UNIVERSITY of VIRGINIA

VIRGINIA DATA SCIENCE AND ANALYTICS (VAST) LAB

GIST: Targeted Data Selection for Instruction Tuning via Coupled Optimization Geometry

Guanghui Min

Department of Computer Science,
University of Virginia

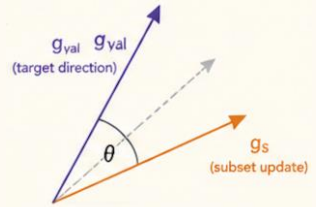
Outline

1 Problem

Targeted data selection

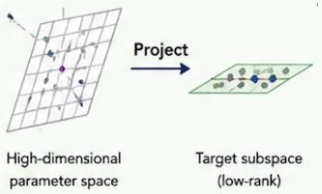
2 Baseline

LESS and its geometry mismatch



3 Method

GIST via target subspace projection



4 Results

Better accuracy with lower cost

Outline

1

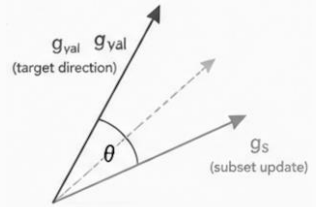
Problem

Targeted data selection

2

Baseline

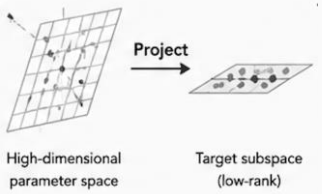
LESS and its geometry mismatch



3

Method

GIST via target subspace projection



4

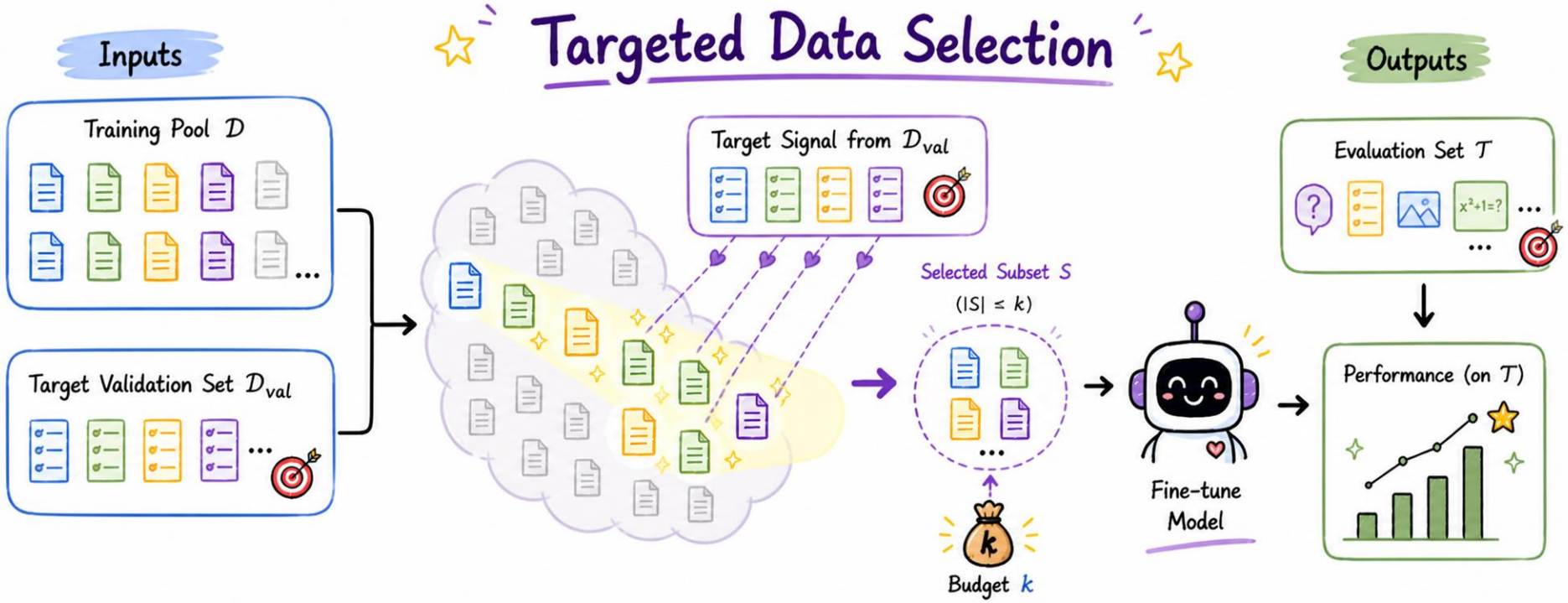
Results

Better accuracy with lower cost

Problem Setup

Targeted Data Selection for Instruction Tuning

- **Given:** a large training pool $\mathcal{D}$, a small target validation set $\mathcal{D}_{\text{val}}$ and a budget k.
- **Goal:** select $S \subset \mathcal{D}, |S| = k$ such that fine-tuning on S achieves the best performance on the target evaluation set $\mathcal{T}$.



Outline

1

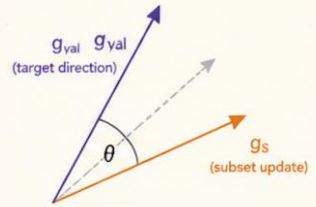
Problem

Targeted data selection

2

Baseline

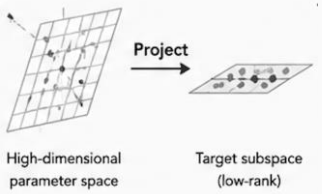
LESS and its geometry mismatch



3

Method

GIST via target subspace projection



4

Results

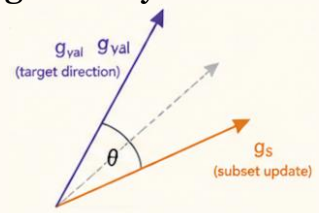
Better accuracy with lower cost

Outline

2

Baseline

LESS and its geometry mismatch



Introduction to LESS and related works

Underline Assumption of LESS

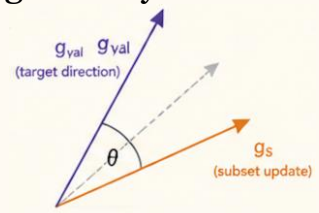
Geometry Mismatch of LESS

Outline

2

Baseline

LESS and its geometry mismatch



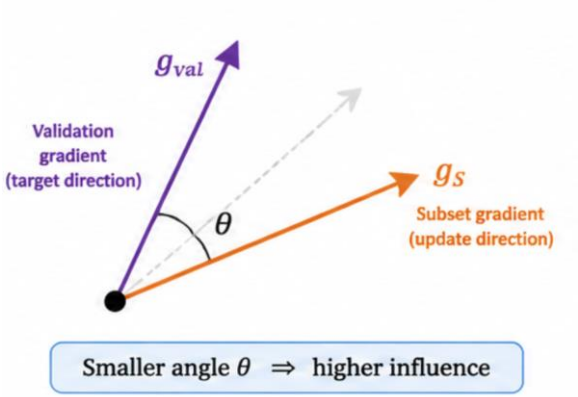
Introduction to LESS and related works

Underline Assumption of LESS

Geometry Mismatch of LESS

LESS: Gradient Alignment

Core idea: Useful data should move the model toward the target task.



Gradient alignment score:

$$\max_{S\subset\mathcal{D},|S|=k} \nabla_{\theta_t}\mathcal{L}(\mathcal{D}_{\text{val}},\theta_t)^{\top}\nabla_{\theta_t}\mathcal{L}(S,\theta_t),$$

LESS implementation:

$$\text{Influence}_{x\in\mathcal{D},z\in\mathcal{D}_{\text{val}}}(x,z)=\sum_{i=1}^{N}\bar{\eta}_i\cos(\nabla\mathcal{L}(z,\theta_i),\Gamma(x,\theta_i)),$$

N: **number of warmup checkpoints**, $\bar{\eta}_i$: **average learning rate**, $\Gamma(x,\theta_i)$: **Adam-preconditioned gradient**.

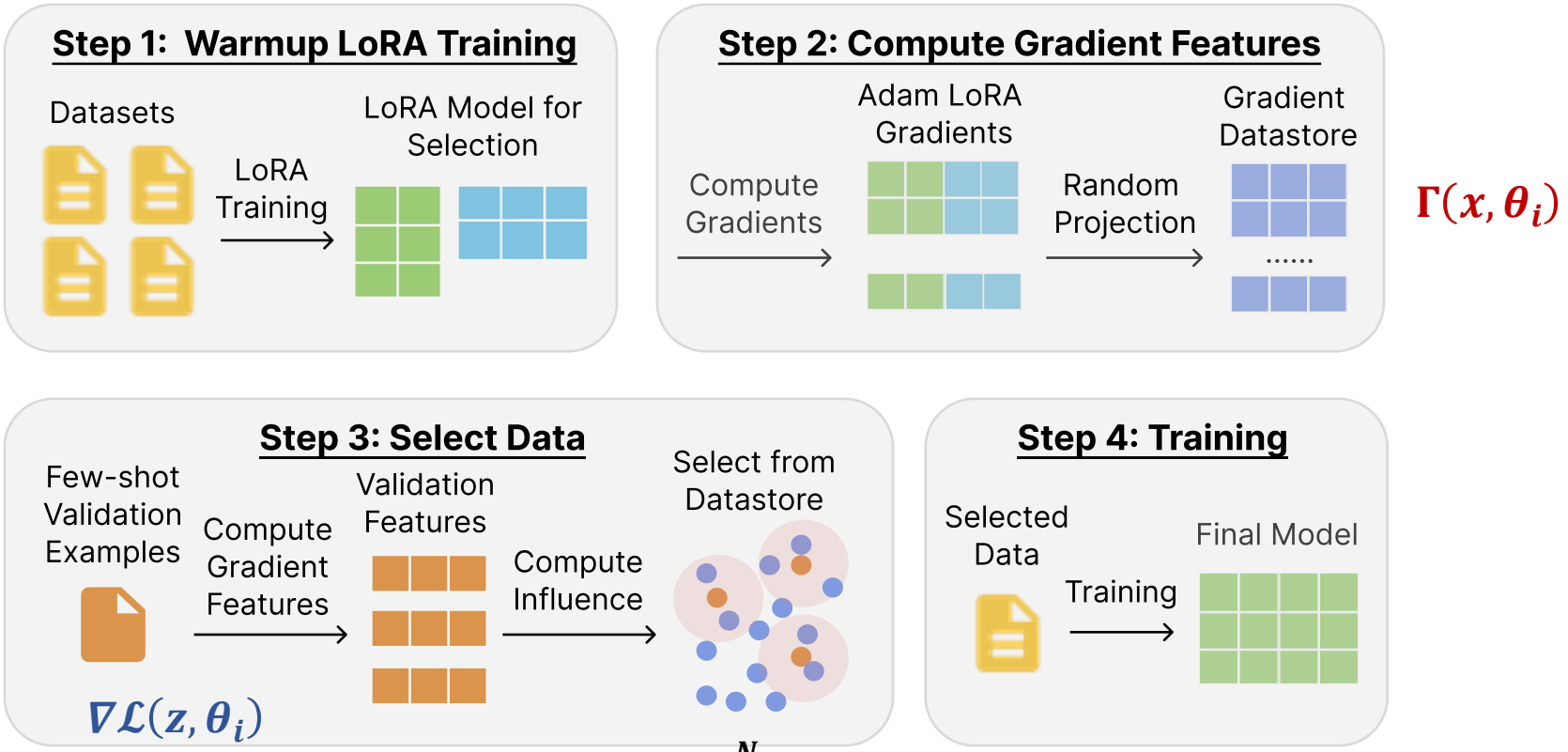
LESS further improves the basic score **with:**

Adam-preconditioned gradients
Capture optimizer geometry.

Multi-checkpoint aggregation
More stable and robust influence scores.

Length normalization
Mitigate bias from variable-length instructions.

LESS Pipeline



$$\textbf{Influence}(x, z) = \sum_{i=1}^{N} \bar{\eta}_i \cos\left(\nabla \mathcal{L}(z, \theta_i), \Gamma(x, \theta_i)\right)$$

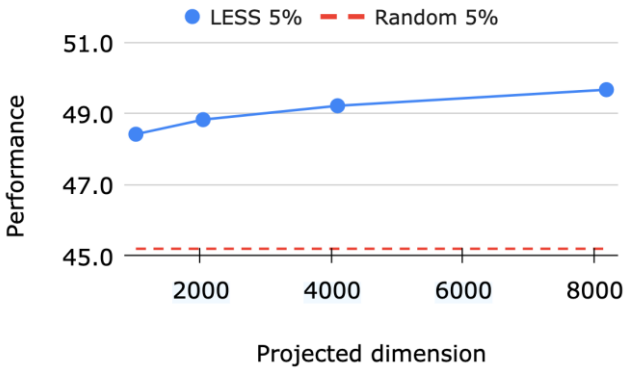
Key idea: precompute gradient features once, then use target validation examples as queries for data selection.

LESS is Expensive

Main bottleneck: computing and storing gradient features

Table 4: Asymptotic complexity, wall-clock runtime (measured as **single** A100 GPU hours) and storage cost associated with each step in LESS. Gradient computation is the most costly step, followed by the warmup LoRA training stage, but this expense is incurred only once. The actual data selection process requires minimal computation.

	Warmup LoRA Training		Gradient Features Computation		Data Selection	
	Complexity	**Actual**	**Complexity**	**Actual**	**Complexity**	**Actual**
Compute	$\mathcal{O}(\lvert\mathcal{D}_{\text{warmup}}\rvert \cdot N)$	6 Hours	$\mathcal{O}(\lvert\mathcal{D}\rvert \cdot N)$	48 Hours	$\mathcal{O}(\lvert\mathcal{D}\rvert \cdot \lvert\mathcal{D}_{\text{val}}\rvert \cdot d)$	< 1 Min
Storage	-	-	$\mathcal{O}(\lvert\mathcal{D}\rvert \cdot N \cdot d)$	17.7 GB	-	-



Key idea: LESS is effective, but its selection performance relies on expensive multi-checkpoint gradient features.

Table 6: Number of checkpoints (N) used for select data with LESS. Using fewer checkpoints still outperforms random selection but is less effective.

	MMLU	TyDiQA	BBH	Avg.
Random	46.5 (0.5)	52.7 (0.4)	38.9 (0.5)	46.0
$N = 1$	48.2 (0.4)	54.9 (0.4)	40.2 (0.2)	47.8
$N = 4$ (default)	50.2 (0.5)	56.2 (0.7)	41.5 (0.6)	**49.3**

Follow-ups Keep LESS Geometry

Recent works mainly improve the LESS framework in two directions:

- **Faster gradient computation:**
  1. **InfDist** (Nikdan et al., NeurIPS'25): Uses JVP-based embeddings to approximate compressed Adam-gradient features, so it avoids explicitly computing all gradient features.
  2. **IProX** (Chen et al., ICLR'26): Uses a smaller proxy model to compute influence features, reducing gradient-computation cost while preserving selection quality.

- **Better retrieval over gradient features:**
  1. **G2IS** (Zhao et al., ACL'25): Builds a graph over gradient features and uses PCA on validation gradients to identify core target knowledge for retrieval.

These methods improve efficiency or retrieval, but largely keep the same Adam-gradient influence formulation.

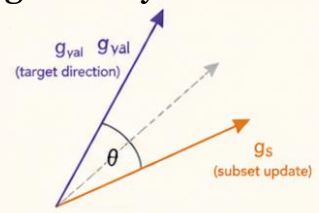
Our question: Is the Adam-preconditioned influence score the right geometry for LoRA fine-tuning?

Outline

2

Baseline

LESS and its geometry mismatch



Introduction to LESS and related works

Underline Assumption of LESS

Geometry Mismatch of LESS

Bi-level View

Given the training set $\mathcal{D}$, the validation set $\mathcal{D}_{\text{val}}$ and the selection budget k, we try to find the optimal subset $S \subset \mathcal{D}$ whose fine-tuned model minimizes target validation loss.

This naturally gives a bi-level optimization problem:

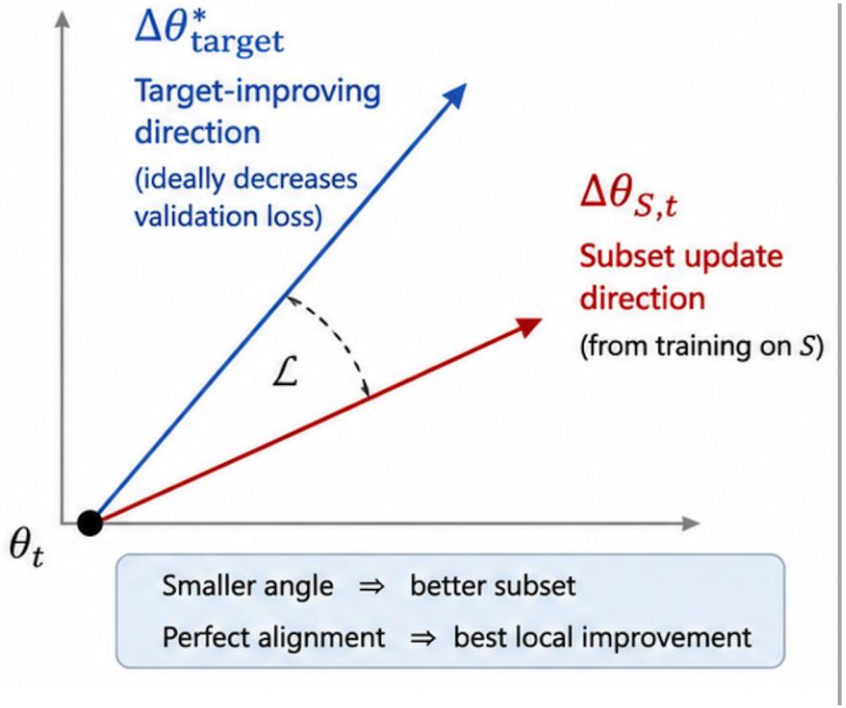
Problem 1 (Targeted Data Selection). *Find the optimal subset $\mathcal{S}^*$ that solves the following bi-level optimization:*

$$\begin{aligned} \min_{S \subseteq \mathcal{D}} \quad & \mathcal{L}(\mathcal{D}_{val}, \boldsymbol{\theta}_S^*) \\ s.t. \quad & |S| = k, \\ & \boldsymbol{\theta}_S^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(S, \boldsymbol{\theta}). \end{aligned} \tag{1}$$

Fine-tune the model on the selected subset

Local Direction Matching

Intuition: choose data whose training update points toward the direction that most reduces the validation loss.



Target-improving (Newton) direction

$$\Delta\theta^*_{\text{target}} = -\mathbf{H}^{\dagger}_{\text{val},t} \nabla_{\theta_t} \mathcal{L}(\mathcal{D}_{\text{val}}, \theta_t)$$

$\mathbf{H}^{\dagger}_{\text{val},t}$: validation loss Hessian at θ_t
†: (pseudo-)inverse

Actual update from training on subset *S* (first-order)

$$\Delta\theta_{S,t} = -\eta \nabla_{\theta_t} \mathcal{L}(S, \theta_t).$$

Objective (local view)

Solve **Problem 1** is locally equivalent to **maximizing the alignment** between the two directions

Ideal Influence

Bi-Level Objective

$$\begin{aligned} \min_{S \subseteq \mathcal{D}} \quad & \mathcal{L}(\mathcal{D}_{val}, \boldsymbol{\theta}^*_S) \\ s.t. \quad & |S| = k, \\ & \boldsymbol{\theta}^*_S = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(S, \boldsymbol{\theta}). \end{aligned}$$

Assume Local Convexity

$$\mathcal{L}(\mathcal{D}_{\text{val}}, \theta_t) + \nabla_{\theta_t} \mathcal{L}(\mathcal{D}_{\text{val}}, \theta_t)^\top \Delta\theta_t + \frac{1}{2} \Delta\theta_t^\top \mathbf{H}_{\text{val},t} \Delta\theta_t$$

Plugging the target-improving direction and the subset update direction into the **local alignment objective** gives:

$$\max_{\substack{S \subset \mathcal{D} \\ |S|=k}} \underbrace{\nabla_\theta L(\mathcal{D}_{\text{val}}, \theta_t)^\top}_{\text{validation gradient target direction}} \underbrace{H^\dagger_{\text{val},t}}_{\text{target Hessian geometry}} \underbrace{\nabla_\theta L(S, \theta_t)}_{\text{subset gradient training update}}$$

Newton Step for target set

up to constants independent of S and higher-order terms.

Influence is not just gradient similarity.
It is gradient similarity under the target geometry.

Adam = Diagonal Scaling

LESS replaces the raw gradient with the **Adam-preconditioned gradient**.

LESS update (per sample x)

$$\Gamma(x, \theta_t) \approx D_{\text{Adam},t} \nabla \mathcal{L}(x, \theta_t)$$

Adam-preconditioned gradient (used by LESS)

raw gradient of sample x

Diagonal scaling matrix

$$D_{\text{Adam},t} = \text{diag}\left(G_{\text{train}} G_{\text{train}}^{\top}\right)^{-\frac{1}{2}}$$

$$G_{\text{train},t} = [\nabla \mathcal{L}(x, \theta_t)]_{x \in \mathcal{D}} \in \mathbb{R}^{d \times |\mathcal{D}|}$$

(stacked training gradient matrix)

Geometric view

Adam multiplies the raw gradient by a diagonal matrix.

$$\Gamma(x, \theta_t) \approx \begin{bmatrix} \frac{1}{g_1^{\top} g_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{g_{|\mathcal{D}|}^{\top} g_{|\mathcal{D}|}} \end{bmatrix} \cdot \nabla \mathcal{L}(x, \theta_t)$$

LESS Uses Diagonal Geometry

Ignoring EMA terms and ϵ in Adam, the influence function in LESS (single epoch) becomes an alignment under **diagonal optimizer geometry**.

Ideal Influence (target geometry)

$$\nabla_{\theta_t} \mathcal{L}(\mathcal{D}_{\text{val}}, \theta_t)^\top \; \mathbf{H}_{\text{val},t}^{\dagger} \; \nabla_{\theta_t} \mathcal{L}(S, \theta_t)$$

validation gradient · target Hessian geometry · subset gradient

LESS Influence (diagonal geometry)

$$\nabla_{\theta_t} \mathcal{L}(\mathcal{D}_{\text{val}}, \theta_t)^\top \; \left[\mathbf{diag}\left(\boldsymbol{G}_{\textbf{train}} \boldsymbol{G}_{\textbf{train}}^\top\right)\right]^{-\frac{1}{2}} \; \nabla_{\theta_t} \mathcal{L}(S, \theta_t)$$

validation gradient · Adam diagonal preconditioner · subset gradient

Key idea

LESS measures influence under a diagonal (axis-aligned) geometry.
It can rescale each coordinate independently, but cannot model rotation or coupling.

Key implication

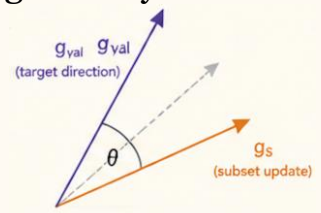
- Parameters are treated as coordinate-wise independent.
- Off-diagonal structure in the target geometry is discarded.

Outline

2

Baseline

LESS and its geometry mismatch



Introduction to LESS and related works

Underline Assumption of LESS

Geometry Mismatch of LESS

Target Geometry is Low-rank

The target (validation) gradients span a low-dimensional subspace.

Stacked validation gradient matrix

$$G_{\mathrm{val},t} = [\nabla \mathcal{L}(x, \theta_t)]_{x \in \mathcal{D}_{\mathrm{val}}} \in \mathbb{R}^{d \times |\mathcal{D}_{\mathrm{val}}|}$$

Target curvature proxy

$$\widehat{H}_{\mathrm{val},t} \approx G_{\mathrm{val},t} G_{\mathrm{val},t}^{\top}$$

empirical Fisher / Gauss-Newton approximation

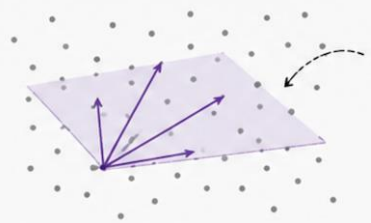
Rank upper bound

$$\mathrm{rank}\left(G_{\mathrm{val},t} G_{\mathrm{val},t}^{\top}\right) = \mathrm{rank}\left(\widehat{H}_{\mathrm{val},t}\right) \leq |\mathcal{D}_{\mathrm{val}}| \ll d$$

What this means

The target geometry lives in a low-dimensional subspace whose dimension is at most $|\mathcal{D}_{\mathrm{val}}|$, which is typically much smaller than d.

Parameter space ($\mathbb{R}^d$)



Low-dimensional subspace spanned by validation gradients (dimension $\leq |\mathcal{D}_{\mathrm{val}}| \ll d$)

Key insight: Low-rank target geometry = a few coupled update directions. Diagonal Adam scaling cannot represent this **coupled subspace**.

Target Geometry is Low-rank

A small principal subspace captures most target-gradient variance.

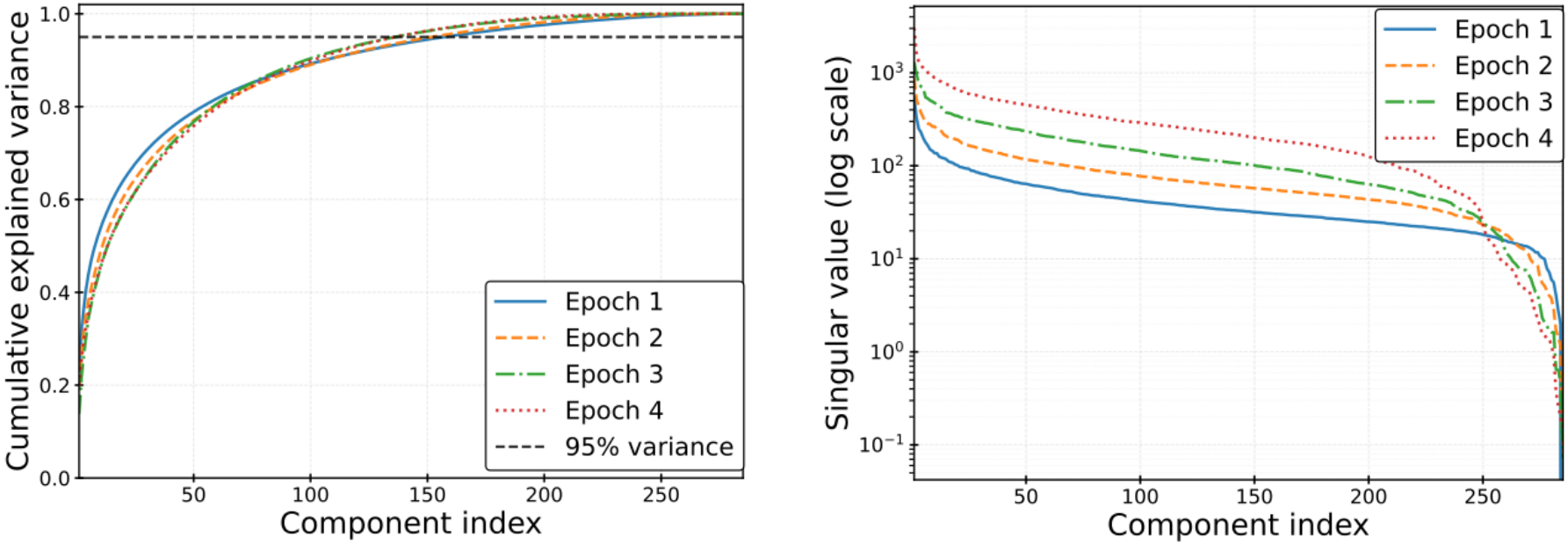


Figure 1. **Spectral analysis of the MMLU validation gradient G_{val} on Llama2-7B**. We decompose the gradient matrix via SVD to obtain singular values σ_i. (a) Cumulative explained variance. A steeper curve indicates that a **smaller principal subspace dimension is sufficient to capture the majority of the variance (e.g., Rank 150 captures 95%)**, confirming high directional information density. (b) The singular values (σ_k) exhibit **precipitous decay, further verifying the intrinsic low-rank structure**.

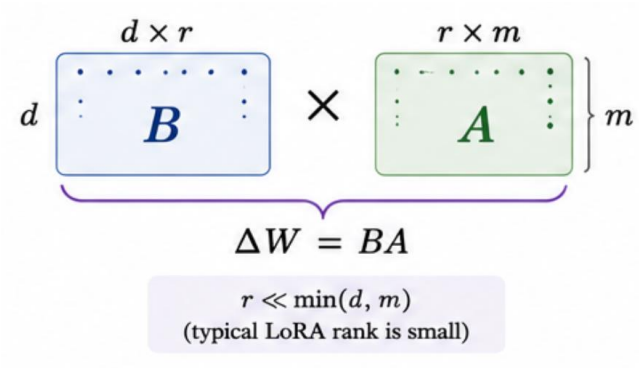
Key insight: The Observation shows that the effective rank can even be smaller than $|\mathcal{D}_{\text{val}}|$.

LoRA Induces Coupling

LoRA uses a bilinear parameterization. Parameters in A and B jointly determine the update ΔW, which leads to off-diagonal curvature in LoRA parameter space.

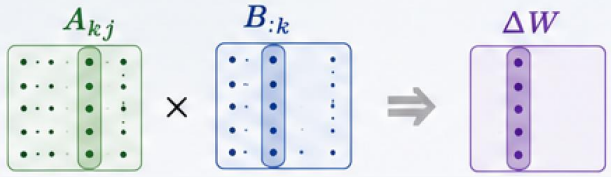
LoRA parameterization

$$W = W_0 + BA$$

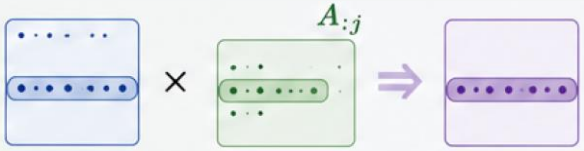


Why are A and B coupled ?

Change one element in A affects ΔW through the entire column of B.

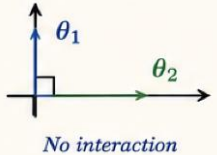


Change one element in B affects ΔW through the entire row of A.



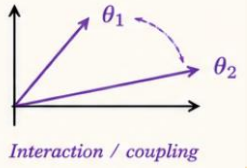
Key insight: LoRA parameters are not coordinate-wise independent.
The bilinear BA structure structurally induces coupling and off-diagonal curvature.

Coordinate-wise independent (LESS's assumption)



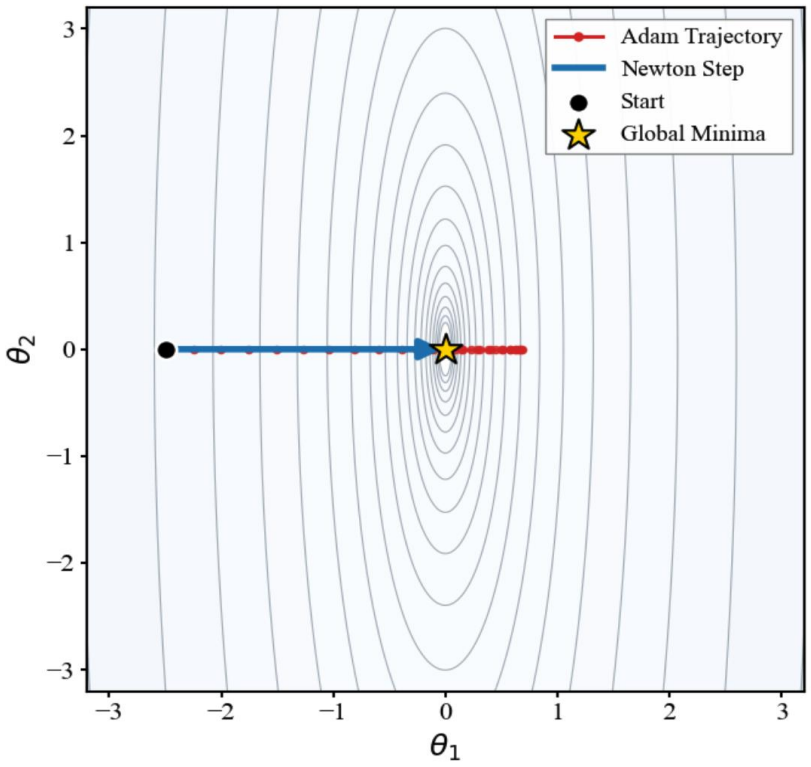
vs.

LoRA coupled geometry

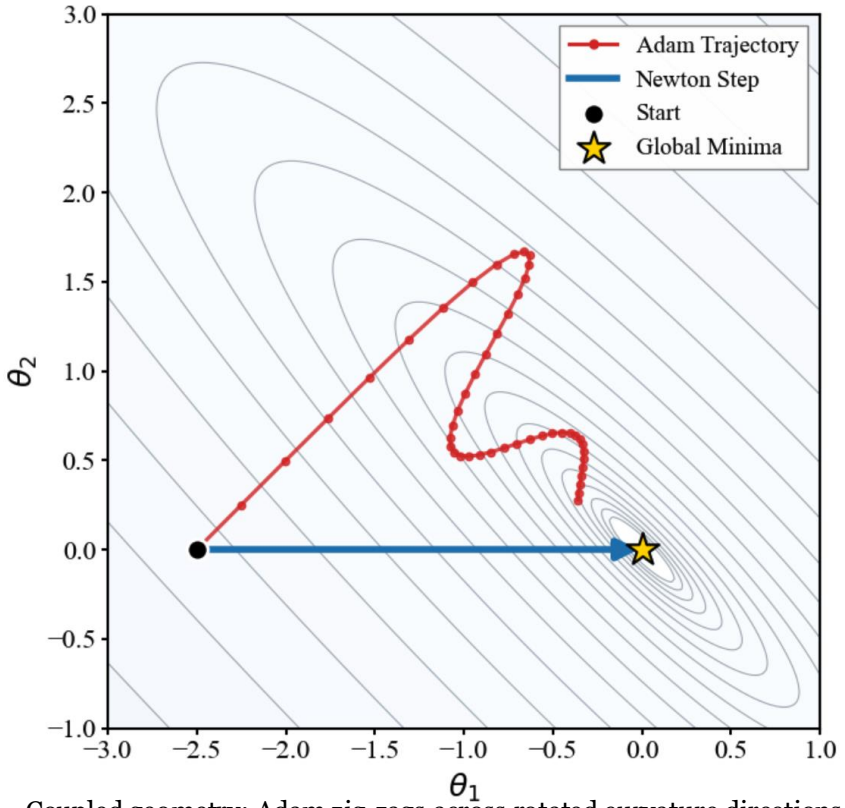


Adam Zig-zags in Coupled Geometry

Toy example: when curvature directions are coupled, diagonal Adam scaling can zig-zag instead of following the target Newton direction.



Axis-aligned geometry: Adam follows the descent direction.

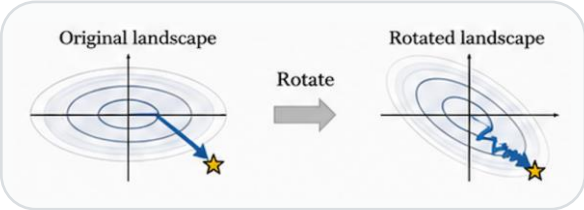


Coupled geometry: Adam zig-zags across rotated curvature directions.

Key insight: diagonal scaling works when the geometry is **axis-aligned**, but becomes **unstable** when curvature is coupled.

Adam is Coupling-sensitive

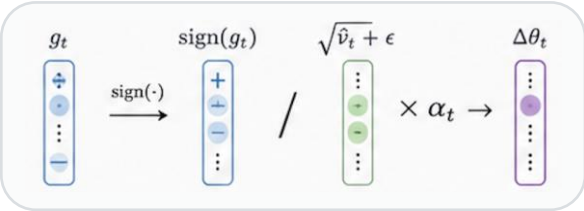
Several optimization studies show that Adam's diagonal adaptive scaling is not rotation-invariant and can be unstable on coupled geometries.



1 Adam is rotation dependent.

SGD is rotation-equivariant, while Adam can produce different trajectories under coupled loss landscapes.

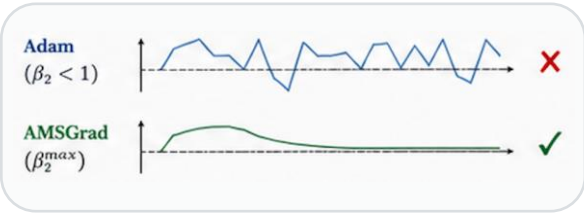
Understanding Adam Requires Better Rotation Dependent Assumptions
Zhang et al., 2025
(arXiv: 2410.19964)



2 Adam is coordinate-wise.

Adam adapts the update using the sign and (second-moment) magnitude of each coordinate independently.

Dissecting Adam: The Sign, Magnitude and Variance of Stochastic Gradients
Balles & Hennig, 2018
(ICML 2018)



3 Adam can fail to converge.

For some convex objectives, Adam does not converge without modifications (e.g., AMSGrad).

On the Convergence of Adam and Beyond
Reddi et al., 2018
(ICLR 2018)

Key insight: Adam is effective in practice, but its diagonal adaptive geometry is **not designed to capture coupled curvature**.

Outline

1

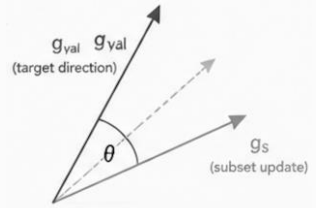
Problem

Targeted data selection

2

Baseline

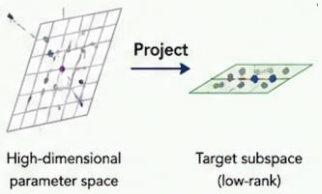
LESS and its geometry mismatch



3

Method

GIST via target subspace projection

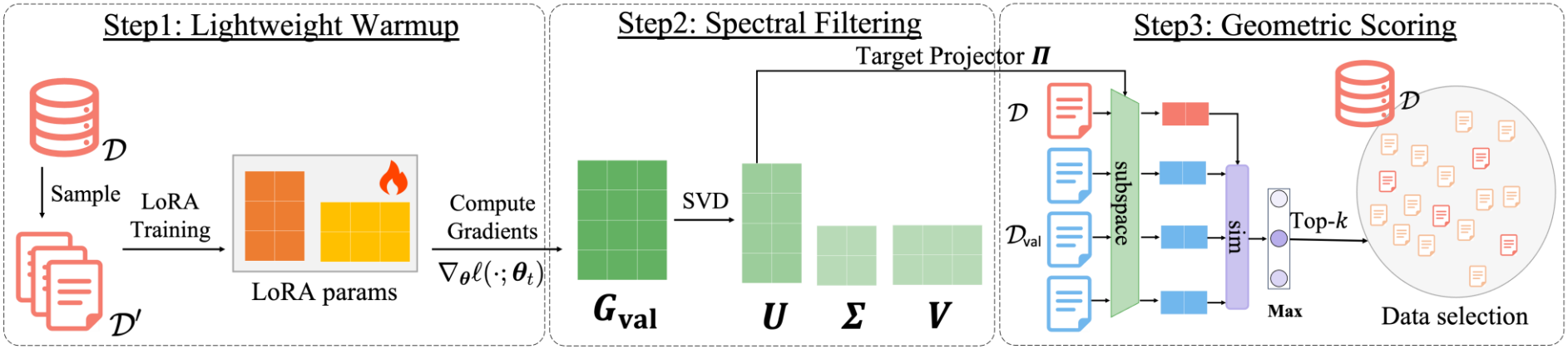


4

Results

Better accuracy with lower cost

GIST Pipeline



- **Step 1: Lightweight Warmup**

Train a LoRA model briefly and compute gradients.

$$\text{Inf}_t(z_i, z_{\text{val}}^{(j)}) \triangleq \frac{(\Pi \mathbf{g}_{i,t})^\top (\Pi \mathbf{g}_{\text{val},t}^{(j)})}{\|\Pi \mathbf{g}_{i,t}\|_2 \, \|\Pi \mathbf{g}_{\text{val},t}^{(j)}\|_2},$$

- **Step 2: Spectral Filtering**

Compute SVD of validation gradients and keep top-r target directions.

- **Step 3: Geometric Scoring**

Project gradients by $\Pi = U_r^\top$, compute cosine scores, select top-k.

Key idea: GIST uses the validation-gradient subspace as the target geometry, then selects training examples by projected alignment.

GIST: Spectral Projection

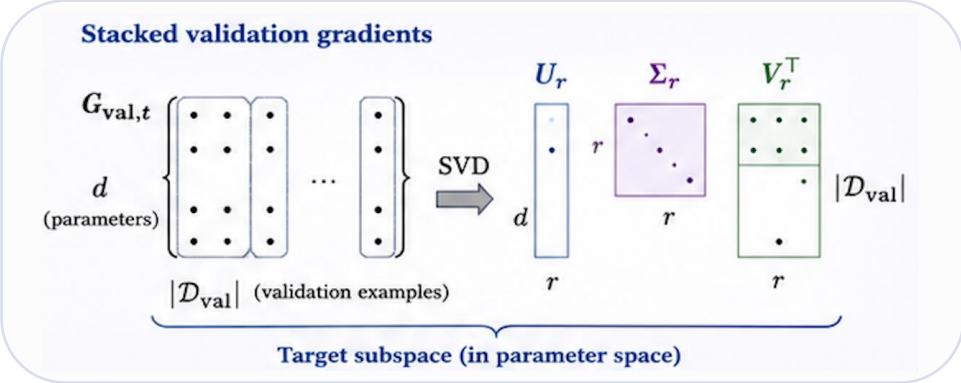
GIST replaces diagonal Adam scaling with **spectral projection onto the target subspace** identified by validation gradients.

1 Identify target subspace from validation gradients

Stack validation gradients at checkpoint θ_t into $G_{\text{val},t}$ and compute compact SVD:

$$G_{\text{val},t} = U_r \Sigma_r V_r^\top$$

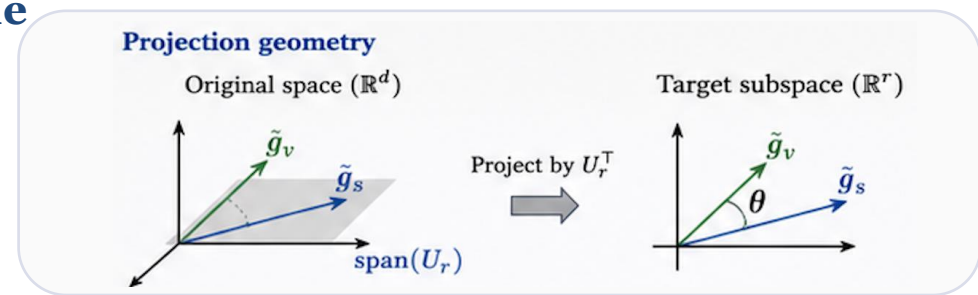
$U_r \in \mathbb{R}^{d \times r}$ (parameter subspace) $\quad \Sigma_r \in \mathbb{R}^{r \times r}$ (singular values) $\quad V_r \in \mathbb{R}^{|\mathcal{D}_{\text{val}}| \times r}$ (example space)



2 Use rotation (projection) to decouple

Take the task projector $\Pi = U_r^\top$.
It diagonalizes the surrogate in the subspace:

$$U_r^\top G_{\text{val},t} G_{\text{val},t}^\top U_r = \Sigma_r V_r^\top V_r \Sigma_r = \Sigma_r^2.$$



3 Project gradients and compare in the subspace

project the per-point gradient by $U_r^\top$, and measure the alignment of the projected training gradient descent direction $U_r^\top \nabla_{\theta_t} \mathcal{L}(S, \theta_t)$ and validation gradient descent direction $U_r^\top \nabla_{\theta_t} \mathcal{L}(\mathcal{D}_{\text{val}}, \theta_t)$.

GIST Approximates Ideal Influence

GIST approximates the ideal target geometry by replacing the Hessian pseudo-inverse with a low-rank projector from validation-gradient SVD.

Ideal Influence (target geometry)

$$\nabla_{\theta_t} \mathcal{L}(\mathcal{D}_{\text{val}}, \theta_t)^\top \quad \mathbf{H}^{\dagger}_{\text{val},t} \quad \nabla_{\theta_t} \mathcal{L}(S, \theta_t)$$

validation gradient | target Hessian geometry | subset gradient

GIST Influence (Coupled geometry)

$$\nabla_{\theta_t} \mathcal{L}(\mathcal{D}_{\text{val}}, \theta_t)^\top [\boldsymbol{U_r U_r^\top}] \quad \nabla_{\theta_t} \mathcal{L}(S, \theta_t)$$

validation gradient | target subspace projector | subset gradient

Key idea: GIST rotates gradients into the validation-gradient subspace (target geometry) and compares alignments there, effectively **decoupling the coupled geometry** that diagonal Adam scaling cannot capture.

Outline

1

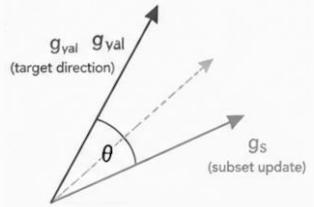
Problem

Targeted data selection

2

Baseline

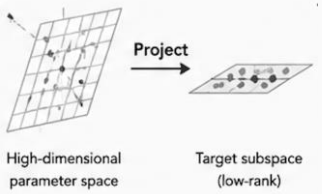
LESS and its geometry mismatch



3

Method

GIST via target subspace projection



4

Results

Better accuracy with lower cost

Evaluation Settings

- **Base Models (Big vs. Small, Modern vs. Old)**
  - Llama-2-7B, Llama-3.2-3B, Qwen-2.5-1.5B

- **Training Sets (~270k samples)**
  - Flan v2 (Longpre et al., 2023), Dolly (Conover et al., 2023), COT (Wei et al., 2022), Open Assistant 1 (Kopf et al., 2023).

- **Target Task**
  - MMLU (Hendrycks et al., 2020)
  - TydiQA (Clark et al., 2020)
  - BBH (Suzgun et al., 2023)

Table 1. Statistics of evaluation datasets. We select tasks covering diverse output formats, including multiple-choice, extractive spans, and generative reasoning.

			Data Size		
Dataset	**Shots**	**Tasks**	**Val.**	**Test**	**Output Format**
MMLU	5	57	285	18,721	Multiple Choice
TyDiQA	1	9	9	1,713	Extractive Span
BBH	3	23	81	920	Generation (CoT)

Main Results

GIST Consistently Improves Targeted Selection

Table 2. Accuracy across datasets and models. *Base* denotes 0% (no selection) and *Full* denotes 100% (full dataset). All other methods select 5% of the data under the same finetuning budget. Gray ± values report standard deviations. **Bold** numbers denote the best selected subset in each row, and underlined numbers denote the second best selected subset. Avg. Δ reports the average absolute improvement over *Base* over MMLU, TYDIQA and BBH.

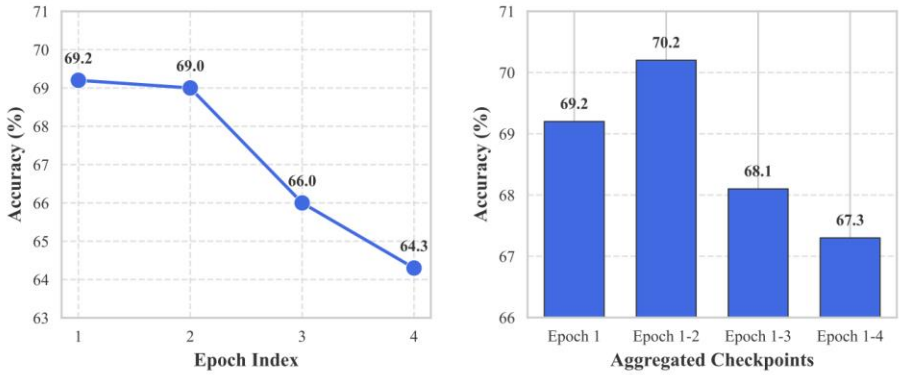
Model	Dataset	Base (0%)	Full (100%)	Rand.	Length	PPL.	Embed.	RDS+	LESS	GIST
Llama2-7B	MMLU	*45.6*	*51.6*	46.5±0.5	50.5±0.3	38.9±0.9	47.3±0.3	46.1±0.6	50.2±0.5	**51.2**±0.7
	TYDIQA	*46.4*	*54*	52.7±0.4	51.1±0.3	32.9±0.4	49.8±0.8	51.0±0.3	**56.2**±0.7	55.8±0.6
	BBH	*38.3*	*43.2*	38.9±0.5	39.8±0.7	34.5±0.2	38.6±0.6	39.0±0.7	41.5±0.6	**41.8**±0.8
	Avg. Δ	–	*+6.2*	+2.6	+3.7	-8.0	+1.8	+1.9	+5.9	**+6.2**
Llama3.2-3B	MMLU	*53.9*	*55.2*	53.2±0.6	**57.6**±0.9	51.8±0.3	53.9±0.6	51.9±0.2	56.5±0.6	56.1±0.4
	TYDIQA	*60.4*	*66.6*	64.1±0.4	63.9±1.3	57.1±0.7	62.8±0.4	62.6±0.5	67.1±0.8	**69.2**±0.3
	BBH	*45.5*	*47.8*	45.1±0.2	45.3±0.3	43.0±0.4	44.6±0.5	45.1±0.4	46.1±0.7	**48.0**±0.5
	Avg. Δ	–	*+3.3*	+0.9	+2.3	-2.6	+0.5	-0.1	+3.3	**+4.5**
Qwen2.5-1.5B	MMLU	*62.0*	*62.3*	61.5±0.3	62.8±0.1	61.7±0.2	62.3±0.2	62.4±0.2	62.2±0.2	**62.9**±0.6
	TYDIQA	*57.8*	*59.9*	55.6±0.3	56.7±0.3	58.4±0.4	54.1±0.3	56.2±0.3	**61.4**±0.8	61.2±0.5
	BBH	*43.8*	*44.0*	43.0±0.4	42.8±0.2	43.1±0.2	43.6±0.2	39.0±0.6	43.2±0.5	**44.1**±0.4
	Avg. Δ	–	*+0.9*	-1.2	-0.4	-0.1	-1.2	-2	+1.1	**+1.5**

Key insights:

- **Best Avg. Δ** across all three backbones.
- 5% selected data can **match or outperform** 100% full fine-tuning.
- **Compared with LESS**: +0.3, +1.2, +0.4 Avg. Δ across the three backbones.

Is One Checkpoint Enough?

Yes. Early single-checkpoint geometry is sufficient, and later/more checkpoints do not consistently help.



(a) Single Epoch Performance (b) Cumulative Performance

Figure 3. **Impact of Checkpoint Selection.** (a) Using single-epoch gradients shows a clear performance drop in later epochs. (b) Aggregating multiple checkpoints (weighted) does not outperform the early-stop strategy, confirming that early gradients contain the essential task optimization directions.

Table 3. Number of checkpoints (N) used for select data with GIST for Llama3.2-3B.

	MMLU	TyDiQA	BBH	Avg.
LESS	56.5±0.6	67.1±0.8	46.1±0.7	56.6
$N = 4$	56.3±0.3	67.3±0.1	46.8±0.4	56.8
$N = 1$ (default)	56.1±0.4	69.2±0.3	48.0±0.2	**57.8**

Table. Warmup ablation with GIST for Llama3.2-3B.

Task	Base	GIST w/o warmup	GIST
MMLU	53.9	53.0±0.2	**56.1**±0.4
TyDiQA	60.4	64.5±1.7	**69.2**±0.3
BBH	45.5	46.2±0.4	**48.0**±0.5

Key insights: GIST only needs a short warmup and one early checkpoint to recover useful target geometry. Later or aggregated checkpoints do not consistently improve selection.

GIST is 4× faster

Table 4. Efficiency comparison between `GIST` (Ours) and LESS. Runtime is measured in **single A100 GPU hours**. `GIST` significantly reduces the overhead in Warmup and Feature Extraction stages (approx. 4× speedup) and requires negligible time for SVD.

Method	Metric	Warmup	Target SVD	Grad. Feats.
LESS	Time	6.0 h	–	48.0 h
	Compl.	$\mathcal{O}(\lvert\mathcal{D}'\rvert \cdot N)$	–	$\mathcal{O}(\lvert\mathcal{D}\rvert \cdot N)$
`GIST`	**Time**	**1.5 h**	**< 1 m**	**12.0 h**
	Compl.	$\mathcal{O}(\lvert\mathcal{D}'\rvert)$	$\mathcal{O}(\lvert\mathcal{D}_{\text{val}}\rvert^2 d)$	$\mathcal{O}(\lvert\mathcal{D}\rvert)$

Key insights:

- GIST reduces the dominant cost of targeted selection by **avoiding multi-checkpoint** feature extraction.
- For each target task, runtime drops from 54.0h to 13.5h, with warmup reduced from 6.0h to 1.5h and gradient-feature extraction reduced from 48.0h to 12.0h.
- The extra target **SVD cost is negligible**, taking less than 1 minute.
- Storage is also reduced from 75GB to 217MB, about a 350× reduction.

The End

VIRGINIA DATA SCIENCE AND ANALYTICS (VAST) LAB

Thanks for listening!