REGIFT: REFRESHED GEOMETRY FOR INSTRUCTION FINE-TUNING VIA DYNAMIC DATA SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Instruction-data selection aims to improve fine-tuning performance under a limited training budget. Static selectors fix a subset before training, whereas dynamic selectors revisit it as the model evolves. Repeated selection therefore requires a ranking criterion that adapts to the model and remains affordable to recompute. We introduce ReGIFT (Refreshed Geometry for Instruction Fine-Tuning), a dynamic selector that constructs and updates a geometric reference from the current model's internal representations of the training pool at every refresh. It estimates layer-wise principal directions from a small uniform pool probe and adjusts loss–entropy rankings using candidate representations obtained during the existing scoring forward passes. This design requires no separately curated reference set, external judge, or learned selection policy. Using Davis–Kahan analysis, we bound changes in the geometric reference between refreshes under bounded covariance change and positive spectral separation. The method selects 10% of the pool per refresh and achieves the highest eight-task average among all compared methods in the main setting, including full-data fine-tuning. Across six backbone–pool settings, it ranks first on the task average in five and second in one among the methods evaluated in each. In the main setting, total runtime is 33.0% lower than Data Agent's.

1 INTRODUCTION

Instruction-data selection can improve model performance while reducing the amount of data used for supervised fine-tuning (SFT). Static methods fix a subset before training (Chen et al., 2024; Li et al., 2024; Liu et al., 2024a; Xia et al., 2024; Chen et al., 2026); dynamic selection revisits data usefulness as the model learns (Kung et al., 2023; Zhang et al., 2026; Yang et al., 2026). In repeated SFT workflows, the practical goal is to improve performance while keeping the combined cost of selection and training manageable.

We introduce ReGIFT, *Refreshed Geometry for Instruction Fine-Tuning*, a dynamic data selector. The construction adapts NAIT's activation principal component analysis (PCA) reference (Chen et al., 2026), using a uniform pool probe in place of a separately curated in-domain seed set. Our contribution combines a pool-derived geometric reference, refitted under the current model at every refresh, with a bounded adjustment to loss–entropy rankings. Layer-wise PCA directions provide candidate projection scores, yielding a model-dependent selection rule without a curated seed set or external judge.

Candidate representations reuse the forward passes already needed for loss and entropy, allowing geometric selection to fit the existing scoring workflow. Geometry adds probe extraction, PCA, pooling, and projection (Appendix A), without a separate selection-policy training loop. With 10% of the pool selected per refresh, our approach achieves the highest eight-task average among all compared methods, including full-data tuning, in the main comparison (Table 2). Including repeated pool scoring and geometric computation, the total runtime is 33.0% lower than Data Agent's (Table 3).

Figure 1(a) illustrates this reference construction and ranking adjustment.

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.

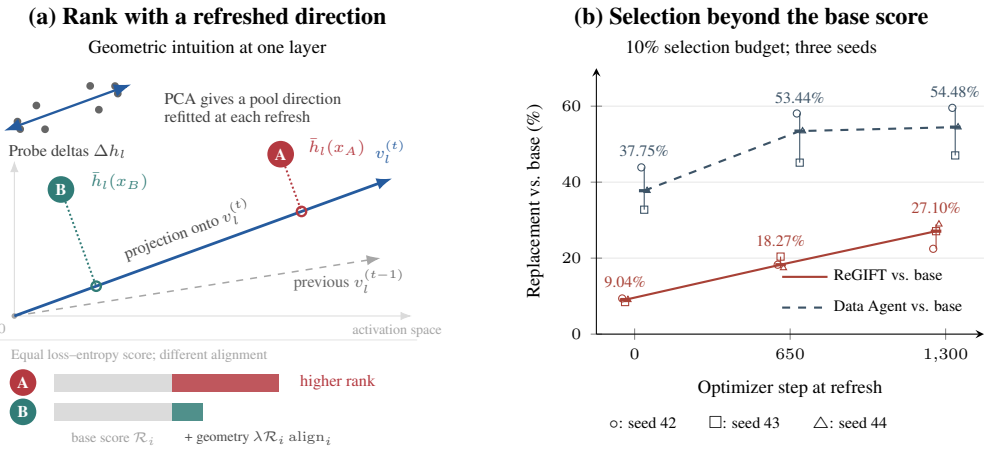


Figure 1: **ReGIFT: refresh pool geometry and adjust selection.** (a) With equal loss–entropy scores, better alignment with the refreshed direction gives a higher rank. (b) In a small-scale diagnostic (Qwen2.5-0.5B + LoRA, cosine schedule; main experiments: LLaMA-2-7B, Section 3), ReGIFT departs less from its base ranking, with narrower seed spread, than Data Agent. Lines and spans show three-seed medians and ranges.

At refresh t, candidate x_i receives score $s_i^{(t)} = \mathcal{R}_i^{(t)}(1 + \lambda\,\mathrm{align}_i^{(t)})$, where $\mathcal{R}_i^{(t)}$ is the loss–entropy base score (the *carrier*) drawn from Data Agent's reward (Yang et al., 2026), $\mathrm{align}_i^{(t)} \in [0, 1]$ measures alignment with directions estimated from a small pool probe, and $\lambda \geq 0$. Setting $\lambda = 0$ recovers the *Carrier-Only* (CO) baseline. The selected examples receive SFT updates until the next refresh.

Figure 1(b) measures how each selector departs from its current loss–entropy base ranking. Using a bounded deterministic multiplier, ReGIFT departs less and more consistently than Data Agent's stochastic policy in the three-seed diagnostic. Appendix F.1 isolates the variation from Data Agent's action sampling.

Our contributions: (i) a deterministic ranking rule that refits a layer-wise reference from the pool and adjusts loss–entropy scores through a bounded multiplier (Section 2); ReGIFT outperforms the $\lambda = 0$ Carrier-Only ablation on every displayed task in the main comparison (Table 2); (ii) a conditional Davis–Kahan characterization of reference movement between refreshes, identifying the measurable covariance change and cross-time separation that the diagnostic in Section 2.4 monitors, with the proof and extended diagnostics in Appendices B.2 and F.2; (iii) a cost-and-performance evaluation (Section 3) at a nominal 10% selection ratio: ReGIFT achieves the highest task average in five of six backbone–pool settings and ranks second in the sixth (Tables 2, 8, and 9), with selection and SFT costs accounted for separately in Appendix G.

Related approaches. Selection signals range from human or external-model judgments (Zhou et al., 2023; Chen et al., 2024) to target-model difficulty and uncertainty (Li et al., 2024; Liu et al., 2024a), activation geometry from curated in-domain seeds (Chen et al., 2026), gradient similarity to target-task examples (Xia et al., 2024; Gharehhasanloo et al., 2025), and learned auxiliary scorers (Ren et al., 2026; Chang et al., 2026). DiverseEvol iteratively expands its training subset using diversity in the current model's embedding space (Wu et al., 2023). ReGIFT instead uses a refreshed layer-wise PCA reference to adjust loss–entropy rankings within a fixed per-refresh subset budget. Response length is itself a strong static criterion: selecting the longest responses is a tough-to-beat baseline for instruction fine-tuning (Zhao et al., 2024), and DEITA scores complexity and quality with external models (Liu et al., 2024b). The length-predicted control in Appendix E.4 relates ReGIFT's alignment score to this length criterion. AtteNT performs loss-based re-ranking after an initial full-data epoch (Zhang et al., 2026); Data Agent trains a PPO selection policy (Yang et al., 2026). Appendix C.3 details the resources and adaptations used in our baseline implementations. Beyond instruction tuning, model-aware and online selection is studied for pretraining, preference, and reinforcement fine-tuning data (Yu et al., 2024; Wang et al., 2024; Albalak et al., 2023; Sun et al., 2025; Jha et al., 2025), with surveys of static criteria (Zhang et al., 2025; Qin et al., 2024).

Example reweighting (Zhao et al., 2026) and response-token selection (Qin et al., 2026) retain input examples and modify supervision instead.

2 REGIFT: REFRESHED GEOMETRY FOR INSTRUCTION FINE-TUNING

2.1 PROBLEM SETUP AND BASE RANKING

Let $\mathcal{D} = \{x_i\}_{i=1}^N$ be a pool of N instruction–response pairs, indexed by i. Let T be the number of scheduled selection refreshes, $t \in \{1, \dots, T\}$ their index, and θ_{t-1} the model parameters immediately before refresh t, starting from θ_0. The per-refresh budget is $B \in \{1, \dots, N\}$ examples, with selection fraction $\rho = B/N$. We use $\rho = 0.1$ in the main setting; the union of examples visited across refreshes can exceed this fraction.

For candidate x_i, let $\mathcal{L}_i^{(t)}$ and $\mathcal{H}_i^{(t)}$ denote cross-entropy loss and predictive entropy under θ_{t-1}, both averaged over response-token positions. Write $\mathrm{Var}_{\mathcal{D}}$ for variance across candidates, $\varepsilon > 0$ for a denominator stabilizer, $w^{(t)}$ for the mixing weight, and $\mathcal{R}_i^{(t)}$ for the resulting loss–entropy base score, or *carrier*. The difficulty and uncertainty signals (Yang et al., 2026) are combined as

$$w^{(t)} = \frac{\mathrm{Var}_{\mathcal{D}}(\mathcal{L}^{(t)})}{\mathrm{Var}_{\mathcal{D}}(\mathcal{L}^{(t)}) + \mathrm{Var}_{\mathcal{D}}(\mathcal{H}^{(t)}) + \varepsilon}, \tag{1}$$

$$\mathcal{R}_i^{(t)} = w^{(t)}\mathcal{L}_i^{(t)} + (1 - w^{(t)})\mathcal{H}_i^{(t)}. \tag{2}$$

Carrier-Only (CO) selects the top-B candidates by $\mathcal{R}_i^{(t)}$, recomputed at each refresh.

2.2 POOL DIRECTIONS AND CANDIDATE ALIGNMENT

Probe the current model. Let $l = 1, \dots, L$ index the layers used for alignment (all transformer layers in our settings; Table 5) with hidden size d, and let $h_l^{(k)}(x;\theta) \in \mathbb{R}^d$ denote the hidden state at token k of the full instruction–response sequence of length K_x under parameters θ. Define its last-minus-first hidden-state difference Δh_l and sequence mean $\bar{h}_l$ by

$$\Delta h_l(x;\theta) := h_l^{(K_x)}(x;\theta) - h_l^{(1)}(x;\theta), \qquad \bar{h}_l(x;\theta) := \frac{1}{K_x}\sum_{k=1}^{K_x} h_l^{(k)}(x;\theta). \tag{3}$$

At refresh t, sample a uniform n_p-example probe $\widetilde{\mathcal{D}}_t \subset \mathcal{D}$ and extract its deltas under θ_{t-1}.

Refit the pool direction. Let $\overline{\Delta h}_l^{(t)}$ and $\widehat{\Sigma}_l^{(t)}$ be the probe deltas' mean and centered empirical covariance. Following NAIT's PCA construction (Chen et al., 2026), take a top unit eigenvector $\tilde{v}_l^{(t)}$ of $\widehat{\Sigma}_l^{(t)}$. Orient it toward the probe mean initially and toward the preceding direction thereafter, yielding $v_l^{(t)}$. These per-layer directions form the *geometric reference*, refitted from the pool under the current model at every refresh. A single direction requires one projection per candidate per layer. Appendices A.1 and A.2 specify the PCA and sign conventions.

Project candidates and adjust ranking. Let $\tilde{a}_i^{(t)}$ denote raw alignment, and use $\langle\cdot,\cdot\rangle$ and $\|\cdot\|_2$ for the Euclidean inner product and norm. Project candidate means onto the pool directions:

$$\tilde{a}_i^{(t)} = \sum_{l=1}^{L} \langle \bar{h}_l(x_i;\theta_{t-1}), v_l^{(t)} \rangle. \tag{4}$$

Pool-wise min–max normalization gives $\mathrm{align}_i^{(t)} \in [0,1]$, with a common value $1/2$ when the range is negligible (Appendix A.3). Probe deltas construct the reference; candidate means, obtained during the existing scoring forward pass, supply the projection scores (Figure 1(a)).

2.3 GEOMETRY-ADJUSTED SELECTION

Let $\lambda \geq 0$ control the geometric adjustment and $s_i^{(t)}$ denote the final score. Combining alignment with the carrier gives

$$s_i^{(t)} = \mathcal{R}_i^{(t)}\big(1 + \lambda\,\mathrm{align}_i^{(t)}\big). \tag{5}$$

The multiplier lies in $[1, 1+\lambda]$: geometry adjusts the base ranking, while $\lambda = 0$ or a common alignment value preserves it. Candidates satisfying $\mathcal{R}_i^{(t)} > (1+\lambda)\mathcal{R}_k^{(t)}$ retain their order (Appendix B.1).

Let $\mathcal{S}_t$ be the selected subset, TopB return the B highest-scoring candidates with fixed candidate-index tie handling, and U_t denote the interval's SFT update count. Select $\mathcal{S}_t = \mathrm{TopB}_{x_i \in \mathcal{D}}\, s_i^{(t)}$ and fine-tune for U_t updates to obtain θ_t; the next refresh recomputes directions and scores under this model. Selection is deterministic given the probe and model-derived scores, yet adapts to their changes across refreshes.

2.4 Reference stability

Write $\Sigma_l^{(t)} = \widehat{\Sigma}_l^{(t)}$; let $\lambda_r(A)$ denote the r-th largest eigenvalue of a symmetric matrix A (distinct from the adjustment weight λ) and $\|\cdot\|_F$ the Frobenius norm. For a layer l and transition $t > 1$, let $\gamma_l > 0$ be a lower bound on cross-time spectral separation.

Theorem 1 (Direction stability)**.** *For consecutive refreshes, suppose*

$$\lambda_1(\Sigma_l^{(t)}) - \lambda_2(\Sigma_l^{(t-1)}) \geq \gamma_l > 0. \tag{6}$$

With temporal sign calibration $\langle v_l^{(t)}, v_l^{(t-1)} \rangle \geq 0$*,*

$$\|v_l^{(t)} - v_l^{(t-1)}\|_2 \leq \frac{\sqrt{2}\,\|\Sigma_l^{(t)} - \Sigma_l^{(t-1)}\|_F}{\gamma_l}. \tag{7}$$

The bound follows from a Davis–Kahan residual argument (Davis & Kahan, 1970; Stewart & Sun, 1990) (Appendix B.2), applied to the refreshed pool covariances. Their change includes both model updates and probe resampling.

Connection to the geometric adjustment. Let $z_{il}^{(t)} = \bar{h}_l(x_i; \theta_{t-1})$ abbreviate the candidate mean, and let $b_l^{(t)}$ be a valid bound on consecutive direction movement from Theorem 1. Adding and subtracting $\langle z_{il}^{(t-1)}, v_l^{(t)} \rangle$ in Eq. (4) and applying Cauchy–Schwarz yields

$$|\tilde{a}_i^{(t)} - \tilde{a}_i^{(t-1)}| \leq \underbrace{\sum_{l=1}^{L} \|z_{il}^{(t)} - z_{il}^{(t-1)}\|_2}_{\text{candidate change}} + \underbrace{\sum_{l=1}^{L} \|z_{il}^{(t-1)}\|_2\, b_l^{(t)}}_{\text{reference change}}. \tag{8}$$

Raw alignment changes through both candidate representations and reference directions. Theorem 1 bounds the latter contribution. The selected subset also depends on carrier scores, normalization ranges, and selection-boundary margins (Appendix B.4).

Empirical check of the conditional bound. Let j index passive diagnostic observations, and let $\tilde{q}_l^{(j)}$ and $q_l^{(j)}$ be the raw and temporally sign-calibrated unit PCA directions at observation j. Denote adjacent displacement by $d_l^{(j)}$. We monitor Qwen2.5-0.5B + LoRA every 25 optimizer steps on a fixed 2,000-example probe (Appendix F.3):

$$d_l^{(j)} = \|q_l^{(j)} - q_l^{(j-1)}\|_2 = \sqrt{2 - 2|\langle \tilde{q}_l^{(j)}, \tilde{q}_l^{(j-1)} \rangle|} \in [0, \sqrt{2}]. \tag{9}$$

These diagnostic directions are distinct from the selection directions $v_l^{(t)}$, which use resampled probes at scheduled refreshes.

Let A and C be the preceding and current diagnostic covariances for one layer, and define the larger cross-time separation $g = \max\{\lambda_1(C) - \lambda_2(A), \lambda_1(A) - \lambda_2(C)\}$. For $g > 0$, apply the residual argument in the temporal direction attaining g (Appendix B.3):

$$d_l^{(j)} \leq b_{\mathrm{diag},l}^{(j)} := \frac{2\|C - A\|_F}{g}. \tag{10}$$

The diagnostic keeps the looser constant 2; the $\sqrt{2}$ constant of Theorem 1 tightens every reported bound by the same factor, so the fractions below are conservative. A bound below $\sqrt{2}$ is *informative* (nontrivial): it gives an angular upper bound below the generic 90° limit for sign-aligned

unit vectors. The separation condition held for all 1,496 retained layer-transition pairs in every run (Appendix F.3).

Table 1 tracks ReGIFT's reference under two learning-rate schedules. The reported displacement summaries correspond to about $1.65°$ under cosine decay and $3.18°$ under a constant rate. Informative bounds narrow the generic $90°$ limit: for illustration, a $30°$ upper bound on an observed $3°$ rotation provides such a restriction. The 86% and 74% summarize the per-run fractions of observed pairs receiving a bound below the generic limit.

Appendix F.2 explains these two quantities and visualizes all three seeds in Figure 5.

Table 1: **Reference diagnostics.** Qwen2.5-0.5B + LoRA; fixed 2,000-example probe, 25-step observations, shared warm-up. Three-seed medians of per-run summaries: time median of layer-median displacement (Eq. (9)) and fraction of 1,496 post-warm-up layer-transition pairs with bound $< \sqrt{2}$ (Appendix F.3).

Schedule	Direction change	Informative bounds
Cosine	0.029	86%
Constant	0.055	74%

2.5 Algorithm and Computation

For the complete procedure, let $H_l^{(t)} \in \mathbb{R}^{N\times d}$ denote the conceptual matrix with rows $\bar{h}_l(x_i;\theta_{t-1})^\top$; its product with $v_l^{(t)}$ collects candidate projections. Figure 2 uses this matrix view (refresh superscript suppressed); Algorithm 1 combines the preceding definitions. The refresh schedule specifies U_t, independently of how often passive diagnostics are recorded.

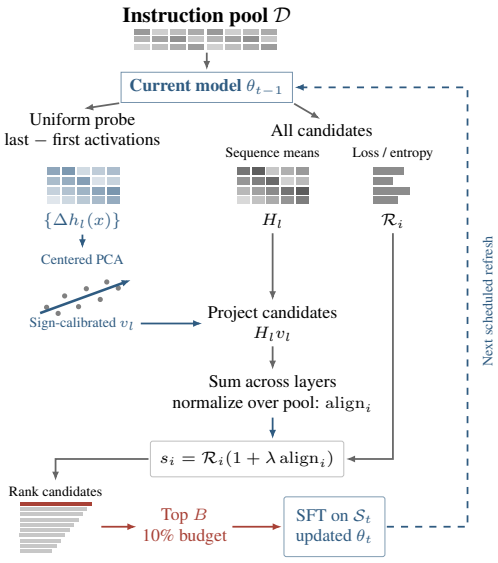


Figure 2: **ReGIFT refresh cycle:** refit, rerank, and fine-tune.

Algorithm 1 ReGIFT: scheduled selection

Input: pool $\mathcal{D}$; model θ_0; budget B; layers $1{:}L$; weight λ; probe size n_p; refreshes T; updates $\{U_t\}$.

for $t = 1,\dots,T$ **do**
  Sample uniform probe $\widetilde{\mathcal{D}}_t$ of size n_p.
  Extract probe deltas Δh_l under θ_{t-1}.
  for $l = 1,\dots,L$ **do**
    Compute $\overline{\Delta h}_l^{(t)}$, $\widehat{\Sigma}_l^{(t)}$ (probe deltas' mean and covariance; Appendix A.1).
    $\tilde{v}_l^{(t)} \leftarrow$ top unit eigenvector of $\widehat{\Sigma}_l^{(t)}$.
    Sign-calibrate $\tilde{v}_l^{(t)} \to v_l^{(t)}$.
  end for
  for each batch in $\mathcal{D}$ **do**
    One forward pass: $\mathcal{L}_i^{(t)}$, $\mathcal{H}_i^{(t)}$, $\bar{h}_l(x_i;\theta_{t-1})$.
    $\tilde{a}_i^{(t)} \leftarrow \sum_{l=1}^{L}\langle \bar{h}_l(x_i;\theta_{t-1}), v_l^{(t)}\rangle$.
  end for
  Pool-normalize $\tilde{a}_i^{(t)} \to \text{align}_i^{(t)}$ (min–max over the pool; Appendix A.3).
  Compute $w^{(t)}$, $\mathcal{R}_i^{(t)}$ (Eqs. (1)–(2)).
  $s_i^{(t)} \leftarrow \mathcal{R}_i^{(t)}(1+\lambda\,\text{align}_i^{(t)})$.
  $\mathcal{S}_t \leftarrow$ top-B candidates by $s_i^{(t)}$.
  SFT on $\mathcal{S}_t$ for U_t updates: $\theta_{t-1} \to \theta_t$.
end for
return θ_T.

Computation at a refresh. Each candidate-scoring forward pass supplies both logits for loss and entropy and hidden states for alignment. Mean pooling and batched projections produce scalar scores without materializing the full matrix $H_l^{(t)}$. Probe extraction and PCA add computation, while retaining hidden states within a scoring batch adds memory overhead. No auxiliary selector network, policy optimizer, or action-sampling loop is required. Between refreshes, SFT reuses the selected IDs without rescoring the pool, so the refresh interval amortizes selection work over subsequent updates. Section 3.2 reports selection and SFT costs.

3 EXPERIMENTS

Setting. We fine-tune all parameters of LLaMA-2-7B (Touvron et al., 2023) on Alpaca-GPT4 (Peng et al., 2023), selecting $\rho = 10\%$ of the pool per refresh. The eight-task suite covers knowledge and reasoning (MMLU, BBH), math (SVAMP, GSM8K), code (MBPP, HumanEval), and multilingual QA (TyDiQA, XQuAD). We report arithmetic task means: AVG over all eight tasks and, to reduce GPU evaluation cost in transfer and ablation studies, AVG4 over four tasks, one per domain (MMLU, SVAMP, MBPP, TyDiQA). Main and transfer comparisons use one fixed-seed run per method and setting. Benchmarks, training protocols, and the budget rationale are in Appendices C.2, C.1, and D.1.

Table 2: **Eight-task performance by selection method. Bold**/underline: best/second-best among ranked subset-selection methods (Random; S2, S3, D). Δ: relative AVG change versus Full FT.

	Knowledge & reasoning		Math		Code		Multilingual QA		Overall	
Method	MMLU	BBH	SVAMP	GSM8K	MBPP	H-Eval	TyDiQA	XQuAD	AVG	Δ (%)
References										
No FT	45.82	36.35	36.67	13.87	0.00[†]	2.38[†]	61.70	55.08	31.48	−20.96%
Full FT (100%)	47.86	37.19	40.33	16.60	37.35	25.60	60.74	52.99	39.83	—
Random (10%)	45.04	35.54	36.80	13.57	35.41	24.51	60.83	52.28	38.00	−4.61%
S. Static selection: subset fixed before SFT (S1 external judgment; S2 target-model signal; S3 auxiliary scorer)										
S1 LIMA[‡]	46.44	36.68	36.20	16.38	33.07	22.09	59.28	51.14	37.66	−5.45%
S1 AlpaGasus[‡]	46.17	36.36	38.33	16.22	37.50	23.94	60.81	53.87	39.15	−1.71%
S2 Q2Q (IFD)[§]	46.18	36.84	34.33	17.82	36.19	25.19	60.42	52.34	38.66	−2.93%
S2 SelectIT[§]	46.02	38.37	36.33	16.98	36.58	28.17	60.44	**54.12**	39.63	−0.52%
S2 NAIT (All)	46.08	40.02	39.67	17.06	35.41	26.44	60.03	52.87	39.70	−0.34%
S3 CPQS[§]	46.35	36.76	36.67	16.60	36.19	**31.18**	51.84	51.95	38.44	−3.49%
D. Dynamic selection: 10% subset refreshed during SFT										
AtteNT (full first epoch)[§]	45.05	35.68	38.67	14.71	37.74	28.25	50.53	50.50	37.64	−5.50%
Data Agent (PPO)	46.49	38.37	37.67	16.48	**39.69**	28.61	60.44	50.85	39.83	−0.02%
CO ($\lambda = 0$)	45.90	40.21	36.67	16.53	34.63	24.38	60.12	50.70	38.64	−2.99%
REGIFT ($\lambda = 1$)	**47.71**	**40.36**	**40.33**	**18.27**	38.52	25.87	**61.13**	53.43	**40.70**	**+2.18%**
F. Full-input methods: different supervision budgets (unranked)										
ADAPT[§]	46.72	35.36	37.33	15.24	35.41	27.99	46.37	39.51	35.49	−10.90%
ssToken (60% tokens)	44.60	25.01	29.00	12.66	31.52	26.91	43.28	41.86	31.86	−20.03%

$\Delta = 100(\mathrm{AVG}/\mathrm{AVG}_{\mathrm{Full\ FT}} - 1)$, computed before rounding.
[†] MBPP and HumanEval execute generated code; the non-instruction-tuned base emits free-form text, hence near-zero scores.
[‡] Published LIMA and AlpaGasus corpora, not 10% subsets of the pool; details in Appendix C.3.
[§] Adaptations of the original recipes: Appendix C.3.

Comparison scope and budgets. Table 2 compares static and dynamic selectors, with Full FT as the full-data training baseline. Ranked subset methods share the SFT recipe and a nominal 10% per-selection budget; dynamic methods can visit more unique examples across refreshes, and AtteNT first trains for one full-data epoch. LIMA and AlpaGasus use their published corpora, while ADAPT and ssToken use different supervision budgets; these rows are unranked. Baseline resources, scoring rules, and adaptations are detailed in Appendix C.3.

3.1 MAIN RESULTS AND PRACTICAL IMPLICATIONS

With approximately 5,200 examples selected per refresh, ReGIFT achieves the **highest eight-task average among all compared methods**, with **40.70 AVG** versus 39.83 for both **Full FT and Data Agent**. The gain over Full FT is **2.18% relative** (+0.87 points). For teams that periodically update instruction-tuned models, this supports dynamic selection as a way to obtain stronger average performance from a small selected training set.

Across all six evaluated backbone–pool settings, ReGIFT exceeds Full FT and Data Agent, achieves the highest task average in **five settings**, and ranks second in the remaining setting (Tables 2, 8, and 9). With the alignment weight fixed, these results support reuse of the selection rule across the evaluated backbones and instruction pools. Random 10% also exceeds Full FT in four of the six settings under the shared SFT recipe, so the Random and Data Agent rows are the more informative references for selection quality.

Selection within an existing SFT workflow. NAIT uses a static reference fitted on a supplied in-domain seed set. ReGIFT refits its reference at every refresh using the current model and a uniform probe of the available pool. This removes reference curation from the preparation needed when a team changes checkpoints or instruction corpora. Direct score updates also avoid Data Agent's separate PPO policy-training loop, making repeated selection easier to integrate into an existing SFT pipeline.

3.2 Cost of model-internal selection

ReGIFT achieves its higher main-setting average with **33.0% less total time than Data Agent** (Table 3). Selection time falls by 37.9%, while target-SFT times remain similar. These totals include the overhead of repeated pool scoring, probe extraction, and PCA. For recurring fine-tuning jobs, lower total runtime can shorten retraining cycles or leave time for additional model updates within the same budget.

Table 3: **Selection and SFT time for LLaMA-2-7B (hours).**

Method	Selection	SFT	Total	Selection mechanism
Full FT (100%)	0	6.40^{E}	6.40^{E}	None
NAIT	1.18	0.66	1.84	Curated-seed geometry (static)
Data Agent	3.77	0.68	4.45	Learned PPO policy
ReGIFT	2.34	0.64	2.98	Refreshed pool geometry

10% selectors; E: Full FT linear estimate ($0.64/0.1$); other entries: single-run A100 times.
ReGIFT vs. Data Agent: 37.9% less selection time; 33.0% less total time (Appendix G).
Human curation and evaluation excluded; full accounting: Appendix G.

Among the methods in Table 3, NAIT has the lowest total time with its curated seed supplied. CO (2.24 h, Appendix G) is also cheaper than ReGIFT (2.98 h). ReGIFT combines higher average performance with automatic reference updates from the current pool, making it useful when selection must adapt during training without a separate curation step. Full cost accounting is in Appendix G.

3.3 Sensitivity, model scale, and dataset transfer

These comparisons examine how the same selection rule behaves as its alignment weight, the model, and the instruction pool change.

Alignment-weight sensitivity. All four positive weights in the tested grid improve AVG4 over CO, with the best score at $\lambda = 1$ (Figure 3, Appendix E). The decline at $\lambda = 5$ supports using geometry as a moderate adjustment to the base score. We selected $\lambda = 1$ using this reported AVG4 sweep; baselines received no analogous sweep. All model and pool transfers retain this value without an additional alignment-weight search.

What the geometric adjustment changes. Adding the geometric multiplier to CO improves all eight task scores and raises AVG by 2.06 points under the same training configuration (Table 2). In the ReGIFT-M controls, a separately configured variant (Appendix C.1), removing geometry at four fixed checkpoints of one run replaces 16.6–34.1% of selected examples, and the full selector exceeds length-matched Random by 1.04 AVG4 points over three seeds. A five-fold out-of-fold predictor of the geometry-derived alignment from input and response lengths reaches $R^2 = 0.991$–0.992 with 93.2–97.9% same-checkpoint selection overlap, and its mean AVG4 is 46.16 versus 46.04 for ReGIFT-M. Because this predictor is supervised by the geometric score at each check-

point, it measures how much of that score is explained by length rather than replacing the geometric computation (Appendices E.2–E.4).

Transfer across model families and scales. On Alpaca-GPT4, ReGIFT achieves the highest AVG4 on Mistral-7B, Qwen2.5-7B, and Qwen2.5-0.5B with $\lambda = 1$ unchanged (Table 8). This consistency across model families and the 0.5B–7B range supports a shared selection procedure for teams maintaining both compact and larger models, without an additional alignment-weight sweep for each backbone.

Replacing the instruction-tuning dataset. Alpaca-GPT4 pairs the original Alpaca instructions with GPT-4 responses; WizardLM Evol-Instruct iteratively rewrites instructions to increase complexity and broaden task coverage (Peng et al., 2023; Xu et al., 2024). Replacing the pool therefore tests reuse under a different instruction generation process. ReGIFT exceeds Full FT and Data Agent on both LLaMA-2-7B and Qwen2.5-7B, ranking first and second, respectively (Table 9). Random is competitive here: ReGIFT's AVG4 margins over it are $+0.17$ and -0.03. The practical benefit is that a pipeline can change how its instruction data are generated while retaining the same selection rule, without curating a new reference set or retuning the alignment weight.

4 Discussion

ReGIFT makes the geometric reference part of the evolving training process. Reconstructing it from the current model and candidate pool lets the same selection rule adapt to changing representations. The transfer results with a fixed alignment weight support reusing this procedure across model families and instruction sources, without preparing a new reference set or learning a separate selection policy.

The bounded adjustment lets geometry refine loss–entropy rankings while preserving sufficiently separated base preferences (Appendix B.1). The conditional Davis–Kahan analysis relates reference movement to covariance change and spectral separation, and the fixed-probe diagnostics show how movement varies with the learning-rate schedule (Appendix F.2). These observations motivate studying refresh intervals that respond to representation change: more frequent refreshes update the selection criterion sooner, while longer intervals spread the cost of each full-pool scoring pass over more training updates.

The ReGIFT-M controls reveal a strong length-related structure in the model-derived alignment score, motivating simpler approximations of that score. Fixed-reference comparisons would isolate the value of reference updates within the complete selector. Extending the design to pretraining and mid-training would test its scalability with larger pools and longer training horizons.

5 Conclusion

ReGIFT combines a geometric reference refitted from the current model and training pool with a bounded adjustment to loss–entropy rankings, without a curated reference set, external judge, or learned selection policy. At 10% selection per refresh, it achieves the highest eight-task average among all compared methods in the main setting, including Full FT and Data Agent. Across six backbone–pool settings, it ranks first on the task average in five and second in one among the methods evaluated in each. In the main setting, total runtime is 33.0% lower than Data Agent's. These results support pool-derived geometric scoring as a practical approach to instruction-data selection.

AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript for readability and to check the consistency of cross-references, reported numbers, and bibliography entries. The research questions, method, experimental design, mathematical results and proofs, and analysis are the authors' own. The authors take full responsibility for the content of this paper.

ETHICS STATEMENT

Instruction-data selection may amplify existing biases or underrepresent tasks and languages in the source pool. Applications should assess these effects and respect the usage terms of the datasets and models.

REPRODUCIBILITY STATEMENT

All configurations behind the reported numbers are specified in the paper: (i) the selection rule and refresh procedure in Section 2, with the proof of Theorem 1 in Appendix B.2; (ii) per-cohort training configurations in Tables 4 and 5, the ReGIFT-M control recipe in Appendix C.1, the diagnostic recipe and its seeds (42/43/44) in Appendix F.3, and the length-matched control seeds (42/1/7) in Appendix E.3; (iii) baseline implementations, adaptations, and the associated disclosures in Appendix C.3 and Table 6; (iv) the evaluation harness and per-benchmark scoring protocols in Appendix C.2; and (v) the cost-accounting assumptions in Appendix G, including which times are measured and which are estimated.

REFERENCES

Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. Efficient online data mixing for language model pre-training. *arXiv preprint arXiv:2312.02406*, 2023.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4623–4637, 2020. doi: 10.18653/v1/2020.acl-main.421.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. URL https://arxiv.org/abs/2108.07732.

Powei Chang, Jinpeng Zhang, Bowen Chen, Chenyu Wang, Chenlu Guo, Yixing Zhang, Yukang Gao, JianXiang Xiang, Yue Gao, Chaoqun Sun, Yiyi Chen, and Dongying Kong. SPICE: Submodular penalized information–conflict selection for efficient large language model training. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/7cc1005ec73cfbaac9fa21192b622507-Abstract-Conference.html.

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 (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/9543942c237ded1b39b1fd37259ff88e-Abstract-Conference.html.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Xin Chen, Junchao Wu, Shu Yang, Runzhe Zhan, Zeyu Wu, Min Yang, Shujian Huang, Lidia S. Chao, and Derek F. Wong. Neuron-aware data selection in instruction tuning for large language models. In *The Fourteenth International Conference on Learning Representations*,

2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/435aa4e377a269a95d58bf21204c91c4-Abstract-Conference.html.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470, 2020. doi: 10.1162/tacl_a_00317.

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. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Chandler Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970. doi: 10.1137/0707001.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 2023. lm-evaluation-harness; https://github.com/EleutherAI/lm-evaluation-harness.

Mohammad Gharehhasanloo, Yueting Chen, Nick Koudas, and Xiaohui Yu. Approximating gradient-based influence for scalable instruction data selection. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 4733–4737. ACM, 2025. doi: 10.1145/3746252.3760808.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2009.03300.

Animesh Jha, Harshit Gupta, and Ananjan Nandi. Rl-guided data selection for language model finetuning. In *NeurIPS 2025 Workshop on Constrained Optimization for Machine Learning*, 2025. arXiv:2509.25850.

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. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.112.

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, Mexico City, Mexico, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.421.

Liangxin Liu, Xuebo Liu, Derek F. Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. SelectIT: Selective instruction tuning for LLMs via uncertainty-aware self-reflection. In *Advances in Neural Information Processing Systems*, volume 37, pp. 97800–97825. Curran Associates, Inc., 2024a. doi: 10.52202/079017-3102. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/b130a5691815f550977e331f8bec08ae-Abstract-Conference.html.

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 *International Conference on Learning Representations (ICLR)*, 2024b.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, 2021. doi: 10.18653/v1/2021.naacl-main.168.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with GPT-4. *arXiv preprint arXiv:2304.03277*, 2023. doi: 10.48550/arXiv.2304.03277.

Xiaohan Qin, Xiaoxing Wang, Ning Liao, Cancheng Zhang, Xiangdong Zhang, Mingquan Feng, Jingzhi Wang, and Junchi Yan. ssToken: Self-modulated and semantic-aware token selection for LLM fine-tuning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/801ec05b0aae9fcd2ef35c168bd538e0-Abstract-Conference.html.

Yulei Qin, Yuncheng Yang, Pengcheng Guo, Gang Li, et al. Unleashing the power of data tsunami: A comprehensive survey on data assessment and selection for instruction tuning of language models. *arXiv preprint arXiv:2408.02085*, 2024.

Yi Ren, Yanhui Li, Tianyi Zhang, and Diandong Liu. CPQS-tuning: A model self-perception-based data filtering algorithm for efficient instruction fine-tuning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/65b721a1df04c1098567f70d483d6468-Abstract-Conference.html.

G. W. Stewart and Ji-guang Sun. *Matrix Perturbation Theory*. Computer Science and Scientific Computing. Academic Press, 1990.

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving data efficiency for LLM reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.05316.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.824.

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

Jiachen T. Wang, Tong Wu, Dawn Song, Prateek Mittal, and Ruoxi Jia. GREATS: Online selection of high-quality data for LLM training in every iteration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Shengguang Wu, Keming Lu, Benfeng Xu, Junyang Lin, Qi Su, and Chang Zhou. Self-evolved diverse data sampling for efficient instruction tuning, 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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54104–54132, 2024. URL https://proceedings.mlr.press/v235/xia24c.html.

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 *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/82eec786fdfbbfa53450c5feb7d1ac92-Abstract-Conference.html.

Suorong Yang, Fangjian Su, Hai Gan, Ziqi Ye, Jie Li, Baile Xu, Furao Shen, and Soujanya Poria. Data agent: Learning to select data via end-to-end dynamic optimization. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=fKe6zvEZEE.

Zichun Yu, Spandan Das, and Chenyan Xiong. MATES: Model-aware data selection for efficient pretraining with data influence models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Bolin Zhang et al. A survey on data selection for LLM instruction tuning. *Journal of Artificial Intelligence Research*, 83, 2025.

Chen Zhang, Jianghui Wang, Bingyang Cheng, Zhongtao Chen, Wendong Xu, Cong Wang, Marco Canini, Francesco Orabona, Yik-Chung Wu, and Ngai Wong. Nonparametric teaching of attention learners. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/54d2d38a56a74387d5916ee40e462295-Abstract-Conference.html.

Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning. In *International Conference on Machine Learning (ICML)*, 2024.

Wanru Zhao, Yihong Chen, Yuzhi Tang, Wentao Ma, Shengchao Hu, Shell Xu Hu, Alex Iacob, Abhinav Mehrotra, and Nicholas D. Lane. Rethinking data curation in LLM training: Online reweighting offers better generalization than offline methods. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/b77e87be2a7caca996da3c79191e2e88-Abstract-Conference.html.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ac662d74829e4407ce1d126477f4a03a-Abstract-Conference.html.

CONTENTS

1 Introduction **1**

2 ReGIFT: Refreshed Geometry for Instruction Fine-Tuning **3**
2.1 Problem setup and base ranking . . . 3
2.2 Pool directions and candidate alignment . . . 3
2.3 Geometry-adjusted selection . . . 3
2.4 Reference stability . . . 4
2.5 Algorithm and computation . . . 5

3 Experiments **6**
3.1 Main results and practical implications . . . 6
3.2 Cost of model-internal selection . . . 7
3.3 Sensitivity, model scale, and dataset transfer . . . 7

4 Discussion **8**

5 Conclusion **8**

A Implementation Details **14**
A.1 Pool probing and PCA . . . 14
A.2 Sign calibration and numerical conventions . . . 14
A.3 Batched scoring and scheduled selection . . . 14

B Supporting Mathematical Analysis **14**
B.1 Properties of the bounded geometric adjustment . . . 14
B.2 Proof of Theorem 1 . . . 15
B.3 Two-sided spectral separation . . . 15
B.4 From reference movement to selection scores . . . 15

C Experimental Protocols **16**
C.1 Training configurations and experimental cohorts . . . 16
C.2 Instruction pools and evaluation . . . 16
C.3 Baseline implementations and adaptations . . . 17

D Additional Performance Results **18**
D.1 Selection-budget sensitivity . . . 18
D.2 Transfer across model families and scales . . . 19
D.3 Transfer to Evol-Instruct . . . 20

E Ablations of Geometric Selection **20**
E.1 Alignment-weight sensitivity . . . 20
E.2 Removing geometry at a fixed checkpoint . . . 21
E.3 Comparison with length-matched random selection . . . 22
E.4 Length-predicted and shuffled alignment . . . 23

F Selection Changes and Reference Movement **23**
F.1 Why do selected examples change? . . . 24
F.2 How far does the reference move? . . . 24
F.3 Diagnostic protocol . . . 24

G Computational Cost **25**
G.1 Measured time and estimated budgets . . . 25
G.2 Accounting assumptions for additional methods . . . 25
G.3 External preparation and sensitivity analysis . . . 27

A Implementation Details

A.1 Pool probing and PCA

At each refresh t, draw a uniform n_p-example subset $\widetilde{\mathcal{D}}_t$ from the pool without replacement. Under the current model θ_{t-1}, use the deltas in Eq. (3) to compute each layer's mean, centered covariance, and leading unit PCA direction:

$$\overline{\Delta h}_l^{(t)} := \frac{1}{n_p} \sum_{x \in \widetilde{\mathcal{D}}_t} \Delta h_l(x; \theta_{t-1}), \tag{11}$$

$$\widehat{\Sigma}_l^{(t)} := \frac{1}{n_p} \sum_{x \in \widetilde{\mathcal{D}}_t} (\Delta h_l(x; \theta_{t-1}) - \overline{\Delta h}_l^{(t)})(\Delta h_l(x; \theta_{t-1}) - \overline{\Delta h}_l^{(t)})^\top, \tag{12}$$

$$\tilde{v}_l^{(t)} \in \arg\max_{\|v\|_2=1} v^\top \widehat{\Sigma}_l^{(t)} v. \tag{13}$$

If the first-token state is constant across probe examples, Eq. (12) equals the covariance of final-token states. Orient $\tilde{v}_l^{(t)}$ toward the probe mean at the first refresh and toward $v_l^{(t-1)}$ thereafter, yielding $v_l^{(t)}$ for candidate scoring in Eq. (4). Numerical conventions are specified in Appendix A.2.

A.2 Sign calibration and numerical conventions

Our implementation computes PCA with a symmetric eigensolver, using the smaller Gram matrix when the probe has fewer examples than hidden dimensions. If the leading eigenvalue is at most 10^{-12}, it reuses the previous direction, or the probe mean on the first refresh, with a fixed coordinate axis if that vector is also numerically zero. It then normalizes and orients the vector. Inner products within 10^{-12} of zero use a fixed largest-magnitude-coordinate sign convention. Theorem 1 applies to consecutive PCA eigenvectors (Appendix B.2). Reusing the previous direction gives zero movement; the initial probe-mean and coordinate-axis fallbacks complete the numerical procedure.

A.3 Batched scoring and scheduled selection

The directions are constructed before scoring candidate batches. Each candidate-scoring forward pass supplies response loss, predictive entropy, and sequence-mean activations. Projections are reduced to one raw alignment per candidate, so the full pool's mean-activation matrices need not be stored.

For raw candidate alignments $\tilde{a}_i^{(t)}$, define pool extrema $m_t = \min_{1 \le i \le N} \tilde{a}_i^{(t)}$ and $M_t = \max_{1 \le i \le N} \tilde{a}_i^{(t)}$. The normalized score is

$$\mathrm{align}_i^{(t)} = \begin{cases} (\tilde{a}_i^{(t)} - m_t)/(M_t - m_t), & M_t - m_t \ge 10^{-8}, \\ 1/2, & \text{otherwise.} \end{cases} \tag{14}$$

The fallback makes the geometric multiplier common to all candidates and therefore preserves the carrier ranking.

Top-B selection breaks score ties by candidate index. The selected IDs are reused for U_t updates before rescoring (Algorithm 1).

B Supporting Mathematical Analysis

B.1 Properties of the bounded geometric adjustment

At a fixed refresh, let i, k index candidates and suppress the time index. For Eq. (5) with $\lambda \ge 0$, since $\mathcal{R}_i \ge 0$ and $\mathrm{align}_i \in [0, 1]$, $\mathcal{R}_i > (1+\lambda)\mathcal{R}_k$ implies $s_i \ge \mathcal{R}_i > (1+\lambda)\mathcal{R}_k \ge s_k$. Geometry can thus reorder closer base scores while preserving sufficiently separated preferences. Appendix B.4 gives a condition for preserving the entire selected set.

B.2 Proof of Theorem 1

We apply a Davis–Kahan residual argument (Davis & Kahan, 1970; Stewart & Sun, 1990) to the refreshed-probe covariances.

Proof. Fix a layer and refresh transition and write $A = \Sigma_l^{(t-1)}$, $C = \Sigma_l^{(t)}$, and $E = C - A$. Let $u = v_l^{(t-1)}$ and $v = v_l^{(t)}$ be their top unit eigenvectors, and set $\beta = \lambda_1(C)$. Choose an orthonormal eigenbasis $q_1 = u, q_2, \ldots, q_d$ of A, with eigenvalues in decreasing order, and expand $v = \sum_j c_j q_j$. Since $Cv = \beta v$,

$$\begin{aligned}\|Ev\|_2^2 &= \|(A - \beta I)v\|_2^2 \\ &\geq \sum_{j=2}^{d} (\beta - \lambda_j(A))^2 c_j^2 \\ &\geq \gamma_l^2 \sum_{j=2}^{d} c_j^2 = \gamma_l^2 (1 - c_1^2).\end{aligned}$$

The last inequality uses the cross-time separation $\beta - \lambda_2(A) \geq \gamma_l > 0$. Temporal sign calibration ensures $c_1 = \langle u, v \rangle \geq 0$. Consequently,

$$\|u - v\|_2^2 = 2(1 - c_1) \leq 2(1 - c_1^2) \leq \frac{2\|Ev\|_2^2}{\gamma_l^2} \leq \frac{2\|E\|_F^2}{\gamma_l^2}.$$

Taking square roots gives

$$\|u - v\|_2 \leq \frac{\sqrt{2}\,\|E\|_F}{\gamma_l},$$

yielding the stated bound. □

Spectral separation. Weyl's inequality gives

$$\lambda_1(C) - \lambda_2(A) \geq \lambda_1(A) - \lambda_2(A) - \|E\|_2.$$

Thus $\lambda_1(A) - \lambda_2(A) > \|E\|_2$ is sufficient for positive cross-time separation. The proof permits nonunique leading eigenvectors under its stated assumptions.

B.3 Two-sided spectral separation

For sign-aligned top unit eigenvectors u, v of covariances A, C, the preceding argument applies in either temporal ordering. Define

$$\begin{aligned} g_{\leftrightarrow}(A, C) &= \max\{\lambda_1(C) - \lambda_2(A),\ \lambda_1(A) - \lambda_2(C)\}, \\ b_{\leftrightarrow}(A, C) &= \frac{2\|C - A\|_F}{g_{\leftrightarrow}(A, C)} \quad (g_{\leftrightarrow} > 0). \end{aligned} \tag{15}$$

For $g_{\leftrightarrow} > 0$, apply the proof in Appendix B.2 in the temporal ordering attaining the maximum. Since displacement and $\|C - A\|_F$ are symmetric, $\|u - v\|_2 \leq b_{\leftrightarrow}(A, C)$. The fixed-probe diagnostic in Eq. (10) evaluates this bound retrospectively for $g_{\leftrightarrow} > 10^{-12}$, without clipping the denominator (Appendix F.3).

Covariances and implementation. At selection refreshes, covariance changes include model updates and probe resampling; the diagnostic holds probe membership fixed. The argument also applies to rank-deficient covariances; Appendix A.2 gives the Gram-matrix implementation.

B.4 From reference movement to selection scores

Consider consecutive refreshes on the same pool, with fixed budget B and weight $\lambda \geq 0$. Equation (8) bounds raw alignment changes from candidate and reference movement.

From raw alignment to normalized scores. Let a_i, a'_i denote raw alignments and α_i, α'_i their normalized values from Eq. (14). Let m, M and m', M' be the respective pool minima and maxima, and set $\epsilon_a = \max_i |a'_i - a_i|$. Suppose both ranges are at least $r_0 \geq 10^{-8}$, so both use the nonconstant normalization branch. The extrema each change by at most ϵ_a, and

$$\alpha'_i - \alpha_i = \frac{(a'_i - a_i) - (1 - \alpha_i)(m' - m) - \alpha_i(M' - M)}{M' - m'}, \qquad \max_i |\alpha'_i - \alpha_i| \leq \frac{2\epsilon_a}{r_0}.$$

From score changes to subset membership. Set $\epsilon_R = \max_i |\mathcal{R}'_i - \mathcal{R}_i|$, $R_{\max} = \max_i\{\mathcal{R}_i, \mathcal{R}'_i\}$, and $\epsilon_s = \max_i |s'_i - s_i|$. Equation (5) and the preceding bound give

$$\epsilon_s \leq (1 + \lambda)\epsilon_R + \frac{2\lambda R_{\max}\epsilon_a}{r_0}.$$

For $B < N$, denote the previous scores in descending order by $s_{(1)}, \ldots, s_{(N)}$, and define the boundary margin $\Delta_B = s_{(B)} - s_{(B+1)}$. If $\Delta_B > 2\epsilon_s$, then $s'_i - s'_k \geq \Delta_B - 2\epsilon_s > 0$ for every previously selected i and excluded k, so the selected subset is unchanged. This is a sufficient condition and is not expected to bind in practice; the observed 26–33% replacement between refreshes (Appendix F) is consistent with it not holding.

C Experimental Protocols

The main performance runs, fixed-probe diagnostics, and ReGIFT-M controls answer different questions and use the configurations below.

Table 4: **Experimental cohorts and their roles.**

Cohort	Purpose	Configuration reference
Main and transfer runs	Downstream performance and reported costs	Table 5; Appendix C.3
Fixed-probe diagnostics	Reference movement and selection behavior	Qwen2.5-0.5B + LoRA; Appendix F.3
ReGIFT-M controls	Geometric intervention and length controls	Separate LLaMA-2-7B recipe; current-probe-mean sign calibration

C.1 Training configurations and experimental cohorts

Table 5 gives the common target-SFT settings and ReGIFT parameters for the main and transfer runs in Section 3. Method-specific data, supervision, and preparation differences are specified in Appendix C.3.

ReGIFT-M control configuration. The controls in Appendices E.2–E.4 use a separate LLaMA-2-7B / Alpaca-GPT4 setting with $N = 52{,}002$, $B = 5{,}200$ per epoch, and $\lambda = 1$. We label this mean-calibrated implementation ReGIFT-M (M for mean): at every refresh, the PCA direction's sign is aligned with the *current probe mean delta*. The main ReGIFT implementation instead aligns the sign with the previous direction after the first refresh (Section 2.2). Both recompute the PCA reference at each refresh. These controls refresh once per epoch, using 2,048 probe examples, all 32 layers, and PCA batch size 16. The ranking is $\mathcal{R}_i(1 + \text{align}_i)$.

These control runs use four epochs of response-only, full-parameter SFT, cutoff 512, the `alpaca_default` prompt, bf16 weights, and 8-bit AdamW. The learning rate is 2×10^{-5}, with cosine decay to zero, warmup ratio 0.03, weight decay 0.1, and gradient clip 1.0. Effective batch size is $8 \times 16 \times 1 = 128$; gradient checkpointing is enabled, and attention uses SDPA. The control runner retains PPO bookkeeping, with rankings determined by ReGIFT-M; the reported runtime comparison uses the main runs (Appendix G).

C.2 Instruction pools and evaluation

Evaluation uses `lm-evaluation-harness` (Gao et al., 2023); AVG and AVG4 follow Section 3.

Table 5: **Main and transfer hyperparameter settings.**

Supervised fine-tuning		*Dynamic selection & ReGIFT*	
Parameter	Value	Parameter	Value
Optimizer	AdamW (8-bit)[a]	Selection ratio ρ	0.10
Precision	bf16	Alignment weight λ	1.0
Learning rate	2×10^{-5} [b]	Probe size n_p	1024
LR scheduler	cosine	PCA layers	all
Warmup ratio	0.03	PCA batch size	4
Weight decay	0.1	Episode batch size	16
Gradient clip	1.0		
Gradient checkpoint	enabled		
Cutoff length	1024		
Epochs	3		
Effective batch size	128 [c]		

[a] Bitsandbytes 8-bit AdamW; falls back to fp32 AdamW if the library is unavailable.
[b] Qwen2.5-7B uses 1×10^{-5} per its official SFT recipe.
[c] Per-GPU batch $8\times$ grad-accum $4\times$ DDP world-size 4.

Instruction-tuning pools. Alpaca-GPT4 pairs 52K original Alpaca instructions with GPT-4 responses (Peng et al., 2023). WizardLM Evol-Instruct increases instruction complexity and broadens task coverage through iterative rewriting (Xu et al., 2024).

Benchmark scoring. MMLU covers academic and professional subjects and is scored by multiple-choice accuracy (Hendrycks et al., 2021). BBH evaluates challenging symbolic, logical, and commonsense reasoning tasks using exact-match accuracy (Suzgun et al., 2023).

GSM8K and SVAMP are arithmetic word-problem benchmarks. GSM8K focuses on grade-school math reasoning; SVAMP tests robustness to variations in problem wording and structure. Both use final-number exact match after normalization (Cobbe et al., 2021; Patel et al., 2021).

HumanEval and MBPP evaluate Python code generation from task descriptions, function signatures, or examples. Both use pass@1 under execution-based evaluation (Chen et al., 2021; Austin et al., 2021).

TyDiQA and XQuAD evaluate multilingual question answering from context passages. TyDiQA covers typologically diverse languages; XQuAD provides parallel SQuAD-style examples across languages. Both use normalized token-overlap F1 (Clark et al., 2020; Artetxe et al., 2020).

For TyDiQA Gold Passage, we use all nine languages and all 5,077 development examples, with five training demonstrations drawn from the same language as each query. We use greedy batch-one decoding and report normalized token-overlap F1 pooled over development examples.

Evaluation of the ReGIFT-M controls. We evaluate each control run at the final epoch using the same pipeline: MMLU accuracy (14,042 examples), SVAMP exact match (300), sanitized MBPP pass@1 (257), and TyDiQA pooled F1 (5,077). Generative evaluation uses batch size one. MBPP uses one sampled completion per problem, temperature 0.8 and $p = 0.95$, so seed variation includes decoding randomness. We compute each run's AVG4 from full-precision percentage scores, then report the across-seed mean and sample SD (denominator $n - 1$).

C.3 Baseline implementations and adaptations

Table 2 reports the implementations described here. Ranked subset methods use a 10% selection ratio (AtteNT after its full-data first epoch). Table 6 records adaptations of CPQS (Ren et al., 2026), AtteNT (Zhang et al., 2026), ADAPT (Zhao et al., 2026), and ssToken (Qin et al., 2026); other implementation details follow below.

Other implementation details. LIMA (Zhou et al., 2023) trains on its published 1,030-example human-curated corpus. AlpaGasus (Chen et al., 2024) uses the official ChatGPT-filtered set (rat-

Table 6: **Adaptations of the additional baselines.** Input budgets are fractions of the example pool.

Method	Intervention / refresh	Auxiliary resource	Input / supervision
CPQS	Example subset; classifier trained once, pool scored once, subset fixed before target SFT	Quality classifier over frozen target-model hidden states, trained on 5K Alpaca-GPT4 (GPT-4) responses as positives and 10K Qwen2.5-0.5B regenerated responses as negatives (the paper uses Llama-3.2-1B and Qwen2.5-1.5B negatives)	10% examples; all response tokens
AtteNT	Example subset; response-loss re-ranking at every epoch boundary after a full-data first epoch (the paper's LLM recipe trains on all data in epoch 1, then on a 70% loss-selected subset)	Current target-model response loss; no other resource	First epoch 100%; later epochs 10%
ADAPT	Example-loss weights recomputed every optimizer step; reference anchors refreshed every step	256 single-turn UltraFeedback-binarized examples, disjoint from the pool and from all evaluation tests (the paper uses MMLU as its reference)	100% inputs; weighted example losses
ssToken	Response-token selection per sample and step	Per-token loss cache from the initial model; current-model attention	100% inputs; 60% response tokens

ing threshold 4.5; 9,229 examples filtered from the original Alpaca-52K data) instead of a top-K selection of 5,200. Both rows are therefore shown without a rank in Table 2.

Q2Q (Li et al., 2024) scores instruction-following difficulty with the base model directly, whereas the paper's default first trains a 1,000-example, one-epoch "pre-experienced" model. SelectIT (Liu et al., 2024a) uses token- and sentence-level self-reflection with the 7B model only and selects 10%; the paper's model-level variant ensembles 7B, 13B, and 70B models and selects 20%. NAIT (Chen et al., 2026) follows the paper's setting (10% of Alpaca-GPT4 on LLaMA-2-7B) and requires a human-chosen in-domain seed set per capability.

AtteNT is an independent implementation of the paper's response-loss LLM recipe, adapted from five-epoch LoRA to our three-epoch full-parameter SFT (Zhang et al., 2026). Our ssToken implementation uses eager attention rather than the original attention-recomputation optimization (Qin et al., 2026); full-input forward and attention computation remain part of its workload. Input, supervision, and preparation budgets differ across these methods; Appendix G accounts for their costs.

D Additional Performance Results

We vary the selection budget, model backbone, and instruction pool to examine the operating choices and reuse of the selector in Section 3. The ratio sweep reports eight tasks; transfer results average MMLU, SVAMP, MBPP, and TyDiQA (AVG4). Training and evaluation protocols are given in Appendix C.

D.1 Selection-budget sensitivity

This experiment asks how much performance improves simply by selecting more examples at random. We vary Random's selection ratio from 10% to 50% while holding the other training hyperparameters fixed. The sweep provides a budget-only comparison for the 10% operating point used in the main experiments.

Random's AVG rises from 38.00 at 10% to its highest value of 38.86 at 30%, then reaches 38.83 and 38.60 at 40% and 50%, respectively (Table 7). The task-level responses differ: MMLU and BBH

Table 7: **Random selection-ratio sweep on LLaMA-2-7B + Alpaca-GPT4.** Budget-only controls with all other hyperparameters held fixed. AVG is the unweighted mean of the eight displayed task scores.

	Know.		Math		Code		MLing.		
ρ	MMLU	BBH	SVAMP	GSM8K	MBPP	H-Eval	TyDiQA	XQuAD	AVG
10%	45.04	35.54	36.80	13.57	35.41	24.51	60.83	52.28	38.00
20%	45.53	35.72	38.33	15.92	34.24	24.13	61.31	53.31	38.56
30%	45.91	36.23	37.33	14.10	36.96	25.31	61.32	53.73	38.86
40%	46.21	36.37	40.00	14.33	35.02	24.02	61.29	53.37	38.83
50%	46.31	36.42	38.00	13.95	33.85	25.64	60.98	53.67	38.60

improve across the grid, while the math and code tasks reach their highest scores at different ratios. Increasing the selected fraction therefore changes the task profile as well as the average.

ReGIFT achieves 40.70 AVG with 10% selected per refresh (Table 2), exceeding the best Random result in this sweep by 1.84 points. At the same 10% ratio, the difference is 2.70 points. These comparisons connect the budget choice to the paper's central question: which examples to train on matters beyond how many are selected. The 10% setting exposes this benefit while keeping each selected training set small.

D.2 Transfer across model families and scales

This experiment asks whether the selection rule can be reused when the backbone and its internal representations change. We hold the Alpaca-GPT4 pool, $\rho = 10\%$, and $\lambda = 1$ fixed across Mistral-7B, Qwen2.5-7B, and Qwen2.5-0.5B. Each model constructs its own reference from its current pool representations at every refresh, so the ranking rule and adjustment weight are reused while the geometric directions and candidate scores are recomputed.

Table 8: **Multi-backbone results on Alpaca-GPT4 (0.5B–7B).** AVG4 is the mean of the four displayed tasks. Bold/underline denote best/second-best distinct values within each backbone, including Full FT; ties receive the same rank.

Method	MMLU	SVAMP	MBPP	TyDiQA	AVG4
Mistral-7B					
Full FT (100%)	53.13	<u>58.24</u>	**51.22**	50.16	53.19
Random (10%)	56.80	57.00	48.64	<u>53.29</u>	53.93
Data Agent (PPO, 10%)	<u>57.38</u>	56.67	50.28	**54.36**	<u>54.67</u>
ReGIFT (λ=1, 10%)	**57.93**	**61.33**	<u>50.97</u>	53.09	**55.83**
Qwen2.5-7B					
Full FT (100%)	**73.93**	<u>83.33</u>	74.31	<u>55.78</u>	<u>71.84</u>
Random (10%)	71.92	83.00	**75.48**	54.10	71.13
Data Agent (PPO, 10%)	73.59	83.01	**75.48**	53.66	71.44
ReGIFT (λ=1, 10%)	<u>73.68</u>	**83.66**	<u>75.10</u>	**56.83**	**72.32**
Qwen2.5-0.5B					
Full FT (100%)	**47.85**	41.33	41.63	40.27	42.77
Random (10%)	47.71	40.68	43.19	39.88	42.87
Data Agent (PPO, 10%)	47.46	<u>44.00</u>	<u>46.30</u>	<u>41.04</u>	<u>44.70</u>
ReGIFT (λ=1, 10%)	<u>47.77</u>	**45.33**	**47.08**	**42.73**	**45.73**

ReGIFT has the highest AVG4 on all three backbones (Table 8). Its gains over Full FT are 2.64 points on Mistral-7B, 0.48 on Qwen2.5-7B, and 2.96 on Qwen2.5-0.5B; the corresponding gains over Data Agent are 1.16, 0.88, and 1.03 points. The aggregate gains come from different task combinations: ReGIFT leads MMLU and SVAMP on Mistral-7B, SVAMP and TyDiQA on Qwen2.5-7B, and SVAMP, MBPP, and TyDiQA on Qwen2.5-0.5B.

This variation is relevant to maintaining models of different sizes and families: the same selector configuration works with each model's own evolving geometry. The experiment supports reusing the

selection procedure across these backbones without a new alignment-weight sweep or a separately curated reference set. It connects the model-internal reference design to a concrete preparation benefit when adapting the SFT pipeline to another backbone.

D.3 Transfer to Evol-Instruct

This experiment asks whether the procedure remains useful when the instruction-generation process changes. Alpaca-GPT4 pairs the original Alpaca instructions with GPT-4 responses, whereas Evol-Instruct iteratively increases instruction complexity and broadens task coverage (Peng et al., 2023; Xu et al., 2024). For each backbone, we rerun selection and SFT on Evol-Instruct with $\rho = 10\%$ and $\lambda = 1$. The transferred configuration is the selection rule and its weight; reference directions and subsets are constructed anew from the current model and replacement pool.

Table 9: **Instruction-dataset transfer on Evol-Instruct** at ρ=10%. AVG4 and ranking follow Table 8; training configurations are given in Appendix C.1.

Method	MMLU	SVAMP	MBPP	TyDiQA	AVG4
LLaMA-2-7B					
Full FT (100%)	<u>45.63</u>	37.67	35.41	<u>64.10</u>	45.70
Random (10%)	45.50	**38.67**	**38.52**	64.00	<u>46.67</u>
Data Agent (PPO, 10%)	44.89	38.27	<u>38.44</u>	63.65	46.31
ReGIFT (λ=1, 10%)	**46.25**	<u>38.45</u>	37.69	**64.96**	**46.84**
Qwen2.5-7B					
Full FT (100%)	**71.79**	81.00	75.88	57.80	71.62
Random (10%)	<u>71.71</u>	<u>83.33</u>	75.88	**59.10**	**72.51**
Data Agent (PPO, 10%)	71.02	82.74	**76.45**	<u>58.79</u>	72.25
ReGIFT (λ=1, 10%)	71.68	**84.16**	<u>76.02</u>	58.04	<u>72.48</u>

ReGIFT exceeds Full FT and Data Agent on both backbones (Table 9). On LLaMA-2-7B, its 46.84 AVG4 improves on them by 1.14 and 0.53 points, respectively; on Qwen2.5-7B, its 72.48 AVG4 improves on them by 0.86 and 0.23 points. Thus the average gains over these two baselines persist when the instruction pool is replaced and the selection procedure is rerun.

Random is close in this setting: ReGIFT leads it by 0.17 points on LLaMA-2-7B and trails it by 0.03 on Qwen2.5-7B. On LLaMA-2-7B, higher MMLU and TyDiQA scores offset lower math and code scores; on Qwen2.5-7B, the SVAMP gain is offset mainly by lower TyDiQA performance. These task patterns account for ReGIFT's first- and second-place AVG4 rankings. For a pipeline that changes instruction sources, the practical result is reuse of the selection procedure with newly computed pool geometry, without preparing a new reference set or retuning λ.

E Ablations of Geometric Selection

The main CO comparison connects geometric adjustment to higher downstream performance (Section 3.3). Here we examine how that adjustment operates: how strongly to weight it, which examples it moves across the selection boundary, and how its ranking relates to sequence length. This sequence of experiments connects the score in Eq. (5) to the training data it allocates and the resulting task scores. The weight sweep uses the main LLaMA-2-7B setting; the subsequent controls use mean-calibrated ReGIFT-M (Appendix C.1).

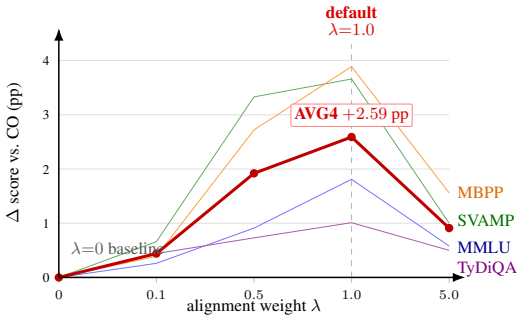
E.1 Alignment-weight sensitivity

We vary λ to assess how strongly geometry should adjust the loss–entropy ranking. At $\lambda = 0$, selection uses the carrier alone; positive values increase the range of the alignment multiplier while retaining the carrier. Figure 3 compares $\lambda \in \{0, 0.1, 0.5, 1, 5\}$ at the same 10% selection budget.

Results and choice of adjustment strength. Every positive weight improves all four task scores over CO. AVG4 rises from 44.33 at $\lambda = 0$ to 46.92 at $\lambda = 1$, with gains of 1.81 points on MMLU,

3.66 on SVAMP, 3.89 on MBPP, and 1.01 on TyDiQA. At $\lambda = 5$, AVG4 is 45.24: increasing the weight beyond the best tested setting reduces the gain. The sweep therefore favors a moderate geometric adjustment, with improvements across all four tested positive weights.

At $\lambda = 1$, the multiplier lies in $[1, 2]$. This gives geometry room to reorder candidates with comparable carrier scores while preserving sufficiently separated carrier preferences (Appendix B.1). At $\lambda = 5$, the range expands to $[1, 6]$, giving alignment more influence over the ordering. The empirical peak at $\lambda = 1$ connects the bounded adjustment in Section 2.3 to a practical default. All model and pool transfers retain $\lambda = 1$, evaluating reuse of this choice across the settings in Appendix D.



λ	MMLU	SVAMP	MBPP	TyDiQA	AVG4
0 (CO)	45.90	36.67	34.63	60.12	44.33
0.1	46.16	37.33	35.02	60.56	44.77
0.5	46.81	40.00	37.35	60.85	46.25
1.0	**47.71**	**40.33**	**38.52**	**61.13**	**46.92**
5.0	46.48	37.67	36.19	60.62	45.24

Figure 3: **Alignment-weight sensitivity.** LLaMA-2-7B + Alpaca-GPT4 at $\rho = 10\%$. Plot: gains over CO; table: absolute scores and four-task AVG4. Each setting is a single run.

E.2 Removing geometry at a fixed checkpoint

We measure whether the bounded multiplier changes which examples receive the fixed training budget. Holding a checkpoint and its carrier scores fixed, removing the geometric adjustment replaces **16.63–34.12% of the selected examples** across four refreshes (Table 10). This intervention isolates geometry's contribution to subset membership at each refresh; Appendix E.3 then evaluates training on the selected data.

Same-checkpoint intervention. For seed 42, we obtain the full-pool signals at each pre-selection checkpoint and verify that they exactly reproduce all four training-time selections. For each refresh t, keep θ_{t-1}, $\mathcal{R}^{(t)}$, the pool, and B fixed, and define

$$S_t = \mathrm{TopB}_i\{\mathcal{R}_i^{(t)}(1 + \mathrm{align}_i^{(t)})\}, \qquad S_t^{\mathrm{CO}} = \mathrm{TopB}_i\{\mathcal{R}_i^{(t)}\}, \tag{16}$$

$$q_t = 1 - \frac{|S_t \cap S_t^{\mathrm{CO}}|}{B}, \qquad J_t = \frac{|S_t \cap S_t^{\mathrm{CO}}|}{|S_t \cup S_t^{\mathrm{CO}}|}. \tag{17}$$

Here CO reranks each fixed checkpoint using only its carrier scores. The replaced count is Bq_t on each side; the symmetric difference has size $2Bq_t$.

Table 10: **Removing the geometric adjustment changes the selected subset.** Four same-checkpoint comparisons along one seed-42 run; 5,200 examples per set.

Refresh	Shared	Replaced per set	Replaced (%)	Jaccard
1	4,335	865	16.63	0.715
2	3,706	1,494	28.73	0.554
3	3,513	1,687	32.44	0.510
4	3,426	1,774	34.12	0.491

Which examples change? The replaced share rises from 16.63% at the first refresh to 34.12% at the fourth: geometry reallocates approximately one in six and one in three of the 5,200 training slots, respectively. Thus the bounded adjustment makes a substantial change at the selection boundary and remains active across the four checkpoints. At epoch 4, admitted examples have mean alignment

0.863 and response loss 1.148, versus 0.479 and 1.289 for displaced examples. Geometry therefore promotes some lower-loss, higher-alignment candidates. This gives the pool-derived reference a concrete role in the learning procedure: it redirects the same example budget toward candidates favored by the current representation geometry. These membership changes explain how the score modification in Section 2.3 translates into different training inputs.

E.3 Comparison with length-matched random selection

At a fixed example count, selecting longer sequences changes both the amount of supervised text and the work per example. We therefore match selected input and response lengths and vary which examples fill that length profile. This tests the complete selector against random choices with comparable sequence-length exposure. ReGIFT-M achieves a three-seed AVG4 of **46.04%**, versus **45.00%** for the matched control, a mean difference of **+1.04 percentage points** (Table 11).

Length matching and paired design. We use seeds $42, 1, 7$ for selection, training, and evaluation. For each seed and epoch, the corresponding ReGIFT-M subset determines quotas in joint input/response length strata. Lengths are measured *after* tokenization and truncation, excluding padding; response length counts supervised tokens. The strata use two-token bins,

$$\left(\lfloor \ell_{\text{in}}/2 \rfloor, \lfloor \ell_{\text{resp}}/2 \rfloor, \mathbf{1}\{\ell_{\text{in}} + \ell_{\text{resp}} = 512\}, \mathbf{1}\{\ell_{\text{resp}} = 0\}\right).$$

The control draws uniformly without replacement within each stratum and preserves the reference stratum at each selected position. Its independent sampling RNG does not consume training RNG draws. We train three controls with the same optimization recipe and pair each with the ReGIFT-M run of the same seed.

Across the 12 matched subsets (three seeds, four epochs), the largest input- or response-length Kolmogorov–Smirnov distance is 0.00673, and the largest absolute mean-length difference is 0.01635 tokens. Fully selected strata force some examples into both sets (at most 11.13% of the budget).

Evaluation and aggregation. Both conditions use the final-epoch evaluation protocol in Appendix C.2. Table 11 reports task means, AVG4 mean and sample SD, and all paired seed results.

Table 11: **Performance after matching selected sequence lengths.** Final-epoch scores (%); $n = 3$ paired seeds. Panel A gives per-task means and AVG4 mean±sample SD. Panel B gives paired AVG4 for each seed; Δ = ReGIFT-M − control is in percentage points. Averages and differences use unrounded scores.

A. Three-seed means

Method	MMLU	SVAMP	MBPP	TyDiQA	AVG4
ReGIFT-M	46.05	38.78	38.26	61.06	46.04 ± 0.46
Length-matched Random	46.18	39.56	33.85	60.39	45.00 ± 0.47
Δ (percentage points)	-0.14	-0.78	+4.41	+0.67	+1.04

B. Paired AVG4 for every seed

Seed	ReGIFT-M	Matched Random	Δ
42	46.170	44.632	+1.538
1	46.418	44.826	+1.592
7	45.524	45.528	-0.004

Paired differences have sample SD 0.907 points.

Interpretation. The mean advantage is concentrated in MBPP (+4.41 points) and TyDiQA (+0.67); MMLU (-0.14) and SVAMP (-0.78) favor the control. The paired AVG4 gains are +1.54 and +1.59 points for seeds 42 and 1; the seed-7 difference is -0.004 points. With closely matched length profiles, ReGIFT-M achieves a higher average than random choices under the same training recipe. This makes the control relevant to pipelines that already manage sequence lengths for batching or token budgets: the combined selector improves average performance while retaining a comparable selected-length profile.

Together with the fixed-checkpoint intervention, these results connect changes in subset membership to the value of the complete selection procedure. Matched Random replaces both carrier and geometric ranking. To examine the information used by the geometric adjustment itself, the next controls retain the carrier and change only the alignment rule.

E.4 LENGTH-PREDICTED AND SHUFFLED ALIGNMENT

These controls separate length-predictable alignment from its assignment to individual candidates. Both retain the carrier-scoring rule, $\lambda = 1$, and the ReGIFT-M training recipe in Appendix C.1.

What each intervention changes. The length-predicted control uses five-fold out-of-fold prediction from input and response lengths, with geometry-derived alignment at the current checkpoint as its target. The control runner retains the original reference computation, reward calculation, and PPO bookkeeping. The shuffled control permutes alignment within length strata. This preserves the alignment values available within each stratum while changing which candidate receives each value. We use the four-task suite, one task per domain, to limit evaluation cost.

Table 12: **Length-based prediction and shuffling of geometric alignment.**

Alignment signal	AVG4 (mean $\pm$ sample SD)
REGIFT-M alignment	46.04 ± 0.46
Length-predicted alignment	46.16 ± 0.12
Within-length shuffled alignment	45.72 ± 0.65

Three seeds; separate ReGIFT-M cohort; four-task scores in percent.

Results and implications for selection design. Length-predicted alignment achieves the highest mean AVG4 (46.16), followed by ReGIFT-M (46.04) and within-length shuffled alignment (45.72). The proxy is 0.12 points above the original alignment, while shuffling is 0.32 points below it.

Across the 12 proxy-run refreshes, out-of-fold alignment prediction has $R^2 = 0.991$–0.992. At each of these checkpoints, the proxy and original alignment select sets with 93.2–97.9% overlap. In this ReGIFT-M cohort, lengths closely reproduce geometry-derived alignment and subset membership, alongside similar observed downstream averages. The predictor is supervised by the geometric alignment at each checkpoint, so this result characterizes the length-related component of the model-derived score; it does not evaluate a selector that omits the geometric computation. Length-only scoring without geometric supervision, and simpler approximations of the score more generally, are natural comparisons for future work.

Source of the length relation. We ran one additional measurement at a single checkpoint to locate this relation: the base Qwen2.5-0.5B model scores a uniform pool of 483 Alpaca-GPT4 examples (probe size 96, cutoff 320, $\lambda = 1$, 10% selection). Input and response lengths and their logarithms predict alignment with out-of-fold $R^2 = 0.79$, and the strongest single correlate is inverse sequence length ($r = -0.90$). In layers 3–20, the first-token state $h_l^{(1)}$ projects onto the reference direction at 33–135 times the pool standard deviation of the sequence-mean projection, and it enters $\bar{h}_l$ with weight $1/K_x$ (Eq. (3)). Excluding this token from the mean lowers the out-of-fold R^2 to 0.22, so mean pooling over the first-token state accounts for a large part of the length relation. At this checkpoint, the geometric adjustment replaces 33% of the carrier-only subset and raises the mean selected response length from 64 to 127 tokens (pool mean 134), while sharing 4% of its subset with longest-response selection.

F SELECTION CHANGES AND REFERENCE MOVEMENT

We examine how ReGIFT updates its training subset and how its geometric reference moves. Figure 4 separates changes across training refreshes from changes caused by random resampling at a fixed model. Figure 5 visualizes the reference summary in Table 1, including all three seeds.

F.1 WHY DO SELECTED EXAMPLES CHANGE?

We compare ReGIFT and Data Agent on Qwen2.5-0.5B with three seeds (setup in Appendix F.3). The replacement rate is the percentage of an old selection absent from a new selection of the same size. Figure 4(a) compares training refreshes; panel (b) repeats Data Agent's action sampling with the model and policy fixed.

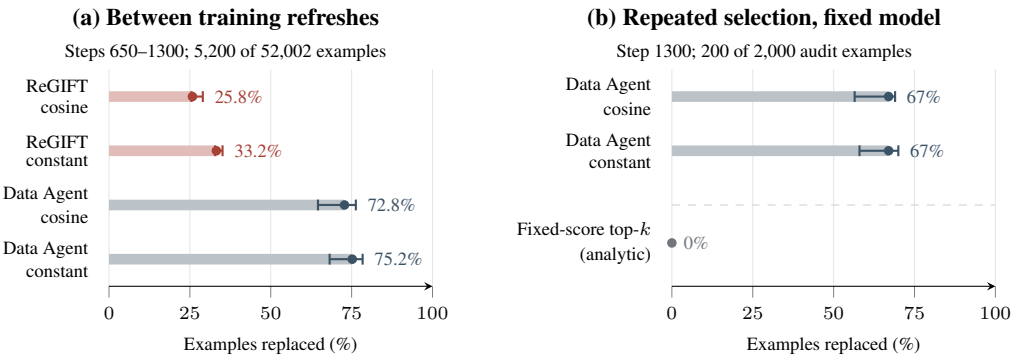


Figure 4: **Subset changes during training and at fixed state.** (a) Replacement between two training refreshes. (b) Data Agent resampling with the model and policy fixed; deterministic top-k with fixed scores provides the analytic zero reference. Dots show three-seed medians; whiskers show seed ranges.

Changes between training refreshes. Between steps 650 and 1300, ReGIFT replaces 25.8% of selected examples under cosine decay and 33.2% under a constant rate; Data Agent replaces 72.8% and 75.2%. ReGIFT thus updates roughly one quarter to one third of its training subset while retaining most previously selected examples. This shows the refreshed ranking in Section 2.3 adapting its data choices during training.

Sampling also changes a selection without a model update. At step 1300, we freeze the model, policy, and a 2,000-example audit pool, then select 200 examples in each of 12 draws. Within each seed, we compare draws 2–12 with draw 1 and take the median replacement; the figure reports the median and range across seeds. The median is 67% under both schedules: **about 134 of the 200 selected examples change when sampling is repeated at the same model**.

Fixed-score top-k with deterministic tie handling returns the same set, giving the analytic zero reference in the figure. ReGIFT uses this deterministic rule once its model, probe, and scores are fixed. This separates adaptation through updated selection inputs from the sampling variation discussed alongside Figure 1(b).

F.2 HOW FAR DOES THE REFERENCE MOVE?

We track the PCA direction on the same probe as the model learns. The typical 25-step direction change in Table 1 corresponds to about 1.65° under cosine decay and 3.18° under a constant rate. The monitored directions exhibit small typical rotations, with larger movement when the learning rate stays constant.

The bound in Eq. (10) is *informative* when it narrows the generic 90° limit. For example, a 30° upper bound on an observed 3° turn restricts the allowable movement. Such informative bounds cover 86% and 74% of the retained layer-transition pairs under cosine and constant schedules, respectively (three-seed medians). These are the two quantities in Table 1: how far the reference actually moves, and how often the bound restricts that movement.

F.3 DIAGNOSTIC PROTOCOL

We use Qwen2.5-0.5B + LoRA on Alpaca-GPT4, with seeds 42/43/44. Each run has 1,950 optimizer updates; selection keeps 5,200 examples and refreshes at steps 0/650/1300. Cosine and constant

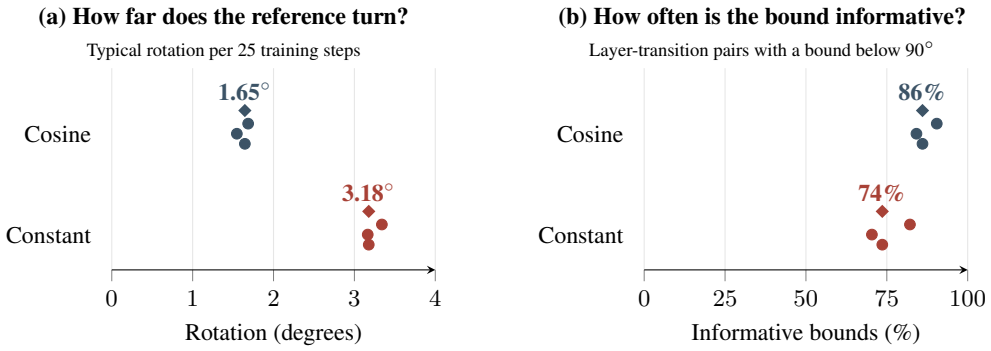


Figure 5: **Reference movement and informative bounds.** Each small dot is one seed; diamonds and labels show medians across three seeds. (a) Rotation angles computed from the per-run displacement summaries. (b) Fraction of the 1,496 layer-transition pairs per seed whose bound is tighter than the generic 90° limit.

schedules share a peak learning rate of 2×10^{-4} and 195 warm-up steps; their cumulative applied rates are 0.1950 and 0.3704, respectively. The common SFT recipe uses standard AdamW, global batch 8, cutoff 512, and LoRA rank 16, scale 32, and dropout 0.05. Data Agent also trains its policy. ReGIFT constructs selection directions from a resampled 1,024-example probe; reference measurements use a fixed 2,000-example probe (seed 4242).

Reference measurements. The passive monitor restores model modes and RNG state and measures temporally sign-aligned directions every 25 updates from step 0 to 1950. The analysis uses layers 2–23 and 68 post-warm-up transitions ending at steps 275–1950: $22 \times 68 = 1,496$ pairs per seed. Selection uses all layers. Each run's displacement is the time median of the layer-median displacement (Eq. (9)); Table 1 takes the median across seeds. The informative fraction counts all 1,496 pairs and requires spectral separation $> 10^{-12}$ and bound $< \sqrt{2}$. All six ReGIFT runs satisfy the separation condition on every retained pair, with zero bound violations at 10^{-7} tolerance.

G Computational Cost

G.1 Measured time and estimated budgets

Figure 6 shows measured selection and SFT times alongside the Full FT linear estimate; Table 13 summarizes analytical budgets. Together they cover all 15 methods in Table 2. Input budgets and local adaptations follow Appendix C.3. The accounting excludes shared corpus construction, evaluation, checkpoint storage, and environment setup.

NAIT requires a separately curated seed set and selection before SFT; its measured 1.18 h selection time excludes seed curation. Data Agent and ReGIFT interleave scheduled selection with SFT updates. ReGIFT obtains candidate representations during the same forward passes used for loss–entropy scoring, avoiding a separate full-pool representation pass. It refreshes its reference from the pool without curated seeds, an external judge, or a learned selection policy.

Probe extraction, PCA, pooling, and projection are included in ReGIFT's reported selection time. Between refreshes, it reuses the selected examples for ordinary SFT, spreading selection work over successive updates. This shared computation and automatic reference refresh fit repeated SFT workflows: selection time is 37.9% lower than Data Agent's, and total time is 33.0% lower, including all refresh costs.

G.2 Accounting assumptions for additional methods

Let $N = 52{,}002$ be the pool size, $\rho = 0.1$ the selected fraction, $e = 3$ the number of SFT passes, and $H_{10} = 0.64$ h the reported ReGIFT target-SFT duration used for calibration. Let q denote the assumed training-to-forward compute ratio. We use $q = 4$ as a checkpointed-training accounting scenario and also show $q = 3$ sensitivity below. Define F as the equivalent budget for one target-

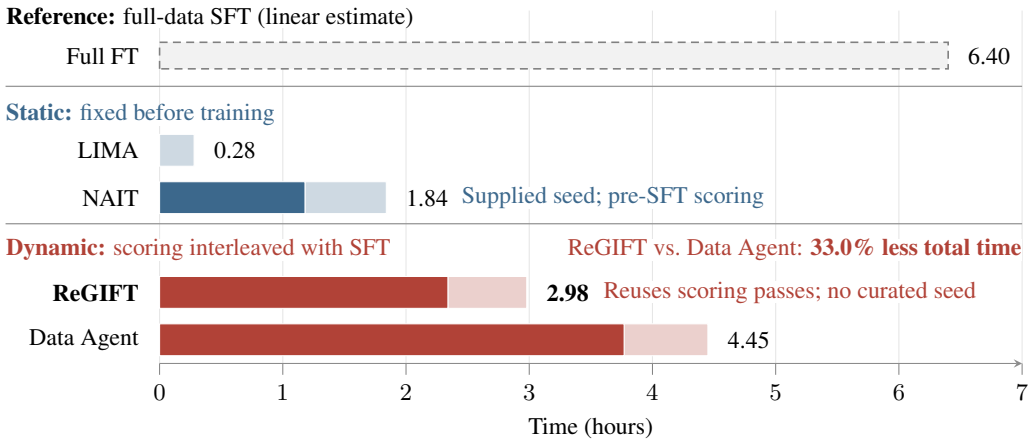


Figure 6: **Selection and fine-tuning time.** Full FT: linear estimate $(0.64/0.1)$; other bars: single-run A100 hours. Dark/light: selection/preparation and SFT. LIMA's corpus and NAIT's seed set are supplied; human preparation and evaluation are excluded.

Table 13: **Estimated budgets for the other methods.** Groups follow Table 2; subtotals include selection and SFT. $\geq$ marks a partial cost floor.

Method	Budget (h)	Basis
References		
No FT	0	No selection or SFT
Full FT	6.40	Linear scaling from 10% SFT
Random	0.64	10% SFT; CPU sampling excluded
S. Static selection: data fixed before SFT		
S1 AlpaGasus	19.70	Published API estimate + measured SFT
S2 Q2Q (IFD)	1.69	Two scoring passes + measured SFT
S2 SelectIT (7B)	3.31	Five scoring prompts + measured SFT
S3 CPQS	$\geq$ 1.33	CNN fitting and negative generation additional
D. Dynamic selection: subset refreshed during SFT		
AtteNT	3.63	Full first epoch, then two 10% epochs
CO ($\lambda = 0$)	2.24	Three pool-scoring passes + 10% SFT
F. Full-input methods: different supervision budgets		
ADAPT	19.22	Full inputs; 256 references refreshed per update
ssToken	$\geq$ 6.93	Full inputs, 60% response supervision; eager attention

model forward pass over the pool:

$$F = \frac{H_{10}}{e\rho q} = 0.533 \text{ h}, \qquad H_{100} = \frac{H_{10}}{\rho} = 6.40 \text{ h}. \tag{18}$$

These *work-normalized budgets* assume equal processed length, linear scaling, and unchanged effective throughput, using the configuration in Appendix C.1.

Per-method accounting. No FT has no selection or SFT workload. Full FT uses H_{100}; Random uses H_{10} without model-based ranking, while CPU sampling remains outside the model-compute estimate. The following operation counts specify the local adaptations in Appendix C.3.

- Q2Q uses conditional and response-only scoring, budgeted as $2F$, plus its reported 0.62 h SFT. The omitted pre-experience stage is documented in Appendix C.3.
- SelectIT uses five scoring prompts with a single 7B model, budgeted as $5F$, plus 0.64 h reported SFT.
- CO uses three full-pool scoring rounds, $3F$, and H_{10} SFT.

- AtteNT's local three-epoch recipe uses the full pool first, then 10% in each subsequent epoch. Target SFT is therefore $4H_{10} = 2.56$ h, with $2F$ for the two rescoring rounds.
- ADAPT retains all inputs and refreshes 256 references at each optimizer update in the registered adaptation. With effective batch 128 and four ranks, $U = 3\lceil N/128 \rceil = 1{,}221$ updates require $4 \cdot 256U/N = 24.04$ pool-forward equivalents because the reference pass is replicated on every rank. This gives $24.04F = 12.82$ equivalent hours plus H_{100} for the local refresh rule (Zhao et al., 2026).
- ssToken incurs one initial loss-cache pass, F, and retains full-input forward/backward computation. We use H_{100} as its SFT compute floor. The local eager-attention implementation adds work beyond this floor; the original method uses optimized attention recomputation (Qin et al., 2026).

G.3 External preparation and sensitivity analysis

Human and API preparation. LIMA reuses its published human-curated corpus (Zhou et al., 2023); its zero preparation entry reflects this reuse. Human preparation is excluded from the reported times.

AlpaGasus scores the candidate pool before retaining about 9K examples (Chen et al., 2024). SelectIT's Table 9 extrapolates 500 API-scored examples to the full Alpaca-GPT4 pool, reporting 19.07 h and $52.02 for ChatGPT scoring (Liu et al., 2024a). We use 19.07 h as a published scoring estimate; adding 0.63 h of measured local SFT gives 19.70 h under serial execution. Cached scores avoid repeat API calls.

CPQS preparation and sensitivity. The local CPQS recipe uses 5K existing positive examples and 10K small-model-generated negatives. One feature pass over these 15K examples and the candidate pool costs $(1 + 15{,}000/N)F = 0.69$ equivalent hours. Adding H_{10} gives the displayed 1.33 h floor, excluding CNN fitting and generation.

Sensitivity of the compute calibration. Using $q = 3$ instead gives $F = 0.711$ equivalent hours. The corresponding subtotals are 2.04 h (Q2Q), 4.20 h (SelectIT), 2.77 h (CO), 3.98 h (AtteNT), and 23.50 h (the local ADAPT rule); the CPQS and ssToken floors become 1.56 and 7.11 h.