From Weeks to Hours: Fast and Principled SFT Data Generation for Reasoning LLMs

Hongyi Henry Jin[†] **Wenhan Yang**[†]
Meysam Ghaffari[⋆] **Carlos Morato**[⋆] **Baharan Mirzasoleiman**[†]
[†]University of California, Los Angeles [⋆]Optum AI
{hongyi,wenhan_yang,baharan}@cs.ucla.edu

Abstract

Reasoning-oriented post-training enables large language models (LLMs) to solve complex tasks via multi-step inference. Supervised fine-tuning (SFT) on high-quality reasoning traces is a particularly efficient approach, but its effectiveness depends critically on careful data curation. Existing pipelines rely on a costly generate-then-filter paradigm: large pools of long reasoning traces are produced by multiple teacher models and subsequently filtered for difficulty and diversity using additional LLMs, often requiring weeks of computation. In this work, we propose SFTPROBE, a fundamentally different approach that bypasses this pipeline by *directly predicting* the quality of reasoning data. We show that a brief LoRA-based adaptation, combined with evaluating loss on only 5% of each reasoning trace, suffices to estimate both difficulty and diversity. SFTPROBE is grounded in a geometric view of post-training, where pretrained LLMs lie in a low-rank, anisotropic loss basin. By probing this basin via low-rank perturbations, we construct compact loss signatures whose clustering captures diversity, while loss at curvature inflection points provides a robust proxy for difficulty. We also provide a simple criterion for teacher selection. Empirically, our approach reduces data curation time from weeks to hours while maintaining or improving performance. Remarkably, on mathematical reasoning, 5k generated examples match the highly filtered 89k-example in OpenThoughts-Math dataset, and 7.5k generated examples outperforms it by 2.5% in accuracy. We further introduce a high-quality physics reasoning dataset, and use it to train a model that outperforms an IPhO 2025 silver-medal-winning model. The code and the dataset are released at https://bigml-cs-ucla.github.io/SFTProbe-page/.

1 Introduction

Large language models (LLMs) have recently demonstrated an emerging ability to solve complex tasks by allocating substantial computation to *reasoning at inference time* [25, 7, 23, 8]. Unlike instruction tuning—where models learn to produce answers in a single pass, akin to fast "System 1" thinking—reasoning fine-tuning encourages slower, deliberative "System 2" behavior, where models generate intermediate steps and refine them iteratively. Enabling such capabilities relies on post-training, either through large-scale reinforcement learning (RL) [7, 25] or supervised fine-tuning (SFT) on high-quality reasoning traces [23, 8, 6]. While RL is effective but costly, SFT is far more efficient, yet highly sensitive to data quality—particularly the diversity and difficulty of reasoning trajectories [9, 10]—making data curation a key bottleneck.

Reasoning data differs fundamentally from standard instruction-tuning corpora. Instead of short question–answer pairs, reasoning traces can span thousands—or even hundreds of thousands—of tokens [23, 6]. Current pipelines therefore rely on a costly generate-then-filter paradigm: multiple strong

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

teacher models are used to produce large pools of candidate traces, which are subsequently filtered using LLM-based heuristics for difficulty and diversity, followed by downstream training to evaluate their utility [8, 6]. This process is not only computationally expensive but also slow, often requiring weeks of iterative generation, filtering, and evaluation. These limitations raise a fundamental question: *can we directly generate high-quality reasoning data without resorting to large-scale sampling and filtering?* This is very challenging because the quality of a reasoning trace—particularly its difficulty and diversity—is typically only observable after full generation and training.

In this work, we introduce a principled and efficient alternative that bypasses this pipeline entirely.SFTPROBE builds on a recent observation that the loss over the initial portion of reasoning traces, evaluated along the training trajectory, provides useful signals about the quality of full reasoning [12]. However, prior methods require training on a large corpus of full reasoning traces to obtain such trajectories. We relax this requirement and show that a brief LoRA-based adaptation on a small set of fully generated examples, is sufficient to reliably estimate both difficulty and diversity. This enables the direct construction of high-quality SFT datasets without exhaustive generation or filtering, reducing data curation time from weeks to minutes.

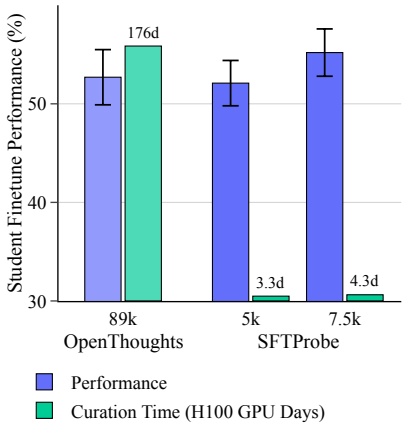


Figure 1: Comparison of the curated math reasoning dataset in OpenThoughts [6] and SFTPROBE. Training on 5k curated data matches the performance with 53x speedup, and on 7.5k improves the performance by 2.5% with 41x speedup.

SFTPROBE is grounded in a geometric perspective of post-training, in which pretrained LLMs are understood to operate within a low-rank loss basin characterized by relatively flat curvature [5]. From a theoretical standpoint, we show that examples with similar loss values at the endpoints of this basin—corresponding to *inflection points* where curvature changes—exhibit tightly bounded differences in their gradients during SFT, and therefore behave similarly under training. However, the loss landscape is inherently anisotropic: curvature varies significantly across directions, implying that reaching the basin boundary requires direction-dependent perturbation magnitudes.

To account for this heterogeneity, we perturb the model along the LoRA directions to reach the boundary of the basin, and thereby obtain a set of representative checkpoints. We then evaluate the loss on only the initial portion of each reasoning trace (as little as 5% of tokens) across these checkpoints, producing a compact *loss signature* for every example. Clustering these signatures provably groups examples with bounded gradient variation during SFT, providing a principled notion of diversity without requiring full training. In addition, we identify a highly reliable proxy for difficulty based on the loss at the inflection boundary of the basin, again computable from only a tiny prefix of the reasoning trajectory. Finally, this geometric framework yields a simple and efficient criterion for teacher selection: the best teacher is the one that minimizes early-token loss under the pretrained model, aligning data quality with the intrinsic structure of the loss landscape.

We validate SFTPROBE on medical, mathematical, and physical reasoning tasks. Remarkably, with only 5K directly generated examples, SFTPROBE matches the performance of large-scale datasets such as OpenThoughts, which contains 89K carefully filtered examples constructed through expensive pipelines; and with 7.5k generated examples, out approach outperforms it by 2.5% in accuracy. In the domain of physics, we further present, to our knowledge, the first high-quality SFT dataset for reasoning. For medical reasoning, SFTPROBE outperforms data selection baselines, with substantially lower costs. Together, these results demonstrate that high-quality reasoning data can be *predicted rather than exhaustively searched*, offering a new direction for scalable and principled post-training of LLMs.

2 Related works

2.1 Data Curation for Reasoning Models

Supervised fine-tuning (SFT) is highly sensitive to the quality—i.e. difficulty and diversity—[9] of reasoning traces, and simply increasing the amount of data without proper curation can even degrade performance [1]. Unlike instruction tuning data which consists of short (instruction, output) pairs, reasoning trajectories are highly exploratory and noisy: Throughout the long chain-of-thought, the model frequently makes multiple failed attempts, pauses to re-evaluate (e.g., generating "Wait, but the problem says..." or "Hmm, let me double check..."), and tries multiple ways to approach the problem before arriving at a solution (c.f. Figure 9). This inherent noise disrupts data selection metrics that are applicable to instruction tuning, including difficulty-filtering based on perplexity [16, 21], learnability-based (loss reduction during training) [35], and gradient magnitude [19], and diversity-filtering based on embedding similarity [18, 2] and the superior gradient similarity [33, 32].

Due to these limitations, reasoning traces are often either manually curated [23] or generated by strong teacher models (e.g., DeepSeek-R1), and then filtered by two key properties: diversity and difficulty [6, 23, 8], meaning they include a broad range of examples that are sufficiently challenging for the target model. In practice, this filtering relies heavily on LLM-based heuristics. Diversity is encouraged by categorizing and selecting examples across different groups, while difficulty is commonly enforced by prompting a strong model to score question hardness and keeping only the most challenging ones, which outperforms the reasoning length heuristic. Despite their effectiveness, these approaches are computationally expensive: generating large numbers of long reasoning traces with strong models, followed by extensive filtering, makes the construction of high-quality SFT datasets costly and difficult to scale. Recent study of Jin et al. [12] observed that the loss over the initial portion (100-1K tokens) of reasoning traces, evaluated along the fine-tuning trajectory, provides useful signals about the quality of full reasoning. However, this requires training on a large corpus of *full* reasoning traces to obtain the fine-tuning trajectory and does not enable efficient SFT data curation.

In our work, we develop a principled method to directly *predict* the quality of reasoning without having to fully generate them and without relying on accurate fine-tuning trajectories.

2.2 Teacher Selection for LLMs

Selecting appropriate teacher models commonly prioritizes alignment with student capabilities. For example, GRACE [27] leverages a cross-validation structure, where it takes gradients from one split of data and measures their magnitudes re-weighted under the spectrum of the other split. The gradient spectrum reflects data diversity, while the gradient norm encodes teacher-student alignment. Most recently, [14] introduced Local Naturalness, which scores a response by measuring the student's log-probabilities over short, sequential reasoning steps (e.g., sentences) conditioned only on a small local window, and used aggregated local scores across prompts for teacher selection. These methods, however, require generating full reasoning traces for all the examples for every teacher model, and hence incur substantial computational cost.

In our work, we show that the best teacher model can be efficiently selected as the one that minimizes average early-token losses under the pretrained model.

3 Efficient SFT Data Curation

In this section, we discuss generating high-quality—i.e., diverse and difficult—SFT dataset, given a teacher model, without having to fully generate reasoning traces and training on them.

3.1 Predicting Similarity of Reasoning Traces

Gradient similarity. During training, each example updates the model parameters through its gradient. Consequently, examples with similar gradients induce similar parameter updates and therefore provide largely redundant training signals [22, 24]. Motivated by this observation, prior works [32, 12] estimate data diversity by clustering examples according to their gradient similarity across *multiple checkpoints along the fine-tuning trajectory*. While effective, this approach requires fully training the model on fully generated reasoning traces in order to obtain the trajectory itself, making it compu-

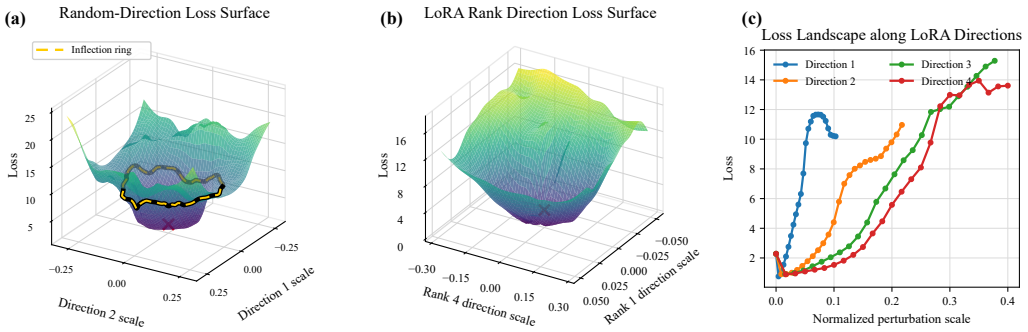


Figure 2: (a) Loss landscape of Qwen2.5-3B-Instruct along two random directions. (b) LoRA directions have different curvatures. (c) Directions with lower curvature should be perturbed more to reach the inflection point.

Algorithm 1 Inflection point finder. Detailed implementation in Algorithm 3

Require: Perturbation direction v

1: Incrementally perturb base model θ_0 while $\mathbb{E}_{x,y}\,\mathcal{L}(x, y, \theta_0 + \epsilon \cdot v)$ is monotone
2: Fit a cubic spline surrogate $f(\epsilon)$ on $\{(\epsilon, \mathbb{E}_{x,y}\,\mathcal{L}(x, y, \theta + \epsilon \cdot v))\}$.
3: Find the inflection point as $\epsilon^\star \leftarrow \text{Root}\left(\frac{d^2}{d\epsilon^2} f(\epsilon) = 0\right)$
4: **return** $\theta_0 + \epsilon^\star \cdot v$

tationally expensive. Moreover, because the training trajectory is inherently stochastic—due to mini-batch ordering, optimizer noise, and other sources of randomness—the resulting gradient similarities can be noisy and unstable. Here, we develop a significantly more efficient and reliable approach for predicting gradient similarity without having to fully generate reasoning traces and training on them.

Fine-tuning loss. Our key observation is that fine-tuning occurs within a *low-rank*, relatively *flat*, and smooth region of the loss landscape, where the loss around the pretrained initialization is well approximated by a *near-parabolic* geometry [5]. Figure 2 (a) illustrates the loss landscape of Qwen2.5-3B-Instruct around the pretrained checkpoint. In this regime, the geometry of the loss basin provides substantially richer information than the stochastic training trajectory itself. Specifically, we show that loss difference at the pretrained parameters and *boundaries* of the parabolic basin—corresponding to inflection points where the curvature changes (shown as the yellow band in Figure 2(a))—induces the tightest bound on gradient similarity throughout fine-tuning. Intuitively, if two examples behave similarly at the extremities of the basin, then their optimization dynamics remain similar across the entire training process. Importantly, bounded loss differences at the pretrained checkpoint and the inflection points yield significantly tighter estimates of gradient similarity than measuring loss similarity along noisy training trajectories. This shifts the problem from tracking optimization dynamics to accurately characterizing the local geometry of the loss basin, as we discuss next.

3.1.1 Approximating the Near-Parabolic Low-rank Loss Basin

Since fine-tuning happens in a low-rank region, only a small number of parameter directions dominate learning. We exploit this structure by briefly training a LoRA adapter on a very small subset of reasoning traces to identify these dominant directions. Importantly, this short adaptation phase is not intended to meaningfully optimize the model itself; rather, it serves only to estimate the principal directions of the local loss geometry. Thus, very brief training on a small subset is sufficient.

However, the loss landscape is anisotropic: as shown in Figure 2(b), curvature varies substantially across directions, implying that the extent of the near-parabolic region differs from one direction to another. To accurately estimate the boundary of the basin, we therefore perturb the model incrementally along each direction, progressively increasing the perturbation magnitude while the loss continues to rise smoothly, and stopping once this behavior breaks down (*c.f.* Figure 2(c)). We then fit a cubic spline to the resulting loss profile and identify the inflection point where the curvature changes sign. Repeating this procedure across all LoRA directions provides a fast and reliable approximation of the inflection boundaries of the near-parabolic loss basin. We provide the pseudocode for finding the inflection point in Algorithm 3.

3.1.2 Upper-bounding Pairwise Gradient Distances

Next, we show that bounded loss difference at two perturbed checkpoints along some direction v upper-bounds their gradient difference projected onto v. Crucially, this upper-bound is *tightest* when the first checkpoint is the pretrained model and the other is the inflection point along direction v.

Theorem 3.1. *Assume a smooth fine-tuning loss with bounded curvature $0 \leq c \leq c_H$. Starting from initialization θ_0, for any checkpoint $\theta \in \Theta$ during training define $\epsilon_\theta = \|\theta - \theta_0\|_2$. Assume that fine-tuning gradients are predominantly captured by an r-dimensional subspace $\mathcal{V} \subset \mathbb{R}^d$ with orthonormal basis $\{v_1, \ldots, v_r\}$, such that for every training example z, $\|Proj_{\mathcal{V}^\perp} \nabla \mathcal{L}_z(\theta)\| \leq \gamma$. For each basis direction v_i, denote $c_{H,i} \triangleq c_{H,v_i}$, and let*

$$\theta_{i,1} = \theta_0 + \epsilon_{i,1} v_i, \qquad \theta_{i,2} = \theta_0 + \epsilon_{i,2} v_i, \qquad \textit{where } \; 0 \leq \epsilon_{i,1} < \epsilon_{i,2}.$$

If any pair of training examples z_1 and z_2 have bounded loss difference at all checkpoints $\theta_{i,1}, \theta_{i,2}$:

$$|\mathcal{L}_{z_1}(\theta_{i,1}) - \mathcal{L}_{z_2}(\theta_{i,1})| \leq \delta, \quad |\mathcal{L}_{z_1}(\theta_{i,2}) - \mathcal{L}_{z_2}(\theta_{i,2})| \leq \delta, \quad \forall i \in \{1, \ldots, r\},$$

then for any $\theta \in \Theta$ during training, the (full-rank) gradient difference is bounded by

$$\|\nabla \mathcal{L}_{z_1}(\theta) - \nabla \mathcal{L}_{z_2}(\theta)\| \leq \sqrt{\sum_{i=1}^{r} \left(\frac{2\delta}{\epsilon_{i,2} - \epsilon_{i,1}} + \frac{1}{2}(\epsilon_{i,1} + \epsilon_{i,2}) c_{H,i} + 2\epsilon_\theta \sqrt{c_H c_{H,i}} \right)^2} + 2\gamma.$$

Corollary 3.2. *The RHS upper-bound of the gradient difference in Theorem 3.1 is tightest at:*

$$\epsilon^\star_{i,1} = 0, \qquad \epsilon^\star_{i,2} = 2\sqrt{\frac{\delta}{c_{H,i}}}.$$

At this minimizer, the full-parameter gradient difference is bounded by:

$$\|\nabla \mathcal{L}_{z_1}(\theta) - \nabla \mathcal{L}_{z_2}(\theta)\| \leq 2\left(\sqrt{\delta} + \epsilon_\theta \sqrt{c_H}\right) \sqrt{\sum_{i=1}^{r} c_{H,i}} + 2\gamma.$$

Remark. Corollary 3.2 suggests that to get the tightest upper-bound on pairwise gradient differences, LoRA directions with smaller curvature should receive larger perturbations (i.e., up to their inflection point).

3.1.3 Diversity Filtering

To predict which reasoning traces will be diverse if fully generated, we first find the fixed low-dimensional subspace $\mathcal{V}$ by finetuning the model on a small subset of the data using LoRA with a small rank r. The use of LoRA here is merely for finding the dominant subspace, and we are not restricted to LoRA during the actual SFT. Performing QR decomposition on the LoRA adaptors gives us the orthogonal basis vectors $\{v_1, \ldots, v_r\}$ of $\mathcal{V}$. Then, following Corollary 3.2, we find $\epsilon_{i,1}, \epsilon_{i,2}$ in every direction v_i using Algorithm 3. This yields two checkpoints for each direction v_i. Following [12], we only generate and evaluate the loss of the initial 1k tokens of every reasoning trace on the $2r$ checkpoints to get the *loss signature* for every example j:

$$\textbf{\textit{Loss signature:}} \; L_j = [\mathcal{L}_j^{\text{1k}}(\theta_0 + \epsilon_{1,1} v_1), \mathcal{L}_j^{\text{1k}}(\theta_0 + \epsilon_{1,2} v_1), \cdots, \mathcal{L}_j^{\text{1k}}(\theta_0 + \epsilon_{r,1} v_r), \mathcal{L}_j^{\text{1k}}(\theta_0 + \epsilon_{r,2} v_r)] \quad (1)$$

Theoretically, $\epsilon^*_{i,1} = 0$ for all i. To avoid all $\theta_0 + \epsilon_{i,1} v_i$ collapsing to θ_0 and have more informative loss signatures, we perturb $\epsilon_{i,1}$ until it satisfies:

$$\mathbb{E}_j \left[\mathcal{L}_j^{\text{1k}}(\theta_0 + \epsilon_{i,1} v_i)\right] = (\mathbb{E}_j \left[\mathcal{L}_j^{\text{1k}}(\theta_0)\right] + \mathbb{E}_j \left[\mathcal{L}_j^{\text{1k}}(\theta_0 + \epsilon_{i,2} v_i)\right])/2, \quad (2)$$

and use the obtained $\epsilon_{i,1}$ for every v_i to calculate the loss signatures. Detailed implementation is in Algorithm 4.

Clustering loss signatures. We standardize each samples to zero mean and unit variance, and use FAISS [4] to cluster the data points into k clusters. We then select an equal number of data points from each cluster to ensure that the data points are diverse. Empirical gains of diversity filtering is presented in Section F.

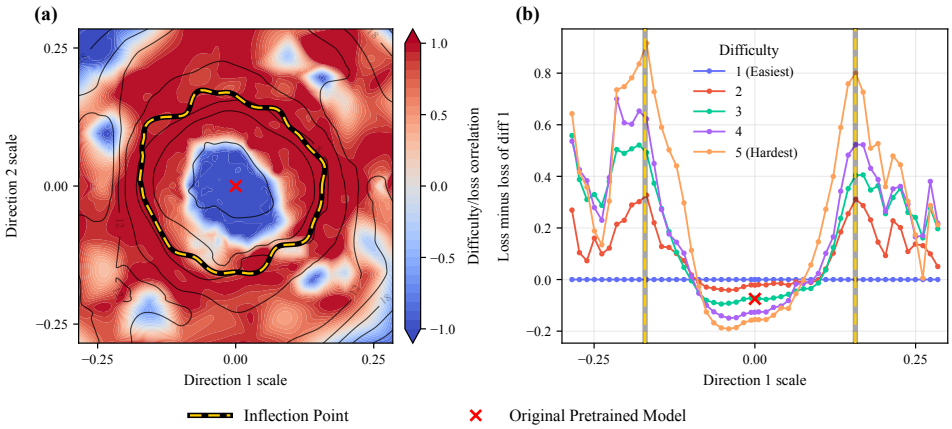


Figure 3: (left) correlation between the initial 100 token and the problem difficulty in the 2D contour plot of the loss. While the loss evaluated at the pretrained model does not have a good correlation with difficulty (blue region near the origin); the loss of perturbed parameters correlates well with difficulty (red region outside the blue circle). Crucially, the *best separation* of difficulty levels occurs at the *inflection* point (yellow band). (right) examples that are labeled more difficult by LLM have a higher curvature for their initial 100 token loss around the pretrained model parameters.

Algorithm 2 Complete data curation pipeline

Require: Mixture of problems from different sources
Ensure: High-quality SFT data

1: Select the teacher model according to Section 4
2: Generate the initial 100 reasoning tokens for all the examples
3: Calculate problem difficulties in each source based on the first 100 tokens (Eq. 3)
4: Cluster examples in each source based on their difficulty scores into two clusters and drop the easy cluster with lower loss
5: Generate the initial 1k reasoning tokens for the remaining examples
6: Calculate loss signature (Eq. 1) in every source based on the initial 1k tokens and cluster them to find similar reasoning traces
7: Sample a balanced subset of N_s (Eq. 4) examples from clusters in every source
8: Generate full reasoning traces for the sampled examples

3.2 Predicting Reliable Difficulty Scores

Next, we discuss predicating reliable difficulty scores for all examples. When generating reasoning traces, the teacher model starts its reasoning by summarizing and rephrasing the problem. Thus, the loss of the initial 100 tokens along the training trajectory provides high-quality signal for problem difficulty [12]. Here, we show how reliable difficulty scores can be calculated without having to train the model on full reasoning traces to obtain the training trajectory.

Difficult problems induce higher loss curvature. Our key observation is that examples with different difficulty levels have different loss curvature for their initial 100 tokens. As shown in Figure 3 (right), examples that are labeled more difficult by LLM have a higher curvature for their initial 100 token loss around the pretrained model parameters. Figure 3 (left) puts the correlation between this loss on individual problem and the difficulty of that problem in the 2D contour plot of the loss. We see that while the loss evaluated at the pretrained model does not have a good correlation with the difficulty of the problem (blue region near the origin); once perturbed, the loss correlates well with the difficulty of the problem (red region outside the blue circle). Crucially, the *best separation* of difficulty levels occurs at the *inflection* point along a random direction $\hat{u}$ found by Algorithm 3.

$$\textit{Difficulty score:} \qquad d_j = \mathcal{L}_j^{100}(\theta_0 + \epsilon_{i,2}\hat{u}), \tag{3}$$

Unlike diversity filtering, we do not require the LoRA directions and any random direction $\hat{u}$ can be used for calculating difficulty scores.

3.2.1 Difficulty Filtering

SFT data is often curated using mixture of problems from different sources. For example, Numina-Math [15], which is a mathematical reasoning dataset contains international Olympiad problems, high school math, math exams, online forums & repositories, and synthetic data.

Filtering easy problems in every source. For problems in every source, we calculate their difficulty scores using Eq. 3. Then, we use FAISS [4] to cluster difficulty scores in each source into two clusters and discard the easy cluster with lower average loss.

Sampling more from more difficult sources. Next, we calculate the average difficulty of problems d_j remained in every source V_s. For a total budget of N data examples to curate, we sample more examples from sources with higher average difficulty, as follows:

$$N_s = N \cdot \text{softmax}_s(\mathbb{E}_{j \in V_s}[d_j]). \tag{4}$$

The complete pseudocode of the method is illustrated in Alg. 2 with further detail in Alg. 5.

4 Teacher Selection

Finally, we describe how to select the best teacher model for a student, without having to train the model or generate full reasoning traces.

Student learns better from teachers with lower rephrasing loss. We aim to select the teacher whose language is most familiar to the student, allowing the student to focus on solving the problem rather than parsing unfamiliar phrasing. To this end, we randomly sample a set of questions and prompt each candidate teacher model to rephrase the same problems. We cap each response at 100 tokens, as in difficulty filtering, and evaluate the rephrasings using the *pretrained* student parameters rather than the perturbed ones. Specifically, let y_{ij}^{rephr} denote the rephrasing generated by teacher model i for problem j. We define its rephrasing loss under the pretrained student model θ_0 as:

$$\textit{Rephrasing Loss:} \qquad R_i = \mathbb{E}_j \left[\mathcal{L}^{100}(x_j, y_{ij}^{rephr}, \theta_0) \right], \tag{5}$$

As shown in Figure 4, the teachers with the lowest average loss can best train the student model. Additional details and discussions are provides in Section E.

We emphasize the distinction between the loss on individual problems and the average loss across problems. For a fixed set of problems, varying the teacher allows us to identify the teacher whose rephrasings are most compatible with the student's existing language patterns. A teacher with lower loss produces phrasing that the pretrained student already understands well, while preserving the underlying problem to be solved. This allows the student to devote its capacity to learning the solution rather than adapting to a new linguistic pattern. Conversely, a high loss indicates that the student struggles to interpret the teacher's rephrasing. Training on such teachers can therefore push the student away from its pretrained optimum, potentially degrading existing capabilities or even inducing catastrophic forgetting. This interpretation is consistent with observations in [27, 14].

5 Experiments

In this section, we evaluate the performance of SFTPROBE on the following tasks:

1. **Data selection for medical reasoning**: First, we apply SFTPROBE to *select* 1k data points from the pre-generated M23k dataset [8] and evaluate the performance of the student model finetuned on the selected data points. This serves as a demonstration for SFTPROBE's applicability in the setting of data selection as well and to provide a controlled comparison against the data selection baselines.
2. **Data curation for medical reasoning**: Then, we use the original, unfiltered set of 37,698 decontaminated *questions* given in Huang et al. [8]—that is used to curate the M23k datase—to generate a new reasoning SFT data. We applied SFTPROBE to select the teacher model (among the smaller models we could fit in our GPU memory) and QwQ-32B is selected. We generated a dataset containing 1000 reasoning examples using our method.

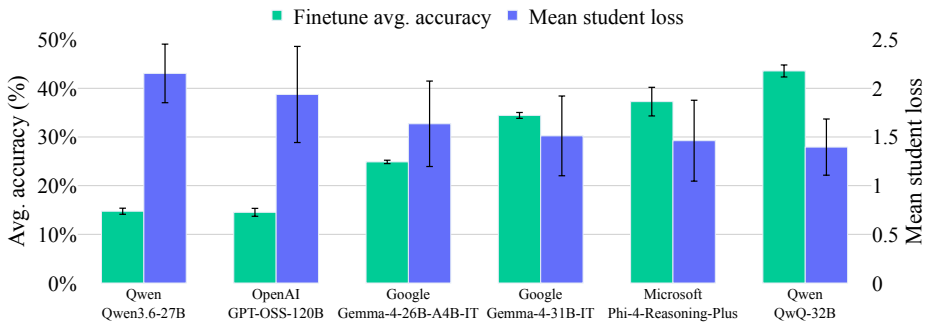


Figure 4: We evaluate the loss of the student model and the accuracy of the finetuned on the dataset generated from the teacher model on the NuminaMath dataset. We also train the student using the same randomly selected 1000 questions with responses generated from the teacher model. We find that the higher the rephrasing loss is, the worse the final finetuned student performs.

3. **Data curation for math reasoning**: We generate reasoning traces from a subset of the NuminaMath dataset [15] containing 192k questions using the best teacher model selected by Section 4 and evaluate the performance of the student model finetuned on the generated dataset. We compare with OpenThinker-7B [6] and other data selection methods that select data from OpenThoughts-Math, a reasoning dataset generated from the same subset of NuminaMath.
4. **Data curation for physics reasoning**: We sanitized the problems and ground-truth solutions from several physics olympics from 1967 to 2018 into 3084 pairs of problems and solutions. We combine it with physics problems from PhysicsEval [29] to form a mixed dataset of 40k problems and curate a physics reasoning dataset.

Additional details on datasets, benchmarks, baselines and hyperparameters are in Section D.

5.1 Results

Our data curation pipeline provides substantial increment in performance across multiple datasets. When selecting from generated reasoning traces, given the same data budget, we're see a 1.1% increase in performance in M23k dataset in Figure 5 when using Llama3.1-8B-Instruct model compared to the best baseline. When the reasoning traces are curated, training on 1k data curated with SFTPROBE matches the full curated M23k dataset within 0.6% as shown in Figure 6. On math reasoning, we see a 3.4% increase in performance in math reasoning in Figure 7 compared to the best baseline. Evaluation on the physics dataset in Table 1 shows a performance close to PRIME-RL/P1-30B-A3B, a model capable of winning the silver model for IPhO 2025. With a higher budget, we curated a 5k math reasoning dataset. Using the same student model, we match OpenThinker-7B as shown in Figure 1, which used 89k math reasoning data to train.

Apart from effective, SFTPROBE is also highly efficient. We were able to obtain the same math performance using 53x less computation compared to OpenThinker-7B as shown in Figure 1. SFTPROBE even requires less time to run compared to selection methods where the long reasoning traces have aleady been generated, as shown in Figure 7.

5.2 Ablation Studies

In this section, we conduct ablation studies to demonstrate the insensitivity of the hyperparameters and the effects of individual components in SFTPROBE.

LoRA Rank. The most significant hyperparameter in SFTPROBE is the rank of the brief LoRA training, which controls the dimension of the subspace and the length of the loss signatures. In theory, LoRA automatically finds the most important dimensions so that increasing the rank has diminishing returns. This is also supported by the Figure 2(b) and (c), which demonstrates that the model is significantly more dependent on the first few dimensions. Results in Table 3 demonstrate that reducing rank from 4 to 2 does not significantly affect performance.

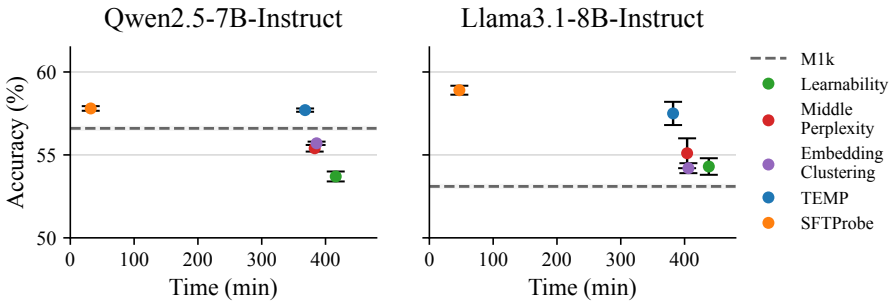


Figure 5: Results on data selection for medical reasoning with different models as the base model. Wall time on 4×H100 is reported. SFTPROBE performs comparable to best selection baseline on Qwen2.5-7B and outperforms selection baselines on Llama3.1-8B, while being substantially faster.

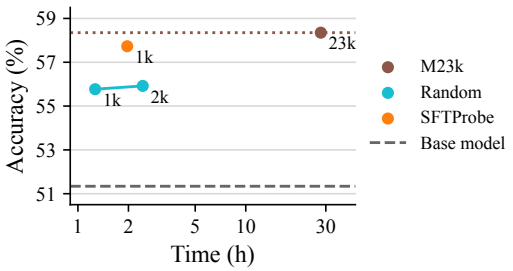


Figure 6: Results on data curation for medical reasoning. Wall time on 4×H100 is reported. Compared to the full 23k data finetune, our curated 1k data closes 76% of the gap between random sampling and full dataset training, while only requiring 7% of the compute cost.

LoRA Training Steps. Unlike Jin et al. [12] and Yang et al. [32] that requires a full training run to get the training trajectory, SFTPROBE relaxes this assumption and works well even with just a few iterations of LoRA training. Table 3 show that using only 100 - 400 iterations is sufficient to identify the dominant training directions.

Individual steps. We ablate the effectiveness of different components of SFTPROBE in Table 2. For diversity filtering, replacing the inflection points with checkpoints along the training trajectory leads to a 0.7% decrease in performance. Furthermore, using checkpoints along the brief (approximate) training trajectory used in our LoRA training (instead of checkpoints along the full training trajectory) leads to additional 0.8% decrease in accuracy.

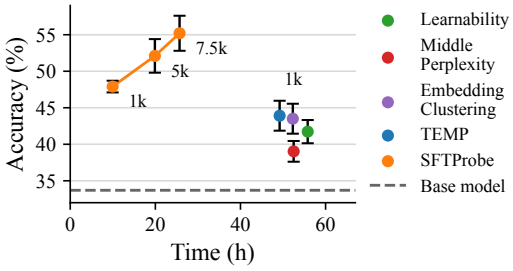


Figure 7: Results on data curation for math reasoning. Data selection methods select 1k data from Openthoughts-Math dataset containing pre-generated teacher examples. SFTPROBE generates data with QwQ-32B as the teacher, and even when taken into consideration the generation time, takes less time compared to the data selection baselines. Wall time on 4×H100 is reported.

Table 1: Experiment results on data curation for physics reasoning with Qwen2.5-7B-Instruct as the student model. We outperform PRIME-RL/P1-30B-A3B, a model capable of winning the silver model for IPhO 2025, although we start with a much weaker base model.

Model	Average Score
Qwen2.5-7B-Instruct	3.58
Google/Gemma-4-26B-A4B-it	9.65
PRIME-RL/P1-30B-A3B [30]	9.05
SFTPROBE (1k data generated)	9.14 ± 0.29
SFTPROBE (2k data generated)	9.32 ± 0.11
SFTPROBE (5k data generated)	9.76 ± 0.74

Table 2: Ablation Study for the effect of inflection search and brief LoRA training.

Method	ID (3 sets)	OOD (7 sets)	Avg (10 sets)
Jin et al. [12]	**69.3%**	52.9%	**57.8%**
SFTPROBE	68.7%	**53.1%**	**57.8%**
without Inflection search	67.9%	52.5%	57.1%
without LoRA	68.1%	51.3%	56.3%

Table 3: Ablation study on the key hyperparameters. We vary the number of training steps and LoRA rank and evaluate the performance of the student model finetuned on the generated dataset. A small number of training steps and LoRA rank is sufficient to capture the dominant directions of the fine-tuning loss landscape and yield satisfactory performance.

Hyperparameter	ID (3 sets)	OOD (7 sets)	Avg (10 sets)
Lora Rank = 4, Train with 200 iters	68.7%	**53.1%**	**57.8%**
Lora Rank = 2, Train with 200 iters	68.5%	52.6%	57.3%
Lora Rank = 4, Train with 100 iters	68.9%	51.9%	57.0%
Lora Rank = 4, Train with 400 iters	**69.0%**	52.0%	57.1%

6 Conclusion

In this work, we introduced a novel paradigm that directly predicts the quality of reasoning data from local loss geometry. SFTPROBE is based on the observation that pretrained LLMs operate within a low-rank, anisotropic loss basin. By briefly adapting LoRA directions and evaluating loss on only 5% of reasoning tokens, we construct compact loss signatures that reliably estimate both diversity and difficulty. We further showed that clustering these signatures captures gradient similarity during SFT, while loss at curvature inflection points provides a robust proxy for difficulty. We also provide a simple criterion for teacher selection.

Empirically, SFTPROBE reduces data curation time from weeks to minutes while maintaining or improving downstream performance. In mathematical reasoning, 5K selected examples match the heavily curated 89K-example OpenThoughts dataset, and we further introduce a high-quality physics reasoning dataset. Overall, our results suggest that reasoning data quality can be predicted rather than exhaustively searched, opening a scalable and principled direction for LLM post-training.

Acknowledgements

This research was supported by Optum AI Science, NSF CAREER Award 2146492, NSF-Simons AI Institute for Cosmic Origins (CosmicAI), NSF AI Institute for Foundations of Machine Learning (IFML), and an Apple Research Award.

References

[1] Syeda Nahida Akter et al. "Front-loading reasoning: The synergy between pretraining and post-training data". In: *arXiv preprint arXiv:2510.03264* (2025).

[2] Gantavya Bhatt et al. "An experimental design framework for label-efficient supervised finetuning of large language models". In: *Findings of the Association for Computational Linguistics: ACL 2024*. 2024, pp. 6549–6560.

[3] Hanjie Chen et al. "Benchmarking large language models on answering and explaining challenging medical questions". In: *arXiv preprint arXiv:2402.18060* (2024).

[4] Matthijs Douze et al. "The FAISS Library". In: *arXiv preprint arXiv:2401.08281* (2024).

[5] Stanislav Fort et al. "Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5850–5861.

[6] Etash Guha et al. "OpenThoughts: Data Recipes for Reasoning Models". In: *arXiv preprint arXiv:2506.04178* (2025).

[7] Daya Guo et al. "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning". In: *arXiv preprint arXiv:2501.12948* (2025).

[8] Xiaoke Huang et al. *m1: Unleash the Potential of Test-Time Scaling for Medical Reasoning in Large Language Models*. 2025.

[9] Adel Javanmard, Baharan Mirzasoleiman, and Vahab Mirrokni. "Theoretical Perspectives on Data Quality and Synergistic Effects in Pre-and Post-Training Reasoning Models". In: *International Conference on Machine Learning* (2026).

[10] Adel Javanmard, Baharan Mirzasoleiman, and Vahab Mirrokni. "Understanding the Role of Training Data in Test-Time Scaling". In: *The Fourteenth International Conference on Learning Representations*.

[11] Di Jin et al. "What disease does this patient have? a large-scale open domain question answering dataset from medical exams". In: *Applied Sciences* 11.14 (2021), p. 6421.

[12] Hongyi Henry Jin et al. "Reasoning Quality Emerges Early: Data Curation for Reasoning Models". In: *Forty-third International Conference on Machine Learning*. 2026.

[13] Qiao Jin et al. "Pubmedqa: A dataset for biomedical research question answering". In: *arXiv preprint arXiv:1909.06146* (2019).

[14] Hoang Anh Just, Myeongseob Ko, and Ruoxi Jia. "The Signal is in the Steps: Local Scoring for Reasoning Data Selection". In: *Proceedings of the 43rd International Conference on Machine Learning*. Ed. by Tong Zhang et al. Vol. 306. Proceedings of Machine Learning Research. PMLR, June 2026, pp. 55125–55155.

[15] J. Li et al. *NuminaMath: The largest public dataset in AI4Maths with 860k pairs of competition math problems and solutions*. `https://huggingface.co/datasets/AI-MO/NuminaMath-CoT`. 2024.

[16] Ming Li et al. "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)*. Ed. by Kevin Duh, Helena Gomez, and Steven Bethard. June 2024.

[17] Hunter Lightman et al. "Let's Verify Step by Step". In: *arXiv preprint arXiv:2305.20050* (2023).

[18] Wei Liu et al. "What Makes Good Data for Alignment? A Comprehensive Study of Automatic Data Selection in Instruction Tuning". In: *The Twelfth International Conference on Learning Representations*. 2024.

[19] Yibai Liu et al. *Learn More, Forget Less: A Gradient-Aware Data Selection Approach for LLM*. 2025.

[20] MAA. *Aime 2024 problems*. Accessed: 2026-01-22. 2024.

[21] Max Marion et al. "When less is more: Investigating data pruning for pretraining llms at scale". In: *arXiv preprint arXiv:2309.04564* (2023).

[22] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. "Coresets for data-efficient training of machine learning models". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 6950–6960.

[23] Niklas Muennighoff et al. “s1: Simple test-time scaling”. In: *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Ed. by Christos Christodoulopoulos et al. Nov. 2025.

[24] Dang Nguyen et al. “Mini-batch Coresets for Memory-efficient Language Model Training on Data Mixtures”. In: *The Thirteenth International Conference on Learning Representations*.

[25] OpenAI. “Learning to reason with llms”. In: *https://openai.com/index/learning-to-reason-with-llms/* (2024).

[26] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. “Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering”. In: *Conference on health, inference, and learning*. PMLR. 2022, pp. 248–260.

[27] Abhishek Panigrahi et al. “In Good GRACEs: Principled Teacher Selection for Knowledge Distillation”. In: *arXiv preprint arXiv:2511.02833* (2025).

[28] David Rein et al. “Gpqa: A graduate-level google-proof q&a benchmark”. In: *First Conference on Language Modeling*. 2024.

[29] Oshayer Siddique et al. *PhysicsEval: Inference-Time Techniques to Improve the Reasoning Proficiency of Large Language Models on Physics Problems*. 2025.

[30] P1 Team. *P1: Mastering Physics Olympiads with Reinforcement Learning*. 2025.

[31] Yubo Wang et al. “Mmlu-pro: A more robust and challenging multi-task language understanding benchmark”. In: *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2024.

[32] Yu Yang et al. “SmallToLarge (S2L): Scalable Data Selection for Fine-tuning Large Language Models by Summarizing Training Trajectories of Small Models”. In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024.

[33] Jipeng Zhang et al. “TAGCOS: Task-agnostic Gradient Clustered Coreset Selection for Instruction Tuning Data”. In: *Findings of the Association for Computational Linguistics: NAACL 2025*. Ed. by Luis Chiruzzo, Alan Ritter, and Lu Wang. Apr. 2025.

[34] Shunfeng Zheng et al. “Benchmarking Foundation Models with Retrieval-Augmented Generation in Olympic-Level Physics Problem Solving”. In: *Findings of the Association for Computational Linguistics: EMNLP 2025*. 2025.

[35] Haotian Zhou et al. “DavIR: Data Selection via Implicit Reward for Large Language Models”. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Wanxiang Che et al. July 2025.

[36] Yuxin Zuo et al. “MedXpertQA: Benchmarking Expert-Level Medical Reasoning and Understanding”. In: *arXiv preprint arXiv:2501.18362* (2025).

A Proofs omitted in the main body

Proof of Theorem 3.1. Let $\theta \in \Theta$ be an arbitrary parameter vector. Let $f(\theta) \triangleq \mathcal{L}_{z_1}(\theta) - \mathcal{L}_{z_2}(\theta)$ be the difference in loss of the two data points. Let $g(\theta) \triangleq \nabla f(\theta) = \nabla \mathcal{L}_{z_1}(\theta) - \nabla \mathcal{L}_{z_2}(\theta)$ and $H(\theta) \triangleq \nabla^2 f(\theta) = \nabla^2 \mathcal{L}_{z_1}(\theta) - \nabla^2 \mathcal{L}_{z_2}(\theta)$ be the difference of gradient and the Fessian of f, respectively. Define

$$v_\theta \triangleq \frac{\theta - \theta_0}{\|\theta - \theta_0\|_2}.$$

Fix any basis direction v_i. Let $\theta_{i,1} = \theta_0 + \epsilon_{i,1} v_i$ and $\theta_{i,2} = \theta_0 + \epsilon_{i,2} v_i$. Let $u_{i,1} = \theta_{i,1} - \theta$ be the displacement vector from θ to the first perturbed checkpoint. Then $\theta_{i,2} - \theta = u_{i,1} + (\epsilon_{i,2} - \epsilon_{i,1}) v_i$. Expanding $f(\theta_{i,1})$ and $f(\theta_{i,2})$ around θ via Taylor series:

$$f(\theta_{i,1}) = f(\theta) + \langle g(\theta), u_{i,1} \rangle + \frac{1}{2} u_{i,1}^\top H(\theta) u_{i,1} + R_{i,1}$$

$$f(\theta_{i,2}) = f(\theta) + \langle g(\theta), u_{i,1} + (\epsilon_{i,2} - \epsilon_{i,1}) v_i \rangle + \frac{1}{2} (u_{i,1} + (\epsilon_{i,2} - \epsilon_{i,1}) v_i)^\top H(\theta) (u_{i,1} + (\epsilon_{i,2} - \epsilon_{i,1}) v_i) + R_{i,2},$$

where we assume the Lagrange remainder terms are negligible. Thus, subtracting the first equation from the second eliminates $f(\theta)$, $\langle g(\theta), u_{i,1} \rangle$, and the $\frac{1}{2} u_{i,1}^\top H(\theta) u_{i,1}$ term:

$$f(\theta_{i,2}) - f(\theta_{i,1}) = (\epsilon_{i,2} - \epsilon_{i,1}) \langle g(\theta), v_i \rangle + (\epsilon_{i,2} - \epsilon_{i,1}) u_{i,1}^\top H(\theta) v_i + \frac{1}{2} (\epsilon_{i,2} - \epsilon_{i,1})^2 v_i^\top H(\theta) v_i$$

Rearranging gives:

$$\langle g(\theta), v_i \rangle = \frac{f(\theta_{i,2}) - f(\theta_{i,1})}{\epsilon_{i,2} - \epsilon_{i,1}} - u_{i,1}^\top H(\theta) v_i - \frac{1}{2} (\epsilon_{i,2} - \epsilon_{i,1}) v_i^\top H(\theta) v_i$$

Applying triangle inequality:

$$|\langle g(\theta), v_i \rangle| \leq \frac{|f(\theta_{i,2}) - f(\theta_{i,1})|}{\epsilon_{i,2} - \epsilon_{i,1}} + |u_{i,1}^\top H(\theta) v_i| + \frac{1}{2} (\epsilon_{i,2} - \epsilon_{i,1}) |v_i^\top H(\theta) v_i|$$

Given $|f(\theta_{i,1})| \leq \delta$ and $|f(\theta_{i,2})| \leq \delta$, we have $|f(\theta_{i,2}) - f(\theta_{i,1})| \leq 2\delta$.

$$u_{i,1} = \theta_{i,1} - \theta = (\theta_0 + \epsilon_{i,1} v_i) - (\theta_0 + \epsilon_\theta v_\theta) = \epsilon_{i,1} v_i - \epsilon_\theta v_\theta.$$

Therefore,

$$\left| u_{i,1}^\top H(\theta) v_i \right| \leq \epsilon_{i,1} \left| v_i^\top H(\theta) v_i \right| + \epsilon_\theta \left| v_\theta^\top H(\theta) v_i \right|.$$

For the first term, since $0 \leq v_i^\top \nabla^2 \mathcal{L}_{z_j}(\theta) v_i \leq c_{H,i}$ for $j \in \{1, 2\}$,

$$\left| v_i^\top H(\theta) v_i \right| = \left| v_i^\top \nabla^2 \mathcal{L}_{z_1}(\theta) v_i - v_i^\top \nabla^2 \mathcal{L}_{z_2}(\theta) v_i \right| \leq c_{H,i}.$$

For the mixed term, by triangle inequality and generalized Cauchy-Schwarz for PSD matrices,

$$\left| v_\theta^\top H(\theta) v_i \right| \leq \sum_{j=1}^{2} \left| v_\theta^\top \nabla^2 \mathcal{L}_{z_j}(\theta) v_i \right| \leq \sum_{j=1}^{2} \sqrt{v_\theta^\top \nabla^2 \mathcal{L}_{z_j}(\theta) v_\theta} \sqrt{v_i^\top \nabla^2 \mathcal{L}_{z_j}(\theta) v_i} \leq 2\sqrt{c_H c_{H,i}}.$$

Thus,

$$\left| u_{i,1}^\top H(\theta) v_i \right| \leq \epsilon_{i,1} c_{H,i} + 2\epsilon_\theta \sqrt{c_H c_{H,i}}.$$

Also, $\left| v_i^\top H(\theta) v_i \right| \leq c_{H,i}$, so

$$|\langle g(\theta), v_i \rangle| \leq \frac{2\delta}{\epsilon_{i,2} - \epsilon_{i,1}} + \epsilon_{i,1} c_{H,i} + 2\epsilon_\theta \sqrt{c_H c_{H,i}} + \frac{1}{2} (\epsilon_{i,2} - \epsilon_{i,1}) c_{H,i} \triangleq B_i.$$

Equivalently,

$$B_i = \frac{2\delta}{\epsilon_{i,2} - \epsilon_{i,1}} + \frac{1}{2} (\epsilon_{i,1} + \epsilon_{i,2}) c_{H,i} + 2\epsilon_\theta \sqrt{c_H c_{H,i}}.$$

Since $\{v_1, \ldots, v_r\}$ is an orthonormal basis of $\mathcal{V}$:

$$\|\text{Proj}_{\mathcal{V}}(g(\theta))\| = \sqrt{\sum_{i=1}^{r} \langle g(\theta), v_i \rangle^2} \leq \sqrt{\sum_{i=1}^{r} B_i^2}.$$

By assumption on orthogonal components:

$$\|\text{Proj}_{\mathcal{V}^\perp}(g(\theta))\| \leq \|\text{Proj}_{\mathcal{V}^\perp} \nabla \mathcal{L}_{z_1}(\theta)\| + \|\text{Proj}_{\mathcal{V}^\perp} \nabla \mathcal{L}_{z_2}(\theta)\| \leq 2\gamma.$$

Finally,

$$\begin{aligned}\|g(\theta)\| &\leq \|\text{Proj}_{\mathcal{V}}(g(\theta))\| + \|\text{Proj}_{\mathcal{V}^\perp}(g(\theta))\| \\ &\leq \sqrt{\sum_{i=1}^{r} \left(\frac{2\delta}{\epsilon_{i,2} - \epsilon_{i,1}} + \frac{1}{2}(\epsilon_{i,1} + \epsilon_{i,2}) c_{H,i} + 2\epsilon_\theta \sqrt{c_H c_{H,i}} \right)^2} + 2\gamma.\end{aligned}$$

□

Proof of Corollary 3.2. From Theorem 3.1, the directional term can be written as

$$B_i(\epsilon_{i,1}, \epsilon_{i,2}) = \frac{2\delta}{\epsilon_{i,2} - \epsilon_{i,1}} + \frac{1}{2}(\epsilon_{i,1} + \epsilon_{i,2}) c_{H,i} + 2\epsilon_\theta \sqrt{c_H c_{H,i}}.$$

First, for fixed $\epsilon_{i,2}$,

$$\frac{\partial B_i}{\partial \epsilon_{i,1}} = \frac{2\delta}{(\epsilon_{i,2} - \epsilon_{i,1})^2} + \frac{1}{2} c_{H,i} > 0,$$

so B_i is strictly increasing in $\epsilon_{i,1}$ and is minimized at $\epsilon_{i,1}^\star = 0$. Substituting $\epsilon_{i,1} = 0$ gives

$$B_i(0, \epsilon_{i,2}) = \frac{2\delta}{\epsilon_{i,2}} + \frac{1}{2}\epsilon_{i,2} c_{H,i} + 2\epsilon_\theta \sqrt{c_H c_{H,i}}.$$

Now differentiate with respect to $\epsilon_{i,2}$:

$$\frac{\partial B_i(0, \epsilon_{i,2})}{\partial \epsilon_{i,2}} = -\frac{2\delta}{\epsilon_{i,2}^2} + \frac{1}{2} c_{H,i} = 0,$$

which yields

$$\epsilon_{i,2}^\star = 2\sqrt{\frac{\delta}{c_{H,i}}}.$$

At this minimizer,

$$B_i^\star = \frac{2\delta}{2\sqrt{\delta / c_{H,i}}} + \frac{1}{2}\left(2\sqrt{\frac{\delta}{c_{H,i}}}\right) c_{H,i} + 2\epsilon_\theta \sqrt{c_H c_{H,i}} = 2\sqrt{\delta c_{H,i}} + 2\epsilon_\theta \sqrt{c_H c_{H,i}}.$$

Substitute into Theorem 3.1:

$$\begin{aligned}\|\nabla \mathcal{L}_{z_1}(\theta) - \nabla \mathcal{L}_{z_2}(\theta)\| &\leq \sqrt{\sum_{i=1}^{r} B_i^2} + 2\gamma \\ &= \sqrt{\sum_{i=1}^{r} \left(2\sqrt{c_{H,i}} \left(\sqrt{\delta} + \epsilon_\theta \sqrt{c_H} \right) \right)^2} + 2\gamma \\ &= 2\left(\sqrt{\delta} + \epsilon_\theta \sqrt{c_H} \right) \sqrt{\sum_{i=1}^{r} c_{H,i}} + 2\gamma.\end{aligned}$$

□

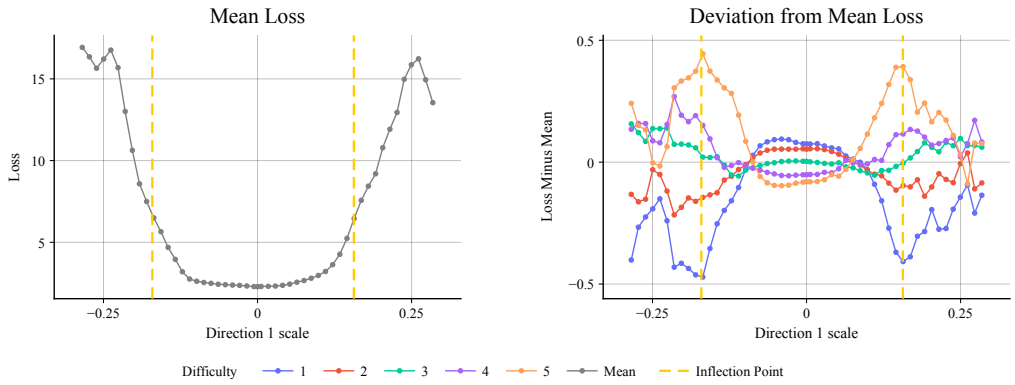


Figure 8: Left: The loss of Qwen2.5-3B-Instruct perturbed along a random direction on M23k dataset. Right: The average loss of problems in each difficulty level minus the average loss of all the problems. Within the pretrained model's proximity, the loss of difficult problems grows faster than the easy problems as the perturbation increases.

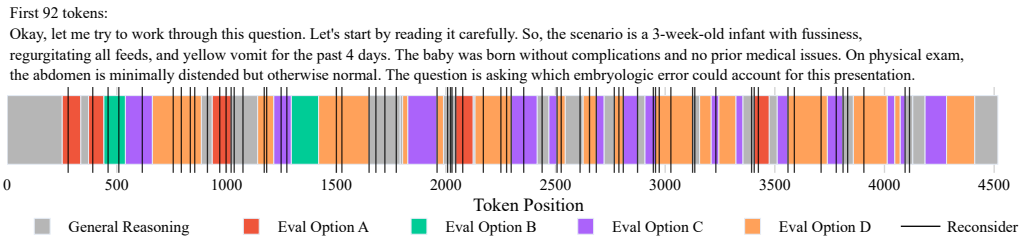


Figure 9: Example reasoning data from M23k dataset. Among the 4517 tokens, the teacher model reason back and forth about different options, and pauses to reconsider its answer multiple times.

B Mechanism of difficulty filtering

We provide further analysis on why the loss on perturbed model indicates difficulty of the problem. We first provide additional context of Figure 3 (right) and Figure 8. Both plots show the loss of the model Qwen2.5-3B-Instruct perturbed along random directions on the first 100 tokens of the M23k dataset to obtain $\mathcal{L}^{100}_{problem}(x, y, \theta_{perturbed})$. We labelled the difficulty of the problems in the dataset with Gemini-2.5-Flash into 5 levels, with level 1 being the easiest and level 5 being the hardest. The loss are grouped by difficulty level and averaged. We calculate the Pierson correlation coefficient between the loss and the difficulty level to obtain the correlation plot in Figure 3 (right). We calculate the average loss of all the problems to obtain the loss curve in Figure 8 (left). For Figure 8 (right), we calculated the average loss on each difficulty level and subtracted mean loss of all the problems.

Figure 8 provides insights into the mechanism of difficulty filtering. When we perturb the model, the loss doesn't increase to the same extent for easy and difficult problems. The subplot on the right shows that loss of easier problems grows slower than average while the loss of difficult problems grows faster as the perturbation increases while keeping the model close to the pretrained checkpoint (within the inflection points). Such difference magnifies the difference of loss between easy and difficult problems to up to 5x, making them more separable. Therefore, within the proximity of the pretrained checkpoints (indicated by the inflection ring), the larger the perturbation is, the more separable the loss of easy and difficult problems are. We utilize this insight to use Algorithm 3 to find the appropriate perturbation scale for the difficulty filtering.

Algorithm 3 Inflection-point search along one direction

Require: Perturbation direction $\hat{u}$, student parameters θ_0, dataset $\mathcal{D}$ of (x, y) pairs
Ensure: Inflection scale $\epsilon^\star$

For $\epsilon \geq 0$, define $\ell(\epsilon) \leftarrow \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathcal{L}^k_{problem}(x, y, \theta_0 + \epsilon\hat{u})\right]$
Initialize coarse scales $\mathcal{E} \subset [0, \epsilon_{\max}]$
Initialize history $\mathcal{H} \leftarrow \{(\epsilon, \ell(\epsilon)) : \epsilon \in \mathcal{E}\}$
for $t = 1, 2, \ldots$ **do**
    Reduce $\epsilon_{\max}$ if queried losses are not monotonically increasing ▷ Evaluations are far away from the inflection point
    Fit cubic spline surrogate $\hat{\ell}_t$ to $\mathcal{H}$
    $\epsilon_{\text{cand}} \leftarrow$ Root of $\frac{d^2}{d\epsilon^2}\hat{\ell}_t(\epsilon) = 0$
    if $\epsilon_{\text{cand}} \notin [0, \epsilon_{\max}]$ **then**
        Expand $\epsilon_{\max}$, query additional scales, and append to $\mathcal{H}$
        continue
    Query $L_{\text{cand}} \leftarrow \ell(\epsilon_{\text{cand}})$ and append $(\epsilon_{\text{cand}}, L_{\text{cand}})$ to $\mathcal{H}$
    if $\epsilon^\star \approx \epsilon_{\text{cand}}$ **then break else** $\epsilon^\star \leftarrow \epsilon_{\text{cand}}$
    if a fixed iteration cap is reached **then break**
$\ell^\star \leftarrow \ell(\epsilon^\star)$
return $(\epsilon^\star, \ell^\star, \hat{\ell}_t)$

C Perturbation selection algorithms

This section details the algorithm described in the main body.

Inflection-point search (Algorithm 3). Given a direction $\hat{u}$ and base parameters θ_0, we approximate the curve $\epsilon \mapsto \ell(\epsilon)$ with a cubic spline fit to queried points and refine the bracket until the second derivative of the surrogate crosses zero. Cubic functions are especially suitable for this purpose: they are smooth and have a second derivative that is easy to compute, which enables us to find the inflection point efficiently by solving numerically for the root. Further, when perturbations are small, the third-order term will be small and the cubic function behaves like a parabola, which is consistent with the loss landscape behavior around the pretrained checkpoint; as we move away from the pretrained checkpoint, the third-order term still dominates other higher order terms, and captures the non-convex behavior of the loss landscape. In the algorithm we actively shrink or expand the search interval to ensure that the cubic function is still a good approximation of the loss curve.

LoRA train-and-perturb (Algorithm 4). To probe multiple axes without relying on a single random direction, we fit a rank-r LoRA adapter, orthogonalize its rank-one factors $\{v_i\}$, and run Algorithm 3 along each v_i, as motivated by Theorem 3.1 in the main body. Theorem 3.1 also suggests that the perturbation scales should be far away from each other. We cap one perturbation scale to the inflection scale of the corresponding direction, as that is the maximum perturbation we are allowed to use to stay within the parabolic shape of the loss landscape; as for the other scale, we set it to the halfway point between the base loss and the inflection loss to maximize the richness of information in the loss signature. It ensures that the loss is different from the previous inflection loss, and also ensures that the perturbation are not too close to the pretrained checkpoint so that the loss of different directions are not too similar.

End-to-end pipeline (Algorithm 5). Algorithm 5 stitches difficulty filtering, and diversity filtering together into a complete pipeline for data curation. GENERATE(θ, D, k) refers to using the teacher model θ to (continue to) generate the response for the questions / partial responses in the dataset D up to k tokens. Note that for perturbations, we only need to sample a small number of N_{sample} data samples to probe the loss landscape or finetune the LoRA model, which is extremely efficient. The perturbed checkpoints will evaluate all the problems that are not filtered out at that stage.

We follow the hyperparameter settings in Jin et al. [12] to use $k_{\text{difficulty}} = 100$ and $k_{\text{diversity}} = 1000$.

Algorithm 4 LoRA train-and-perturb for directional checkpoints

Require: Pretrained checkpoint θ_0, dataset D, LoRA rank r
Ensure: Directional checkpoints $\{\theta_i^{(\text{infl})}, \theta_i^{(\text{half})}\}_{i=1}^r$

Train a rank-r LoRA adapter from θ_0 on D
Decompose the adapter into orthogonal rank-1 directions $\{v_i\}_{i=1}^r$
$L_0 \leftarrow \mathbb{E}_{(x,y)\sim D}\left[\mathcal{L}^k_{problem}(x, y, \theta_0)\right]$
for $i = 1, \ldots, r$ **in parallel do**
    $(\epsilon_i^{(\text{infl})}, L_i^{(\text{infl})}, \hat{\ell}_i) \leftarrow$ Algorithm 3(v_i, θ_0, D)
    $\epsilon_i^{(\text{half})} \leftarrow$ Root of $\hat{\ell}_i(\epsilon) = \left(L_0 + L_i^{(\text{infl})}\right)/2$
    $\theta_i^{(\text{infl})} \leftarrow \theta_0 + \epsilon_i^{(\text{infl})} v_i, \quad \theta_i^{(\text{half})} \leftarrow \theta_0 + \epsilon_i^{(\text{half})} v_i$
return $\{\theta_i^{(\text{half})}, \theta_i^{(\text{infl})}\}_{i=1}^r$

Algorithm 5 Complete SFTPROBE pipeline

Require: Student model θ_0, teacher model θ_{teacher}, seed question set D, target SFT size N
Require: Hyperparameters: Prefix length $k_{\text{difficulty}}$ and $k_{\text{diversity}}$ for teacher rollouts; Sample size N_{sample}; LoRA rank r
Ensure: Curated dataset

$D_{\text{difficulty}} \leftarrow \text{GENERATE}(\theta_{\text{teacher}}, D, k_{\text{difficulty}})$
Sample a random unit direction $\hat{u}$ in parameter space
$(\epsilon^\star, \cdot, \hat{\ell}) \leftarrow$ Algorithm 3$(\hat{u}, \theta_0, \text{SAMPLE}(D_{\text{difficulty}}, N_{\text{sample}}))$
$\theta_{\text{diff}} \leftarrow \theta_0 + \epsilon^\star \cdot \hat{u}$
For each $(x_j, y_j) \in D_{\text{difficulty}}$, compute $d_j \leftarrow \ell^{k_{\text{difficulty}}}(x_j, y_j, \theta_{\text{diff}})$
Cluster d_j by source into two groups; let D_{hard} be the high-loss cluster for each source
$D_{\text{diversity}} \leftarrow \text{GENERATE}(\theta_{\text{teacher}}, D_{\text{hard}}, k_{\text{diversity}})$
$\mathbf{\Theta}_{\text{pert}} \leftarrow$ Algorithm 4$(\theta_0, \text{SAMPLE}(D_{\text{hard}}, N_{\text{sample}}), r)$
Build loss signature: $\forall (x, y) \in D_{\text{diversity}}$, form $L_j \leftarrow \left(\ell^{k_{\text{diversity}}}(x, y, \theta)\right)_{\theta \in \mathbf{\Theta}_{\text{pert}}}$
Cluster $\{L_j\}$ (Equation (1)) by source with number of clusters determined by Equation (4)
$D_{\text{sel}} \leftarrow$ sample the same number of examples from each cluster
return GENERATETOCOMPLETION$(\theta_{\text{teacher}}, D_{\text{sel}})$

D Additional information and results on experiments

D.1 Datasets and benchmarks

Medical Reasoning We follow the setting of Huang et al. [8], which curated a M1k dataset containing 1k examples from the M23k dataset, containing 4 sources form 23k examples. It comes with 10 evaluation benchmarks containing a total of 11k medical reasoning questions. The evaluation benchmark includes 3 in-distribution (ID) ones (MedMCQA [26], MedQA-USMLE [11], and PubMedQA [13]), and 7 out-of-distribution (OOD) ones (MMLU-Pro (Medical) [31], GPQA (Medical) [28], Lancet and NEJM clinical case reports, MedBullets [3], MedXpertQA [36]).

Math Reasoning NuminaMath [15] dataset contains 859k math problems with ground truth answers. Among them, Guha et al. [6] considered only 4 sources, namely "amc_aime", "aops_forum", "math", "olympiads". These 4 sources contain 192k data and they selected 89k of them and have Deepseek-R1 model to generate the reasoning traces. We use the same four sources and considerred QwQ-32B to generate the examples. For evaluation, the 4 benchmarks we use are the 4 math benchmarks presented in Guha et al. [6], namely MATH500 [17], AMC23, AIME24, AIME25 [20]. The latter three benchmarks are relatively small, and we repeat the evaluation 6 times for a more accurate evaluation of mean and std.

Physics reasoning We combine two datasets for training, namely Siddique et al. [29] containing 17,635 textbook physics questions (12 removed for empty questions or ground-truth answers) and 3084 olympiad questions we curated by ourselves. The olympiad dataset originates from Zheng et al. [34], yet their dataset contains OCR errors, subquestions, and images. We prompted LLMs

with few-shot prompt to sanitize the errors, split the subquestions while carrying on all necessary information from presious subquestions, and describe the image or discard the problem depending on the nature of the image. We end up repeating harder but smaller subsets to ensure that we have enough questions to select from these harder questions. Specifically, we repeat the questions from WoPho, RMPh, EuPhO 16 times, IPhO 4 times, NbPhO 9 times, USAPhO 3 times, and APhO 6 times. These numbers are selected based on the desired number of data to select from according to our algorithm. Combined with the 17,635 textbook questions, we end up with 39595 physics questions to curate the SFT dataset from. For evaluation, We use the same evaluation benchmark as Team [30], except that we only evaluated on the text data as the scope of this work is text reasoning. The benchmark contains 102 questions from seven 2024–2025 competitions, and we repeat the evaluation 3 times to take the average.

D.2 Data Decontamination

We follow the same decontamination protocol as the one used in Huang et al. [8] and Guha et al. [6], where a fuzzy search is applied. Specifically, the normalized indel similarity is calculated, which is based on how much insertions and deletion is required to turn an example from the train set into one in the test set. This identifies not only exact matches, but also similar questions with minor variations. We decontaminated ourselves for the numina dataset; we verified that the Physics dataset is not contaminated, which is expected because our training set only incluses textbook problems from Siddique et al. [29] and Olympic questions we curated ourselves from 1967-2018. We also verified that the M23k dataset provided by Huang et al. [8] is already decontaminated as claimed by the authors.

D.3 Hyperparameters

We randomly sample a subset of data containing 800 examples for training with LoRA to obtain the directions. We use LoRA rank 4 for all tasks other than math reasoning (which we use LoRA rank 2 as it's more efficient). We train LoRA for 200 iterations. This is especially efficient: For instance, in the setting of data generation for medical with QwQ-32B, the entire cost for generating LoRA examples and training LoRA is only 16 minutes. Other hyperparameters are consistent with the ones in [12]. Specifically, the temperature was set to 0.7 for math and physics and 0 for medical reasoning, and all other generation configs were engine defaults. Since Qwen2.5 models have a max context of 32k tokens, we cap response length to 27k tokens for math and physics to set aside 5k tokens for prompt. For medical, we cap the max response length to 16k tokens to match the implementation in Huang et al. [8]. Without specification, we repeat the inference on each benchmark twice to get the mean and std.

D.4 Baselines

Baselines in medical reasoning and math reasoning are consistent with Jin et al. [12]. This includes:

- Learnability [35]: Learnability measures how much loss of an example decreases during the training and selects examples that are most learnable by the model, i.e. experience the largest drop in loss during training.
- Middle Perplexity [21]: Given a trained model, it ranks all the data by perplexity and takes the required number of data around the median. The assumption is that easiest examples with lowest perplexity do not benefit training and hardest examples with highest perplexity are noisy or outliers.
- Embedding Clustering [2]: Given a decoder-only LLM, the method takes the last hidden state as the feature, and uses k-center algorithm on the embedding to pick a set of diverse examples.
- TEMP [12]: This method extends Yang et al. [32] to long reasoning datasets by identifying the initial tokens reflects data quality. It fine-tunes the model on the full SFT data and tracks the loss values of the initial reasoning tokens for all examples at noisy checkpoints along the fine-tuning directions. Then, it clusters the loss trajectories and samples examples with higher loss values from the clusters. This approach fundamentally depends on access to the full SFT dataset and thus cannot be applied for data generation.

E Additional details and discussions on teacher selection

In this section, we provide additional details and results on teacher selection.

Loss evaluation We detail how the blue bars in Figure 4 are calculated. We randomly select 1000 questions from the NuminaMath dataset and prompt the teacher model to generate the reasoning traces with up to 150 tokens, with the extra 50 tokens to account for the variation in tokenization, using the recommended generation parameters for each model individually. For teacher selection, we prompt rach teacher model to "Explain what the problem is asking in your own words." The pretrained student model evaluates the first 100 tokens of the teachers generation using the standard answering prompt, namely, "`Return your final response within \boxed{}. {question}`" as if the teacher model started its reasoning with rephrasing the problem.

Finetuning the student We detail how the green bars in Figure 4 are calculated. We randomly select 1000 questions from the NuminaMath dataset and generate with teacher model up to 16384 tokens with the prompt "`Return your final response within \boxed{}. {question}`" using the recommended generation parameters for each model individually to form a SFT dataset for each teacher model. We then finetune the student model (Qwen2.5-7B-Instruct) on the generated dataset and evaluate the accuracy on the test set. The performance is shown in Table 4.

Table 4: Average accuracy (%) of the student model finetuned on the a same set of 1000 randomly selected questions from the NuminaMath dataset, generated by different teacher models.

Teacher	Avg.	AIME24	AIME25	AMC23	MATH500
Qwen/Qwen3.6-27B	14.74 ± 0.62	0.00 ± 0.00	0.00 ± 0.00	16.88 ± 2.39	42.10 ± 0.14
Openai/GPT-OSS-120B	14.52 ± 0.82	2.78 ± 2.51	1.11 ± 1.72	19.58 ± 2.46	34.60 ± 0.00
Google/Gemma-4-26B-A4B-IT	24.88 ± 0.35	7.78 ± 1.72	3.33 ± 0.00	32.50 ± 0.00	55.90 ± 0.71
Google/Gemma-4-31B-IT	34.43 ± 0.60	8.33 ± 1.92	13.33 ± 2.72	46.25 ± 1.44	69.80 ± 0.00
Microsoft/Phi-4-Reasoning-Plus	37.26 ± 2.94	19.44 ± 8.28	13.33 ± 0.00	46.67 ± 3.76	69.60 ± 0.00
Qwen/QwQ-32B	43.56 ± 1.23	15.83 ± 1.67	18.33 ± 5.77	61.88 ± 1.25	78.20 ± 0.28

Discussion on the necessity of teacher rephrasing In Equation (5), we rank the teachers based on the rephrasing loss instead of the the loss of the actual reasoning in the solution given by the teacher. This is because selection based on rephrasing and reasoning optimize for opposing dynamics, and applying the same low-loss selection strategy to the first part of a reasoning chain fundamentally degrades performance. While prompt rephrasing merely changes the problem's external framing, the initial steps of a reasoning trace determine its entire trajectory. Selecting reasoning steps based on low student loss systematically discards the teacher's most valuable contributions–such as novel abstraction pivots, non-obvious problem decompositions, and counter-intuitive intermediate steps-—simply because they are unfamiliar to the student. Instead of encouraging the student to learn new, sophisticated problem-solving pathways, this filtering mechanism forces the model into a local minimum, prioritizing predictability over logical correctness.

While most models start their response with summarizing or rephrasing the problem, there are counterexamples like Gemma-4 series and Qwen3.6 series models, which directly start solving the problem. The rephrasing prompt we use therefore ensures that we're measuring the same metric across different teacher models, namely how good can the student follow the teacher's rephrasing of the problem itself. Our additional ablation on teacher selection shown in Table 5 confirms that, without prompting the teacher to explain the problem, the loss of the initial reasoning tokens does not correlate with finetune accuracy:

F Empirical evidence of the increase of diversity

The diversity metrics we aim to optimize for is the gradient diversity along the LoRA directions. To quantify the increment in diversity, we calculate the norm of pairwise gradient difference projected on the LoRA dimensions. Specifically, we take all the data after diversity filtering of a source, and take the same number of data from the same source before diversity filtering. With the full gradients

Table 5: Loss of initial reasoning tokens vs. the average finetune accuracy reported in Figure 4. Without prompting the teacher to explain the problem, the loss of the initial reasoning tokens does not correlate with finetune accuracy.

Teacher	Mean Loss	Avg. Finetune Acc.
Google/Gemma-4-26B-A4B-IT	1.0045 ± 0.311	$24.88\% \pm 0.35\%$
Google/Gemma-4-31B-IT	1.0049 ± 0.295	$34.43\% \pm 0.60\%$
Qwen/QwQ-32B	1.3022 ± 0.302	$43.56\% \pm 1.23\%$
Qwen/Qwen3.6-27B	1.3467 ± 0.415	$14.74\% \pm 0.62\%$
Openai/GPT-OSS-120B	1.7742 ± 0.445	$14.52\% \pm 0.82\%$
Microsoft/Phi-4-Reasoning-Plus	1.8296 ± 0.479	$37.26\% \pm 2.94\%$

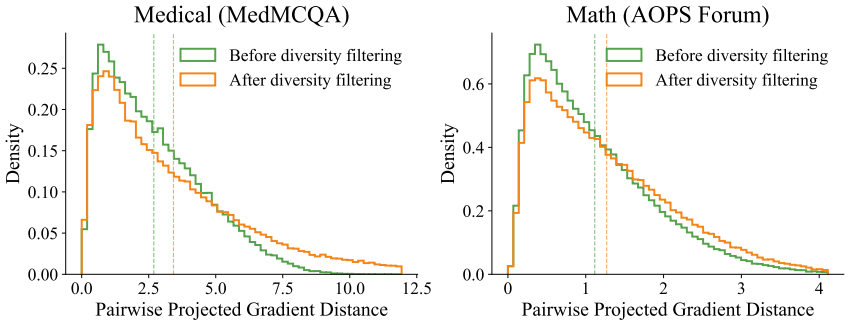


Figure 10: The distribution of the pairwise projected gradient difference before and after diversity filtering. Vertical line measures the mean. Diversity filtering increased the pairwise difference, indicating an increases in the diversity of the gradients.

at the pretrained checkpoint g^i for example i, and the LoRA directiosn being

$$V = \begin{pmatrix} v_1^\top \\ v_2^\top \\ \vdots \\ v_r^\top \end{pmatrix}$$

where r is the LoRA rank, we calculate the distribution of

$$\Delta_{ij} = \|Vg_i - Vg_j\|_2$$

and present in Figure 10.

The root-mean-square of the norm, namely $\sqrt{\mathbb{E}_{i \neq j}\, \Delta_{ij}^2}$, changes from 3.26 to 4.41 (35.2% increase) and 1.36 to 1.55 (13.8% increase) on medical and math respectively, verifying the increased diversity.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All claims have been accurately reflected.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We acknowledge the requirement of minimal training and not curating a larger dataset.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.

- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Complete proof is given in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Detailed hyperparameters have been provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.

- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and data is provided in the link in the appendix, which contains reproduction instructions.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We include all experimental details in the appendix

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We included error bars and stds in our plots.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Details are included in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We conform the Code of Ethics strictly.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We do not have negative impact. Our method facilitates efficient AI and is environmentally friendly.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We do not pose such risk.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited all original works and respected the terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We have released our code and dataset assets with proper documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not involve crowdsource.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We do not involve crowdsource.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We followed the LLM policy.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.