextual influence. Other data curation methods synthesize finetuning data through text-grounded task design (e.g., Chen et al. (2023)). These methods scale well to large finetuning corpora, but their proxy objectives can be misaligned with the downstream metric and generally fail to capture *interaction effects* that emerge only under actual training dynamics.

Train-based (objective-aligned) data selection methods for finetuning. A complementary line of work studies *objective-aligned* data selection, where the value of a training example (or group) is defined by its estimated effect on downstream performance after finetuning. Earlier training-based approaches explore several forms of model-in-the-loop selection, including transferred Shapley-value-based data valuation (Schoch et al., 2023), iterative self-evolving subset refinement (Wu et al., 2023), and active task selection based on prompt sensitivity (Kung et al., 2023). He et al. (2024) further moves closer to the true "fine-tune then evaluate" objective by using a Shapley-style refinement framework for instruction finetuning, but this comes with substantial computational overhead. To reduce the cost of repeated finetuning, Xia et al. (2024) proposes an influence-based approximation that builds a reusable gradient datastore and selects data using optimizer-aware gradient similarity to a small target set, enabling targeted capability induction with only a small fraction of the data. More recently, Wang et al. (2025) (NICE) extends this direction to *non-differentiable* evaluation metrics via policy-gradient-style influence estimation, bringing data selection closer to the true task objective when loss-based proxies are misaligned.

Finetuning of large language models. Supervised finetuning (SFT) adapts pretrained LLMs to follow natural-language instructions using instruction–response data, and has become a standard post-training stage for improving instruction following and generalization. Prior work shows that both scale and mixture design matter: large instruction collections and carefully balanced task mixtures can substantially improve transfer, while parameter-efficient methods reduce the cost of adaptation (Zhang et al., 2024; Chung et al., 2024; Longpre et al., 2023; Hu et al., 2022; Dettmers et al., 2023; Zhang et al., 2023; Xu et al., 2023). At the data-construction layer, synthetic pipelines bootstrap

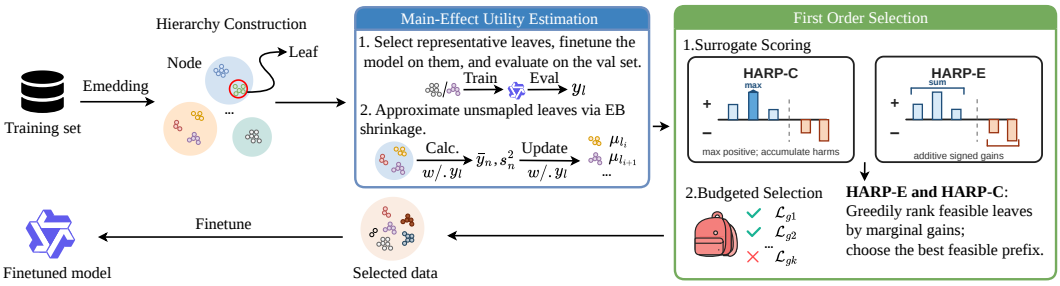

Figure 1: An overview of HARP.

supervision from seed tasks, teacher models, or reverse/task-grounded generation, including Self-Instruct, Stanford Alpaca, GPT-4-based instruction generation, and LongForm, while AlpaGasus filters existing instruction data (Wang et al., 2023; Taori et al., 2023; Peng et al., 2023; Köksal et al., 2024; Chen et al., 2024). Adjacent post-training work further studies preference-based optimization and scaled feedback data, while domain-specific tuning pipelines reinforce that supervision design and data composition are often as important as raw scale Dubois et al. (2023); Rafailov et al. (2023); Cui et al. (2023); Yue et al. (2023); Yu et al. (2023); Luo et al. (2023). Conversely, LIMA shows that a small but carefully curated set can still be highly effective, underscoring SFT's sensitivity to redundancy and example quality Zhou et al. (2023a).

3 Method

3.1 Problem Setup and Overview

Let $\mathcal{D} = \{x_i\}_{i=1}^{N}$ denote the training dataset. Our goal is to select a subset of training examples with total size at most B that maximizes downstream performance. Let $U(A)$ denote the task-averaged downstream utility (e.g., accuracy) after fine-tuning on $A \subseteq \mathcal{D}$. The ideal objective is $\max_{A\subseteq\mathcal{D},\,|A|\le B} U(A)$. However, this ideal objective is not directly tractable in practice. Evaluating a candidate subset requires a train–evaluate iteration, and the number of budget-feasible subsets grows combinatorially with the size of the training data. HARP makes this problem tractable. First, it organizes the training data into a hierarchy and selects leaves rather than individual examples, reducing the number of candidate units that must be measured. Once the hierarchy below defines leaves $\{\mathcal{L}_g\}_{g=1}^{G}$, each selected leaf set $S \subseteq [G]$ corresponds to $A(S) = \bigcup_{g\in S} \mathcal{L}_g$, with cost $c(S) = \sum_{g\in S} |\mathcal{L}_g|$, and we write $U(S) = U(A(S))$. HARP optimizes the leaf-restricted target $\max_{S\subseteq[G],\, c(S)\le B} U(S)$, which is a restriction of the example-level objective above. This leaf-level representation lets us score the leaf-restricted budgeted objective with a first-order utility approximation: HARP estimates the contribution of each leaf and then aggregates these main effects to score candidate selections. Second, we introduce two complementary domain-wise envelopes. **HARP-C** uses a conservative coverage envelope, crediting only the strongest positive leaf in each domain while accumulating all estimated harms; this limits over-counting when several leaves capture the same domain. **HARP-E** uses an expansive additive envelope, summing positive contributions across leaves so that multiple complementary leaves can receive credit within the same domain.

Let $\mathcal{E} = \bigcup_{d=1}^{D} \mathcal{E}_d$ be the fixed full validation set used for selection, partitioned into nonempty domains, and let $n_d = |\mathcal{E}_d|$. Every measured model is scored on all examples in each $\mathcal{E}_d$. For a model f_A fine-tuned on training subset A, the domain utility and base-model reference are

$$u_d(A) = \frac{1}{n_d} \sum_{e\in\mathcal{E}_d} m(f_A(e), y_e), \qquad v_0(d) = \frac{1}{n_d} \sum_{e\in\mathcal{E}_d} m(f_0(e), y_e),$$

where f_0 is the base model and the normalized metric m lies in $[0, 1]$. The domain partition and evaluation examples remain fixed throughout selection.

3.2 HIERARCHICAL NODE–LEAF CONSTRUCTION

We further reduce the number of train–evaluate iterations by treating leaves as the atomic selection units instead of individual examples. Let $\hat{z}_i \in \mathbb{R}^p$ denote the normalized embedding of training example x_i. HARP builds a two-level hierarchy over the training pool: node groups provide coarse local regions for sharing utility estimates, and leaves $\{\mathcal{L}_g\}_{g=1}^{G}$ define the atomic units used for selection. For any subset $Q \subseteq \mathcal{D}$ and a specified target number M_Q of child groups, we use a coverage-oriented anchor partitioning procedure. The first anchor is chosen as the example closest to the centroid of Q. Subsequent anchors are selected by the farthest-first rule $a_r = \arg\max_{x_i \in Q} \min_{a \in A_{r-1}} (1 - \hat{z}_i^\top \hat{z}_a)$, where A_{r-1} is the set of previously selected anchors. Each example in Q is then assigned to its closest anchor by cosine similarity. This procedure is first used to form node groups and then applied within each node to produce leaves, recursively splitting groups that exceed the maximum leaf size $C_{\max}$ and merging leaves with fewer than $C_{\min}$ examples into the nearest valid sibling by centroid similarity. The resulting leaves cover diverse regions of the training pool while keeping each selection unit large enough for stable utility estimation. The minimum size is a construction target; the bound below assumes that the final leaves satisfy it. As an immediate upper bound, replacing individual examples with leaves reduces exhaustive utility estimation from N individual-example queries to at most G leaf-level queries, with $G \leq N/C_{\min}$ whenever the final leaves satisfy the minimum-size constraint. Appendix B.2 formalizes this iteration-reduction bound.

3.3 MAIN-EFFECT ESTIMATION

We then estimate the utility of each leaf. Let $p(g)$ denote the parent node of leaf g, and let $\mathcal{P}_p$ denote the leaves under node p. Each leaf is represented by its mean embedding $\bar{z}_g = |\mathcal{L}_g|^{-1} \sum_{x_i \in \mathcal{L}_g} \hat{z}_i$. For domain d, define the centered main effect $\phi_g(d) = u_d(\mathcal{L}_g) - v_0(d)$, where $u_d(\mathcal{L}_g)$ is the domain utility after fine-tuning on $\mathcal{L}_g$, and $v_0(d)$ is the base-model domain utility. Since evaluating all G leaves is expensive, we measure only representative leaves $\mathcal{R}_p \subseteq \mathcal{P}_p$ with $|\mathcal{R}_p| \geq 1$ for every nonempty parent node p, selected by size stratification and farthest-first sampling over $\bar{z}_g$. For each measured representative r, we observe $y_r(d) = u_d(\mathcal{L}_r) - v_0(d)$, the noisy measured main effect. For an unmeasured leaf g, we first form a local estimate $\widetilde{y}_g(d)$ from measured representatives in the same node: $\widetilde{y}_g(d) = \sum_{r \in \mathcal{R}_{p(g)}} \alpha_{g,r} y_r(d)$. Here $\alpha_{g,r}$ is the similarity weight assigned from representative r to leaf g, defined by $\alpha_{g,r} = \kappa(\bar{z}_g, \bar{z}_r) / \sum_{r' \in \mathcal{R}_{p(g)}} \kappa(\bar{z}_g, \bar{z}_{r'})$, where $\kappa(\bar{z}_g, \bar{z}_r) = \exp(((\bar{z}_g/\|\bar{z}_g\|_2)^\top (\bar{z}_r/\|\bar{z}_r\|_2))/\lambda)$ is a cosine-similarity kernel computed from normalized leaf mean embeddings and $\lambda > 0$ controls locality. The interpolation step gives a local representative-based estimate of $\phi_g(d)$, with its approximation error controlled separately by the representative interpolation bound. We then apply empirical-Bayes shrinkage: $\widehat{\phi}_g(d) = \rho_g(d)\widetilde{y}_g(d) + (1 - \rho_g(d))\mu_0(d)$. Here $\widehat{\phi}_g(d)$ is the final estimated main effect, $\mu_0(d)$ is the global measured-effect mean, and $\rho_g(d) = \tau^2(d)/(\tau^2(d) + \sigma^2_{p(g)}(d)/n_{\text{eff}}(g))$ is the shrinkage weight. The terms $\tau^2(d)$ and $\sigma^2_{p(g)}(d)$ are the across-node and within-node variances, respectively, and $n_{\text{eff}}(g) = 1/\sum_{r \in \mathcal{R}_{p(g)}} \alpha^2_{g,r}$ is the effective number of representatives supporting leaf g. For directly measured leaves, we set $\widehat{\phi}_g(d) = y_g(d)$. This yields the domain-wise main-effect matrix $\widehat{\Phi} = [\widehat{\phi}_g(d)]_{g \in [G], d \in [D]}$. We keep active domains $\mathcal{A}_{\text{dom}} = \{d : \max_g |\widehat{\phi}_g(d)| > \epsilon_{\text{dom}}\}$. If $\mathcal{A}_{\text{dom}} = \varnothing$, we set $\mathcal{A}_{\text{dom}} = [D]$. We then normalize their weights w_d, write $b_d = v_0(d)$, and fix $\mathcal{A}_{\text{dom}}$, w_d, and b_d before defining both true and estimated surrogate objectives. This keeps domains with either estimated gains or estimated harms active, which is necessary because the conservative objective below accumulates negative effects rather than ignoring them. We justify the interpolation error, shrinkage estimator, and train–evaluate iteration reduction in Appendices B.3, B.4, and B.5.

3.4 HARP-C: CONSERVATIVE COVERAGE ENVELOPE

An additive first-order objective can over-count leaves: if multiple leaves improve the same domain, summing their positive effects may predict unrealistically large gains even when domain performance saturates. We therefore use a conservative first-order envelope. For each leaf g and active domain d, write $\widehat{\phi}_g^+(d) = \max(\widehat{\phi}_g(d), 0)$ and $\widehat{\phi}_g^-(d) = \max(-\widehat{\phi}_g(d), 0)$. Given a selected leaf set S, we predict the conservative domain utility as $\widehat{y}_d^{\text{C}}(S) = \text{clip}_{[0,1]}(b_d + \max_{g \in S} \widehat{\phi}_g^+(d) - \sum_{g \in S} \widehat{\phi}_g^-(d))$,

with the convention $\max_{g\in\varnothing}\widehat{\phi}_g^{+}(d)=0$, and define $\widehat{U}_{\mathrm{C}}(S)=\sum_{d\in\mathcal{A}_{\mathrm{dom}}} w_d\widehat{y}_d^{\mathrm{C}}(S)$. This envelope credits only the strongest positive leaf in each domain while accumulating all estimated harms. It therefore favors leaves whose gains remain useful under a conservative view of within-domain redundancy. HARP-C ranks leaves greedily by marginal gain under $\widehat{U}_{\mathrm{C}}$. Let $c_g=|\mathcal{L}_g|$ be the cost of leaf g. Starting from $S_0=\varnothing$, at step k we define the feasible candidate set $\mathcal{F}_k=\{g\notin S_k:\ c(S_k)+c_g\le B\}$. If $\mathcal{F}_k=\varnothing$, the greedy construction stops. Otherwise, we choose $g_{k+1}=\arg\max_{g\in\mathcal{F}_k}\left[\widehat{U}_{\mathrm{C}}(S_k\cup\{g\})-\widehat{U}_{\mathrm{C}}(S_k)\right]$, and set $S_{k+1}=S_k\cup\{g_{k+1}\}$, producing a ranked sequence of feasible leaf prefixes. The final HARP-C selection is the best feasible prefix, $\widehat{S}_{\mathrm{C}}=S_{k^\star},\quad k^\star=\arg\max_{k:\,c(S_k)\le B}\widehat{U}_{\mathrm{C}}(S_k)$, where $c(S_k)=\sum_{g\in S_k}c_g$. The budget value B is set in the experimental protocol.

Theorem 1 (First-order stability of HARP-C)**.** *Let U_{C} be the conservative coverage first-order utility defined with the true main effects $\phi_g(d)$, and let $\widehat{U}_{\mathrm{C}}$ be its estimated version defined with $\widehat{\phi}_g(d)$. Let $S^\star\in\arg\max_{c(S)\le B}U(S)$, and let $\widehat{S}_{\mathrm{C}}$ be the set returned by HARP-C. Suppose that, for every feasible set S, $|U(S)-U_{\mathrm{C}}(S)|\le\epsilon_{\mathrm{C}}$ and $|\widehat{U}_{\mathrm{C}}(S)-U_{\mathrm{C}}(S)|\le\delta_{\mathrm{C}}$. If the greedy procedure returns an ϵ_{opt}-approximate maximizer of $\widehat{U}_{\mathrm{C}}$ under the budget, then $U(S^\star)-U(\widehat{S}_{\mathrm{C}})\le 2\epsilon_{\mathrm{C}}+2\delta_{\mathrm{C}}+\epsilon_{\mathrm{opt}}$. Moreover, if every feasible set contains at most K leaves, the active-domain set is fixed, the active-domain weights sum to one, b_d and w_d are fixed, and $|\widehat{\phi}_g(d)-\phi_g(d)|\le\eta$ for all g,d, then $\delta_{\mathrm{C}}\le(K+1)\eta$.*

Theorem 1 follows from the shared first-order stability template in Theorem 3 and the HARP-C sensitivity bound in Lemma 5. The result shows that HARP-C is stable when the conservative coverage envelope is a good surrogate for the true subset utility, the estimated main effects are accurate, and the greedy optimization gap is small.

3.5 HARP-E: Expansive Additive Envelope

HARP-C is conservative because each domain can receive positive credit from a single selected leaf. This protects against redundant gains, but it can under-credit genuinely complementary leaves. For example, different leaves may improve the same domain, in which case taking only the maximum positive effect discards the useful additional signal. HARP-E addresses this case with an expansive additive envelope that sums positive contributions while still accumulating estimated harms. For each leaf g and active domain d, the predicted expansive domain utility for a selected leaf set S is $\widehat{y}_d^{\mathrm{E}}(S)=\mathrm{clip}_{[0,1]}(b_d+\sum_{g\in S}\widehat{\phi}_g(d))$, which is equivalent to summing positive effects and subtracting estimated harms. The aggregated objective is $\widehat{U}_{\mathrm{E}}(S)=\sum_{d\in\mathcal{A}_{\mathrm{dom}}}w_d\widehat{y}_d^{\mathrm{E}}(S)$. HARP-E uses the same feasible set, greedy update, and best-prefix rule as HARP-C, replacing $\widehat{U}_{\mathrm{C}}$ with $\widehat{U}_{\mathrm{E}}$. The two envelopes are complementary: HARP-C limits over-counting under redundancy, while HARP-E can recover multiple positive contributions when leaves within the same domain provide complementary information.

Theorem 2 (First-order stability of HARP-E)**.** *Let U_{E} be the expansive additive first-order utility defined with the true main effects $\phi_g(d)$, and let $\widehat{U}_{\mathrm{E}}$ be its estimated version defined with $\widehat{\phi}_g(d)$. Let $S^\star\in\arg\max_{c(S)\le B}U(S)$, and let $\widehat{S}_{\mathrm{E}}$ be the set returned by HARP-E. Suppose that, for every feasible set S, $|U(S)-U_{\mathrm{E}}(S)|\le\epsilon_{\mathrm{E}}$ and $|\widehat{U}_{\mathrm{E}}(S)-U_{\mathrm{E}}(S)|\le\delta_{\mathrm{E}}$. If the greedy procedure returns an ϵ_{opt}-approximate maximizer of $\widehat{U}_{\mathrm{E}}$ under the budget, then $U(S^\star)-U(\widehat{S}_{\mathrm{E}})\le 2\epsilon_{\mathrm{E}}+2\delta_{\mathrm{E}}+\epsilon_{\mathrm{opt}}$. Moreover, if every feasible set contains at most K leaves, the active-domain set is fixed, the active-domain weights sum to one, b_d and w_d are fixed, and $|\widehat{\phi}_g(d)-\phi_g(d)|\le\eta$ for all g,d, then $\delta_{\mathrm{E}}\le K\eta$.*

Theorem 2 follows from the same shared stability template as HARP-C, using the HARP-E sensitivity bound in Lemma 5. Since HARP-E is additive in the signed main effects, $\widehat{\phi}_g^{+}(d)-\widehat{\phi}_g^{-}(d)=\widehat{\phi}_g(d)$, its sensitivity to main-effect estimation error is bounded by $K\eta$ under the assumptions of the theorem.

3.6 Computational Complexity

Let $Q=\sum_p|\mathcal{R}_p|$ be the total number of measured representative leaves, $T_L(x,e)$ the cost of finetuning on x training examples for e epochs, and $T_{\mathrm{eval}}(|\mathcal{E}|)$ the cost of scoring one model on the

full validation set. With one epoch per measured leaf and three epochs for final finetuning in our experiments, main-effect estimation costs $\sum_p \sum_{r \in \mathcal{R}_p} T_L(|\mathcal{L}_r|, 1) + (Q+1)T_{\text{eval}}(|\mathcal{E}|)$, including one base-model evaluation. Each representative leaf contains at most $L = C_{\max}$ examples, so the training term is at most $QT_L(L, 1)$. Final finetuning costs $T_L(c(\widehat{S}), 3)$, where $c(\widehat{S}) = |A(\widehat{S})|$ counts selected training examples. HARP reduces the number of train–evaluate iterations; each measured iteration still evaluates the full set. Hierarchy construction is a one-time shared preprocessing cost, treated as a constant in the train–evaluate analysis. Details and baseline comparisons are in Appendix D.

4 Experimental Setup

Models. We evaluate three pre-trained base models: LLAMA-3.1-8B-BASE Grattafiori et al. (2024), QWEN3-4B-BASE and QWEN3-8B-BASE Yang et al. (2025a). All three are fine-tuned with LoRA; LoRA configuration, optimizer, and batch settings shared across HARP, and the baselines are in Appendix Table 3.

Training datasets. Three SFT sources spanning data quality and skill mix: SELF-INSTRUCT Wang et al. (2023) (SI; ∼82k examples, noisy auto-generated), WIZARDLM/EVOL-INSTRUCT-70K Xu et al. (2024) (Wiz; 70k, clean), and TULU-3-SFT-MIXTURE Lambert et al. (2024) (Tulu; 100k subset, mixed domains).

Validation datasets. We use the official MMLU (1,531 examples) and ARC-Challenge (299) validation splits, and hold out 500 training examples each from GSM8K and math12k for the GSM8K and MATH tasks, respectively.

Evaluation datasets. Final evaluation uses four reasoning benchmarks: MMLU Hendrycks et al. (2020) (57 subjects, 14,042 examples), ARC-CHALLENGE Clark et al. (2018) (1,172), GSM8K Cobbe et al. (2021) (1,319), and MATH-500 Hendrycks et al. (2021) (500). MMLU and ARC are scored by the first-token log-probabilities of answer-option letters; GSM8K and MATH use answer-string match on greedy vLLM generations.

Baselines. 0) Base: the untuned model without finetuning. 1) Random (Rand): uniform random selection of 10k examples. 2) DSIR Xie et al. (2023): importance sampling on n-gram features. 3) DQ Zhou et al. (2023b): dataset quantization with submodular bin selection. 4) Full Finetuning (Full FT): LoRA finetuning on the entire training dataset. 5) SHED-QWCS (SHED) and SHED-QOCS (SHED-O) He et al. (2024): two clustering-based Shapley variants. 6) LESS Xia et al. (2024): optimizer-aware influence estimation using low-rank gradient similarity. 7) NICE Wang et al. (2025): policy-gradient influence estimation targeting the downstream evaluation metric. All budgeted methods share a 10,000-example maximum: the baselines use the full budget, whereas HARP treats it as a cap and dynamically selects its best prefix.

Hyperparameters and reproducibility. For final finetuning across methods, we use LoRA (rank 16, $\alpha = 32$) with AdamW, a learning rate of 2×10^{-4}, an effective batch size of 16, and 3 epochs, averaging results over three seeds. Details are provided in Appendix Tables 3 and 4.

5 Experimental Results

In this paper, we first report main results in § 5.1, and then discuss the data efficiency in § 5.2, and finally show the ablation studies in § 5.3.

5.1 Main Results

Table 1 reports three-seed accuracy across 3 models × 3 training sets × 4 evaluation sets = 36 settings. Among the ranked methods, the HARP variants achieve the highest accuracy in **23** of the 36 settings. HARP-C is best in 13 settings and HARP-E in 11 compared with 10 for the strongest baseline NICE, 2 for SHED-O, and 1 for DSIR. Averaged across all settings HARP-C and HARP-E reach 64.2 and 63.5 respectively. Compared with the untuned Base model (50.0), HARP-C and HARP-E improve mean accuracy by 14.2 and 13.5 percentage points, respectively. HARP-C and HARP-E exceed the strongest baseline NICE by +4.9 and +4.2 points and SHED by +7.8 and +7.1 points. For example, on QWEN3-4B/SI/GSM8K, HARP-C achieves 75.1, compared with 32.6 for the

Table 1: Main results (accuracy%). **Bold**/underline: best/second-best among data selectors, excluding the Base, Random, and Full FT reference columns.

			Baselines								HARP	
Train	**Eval**	**Base**	**Random**	**DSIR**	**SHED**	**LESS**	**NICE**	**DQ**	**SHED-O**	**Full FT**	**HARP-C**	**HARP-E**
Llama-8B												
SI	MMLU	$60.1_{\pm 0.0}$	$58.2_{\pm 0.9}$	$56.1_{\pm 0.9}$	$55.3_{\pm 3.1}$	$47.6_{\pm 1.0}$	$62.5_{\pm 0.4}$	$8.3_{\pm 12.6}$	$45.5_{\pm 23.6}$	$49.1_{\pm 2.9}$	$\mathbf{63.2_{\pm 0.3}}$	$\mathbf{63.2_{\pm 0.2}}$
	ARC	$73.3_{\pm 0.2}$	$70.8_{\pm 2.1}$	$71.0_{\pm 0.6}$	$69.0_{\pm 3.7}$	$66.8_{\pm 2.7}$	$74.2_{\pm 1.2}$	$9.0_{\pm 10.5}$	$26.1_{\pm 40.0}$	$58.4_{\pm 4.1}$	$75.6_{\pm 2.4}$	$\mathbf{76.3_{\pm 0.9}}$
	GSM8K	$17.0_{\pm 0.7}$	$8.2_{\pm 1.5}$	$10.5_{\pm 0.8}$	$10.4_{\pm 0.6}$	$6.4_{\pm 0.8}$	$\mathbf{26.6_{\pm 1.0}}$	$9.3_{\pm 2.2}$	$9.4_{\pm 1.1}$	$6.7_{\pm 0.5}$	$23.2_{\pm 2.0}$	$20.6_{\pm 2.4}$
	MATH	$1.6_{\pm 0.3}$	$4.2_{\pm 0.5}$	$4.7_{\pm 0.4}$	$2.8_{\pm 0.5}$	$7.6_{\pm 2.0}$	$6.3_{\pm 0.9}$	$3.8_{\pm 1.2}$	$2.7_{\pm 0.6}$	$2.9_{\pm 0.4}$	$7.7_{\pm 1.2}$	$\mathbf{8.0_{\pm 1.1}}$
Wiz	MMLU	$60.1_{\pm 0.0}$	$62.8_{\pm 0.4}$	$61.6_{\pm 0.4}$	$61.7_{\pm 0.4}$	$62.7_{\pm 0.3}$	$62.2_{\pm 0.7}$	$62.5_{\pm 0.1}$	$62.9_{\pm 0.8}$	$59.5_{\pm 0.4}$	$\mathbf{64.5_{\pm 0.1}}$	$63.0_{\pm 0.2}$
	ARC	$73.3_{\pm 0.2}$	$77.0_{\pm 1.2}$	$76.4_{\pm 1.0}$	$76.9_{\pm 0.7}$	$76.3_{\pm 0.4}$	$77.2_{\pm 0.5}$	$74.4_{\pm 1.6}$	$74.4_{\pm 1.1}$	$72.2_{\pm 1.4}$	$\mathbf{77.9_{\pm 0.5}}$	$77.6_{\pm 0.4}$
	GSM8K	$17.0_{\pm 0.7}$	$49.2_{\pm 2.7}$	$46.4_{\pm 2.3}$	$50.2_{\pm 0.7}$	$42.6_{\pm 4.1}$	$49.1_{\pm 4.0}$	$48.5_{\pm 2.5}$	$\mathbf{51.4_{\pm 2.6}}$	$51.9_{\pm 1.8}$	$49.7_{\pm 2.4}$	$48.8_{\pm 1.5}$
	MATH	$1.6_{\pm 0.3}$	$10.3_{\pm 4.1}$	$10.8_{\pm 0.7}$	$10.6_{\pm 1.0}$	$10.0_{\pm 1.3}$	$10.4_{\pm 1.1}$	$10.7_{\pm 3.0}$	$11.0_{\pm 0.7}$	$9.5_{\pm 1.2}$	$8.5_{\pm 2.1}$	$\mathbf{12.3_{\pm 1.3}}$
Tulu	MMLU	$60.1_{\pm 0.0}$	$62.4_{\pm 0.2}$	$62.7_{\pm 0.5}$	$62.7_{\pm 0.0}$	$61.8_{\pm 1.0}$	$63.1_{\pm 0.6}$	$62.4_{\pm 0.7}$	$62.5_{\pm 1.2}$	$60.2_{\pm 0.5}$	$\mathbf{64.5_{\pm 0.2}}$	$63.8_{\pm 0.6}$
	ARC	$73.3_{\pm 0.2}$	$74.7_{\pm 1.5}$	$74.9_{\pm 0.2}$	$76.3_{\pm 1.3}$	$75.2_{\pm 0.5}$	$76.1_{\pm 0.7}$	$72.7_{\pm 1.9}$	$73.9_{\pm 0.9}$	$72.2_{\pm 1.2}$	$\mathbf{79.0_{\pm 0.4}}$	$76.1_{\pm 0.7}$
	GSM8K	$17.0_{\pm 0.7}$	$29.7_{\pm 10.9}$	$30.5_{\pm 21.2}$	$20.0_{\pm 6.3}$	$21.8_{\pm 16.1}$	$\mathbf{62.5_{\pm 3.0}}$	$29.3_{\pm 3.5}$	$38.8_{\pm 24.6}$	$57.2_{\pm 5.5}$	$57.9_{\pm 0.6}$	$58.4_{\pm 2.2}$
	MATH	$1.6_{\pm 0.3}$	$16.4_{\pm 2.1}$	$13.5_{\pm 1.1}$	$15.1_{\pm 1.9}$	$11.2_{\pm 0.6}$	$\mathbf{19.3_{\pm 1.8}}$	$15.1_{\pm 0.1}$	$15.8_{\pm 1.4}$	$14.3_{\pm 0.9}$	$17.4_{\pm 1.6}$	$17.4_{\pm 1.3}$
Qwen3-8B												
SI	MMLU	$62.4_{\pm 0.1}$	$70.3_{\pm 2.0}$	$68.2_{\pm 1.3}$	$68.7_{\pm 2.0}$	$66.2_{\pm 1.4}$	$71.4_{\pm 0.7}$	$29.1_{\pm 4.6}$	$64.0_{\pm 9.6}$	$64.8_{\pm 1.3}$	$\mathbf{75.4_{\pm 0.2}}$	$73.6_{\pm 0.7}$
	ARC	$72.9_{\pm 0.5}$	$88.3_{\pm 1.0}$	$87.0_{\pm 0.9}$	$86.5_{\pm 0.8}$	$86.0_{\pm 1.8}$	$\mathbf{92.8_{\pm 0.1}}$	$39.5_{\pm 16.7}$	$69.3_{\pm 18.3}$	$82.4_{\pm 1.4}$	$90.3_{\pm 0.4}$	$91.4_{\pm 0.8}$
	GSM8K	$75.2_{\pm 0.6}$	$12.1_{\pm 4.6}$	$8.5_{\pm 0.9}$	$9.6_{\pm 1.9}$	$8.3_{\pm 1.2}$	$16.8_{\pm 0.9}$	$12.3_{\pm 2.7}$	$12.5_{\pm 2.5}$	$8.8_{\pm 2.4}$	$\mathbf{76.2_{\pm 1.51}}$	$40.8_{\pm 1.6}$
	MATH	$34.3_{\pm 0.0}$	$10.8_{\pm 1.7}$	$7.6_{\pm 2.4}$	$10.5_{\pm 1.1}$	$6.1_{\pm 2.7}$	$5.8_{\pm 1.3}$	$11.6_{\pm 1.9}$	$10.6_{\pm 2.5}$	$7.0_{\pm 1.5}$	$35.6_{\pm 3.0}$	$\mathbf{35.8_{\pm 2.8}}$
Wiz	MMLU	$62.4_{\pm 0.1}$	$75.3_{\pm 0.8}$	$74.6_{\pm 0.6}$	$75.5_{\pm 0.5}$	$75.6_{\pm 0.4}$	$\mathbf{75.7_{\pm 0.5}}$	$68.2_{\pm 3.3}$	$71.5_{\pm 2.6}$	$74.0_{\pm 1.2}$	$75.4_{\pm 0.3}$	$74.9_{\pm 0.2}$
	ARC	$72.9_{\pm 0.5}$	$92.5_{\pm 0.4}$	$92.0_{\pm 1.0}$	$92.0_{\pm 0.7}$	$92.1_{\pm 0.4}$	$92.8_{\pm 0.2}$	$74.2_{\pm 14.2}$	$85.5_{\pm 3.1}$	$91.2_{\pm 0.9}$	$92.9_{\pm 0.3}$	$\mathbf{93.1_{\pm 0.4}}$
	GSM8K	$75.2_{\pm 0.6}$	$83.0_{\pm 5.9}$	$85.1_{\pm 0.1}$	$86.3_{\pm 1.5}$	$47.2_{\pm 9.2}$	$86.6_{\pm 0.3}$	$86.5_{\pm 0.8}$	$87.2_{\pm 1.2}$	$84.4_{\pm 1.5}$	$\mathbf{89.3_{\pm 1.8}}$	$85.2_{\pm 4.2}$
	MATH	$34.3_{\pm 0.0}$	$47.9_{\pm 1.0}$	$45.2_{\pm 1.7}$	$47.2_{\pm 1.4}$	$47.5_{\pm 0.5}$	$50.1_{\pm 1.0}$	$46.3_{\pm 1.7}$	$46.7_{\pm 1.6}$	$46.3_{\pm 1.0}$	$48.2_{\pm 0.2}$	$\mathbf{51.6_{\pm 1.2}}$
Tulu	MMLU	$62.4_{\pm 0.1}$	$75.9_{\pm 0.1}$	$75.1_{\pm 0.3}$	$75.8_{\pm 0.7}$	$74.9_{\pm 0.3}$	$\mathbf{75.9_{\pm 0.5}}$	$72.2_{\pm 2.5}$	$72.6_{\pm 2.5}$	$75.0_{\pm 0.4}$	$75.8_{\pm 0.0}$	$75.7_{\pm 0.3}$
	ARC	$72.9_{\pm 0.5}$	$92.4_{\pm 0.5}$	$\mathbf{92.9_{\pm 0.3}}$	$92.6_{\pm 0.4}$	$92.6_{\pm 0.4}$	$92.7_{\pm 0.3}$	$85.5_{\pm 4.4}$	$87.5_{\pm 3.8}$	$91.7_{\pm 0.4}$	$92.2_{\pm 0.2}$	$92.4_{\pm 0.2}$
	GSM8K	$75.2_{\pm 0.6}$	$59.8_{\pm 23.8}$	$74.4_{\pm 5.3}$	$63.4_{\pm 8.4}$	$47.8_{\pm 13.8}$	$70.5_{\pm 9.4}$	$50.2_{\pm 16.6}$	$46.6_{\pm 18.9}$	$61.0_{\pm 15.1}$	$74.3_{\pm 2.7}$	$\mathbf{76.1_{\pm 4.7}}$
	MATH	$34.3_{\pm 0.0}$	$52.9_{\pm 2.0}$	$47.1_{\pm 1.7}$	$52.0_{\pm 1.8}$	$40.9_{\pm 2.4}$	$54.2_{\pm 1.2}$	$51.8_{\pm 0.7}$	$49.7_{\pm 0.7}$	$48.2_{\pm 1.7}$	$43.7_{\pm 3.5}$	$\mathbf{54.4_{\pm 0.7}}$
Qwen3-4B												
SI	MMLU	$52.1_{\pm 0.2}$	$68.1_{\pm 0.5}$	$67.0_{\pm 1.1}$	$65.9_{\pm 0.9}$	$62.1_{\pm 1.7}$	$61.6_{\pm 1.7}$	$30.2_{\pm 17.7}$	$64.1_{\pm 4.3}$	$63.1_{\pm 1.0}$	$\mathbf{71.4_{\pm 0.2}}$	$70.1_{\pm 0.2}$
	ARC	$46.9_{\pm 1.0}$	$85.6_{\pm 0.5}$	$84.1_{\pm 0.7}$	$84.1_{\pm 0.9}$	$81.6_{\pm 0.8}$	$\mathbf{88.8_{\pm 0.3}}$	$29.9_{\pm 25.1}$	$81.9_{\pm 4.4}$	$80.8_{\pm 1.6}$	$87.5_{\pm 0.2}$	$87.6_{\pm 0.3}$
	GSM8K	$70.8_{\pm 0.1}$	$13.0_{\pm 4.1}$	$11.1_{\pm 2.0}$	$21.2_{\pm 5.2}$	$10.1_{\pm 1.2}$	$14.7_{\pm 3.2}$	$18.2_{\pm 5.4}$	$32.6_{\pm 18.4}$	$6.8_{\pm 0.8}$	$\mathbf{75.1_{\pm 4.9}}$	$75.0_{\pm 5.0}$
	MATH	$33.1_{\pm 0.8}$	$13.0_{\pm 0.7}$	$7.2_{\pm 0.7}$	$10.3_{\pm 0.4}$	$4.2_{\pm 2.4}$	$10.8_{\pm 0.7}$	$12.7_{\pm 2.2}$	$13.6_{\pm 1.3}$	$9.4_{\pm 1.5}$	$\mathbf{34.8_{\pm 3.6}}$	$34.6_{\pm 3.3}$
Wiz	MMLU	$52.1_{\pm 0.2}$	$72.2_{\pm 0.2}$	$71.7_{\pm 0.1}$	$72.2_{\pm 0.0}$	$72.0_{\pm 0.3}$	$72.2_{\pm 0.3}$	$53.0_{\pm 11.3}$	$54.6_{\pm 3.9}$	$71.1_{\pm 0.6}$	$\mathbf{72.7_{\pm 0.2}}$	$72.5_{\pm 0.2}$
	ARC	$46.9_{\pm 1.0}$	$88.3_{\pm 0.3}$	$89.0_{\pm 0.3}$	$88.6_{\pm 0.6}$	$88.7_{\pm 0.3}$	$89.0_{\pm 0.2}$	$61.2_{\pm 19.3}$	$54.6_{\pm 10.2}$	$88.1_{\pm 0.1}$	$88.3_{\pm 0.1}$	$\mathbf{89.2_{\pm 0.2}}$
	GSM8K	$70.8_{\pm 0.1}$	$84.3_{\pm 2.7}$	$82.8_{\pm 3.4}$	$85.7_{\pm 1.5}$	$84.2_{\pm 0.7}$	$87.4_{\pm 0.2}$	$85.3_{\pm 0.6}$	$87.7_{\pm 1.0}$	$80.2_{\pm 4.1}$	$88.3_{\pm 0.4}$	$\mathbf{89.6_{\pm 1.1}}$
	MATH	$33.1_{\pm 0.8}$	$43.8_{\pm 0.8}$	$42.9_{\pm 0.2}$	$44.3_{\pm 1.4}$	$43.3_{\pm 1.8}$	$44.4_{\pm 1.8}$	$43.3_{\pm 1.2}$	$43.7_{\pm 1.4}$	$42.8_{\pm 1.5}$	$\mathbf{49.1_{\pm 0.8}}$	$47.4_{\pm 1.6}$
Tulu	MMLU	$52.1_{\pm 0.2}$	$72.2_{\pm 0.2}$	$71.0_{\pm 0.8}$	$72.2_{\pm 0.2}$	$72.0_{\pm 0.3}$	$\mathbf{72.5_{\pm 0.3}}$	$71.5_{\pm 0.7}$	$64.6_{\pm 5.3}$	$70.6_{\pm 0.5}$	$72.2_{\pm 0.2}$	$72.2_{\pm 0.1}$
	ARC	$46.9_{\pm 1.0}$	$88.3_{\pm 0.4}$	$87.7_{\pm 0.1}$	$88.3_{\pm 0.3}$	$88.1_{\pm 0.2}$	$\mathbf{89.1_{\pm 0.3}}$	$86.4_{\pm 0.4}$	$85.5_{\pm 1.6}$	$87.9_{\pm 0.8}$	$88.0_{\pm 0.2}$	$88.6_{\pm 0.2}$
	GSM8K	$70.8_{\pm 0.1}$	$81.6_{\pm 4.5}$	$79.3_{\pm 2.4}$	$79.0_{\pm 2.4}$	$34.0_{\pm 7.7}$	$81.6_{\pm 2.7}$	$75.7_{\pm 4.0}$	$\mathbf{83.0_{\pm 2.1}}$	$74.9_{\pm 3.7}$	$79.6_{\pm 2.1}$	$82.4_{\pm 6.0}$
	MATH	$33.1_{\pm 0.8}$	$45.8_{\pm 1.2}$	$44.3_{\pm 0.2}$	$48.7_{\pm 1.7}$	$43.4_{\pm 1.2}$	$\mathbf{49.6_{\pm 4.0}}$	$44.5_{\pm 0.8}$	$44.8_{\pm 2.6}$	$44.7_{\pm 2.8}$	$44.3_{\pm 1.4}$	$47.9_{\pm 0.6}$
Mean (36 settings)		50.0	56.7	56.0	56.4	51.6	59.3	45.7	52.5	54.7	**64.2**	63.5

strongest ranked baseline, SHED-O. On QWEN3-8B/SI/GSM8K, it reaches 76.2 compared with 16.8 for NICE. HARP also performs well beyond SI: on Wiz, HARP-C and HARP-E both achieve a mean accuracy of 67.1, compared with 66.4 for NICE; on Tulu, HARP-E nearly matches NICE (67.1 versus 67.3). This is because HARP's hierarchy divides the candidate pool into small local leaf groups, and interpolates the measured main effects of representative leaves to obtain local estimates for unmeasured leaves within each parent node. This guides the ranking to favor positively contributing leaves over weakly contributing or potentially harmful ones. Such selectivity may be particularly beneficial on SI where the candidate pool may contain many irrelevant or misleading examples.

HARP-C achieves higher mean accuracy on SI (59.7 versus 56.4) whereas HARP-E performs better on Tulu (67.1 versus 65.7). The two variants are nearly tied on Wiz (both 67.1 after rounding). This pattern is consistent with how their envelopes account for positive contributions. Both accumulate estimated harms but HARP-C credits only the strongest positive leaf within each domain. Its conservative coverage envelope therefore limits additional credit for leaves with overlapping benefits. HARP-E instead sums signed main effects across leaves through its expansive additive envelope. This preserves credit for complementary contributions within the same domain that HARP-C may omit but can also overcount gains when those contributions are redundant. These results suggest using HARP-C when controlling redundancy and keeping the selected subset compact are priorities and HARP-E when retaining multiple complementary contributions within the same domain is more important.

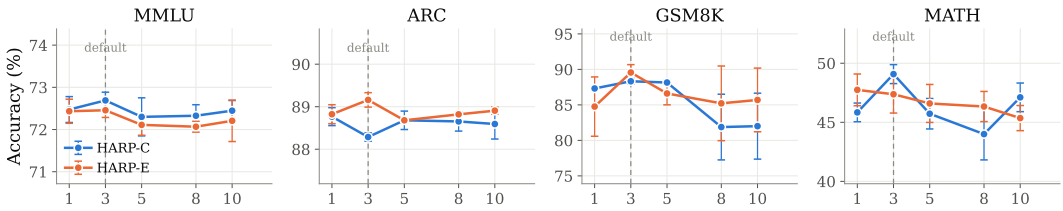

Figure 3: Effect of representatives per node R.

5.2 Data Efficiency

Figure 2 compares mean accuracy and training set size, using median selected sizes for the HARP markers. All budgeted selection methods share the same maximum budget of 10,000 examples. The baselines use this fixed budget, whereas HARP treats it as a cap and dynamically selects a smaller prefix when its estimated utility peaks earlier. HARP-C achieves the highest mean accuracy with the fewest training examples, using a median of 1,087 examples compared with 4,106 for HARP-E, 10,000 for the budgeted baselines, and 81,674 for full fine-tuning. Across settings, HARP-C selects roughly 125–5,467 examples, while HARP-E ranges from about 442 to the full 10,000-example budget. Despite using the full pool, Full FT reaches only 54.7 mean accuracy, trailing HARP-C and HARP-E by 9.5 and 8.8 points, respectively. This highlights the importance of selecting useful training data. HARP obtains its smaller subsets through the best prefix rule, which dynamically selects under the same 10,000-example maximum budget rather than using a smaller preset budget. Leaves are ranked greedily by marginal gain under the corresponding envelope, and the prefix with the highest estimated utility is selected. Under HARP-C, once a domain is already covered by a strong positive leaf, additional leaves in the same domain receive no further positive credit unless they provide a stronger contribution, while estimated harms continue to accumulate. Under HARP-E, contributions from multiple leaves are added, so additional leaves can still improve estimated utility when they provide complementary positive effects. Thus, either envelope may reach its highest estimated utility before the 10,000 example cap and select a smaller subset. This occurs more often for HARP-C because redundant positive contributions receive no additional credit. This provides a plausible explanation for the observed combination of compact subsets and strong downstream performance, further highlighting the importance of efficient data selection.

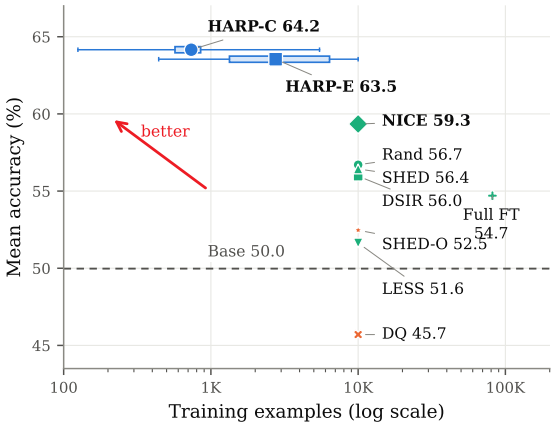

Figure 2: Mean accuracy versus median number of training examples.

5.3 Ablation Studies

In this section, we ablate the number of representatives per node R, leaf size L, and hierarchy partitioning on Qwen3-4B-Base with WizardLM, reporting mean accuracy and standard deviation over three seeds.

Effectiveness of representatives per node R. We vary $R \in \{1, 3, 5, 8, 10\}$ (Figure 3). Larger R directly measures more representative leaves but increases train–evaluate cost (Appendix D). We find that MMLU and ARC remain within 0.5 points of the $R = 3$ setting for both variants, while larger R does not consistently improve GSM8K or MATH. The strong performance at small R supports the effectiveness of our estimation procedure. Interpolation transfers local information from measured representatives within the same parent node, while empirical-Bayes shrinkage stabilizes the resulting estimates when representative support is limited. This allows HARP to preserve downstream

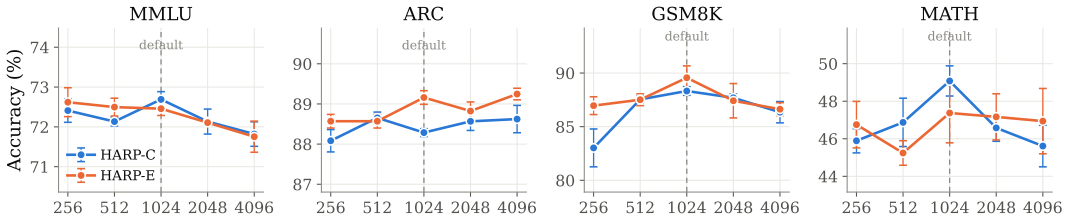

Figure 4: Effect of leaf size L.

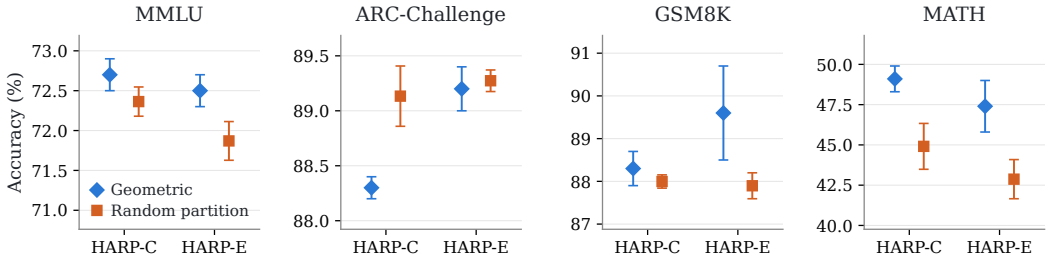

Figure 5: Random partition versus geometric grouping.

performance while directly measuring only a small number of leaves. We therefore use $R = 3$ as the default to balance accuracy and measurement cost.

Effectiveness of leaf size L. We vary $L \in \{256, 512, 1024, 2048, 4096\}$ (Figure 4). Both variants remain within 0.9 points of the $L = 1024$ setting on MMLU and ARC, while GSM8K and MATH are more sensitive. Increasing L from 256 to 1024 improves HARP-C from 83.0 to 88.3 on GSM8K and from 45.9 to 49.1 on MATH, with little additional gain at larger values. HARP-E also reaches its highest GSM8K and MATH accuracy at $L = 1024$. The broad stability across a $16\times$ range of L supports the robustness of HARP's hierarchy to leaf granularity. The coverage-oriented partitioning preserves diverse local regions, while the leaf-size constraints keep selection units sufficiently large for stable utility estimation. Very small leaves can therefore become less stable, while larger leaves make the selection coarser without providing consistent gains. We use $L = 1024$ as the default.

Effectiveness of geometric grouping. We compare geometric grouping with random partitioning on QWEN3-4B-BASE with WIZARDLM. Geometric grouping yields higher accuracy on MMLU, GSM8K, and MATH for both variants, while random partitioning is slightly better on ARC. On MATH, geometric grouping improves HARP-C from 44.9 to 49.1 and HARP-E from 42.9 to 47.4. Averaged across the four benchmarks and both variants, the gain is 1.3 points. Geometric grouping keeps related leaves local, improving representative interpolation and producing coherent selection units with clearer gain and harm estimates. Random partitioning mixes heterogeneous regions, weakening both utility estimation and greedy ranking.

6 CONCLUSION

We propose HARP, a hierarchical data selection method with two complementary envelopes, HARP-C for conservative pruning and HARP-E for broader coverage. Across 36 settings, HARP achieves the best result among ranked methods in 23 settings. HARP-C and HARP-E reach mean accuracies of 64.2 and 63.5, respectively, compared with 59.3 for NICE and 54.7 for full finetuning. HARP-C uses roughly $9\times$ fewer training examples than the budgeted baselines and $75\times$ fewer than full finetuning. The two envelopes show complementary behavior. HARP-C is strongest on SI, with mean accuracy 59.7, while HARP-E performs better on Tulu and the two variants are nearly tied on Wiz. Overall, these results show that HARP can identify compact training subsets while preserving strong downstream performance across different data regimes.

AI USE STATEMENT

We used AI assistants during the preparation of this work for limited coding and writing support. In particular, we use Claude Code for parts of the implementation, including code drafting, debugging, and refactoring. We also used GPT/OpenAI tools to improve the clarity, grammar, and presentation of the manuscript. All research ideas, method design, experimental decisions, result analysis, and final manuscript content were reviewed and verified by the authors. The authors take full responsibility for the correctness, originality, and integrity of the submitted work.

REFERENCES

Alexander Bukharin, Shiyang Li, Zhengyang Wang, Jingfeng Yang, Bing Yin, Xian Li, Chao Zhang, Tuo Zhao, and Haoming Jiang. Data diversity matters for robust instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3411–3425, 2024.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. Instruction mining: Instruction data selection for tuning large language models. *arXiv preprint arXiv:2307.06290*, 2023.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. Alpagasus: Training a better alpaca with fewer data. In *International Conference on Learning Representations*, volume 2024, pp. 34767–34797, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/9543942c237ded1b39b1fd37259ff88e-Paper-Conference.pdf.

Yongrui Chen, Haiyun Jiang, Xinting Huang, Shuming Shi, and Guilin Qi. TeGit: Generating high-quality instruction-tuning data with text-grounded task design. *arXiv preprint arXiv:2309.05447v1*, 2023. URL https://arxiv.org/abs/2309.05447v1.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

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

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

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*, 2023.

Jiale Deng, Yanyan Shen, Ziyuan Pei, Youmin Chen, and Linpeng Huang. Influence guided context selection for effective retrieval-augmented generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069, 2023.

Yuan Ge, Yilun Liu, Chi Hu, Weibin Meng, Shimin Tao, Xiaofeng Zhao, Mahong Xia, Zhang Li, Boxing Chen, Hao Yang, Bei Li, Tong Xiao, and JingBo Zhu. Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 464–478, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.28.

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

Yexiao He, Ziyao Wang, Zheyu Shen, Guoheng Sun, Yucong Dai, Yongkai Wu, Hongyi Wang, and Ang Li. Shed: shapley-based automated dataset refinement for instruction fine-tuning. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *ArXiv*, abs/2103.03874, 2021. URL https://arxiv.org/abs/2103.03874.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. Longform: Effective instruction tuning with reverse instructions. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7056–7078, 2024.

Po-Nien Kung, Fan Yin, Di Wu, Kai-Wei Chang, and Nanyun Peng. Active instruction tuning: Improving cross-task generalization by training on prompt sensitive tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1813–1829, 2023.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14255–14273, 2024a.

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

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Ling-Hao Chen, Junhao Liu, Tongliang Liu, et al. One-shot learning as instruction data prospector for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4586–4601, 2024c.

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

Zifan Liu, Amin Karbasi, and Theodoros Rekatsinas. Tsds: data selection for task-specific model finetuning. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

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

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Stephanie Schoch, Ritwick Mishra, and Yangfeng Ji. Data selection for fine-tuning large language models using transferred shapley values. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 266–275, 2023.

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.

Jingtan Wang, Xiaoqiang Lin, Rui Qiao, Pang Wei Koh, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. NICE data selection for instruction tuning in LLMs with non-differentiable evaluation metric. In *Forty-second International Conference on Machine Learning*, 2025.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754.

Shengguang Wu, Keming Lu, Benfeng Xu, Junyang Lin, Qi Su, and Chang Zhou. Self-evolved diverse data sampling for efficient instruction tuning. *arXiv preprint arXiv:2311.08182*, 2023.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36: 34201–34227, 2023.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *International Conference on Learning Representations*, volume 2024, pp. 30745–30766, 2024. URL `https://proceedings.iclr.cc/paper_files/paper/2024/file/82eec786fdfbbfa53450c5feb7d1ac92-Paper-Conference.pdf`.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 6268–6278, 2023.

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

Yuming Yang, Yang Nan, Junjie Ye, Shihan Dou, Xiao Wang, Shuo Li, Huijie Lv, Tao Gui, Qi Zhang, and Xuanjing Huang. Measuring data diversity for instruction tuning: A systematic analysis and a reliable metric. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18530–18549, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.908.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023.

Zhengxin Zhang, Dan Zhao, Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Qing Li, Yong Jiang, and Zhihao Jia. Quantized side tuning: Fast and memory-efficient tuning of quantized large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1–17, 2024.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023a.

Daquan Zhou, Kai Wang, Jianyang Gu, Xiangyu Peng, Dongze Lian, Yifan Zhang, Yang You, and Jiashi Feng. Dataset quantization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17205–17216, 2023b.

A Artifact Use Consistent With Intended Use

We use pretrained models, finetuning datasets, evaluation benchmarks, and baseline implementations described in Section 4 only for research purposes. Our use is limited to controlled data-selection, finetuning, and benchmark evaluation experiments, consistent with the intended research use of these artifacts. Artifacts produced by this work, including selected training subsets, hierarchy partitions, utility estimates, and code, are intended for research on efficient finetuning data selection, and any derived artifacts should be used or redistributed only when permitted by the licenses and access terms of the original datasets, models, and benchmarks.

B Proof

B.1 Universal Notation Table

Table 2 summarizes the notation used in the method and appendix. The same symbols are used throughout unless a local proof explicitly introduces a temporary dummy variable.

Table 2: Universal notation table for HARP.

Symbol	Meaning
Training data and full-set validation	
$\mathcal{D} = \{x_i\}_{i=1}^{N}$	Candidate training dataset; x_i is one training example and N is the number of candidates.
$\mathcal{E}$, $\mathcal{E}_d$, D	Full validation set and its domain-d subset, and the number of validation domains. The index $d \in [D]$ ranges over validation domains.
n_d	Number of examples in the full validation domain $\mathcal{E}_d$.
$\hat{z}_i$, $\bar{z}_g$	Normalized training-example embedding and mean embedding of leaf g.
$A(S)$, $c(S)$, c_g	Training subset induced by selected leaves, total selected leaf cost, and individual leaf cost: $A(S) = \bigcup_{g \in S} \mathcal{L}_g$, $c(S) = \sum_{g \in S} \lvert\mathcal{L}_g\rvert$, and $c_g = \lvert\mathcal{L}_g\rvert$.
B, K	Training-data budget and an upper bound on the number of leaves in any feasible set.
$C_{\min}$, $C_{\max}$	Minimum leaf-size target and maximum leaf-size cap.
$U(S)$, $u_d(S)$, $v_0(d)$, b_d	Task-averaged utility, domain utility, base-model domain utility, and shorthand $b_d = v_0(d)$.
$\mathcal{L}_g$, G, $p(g)$, $\mathcal{P}_p$	Leaf group, number of leaves, parent node of leaf g, and leaves under parent p.
$\mathcal{R}_p$, r_p, R, Q	Representative leaves in parent p, their count, per-parent representative cap, and total measured representative count $Q = \sum_p r_p$.
$\phi_g(d)$, $y_r(d)$, $\widetilde{y}_g(d)$, $\widehat{\phi}_g(d)$	True main effect, measured representative effect, interpolated estimate, and final estimated main effect.
$\alpha_{g,r}$, $n_{\text{eff}}(g)$, $\Delta_{g,p}$	Interpolation weight, effective representative count, and weighted representative distance.
$\mu_0(d)$, $\tau^2(d)$, $\sigma_p^2(d)$, $\rho_g(d)$	Global measured-effect mean, across-node variance, within-node variance, and empirical-Bayes shrinkage weight.
$\mathcal{A}_{\text{dom}}$, w_d, ϵ_{dom}	Active domains, active-domain weights, and active-domain threshold.
$\mathcal{F}_k$	Feasible candidate set used by greedy selection at step k: $\mathcal{F}_k = \{g \notin S_k : c(S_k) + c_g \le B\}$.
$y_d^{\mathrm{C}}(S)$, $\widehat{y}_d^{\mathrm{C}}(S)$	True and estimated clipped conservative domain utilities.
$U_{\mathrm{C}}(S)$, $\widehat{U}_{\mathrm{C}}(S)$	True and estimated conservative first-order utilities.
$y_d^{\mathrm{E}}(S)$, $\widehat{y}_d^{\mathrm{E}}(S)$	True and estimated clipped expansive additive domain utilities.
$U_{\mathrm{E}}(S)$, $\widehat{U}_{\mathrm{E}}(S)$	True and estimated expansive additive first-order utilities.
$q_d^{\mathrm{C}}(S)$, $q_d^{\mathrm{E}}(S)$	Raw unclipped conservative and expansive domain envelopes.
ϵ_{C}, δ_{C}, ϵ_{E}, δ_{E}, ϵ_{opt}	First-order surrogate errors, estimation errors, and optimization gap for HARP-C and HARP-E.

B.2 Iteration-Reduction Bound for Leaf-Level Estimation

The following lemma isolates the reduction obtained by moving from individual examples to leaf groups. It bounds the number of the utility queries that would be required if every final leaf were measured once; the representative-leaf estimation step in Section 3.3 further reduces this cost in practice. The upper leaf-size parameter $C_{\max}$ controls recursive splitting, while the lower bound $C_{\min}$ controls the counting argument below.

Lemma 1 (Leaf grouping reduces the utility-query count)**.** *Let $\{\mathcal{L}_g\}_{g=1}^{G}$ be the final leaf groups constructed by HARP. Suppose that the leaves form a partition of the candidate pool $\mathcal{D}$, where $|\mathcal{D}| = N$, and that each final leaf satisfies the minimum-size constraint $|\mathcal{L}_g| \geq C_{\min}$. Then*

$$G \leq \frac{N}{C_{\min}}.$$

Therefore, exhaustive utility evaluation at the leaf level requires at most $N/C_{\min}$ train–evaluate iterations, compared with N iterations for exhaustive individual-example singleton estimation.

Proof. Because the final leaf groups form a partition of $\mathcal{D}$, each training example belongs to exactly one leaf group, and therefore

$$N = \sum_{g=1}^{G} |\mathcal{L}_g|.$$

By the minimum-size constraint, $|\mathcal{L}_g| \geq C_{\min}$ for every g. Hence

$$N = \sum_{g=1}^{G} |\mathcal{L}_g| \geq \sum_{g=1}^{G} C_{\min} = GC_{\min}.$$

Rearranging gives $G \leq N/C_{\min}$. Thus, replacing individual examples with leaf groups reduces the number of possible singleton utility measurements from N to at most $N/C_{\min}$ before applying any additional representative-leaf subsampling. □

B.3 Representative Interpolation Error

All utilities in this analysis are defined on the fixed full validation domains of Section 3.1. The measurement-noise model below conditions on that fixed evaluation set; its zero-mean and independence conditions are assumptions about stochastic training or measurement, not consequences of the evaluation protocol.

For the interpolation weights used in Section 3.3, assume $\mathcal{R}_p \neq \varnothing$ for every nonempty parent node p. Define

$$\nu(\bar{z}) = \begin{cases} \bar{z}/\|\bar{z}\|_2, & \|\bar{z}\|_2 > 0, \\ 0, & \|\bar{z}\|_2 = 0, \end{cases}$$

and, for $\lambda > 0$,

$$\kappa(\bar{z}_g, \bar{z}_r) = \exp\left(\frac{\nu(\bar{z}_g)^\top \nu(\bar{z}_r)}{\lambda}\right).$$

Then

$$\alpha_{g,r} = \frac{\kappa(\bar{z}_g, \bar{z}_r)}{\sum_{r' \in \mathcal{R}_{p(g)}} \kappa(\bar{z}_g, \bar{z}_{r'})}$$

is well-defined, nonnegative, and satisfies $\sum_{r \in \mathcal{R}_{p(g)}} \alpha_{g,r} = 1$.

Lemma 2 (Representative interpolation under local smoothness)**.** *Fix an nonempty parent node p, a validation domain d, and a leaf $g \in \mathcal{P}_p$, with $\mathcal{R}_p \neq \varnothing$. Suppose the true domainwise main effect is locally smooth over leaf embeddings inside parent p: for any leaves $g, h \in \mathcal{P}_p$,*

$$|\phi_g(d) - \phi_h(d)| \leq L_{p,d} \|\bar{z}_g - \bar{z}_h\|_2.$$

For each measured representative $r \in \mathcal{R}_p$, suppose

$$\begin{aligned} y_r(d) &= \phi_r(d) + \varepsilon_r(d), \\ \mathbb{E}[\varepsilon_r(d)] &= 0, \\ \mathrm{Var}(\varepsilon_r(d)) &\leq \sigma_p^2(d), \end{aligned}$$

and suppose the noises are independent across representatives. For an unmeasured leaf g, define the local representative interpolation estimate by

$$\widetilde{y}_g(d) = \sum_{r \in \mathcal{R}_p} \alpha_{g,r} y_r(d),$$

$$\sum_{r \in \mathcal{R}_p} \alpha_{g,r} = 1,$$

$$\alpha_{g,r} \geq 0.$$

Let

$$\Delta_{g,p} = \sum_{r \in \mathcal{R}_p} \alpha_{g,r} \|\bar{z}_g - \bar{z}_r\|_2,$$

$$n_{\text{eff}}(g) = \frac{1}{\sum_{r \in \mathcal{R}_p} \alpha_{g,r}^2}.$$

Then the interpolation bias and variance satisfy

$$|\mathbb{E}[\widetilde{y}_g(d)] - \phi_g(d)| \leq L_{p,d} \Delta_{g,p},$$

and

$$\mathrm{Var}(\widetilde{y}_g(d)) \leq \frac{\sigma_p^2(d)}{n_{\text{eff}}(g)}.$$

Consequently,

$$\mathbb{E}\left[(\widetilde{y}_g(d) - \phi_g(d))^2\right] \leq L_{p,d}^2 \Delta_{g,p}^2 + \frac{\sigma_p^2(d)}{n_{\text{eff}}(g)}.$$

For a directly measured representative leaf, HARP sets $\widehat{\phi}_g(d) = y_g(d)$ instead of using interpolation; equivalently, one may take $\alpha_{g,g} = 1$, which makes $\Delta_{g,p} = 0$.

Proof. Using the definition of $\widetilde{y}_g(d)$ and the zero-mean noise assumption,

$$\mathbb{E}[\widetilde{y}_g(d)] = \sum_{r \in \mathcal{R}_p} \alpha_{g,r} \phi_r(d).$$

Since the weights are nonnegative and sum to one,

$$\begin{aligned} &|\mathbb{E}[\widetilde{y}_g(d)] - \phi_g(d)| \\ &\quad = \left| \sum_{r \in \mathcal{R}_p} \alpha_{g,r} (\phi_r(d) - \phi_g(d)) \right| \\ &\quad \leq \sum_{r \in \mathcal{R}_p} \alpha_{g,r} |\phi_r(d) - \phi_g(d)|. \end{aligned}$$

By local smoothness,

$$|\phi_r(d) - \phi_g(d)| \leq L_{p,d} \|\bar{z}_g - \bar{z}_r\|_2.$$

Combining the two inequalities gives

$$\begin{aligned} &|\mathbb{E}[\widetilde{y}_g(d)] - \phi_g(d)| \\ &\quad \leq L_{p,d} \sum_{r \in \mathcal{R}_p} \alpha_{g,r} \|\bar{z}_g - \bar{z}_r\|_2 \\ &\quad = L_{p,d} \Delta_{g,p}. \end{aligned}$$

For the variance, independence gives

$$\begin{aligned} \mathrm{Var}(\widetilde{y}_g(d)) &= \mathrm{Var}\left(\sum_{r \in \mathcal{R}_p} \alpha_{g,r} \varepsilon_r(d) \right) \\ &= \sum_{r \in \mathcal{R}_p} \alpha_{g,r}^2 \mathrm{Var}(\varepsilon_r(d)). \end{aligned}$$

Since $\mathrm{Var}(\varepsilon_r(d)) \le \sigma_p^2(d)$,

$$\mathrm{Var}(\widetilde{y}_g(d)) \le \sigma_p^2(d) \sum_{r \in \mathcal{R}_p} \alpha_{g,r}^2 = \frac{\sigma_p^2(d)}{n_{\mathrm{eff}}(g)}.$$

Finally, the mean-squared interpolation error decomposes into squared bias plus variance:

$$\mathbb{E}\left[(\widetilde{y}_g(d) - \phi_g(d))^2\right] = \left(\mathbb{E}[\widetilde{y}_g(d)] - \phi_g(d)\right)^2 + \mathrm{Var}(\widetilde{y}_g(d)).$$

Substituting the bias and variance bounds proves the stated MSE bound. The directly measured case follows by setting all interpolation mass on g, which eliminates the smoothness-bias term. □

B.4 Empirical-Bayes Shrinkage

Proposition 1 (Normal-normal shrinkage and risk reduction)**.** *Fix an unmeasured leaf g and domain d. Suppose the local representative estimate satisfies*

$$\widetilde{y}_g(d) \mid \theta_g(d) \sim \mathcal{N}(\theta_g(d), V_g(d)),$$

where $\theta_g(d) = \mathbb{E}[\widetilde{y}_g(d)]$ denotes the local interpolation target rather than necessarily the true main effect $\phi_g(d)$. By Lemma 2, this target satisfies $|\theta_g(d) - \phi_g(d)| \le L_{p(g),d}\Delta_{g,p(g)}$. where $V_g(d) = \sigma^2_{p(g)}(d)/n_{\mathrm{eff}}(g)$ and $V_g(d) > 0$, and suppose the latent local interpolation target has prior

$$\theta_g(d) \sim \mathcal{N}(\mu_0(d), \tau^2(d)) \quad \tau^2(d) > 0.$$

Then the posterior mean is

$$\widehat{\phi}_g(d) = \rho_g(d)\widetilde{y}_g(d) + (1 - \rho_g(d))\mu_0(d),$$
$$\rho_g(d) = \frac{\tau^2(d)}{\tau^2(d) + V_g(d)}.$$

Moreover, under squared error loss, this posterior mean is the Bayes estimator, and its integrated mean-squared error is

$$\mathbb{E}\left[(\widehat{\phi}_g(d) - \theta_g(d))^2\right] = \frac{\tau^2(d)V_g(d)}{\tau^2(d) + V_g(d)} \le V_g(d) = \mathbb{E}\left[(\widetilde{y}_g(d) - \theta_g(d))^2\right].$$

Consequently, when the later theory uses $\widehat{\phi}_g(d)$ as an estimator of the true main effect $\phi_g(d)$, the error bound must include both shrinkage risk and interpolation bias:

$$\mathbb{E}\left[(\widehat{\phi}_g(d) - \phi_g(d))^2\right] \le 2\frac{\tau^2(d)V_g(d)}{\tau^2(d) + V_g(d)} + 2L^2_{p(g),d}\Delta^2_{g,p(g)}.$$

For directly measured leaves, HARP sets $\widehat{\phi}_g(d) = y_g(d)$; this shrinkage proposition applies to unmeasured leaves estimated by interpolation.

Proof. For readability, suppress g, d and write $\widetilde{y}$, θ, μ_0, τ^2, and V. The likelihood and prior are

$$\widetilde{y} \mid \theta \sim \mathcal{N}(\theta, V), \theta \sim \mathcal{N}(\mu_0, \tau^2).$$

By conjugacy of the normal-normal model, the posterior precision is

$$\frac{1}{\tau^2} + \frac{1}{V},$$

so the posterior variance is

$$\left(\frac{1}{\tau^2}+\frac{1}{V}\right)^{-1}=\frac{\tau^2 V}{\tau^2+V}.$$

The posterior mean is

$$\frac{\frac{1}{V}\widetilde{y}+\frac{1}{\tau^2}\mu_0}{\frac{1}{V}+\frac{1}{\tau^2}}=\frac{\tau^2}{\tau^2+V}\widetilde{y}+\frac{V}{\tau^2+V}\mu_0.$$

Thus $\rho=\tau^2/(\tau^2+V)$, giving the estimator in the proposition.

Under squared error loss, the posterior mean minimizes posterior expected squared error, so it is the Bayes estimator. Its integrated Bayes risk equals the expected posterior variance:

$$\mathbb{E}\left[(\widehat{\phi}-\theta)^2\right]=\frac{\tau^2 V}{\tau^2+V}.$$

The unshrunk estimator $\widetilde{y}$ has integrated mean-squared error

$$\mathbb{E}\left[(\widetilde{y}-\theta)^2\right]=\mathbb{E}[\varepsilon^2]=V.$$

Since $\tau^2/(\tau^2+V)\le 1$,

$$\frac{\tau^2 V}{\tau^2+V}\le V.$$

Therefore the shrinkage estimator has no larger integrated squared error than the unshrunk local estimate. Finally, the true-effect bound follows from $(a+b)^2\le 2a^2+2b^2$. Since $\widehat{\phi}-\phi=(\widehat{\phi}-\theta)+(\theta-\phi)$ and $|\theta-\phi|\le L_{p(g),d}\Delta_{g,p(g)}$,

$$\begin{aligned}\mathbb{E}\left[(\widehat{\phi}-\phi)^2\right]&\le 2\mathbb{E}\left[(\widehat{\phi}-\theta)^2\right]+2(\theta-\phi)^2\\&\le 2\frac{\tau^2 V}{\tau^2+V}+2L_{p(g),d}^2\Delta_{g,p(g)}^2.\end{aligned}$$

This proves the additional true main-effect error bound stated in the proposition. □

Proposition 2 (MSE-to-uniform main-effect error event)**.** *Suppose that for every leaf-domain pair (g,d) there is a finite bound $M_{g,d}$ such that*

$$\mathbb{E}\left[(\widehat{\phi}_g(d)-\phi_g(d))^2\right]\le M_{g,d}.$$

For an unmeasured leaf, Proposition 1 gives one admissible choice

$$M_{g,d}=2\frac{\tau^2(d)V_g(d)}{\tau^2(d)+V_g(d)}+2L_{p(g),d}^2\Delta_{g,p(g)}^2.$$

For a directly measured leaf with $\widehat{\phi}_g(d)=y_g(d)=\phi_g(d)+\varepsilon_g(d)$, one may take any valid second-moment bound on $\varepsilon_g(d)$, for example $M_{g,d}\ge\mathbb{E}[\varepsilon_g(d)^2]$. Let

$$M_\Sigma=\sum_{g=1}^{G}\sum_{d=1}^{D}M_{g,d}.$$

Then, for any failure probability $\alpha\in(0,1)$, with

$$\eta_\alpha=\sqrt{\frac{M_\Sigma}{\alpha}},$$

we have

$$\Pr\left(\max_{g\in[G],\,d\in[D]}|\widehat{\phi}_g(d)-\phi_g(d)|\le\eta_\alpha\right)\ge 1-\alpha.$$

Consequently, the deterministic condition $|\widehat{\phi}_g(d)-\phi_g(d)|\le\eta$ used in the envelope sensitivity lemma can be read as conditioning on this uniform event, with $\eta=\eta_\alpha$.

Proof. Let

$$Z = \sum_{g=1}^{G} \sum_{d=1}^{D} (\widehat{\phi}_g(d) - \phi_g(d))^2.$$

By the assumed pairwise MSE bounds, $\mathbb{E}[Z] \le M_\Sigma$. If

$$\max_{g,d} |\widehat{\phi}_g(d) - \phi_g(d)| > \eta_\alpha,$$

then $Z > \eta_\alpha^2$. Therefore, by Markov's inequality,

$$\begin{aligned} &\Pr\left(\max_{g,d} |\widehat{\phi}_g(d) - \phi_g(d)| > \eta_\alpha \right) \\ &\le \Pr(Z > \eta_\alpha^2) \le \frac{\mathbb{E}[Z]}{\eta_\alpha^2} \\ &\le \frac{M_\Sigma}{M_\Sigma/\alpha} = \alpha. \end{aligned}$$

Taking complements proves the claim. □

B.5 Train–Evaluate Iteration Reduction from Representative Main-Effect Estimation

Lemma 3 (Representative main-effect estimation reduces train–evaluate iterations)**.** *Let $\{\mathcal{L}_g\}_{g=1}^{G}$ be the training leaves and $\{\mathcal{P}_p\}_{p=1}^{P}$ their parent partition. HARP measures $r_p = |\mathcal{R}_p|$ representatives in each nonempty parent, where*

$$1 \le r_p \le |\mathcal{P}_p|, \qquad \mathcal{R}_p \subseteq \mathcal{P}_p.$$

Set

$$Q = \sum_{p=1}^{P} r_p.$$

Main-effect estimation requires Q train–evaluate iterations instead of G, saving $G - Q$ iterations and reducing their count by a factor of G/Q. If $r_p \le R$ for every parent, then

$$Q \le PR.$$

Each iteration scores its model on the full fixed validation set.

Proof. Exhaustive singleton leaf evaluation fine-tunes one model per leaf and scores that model on every validation domain, requiring G iterations. HARP measures r_p leaves in parent p, for a total of

$$Q = \sum_{p=1}^{P} r_p.$$

Interpolation and empirical-Bayes shrinkage estimate the other $G - Q$ leaves without additional fine-tuning runs, giving the claimed saving and ratio. Finally,

$$Q = \sum_{p} r_p \le \sum_{p} R = PR.$$

The common base-model evaluation does not require a fine-tuning run and is excluded from both iteration counts. □

B.6 Theory for First-Order Envelope Selection

We prove a shared stability template for both HARP-C and HARP-E. The two variants differ only in how estimation error propagates through the envelope: HARP-C uses a maximum over positive effects and therefore has sensitivity $(K + 1)\eta$, while HARP-E is additive in signed main effects and has sensitivity $K\eta$.

The deterministic estimation condition used in Lemma 5 is either an explicit assumption or a high-probability event supplied by Proposition 2.

For each active domain d, define $\phi_g^+(d) = \max(\phi_g(d), 0)$ and $\phi_g^-(d) = \max(-\phi_g(d), 0)$. The raw HARP-C and HARP-E envelopes are

$$q_d^{\mathrm{C}}(S) = b_d + \max_{g \in S} \phi_g^+(d) - \sum_{g \in S} \phi_g^-(d),$$

with the convention $\max_{g \in \varnothing} \phi_g^+(d) = 0$, and

$$q_d^{\mathrm{E}}(S) = b_d + \sum_{g \in S} \phi_g^+(d) - \sum_{g \in S} \phi_g^-(d).$$

Equivalently, since $\phi_g^+(d) - \phi_g^-(d) = \phi_g(d)$,

$$q_d^{\mathrm{E}}(S) = b_d + \sum_{g \in S} \phi_g(d).$$

For $X \in \{\mathrm{C}, \mathrm{E}\}$, define the clipped domain utility $y_d^X(S) = \mathrm{clip}_{[0,1]}(q_d^X(S))$ and the aggregate first-order utility

$$U_X(S) = \sum_{d \in \mathcal{A}_{\mathrm{dom}}} w_d y_d^X(S).$$

The estimated utility $\widehat{U}_X(S)$ is defined analogously by replacing $\phi_g(d)$ with $\widehat{\phi}_g(d)$. In particular,

$$\widehat{q}_d^{\mathrm{E}}(S) = b_d + \sum_{g \in S} \widehat{\phi}_g(d).$$

Lemma 4 (Envelope clipping stability)**.** *Fix $X \in \{\mathrm{C}, \mathrm{E}\}$. For each active domain d, suppose the true domain utility $u_d(S) \in [0, 1]$ satisfies*

$$\left|u_d(S) - q_d^X(S)\right| \le \xi_d^X(S)$$

for every feasible set S. Let $U_{\mathcal{A}}(S) = \sum_{d \in \mathcal{A}_{\mathrm{dom}}} w_d u_d(S)$, using the same active-domain set and weights as U_X. Then

$$\left|u_d(S) - y_d^X(S)\right| \le \xi_d^X(S),$$

and therefore, for nonnegative active-domain weights,

$$|U_{\mathcal{A}}(S) - U_X(S)| \le \sum_{d \in \mathcal{A}_{\mathrm{dom}}} w_d \xi_d^X(S).$$

If the paper-level utility $U(S)$ is evaluated on the full downstream suite rather than only the active domains, then the difference between $U(S)$ and $U_{\mathcal{A}}(S)$ is absorbed into the approximation term ϵ_X in Theorem 3.

Proof. Fix d, S, and $X \in \{\mathrm{C}, \mathrm{E}\}$. Since $u_d(S) \in [0, 1]$, we have $u_d(S) = \mathrm{clip}_{[0,1]}(u_d(S))$. The clipping map is 1-Lipschitz, so

$$\begin{aligned}
&\left|u_d(S) - y_d^X(S)\right| \\
&= \left|\mathrm{clip}_{[0,1]}(u_d(S)) - \mathrm{clip}_{[0,1]}(q_d^X(S))\right| \\
&\le \left|u_d(S) - q_d^X(S)\right| \le \xi_d^X(S).
\end{aligned}$$

Multiplying by w_d, summing over active domains, and applying the triangle inequality gives

$$|U_{\mathcal{A}}(S) - U_X(S)| \le \sum_{d \in \mathcal{A}_{\mathrm{dom}}} w_d \xi_d^X(S).$$

□

Theorem 3 (Shared first-order stability template)**.** *Fix $X \in \{\mathrm{C}, \mathrm{E}\}$. Let U_X be the first-order utility defined with the true main effects $\phi_g(d)$, and let $\widehat{U}_X$ be its estimated version defined with $\widehat{\phi}_g(d)$. Let $S^\star \in \arg\max_{c(S)\le B} U(S)$, where the maximum is taken over feasible leaf sets, and let $\widehat{S}_X$ be the set returned by the corresponding HARP variant. Suppose that, for every feasible set S,*

$$|U(S) - U_X(S)| \le \epsilon_X$$

and

$$|\widehat{U}_X(S) - U_X(S)| \le \delta_X.$$

If the greedy procedure returns an ϵ_{opt}-approximate maximizer of $\widehat{U}_X$ under the budget, where ϵ_{opt} denotes its optimization gap, then

$$U(S^\star) - U(\widehat{S}_X) \le 2\epsilon_X + 2\delta_X + \epsilon_{\mathrm{opt}}.$$

Proof. By the approximation assumption,

$$U(S^\star) \le U_X(S^\star) + \epsilon_X.$$

By the estimation assumption,

$$U_X(S^\star) \le \widehat{U}_X(S^\star) + \delta_X.$$

Since $\widehat{S}_X$ is an ϵ_{opt}-approximate maximizer of $\widehat{U}_X$,

$$\widehat{U}_X(S^\star) \le \widehat{U}_X(\widehat{S}_X) + \epsilon_{\mathrm{opt}}.$$

Applying the estimation and approximation assumptions again,

$$\begin{aligned}\widehat{U}_X(\widehat{S}_X) &\le U_X(\widehat{S}_X) + \delta_X \\ &\le U(\widehat{S}_X) + \delta_X + \epsilon_X.\end{aligned}$$

Combining the four inequalities gives

$$U(S^\star) - U(\widehat{S}_X) \le 2\epsilon_X + 2\delta_X + \epsilon_{\mathrm{opt}}.$$

□

Lemma 5 (Envelope sensitivity to main-effect estimation error)**.** *Suppose every feasible set contains at most K leaves, the active-domain set $\mathcal{A}_{\mathrm{dom}}$ is fixed, the active-domain weights are nonnegative and weights sum to one, b_d and w_d are fixed, and $|\widehat{\phi}_g(d) - \phi_g(d)| \le \eta$ for all g, d. Then*

$$\delta_{\mathrm{C}} \le (K+1)\eta \qquad \textit{and} \qquad \delta_{\mathrm{E}} \le K\eta.$$

Proof. Since $x \mapsto \max(x, 0)$ and $x \mapsto \max(-x, 0)$ are both 1-Lipschitz, the assumption $|\widehat{\phi}_g(d) - \phi_g(d)| \le \eta$ implies

$$|\widehat{\phi}_g^+(d) - \phi_g^+(d)| \le \eta$$

and

$$|\widehat{\phi}_g^-(d) - \phi_g^-(d)| \le \eta.$$

First consider HARP-C. For any feasible set S with $|S| \le K$,

$$\left|\max_{g\in S} \widehat{\phi}_g^+(d) - \max_{g\in S} \phi_g^+(d)\right| \le \eta,$$

where the empty-set convention gives zero on both sides when $S = \varnothing$. For the negative part,

$$\left|\sum_{g\in S} \widehat{\phi}_g^-(d) - \sum_{g\in S} \phi_g^-(d)\right| \le K\eta.$$

Thus the raw HARP-C domain-envelope error is at most $(K+1)\eta$. Since clipping to $[0, 1]$ is 1-Lipschitz, the clipped domain-utility error is also at most $(K+1)\eta$. Summing over active domains with nonnegative weights that sum to one gives

$$|\widehat{U}_{\mathrm{C}}(S) - U_{\mathrm{C}}(S)| \le (K+1)\eta.$$

Hence $\delta_{\mathrm{C}} \le (K+1)\eta$.

Now consider HARP-E. Because

$$\phi_g^+(d) - \phi_g^-(d) = \phi_g(d) \text{ and } \widehat{\phi}_g^+(d) - \widehat{\phi}_g^-(d) = \widehat{\phi}_g(d),$$

we have

$$q_d^{\mathrm{E}}(S) = b_d + \sum_{g \in S} \phi_g(d), \quad \widehat{q}_d^{\mathrm{E}}(S) = b_d + \sum_{g \in S} \widehat{\phi}_g(d).$$

Therefore, for any feasible set S with $|S| \le K$,

$$\begin{aligned} \left|\widehat{q}_d^{\mathrm{E}}(S) - q_d^{\mathrm{E}}(S)\right| &= \left|\sum_{g \in S} \Big(\widehat{\phi}_g(d) - \phi_g(d)\Big)\right| \\ &\le \sum_{g \in S} |\widehat{\phi}_g(d) - \phi_g(d)| \\ &\le K\eta. \end{aligned}$$

Again, clipping to $[0,1]$ is 1-Lipschitz, so the clipped HARP-E domain-utility error is at most $K\eta$. Summing over active domains with nonnegative weights that sum to one gives

$$|\widehat{U}_{\mathrm{E}}(S) - U_{\mathrm{E}}(S)| \le K\eta.$$

Thus $\delta_{\mathrm{E}} \le K\eta$. □

Combining Theorem 3 with Lemma 5 gives the stability statements for HARP-C and HARP-E in the main text. The shared theorem explains why both variants have the same high-level stability form. The sensitivity lemma shows that HARP-C incurs one positive-side maximum error and up to K negative-side errors, giving $(K+1)\eta$. HARP-E, although written with positive and negative parts, is additive in the signed main effects and therefore incurs only K signed-effect errors, giving $K\eta$.

C EXPERIMENTAL HYPERPARAMETERS

We split the hyperparameters into two tables. Table 3 lists the shared final-finetuning settings (LoRA configuration, optimizer, and per-pool sequence-length prefilter) and the evaluation protocol. Table 4 lists the HARP-specific hyperparameters (training-leaf size bounds, representative count, shrinkage settings, and training budget). Section 5.3 varies the leaf-size cap $L = C_{\max}$ and the representative cap R. All measured models use the same full validation set during selection.

Table 3: Shared final-finetuning settings and the evaluation protocol.

Hyperparameter	**Value**
LoRA fine-tuning (Stage 2)	
LoRA rank r	16
LoRA α	32
LoRA dropout	0.05
LoRA target modules	all linear projections
Optimizer	AdamW
Learning rate	2×10^{-4}
Effective batch size	16
Epochs	3
Prefilter: sequence-length cap per training pool	
Self-Instruct (SI)	512
WizardLM-Evol-70k (Wiz)	1,024
Tulu-3-SFT-mixture (Tulu)	2,048
Per-(model, seq-len) micro batch / grad-accum	
QWEN3-4B-BASE, ≤ 1024	16/1
QWEN3-4B-BASE, 2048	8/2
LLAMA-3.1-8B-BASE / QWEN3-8B-BASE, 512	8/2
LLAMA-3.1-8B-BASE / QWEN3-8B-BASE, 1024	4/4
LLAMA-3.1-8B-BASE / QWEN3-8B-BASE, 2048	2/8
Evaluation	
MMLU/ARC scoring	option-letter log-probability
GSM8K/MATH decoding	greedy (temp 0)
Max new tokens, GSM8K/MATH	512
GPU memory utilization	0.95
Repetition	
Random seeds per cell	$\{42, 1, 2\}$

Table 4: HARP-specific hyperparameters.

Hyperparameter	**Value**
Hierarchy (Sec. 3.2)	
Max leaf size $C_{\max}$	1024
Target min leaf size $C_{\min}$	$\lfloor L/4 \rfloor = 256$
Main-effect estimation (Sec. 3.3)	
Representative cap per parent R	3
EB prior variance $\tau^2(d)$	0.01
SE floor	10^{-3}
Envelope selection (Secs. 3.4–3.5)	
Training budget B (best-prefix cap)	10,000

C.1 Selected Subset Sizes

Define $n_{\text{sel}} = c(\widehat{S}) = |A(\widehat{S})|$, the number of selected training examples. The HARP-selected subset size n_{sel} is variable and is fixed by the best-prefix cut rather than by the budget: across all 36 cells of Table 1, HARP-C selects $1{,}087$ examples on average (median 735.5, range 125–$5{,}467$) and HARP-E selects $4{,}106$ (median $2{,}752.5$). These means match Section 5.2 and correspond to $9.2\times$ and $2.4\times$ fewer examples than the $10{,}000$-example baseline budget, respectively. Relative to the full-finetuning mean of $81{,}674$ examples, HARP-C uses approximately $75\times$ fewer examples. Full FT reaches 54.7 mean accuracy, compared with 64.2 for HARP-C and 63.5 for HARP-E (Table 1). The cap B binds in 6 of 36 HARP-E cells and in no HARP-C cell. The construction target $C_{\min}$ in Table 4 is not a lower bound on the final training-example count n_{sel}.

D Computational Complexity

Hierarchy construction is a one-time shared cost C_{hier}, independent of train–evaluate iterations and excluded from the terms below. Table 5 summarizes the per-cell selection and final-training costs of HARP and the baselines. Let N denote the full training set size and $B{=}10{,}000$ the selection budget used by all budgeted baselines. Let $T_L(x, e)$ denote the finetuning cost on x examples for e epochs, which is linear in x at fixed e. HARP uses one epoch per measured leaf and three epochs for final finetuning. Let $T_{\text{eval}}(|\mathcal{E}|)$ denote one evaluation on the full validation set, and let $Q = \sum_p |\mathcal{R}_p|$ be the number of measured training leaves. HARP's dominant selection cost is

$$\sum_p \sum_{r \in \mathcal{R}_p} T_L(|\mathcal{L}_r|, 1) + (Q+1) T_{\text{eval}}(|\mathcal{E}|).$$

The extra evaluation scores the base model. The training term is bounded by $QT_L(L, 1)$ when every leaf has at most $L = C_{\max}$ examples. The selection-time variables are: F for the n-gram feature dimension in DSIR; K_c and $\bar{c}$ for the number and average size of clusters in DQ and SHED; L for the leaf size cap; R for the number of representative leaves per parent used in main-effect estimation; T_∇ for the cost of computing gradient features for one example in LESS; and T_{PG} for the cost of one policy-gradient influence estimate in NICE.

Table 5: Selection and final finetuning complexity per setting. The HARP row bounds representative training and validation, excluding embedding computation, hierarchy construction, and selection search.

Method	**Selection cost**	**Final training**
Random	$O(B)$ sampling	$T_L(B, 3)$
DSIR	$O(NF)$ feature pass	$T_L(B, 3)$
Full-FT	—	$T_L(N, 3)$
SHED	$O(K_c[T_L(\bar{c}, 1) + T_{\text{eval}}(\lvert\mathcal{E}\rvert)])$	$T_L(B, 3)$
LESS	$O(N \cdot T_\nabla)$ gradient datastore	$T_L(B, 3)$
NICE	$O(N \cdot T_{\text{PG}})$ policy-grad. inf.	$T_L(B, 3)$
HARP-C/E	$QT_L(L, 1) + (Q+1)T_{\text{eval}}(\lvert\mathcal{E}\rvert)$	$T_L(n_{\text{sel}}, 3)$

LESS builds a low-dimensional gradient datastore by running a forward–backward pass on each training example, then selects examples by target-task gradient similarity. NICE extends influence estimation to non-differentiable evaluation metrics through policy-gradient-style estimates, requiring repeated metric evaluation for candidate examples. Both methods scale linearly in N with a large per-example constant: LESS costs $O(N \cdot T_\nabla)$, while NICE costs $O(N \cdot T_{\text{PG}})$.