

When the Strongest Teacher Is Not the Best Teacher: Student-Centric Answer Selection

Zhengyu Hu^{1,*}, Zheyuan Xiao^{2,*}, Linxin Song³, Fengqing Jiang¹, Yuetai Li¹,
Zhihan Xiong¹, Yue Liu⁴, Junhao Lin⁵, Yao Su⁶, Lijie Hu⁷,
Kaize Ding⁸, Teng Xiao^{1,9,✉}, Radha Poovendran¹

¹ University of Washington ² University of Texas at Austin ³ USC

⁴ National University of Singapore ⁵ Microsoft ⁶ WPI

⁷ MBZUAI ⁸ Northwestern University ⁹ Allen Institute for AI (AI2)

*Equal contribution. ✉Corresponding author.

Abstract

LLM training increasingly relies on teacher-generated supervision, from synthetic responses to reasoning traces and tool-use demonstrations. Current practice often chooses the highest-performing teacher to generate student training data, implicitly treating teacher test performance as a proxy for teaching quality. We show that this assumption can fail: even when multiple teachers provide correct answers to the same question, the answer from the strongest teacher is not necessarily the best supervision for a given student. To address this gap, we propose *Student-Centric Answer Sampling* (SCAS), a framework that selects from verified teacher-generated answers according to their estimated student-centric learning cost. Motivated by a token-wise gradient decomposition, we derive an efficient forward-only proxy for this cost and use it to guide answer selection during training. Experiments across 30 teacher models, 6 student base models, and 6 tasks show that SCAS consistently improves student performance, suggesting that effective distillation should prioritize supervision matched to the current student rather than teacher strength alone.

1 Introduction

Modern LLM training increasingly relies on model-generated supervision, as high-quality human annotation is expensive to collect and limited in coverage (Hu et al., 2024d; Chen et al., 2025b; Hu et al., 2025b; Djuhera et al., 2025; Nadăș et al., 2025). Teacher models now produce synthetic responses (Wang et al., 2022; Xu et al., 2024, 2026b), reasoning traces (Yu et al., 2025; Yang et al., 2026b), agent trajectories (Li et al., 2025; Wang et al., 2025b; Chen et al., 2025a), and tool-use demonstrations (Liu et al., 2024; Xu et al., 2025c; Luo et al., 2025b) at scale, making teacher-generated data central to distillation-style LLM training (Fang et al., 2025; Shen et al.,

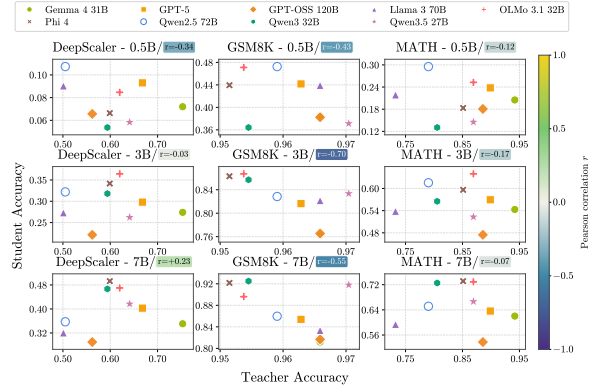


Figure 1: **Teacher-student Accuracy Comparison.** Higher teacher accuracy does not consistently yield higher student accuracy. Colored r values denote the per-panel teacher-student rank correlation.¹

2025; Burns et al., 2023; Gu et al., 2023; Agarwal et al., 2023). In practice, teacher selection is often teacher-centric: the strongest available model is chosen to generate student training data (Xu et al., 2024; Abdin et al., 2024; Xu et al., 2025b; Li et al., 2025), implicitly assuming that *teacher competence* is a reliable proxy for *teaching quality*.

However, our teacher-student comparison as shown in Figure 1 challenges this assumption. We evaluate nine teacher models on the test set of each task, then use each teacher to generate training answers for Qwen2.5 students of three sizes across DeepScaleR, GSM8K, and MATH. To isolate teaching quality from answer correctness, the training data for this comparison is restricted to the largest common subset of questions for which all nine teachers produce correct answers; details are provided in Appendix A.4. Despite this controlled setting, higher teacher accuracy does not consistently translate into higher student accuracy. A teacher may have stronger absolute task performance, yet the supervision it provides need not be equally effective for every student.

¹See Appendix A.4 for the definition of the correlation coefficient.

These observations suggest that supervision quality should be viewed as student-centric. Given multiple teachers for the same question, the best supervision may not come from the teacher with the highest test performance, but from the answer that best matches the current student.

To make this idea concrete, we view the suitability of a teacher-generated answer through the update it would induce on the current student. In supervised fine-tuning, pairing the same question with different correct answers creates different token sequences, which lead to different hidden states, prediction errors, and gradients. We therefore analyze the gradient induced by a question–answer sequence and decompose its squared norm into answer–answer (AA), answer–conditioning (AQ/QA), and conditioning–conditioning (QQ) interaction blocks. This decomposition clarifies why correctness alone does not determine supervision quality: two correct teacher answers can require different changes from the student, depending on how their answer tokens interact with one another and with the question context. We refer to this student-dependent update burden as **student-centric learning cost**. To avoid expensive backpropagations for computing these gradient interactions, we derive an efficient forward-only proxy for estimating student-centric learning cost. Then, building on this proxy, we propose **Student-Centric Answer Sampling (SCAS)**, a training framework that selects from verified teacher-generated answers according to their estimated student-centric learning cost. Rather than defaulting to the highest-performing teacher, SCAS prioritizes lower-cost answers that better match the current student’s learning state. To evaluate this idea, we collect candidate answers from 30 widely used open- and closed-source models and conduct experiments across 6 different student base models and 6 representative tasks. Our results show that student-centric answer selection is an effective principle for turning diverse teacher generations into better training data.

2 Method

We organize this section as follows. Section 2.1 first formalizes the candidate-answer setting and motivates student-centric learning cost, starting from the verified answer pool illustrated in ❶. Section 2.2 then derives an efficient forward-only proxy for estimating this cost under the current student, corresponding to the scoring module in ❷. Section 2.3 finally describes how these scores

are used to sample training answers and iteratively update the student, as shown in ❸. The overall framework is shown in Figure 2.

2.1 Formulation and Motivation

Setup and notation. Let $\mathcal{D} = \{q_i\}_{i=1}^N$ denote the training questions. For each $q \in \mathcal{D}$, let $\mathcal{A}(q)$ denote the set of verified correct teacher-produced answers² for q . The student is an autoregressive language model $p_\theta(\cdot)$ with parameters θ . For a candidate answer $a \in \mathcal{A}(q)$, we form the training-templated sequence $x = [q; a]$. Let $I = \{1, \dots, |x|\}$ denote the token positions of x , and write $I = I_Q \cup I_A$, where $I_Q = \{1, \dots, |q|\}$ indexes the question tokens and $I_A = \{|q| + 1, \dots, |x|\}$ indexes the answer tokens. For each token position $t \in I$, let h_t denote the last-layer hidden state at position t under the training template. A summary of notation is provided in Appendix A.6.

Token-wise gradient decomposition. We first characterize how token-level interactions contribute to the gradient of a generic student weight matrix. Let $W \in \mathbb{R}^{d \times m}$ be any weight matrix in the student, let $X = [z_t^\top]_{t \in I} \in \mathbb{R}^{|I| \times d}$ denote the token representations feeding into W , and let $A = XW \in \mathbb{R}^{|I| \times m}$ be the corresponding pre-activations. For a given layer, the rows of X can be taken as the token representations at that layer feeding into W ; our derivation is agnostic to the specific choice. Writing $G = [g_t^\top]_{t \in I} \in \mathbb{R}^{|I| \times m}$ for the loss gradient with respect to A , standard matrix calculus gives

$$\nabla_W \mathcal{L} = X^\top G. \quad (1)$$

We are interested in the squared Frobenius norm of this gradient,

$$\|\nabla_W \mathcal{L}\|_F^2 = \|X^\top G\|_F^2. \quad (2)$$

Expanding the Frobenius norm into pairwise inner products (Appendix C), and writing $z_i^\top$ and $g_i^\top$ for the i -th rows of X and G , yields

$$\|\nabla_W \mathcal{L}\|_F^2 = \sum_{i,j \in I} (z_i^\top z_j) (g_i^\top g_j). \quad (3)$$

We now split token indices into conditioning and answer positions, $I = I_A \cup I_Q$, and separate

²See Appendix G for verification details.

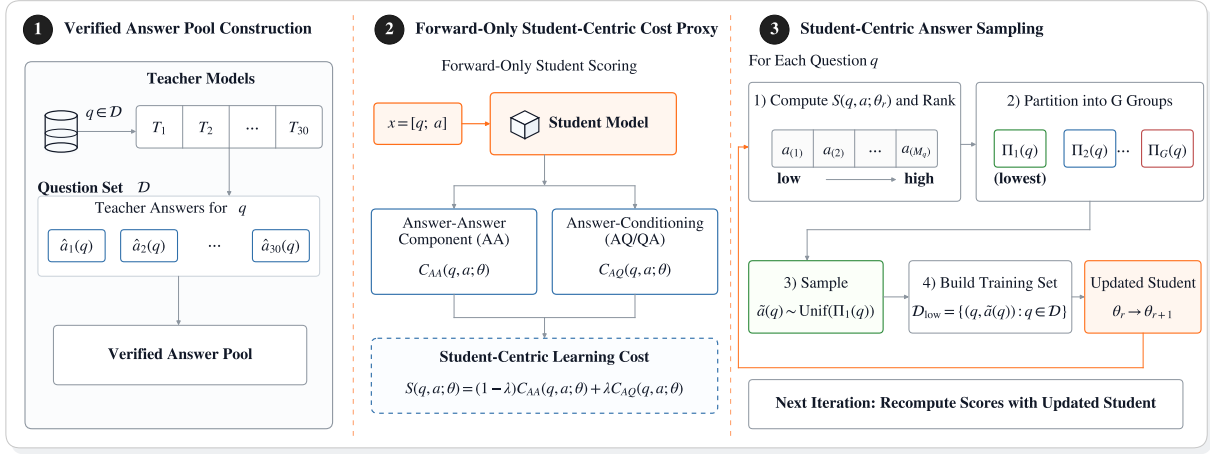


Figure 2: **Framework overview of Student-Centric Answer Sampling (SCAS).** ① A verified answer pool is constructed from multiple teacher-generated answers for each question. ② The current student estimates a forward-only student-centric learning cost for each candidate answer. ③ Candidates are ranked and grouped by the estimated cost, and low-cost answers are sampled to form a round-specific training set for updating the student. Scores are recomputed with the updated student in the next iteration.

Eq. (3) into answer–answer, answer–conditioning, and conditioning–conditioning contributions:

$$\begin{aligned}
 \|\nabla_W \mathcal{L}\|_F^2 = & \underbrace{\sum_{i,j \in I_A} (z_i^\top z_j)(g_i^\top g_j)}_{\text{AA}} \\
 & + 2 \underbrace{\sum_{i \in I_A, j \in I_Q} (z_i^\top z_j)(g_i^\top g_j)}_{\text{AQ/QA}} \\
 & + \underbrace{\sum_{i,j \in I_Q} (z_i^\top z_j)(g_i^\top g_j)}_{\text{QQ}}.
 \end{aligned} \quad (4)$$

This decomposition makes explicit how different parts of the input sequence jointly contribute to the update induced on the student.

Motivation. Eq. (4) shows that the update induced by a question–answer sequence depends on how answer tokens interact with one another and with the conditioning tokens. Thus, for a fixed question, different verified teacher answers can induce different gradient interaction patterns and impose different update burdens on the current student. We refer to this student-dependent update burden as *student-centric learning cost*. This cost is the quantity we seek to estimate when ranking candidate answers: lower-cost answers provide training signals that better match the current student. Directly computing these blockwise gradient interactions requires backpropagated token gradients g_i for every candidate answer, which is too expensive for

large-scale answer selection. We therefore derive an efficient forward-only proxy for student-centric learning cost in the next subsection.

2.2 A Forward-only Proxy for Student-Centric Learning Cost

Having defined student-centric learning cost as the update burden induced by a candidate answer, we now derive a scalable proxy for estimating it. Guided by the block structure in Eq. (4), we focus on the answer-involving AA and AQ/QA blocks. For a fixed question, the QQ term exhibits much smaller cross-candidate variance than the answer-involving blocks, and therefore contributes little to answer ranking; empirical evidence is provided in Section 4.6.

The remaining challenge is to approximate these retained interactions using only forward-pass signals. We adopt a two-step reduction. First, we replace residual magnitudes in the retained last-layer interactions with token-level forward uncertainty, yielding a token-sensitive proxy that preserves the AA and AQ/QA structure. Second, we replace token-level uncertainty within each block by its mean, obtaining a more stable sequence-level proxy. Concretely, this leads to using mean token NLL to summarize candidate difficulty under the current student, while hidden-state interactions capture the retained AA and AQ/QA structure in a blockwise manner. This yields an efficient forward-only score for ranking multiple correct answers. A theoretical analysis is provided in Appendix I.

We instantiate the forward-only proxy using final-layer hidden states, for which the corresponding LM-head residuals admit a direct connection to token-level NLL; layer ablations in Section E further support this choice. Given $x = [q; a]$, we run the student once and extract the final-layer hidden states $H_\theta(x) = [h_t]_{t \in I}$. We normalize each token representation as $u_t = h_t / \|h_t\|$, and define the answer and conditioning representation aggregates

$$\mu_A = \frac{1}{|I_A|} \sum_{i \in I_A} u_i, \quad \mu_Q = \frac{1}{|I_Q|} \sum_{j \in I_Q} u_j. \quad (5)$$

We further define sequence-level difficulty signals using mean token NLL:

$$\begin{aligned} d_A &= \frac{1}{|I_A|} \sum_{i \in I_A} -\log p_\theta(x_i \mid q, a_{<i}), \\ d_Q &= \frac{1}{|I_Q|} \sum_{j \in I_Q} -\log p_\theta(x_j \mid x_{<j}). \end{aligned} \quad (6)$$

Our score is based on a blockwise difficulty approximation: within each interaction block, we use a shared sequence-level difficulty scalar to modulate representation-side interactions. For the AA block, we define

$$C_{AA}(q, a; \theta) = d_A^2 \|\mu_A\|^2, \quad (7)$$

where $\|\mu_A\|^2$ summarizes answer-side pairwise similarity structure. For the AQ block, we define

$$C_{AQ}(q, a; \theta) = d_A d_Q (\mu_A^\top \mu_Q), \quad (8)$$

where $\mu_A^\top \mu_Q$ summarizes answer-conditioning interaction structure.

Putting these together, we define the student-centric learning cost as

$$S(q, a; \theta) = (1 - \lambda) C_{AA}(q, a; \theta) + \lambda C_{AQ}(q, a; \theta), \quad (9)$$

where $\lambda \in [0, 1]$ controls the trade-off between the two terms. $S(q, a; \theta)$ is a forward-only proxy for ranking multiple already-correct answers under the current student. It quantifies how costly a correct answer is for the student to learn from at its current stage. Lower-cost candidates are interpreted as answers that the current student can absorb with less effort, and are therefore preferred for data selection. A theoretical justification of the proposed score is provided in Appendix I.

Algorithm 1 Student-Centric Answer Sampling (SCAS)

Require:

training questions $\mathcal{D}$; candidate answer sets $\{\mathcal{A}(q)\}_{q \in \mathcal{D}}$; number of groups G ; initial student parameters θ_0 ; number of selection-training rounds R

```

1: for selection-training round  $r = 0, 1, \dots, R - 1$  do
2:   for all  $q \in \mathcal{D}$  do
3:     Compute  $S(q, a; \theta_r)$  for all  $a \in \mathcal{A}(q)$ 
4:     Sort  $\mathcal{A}(q)$  in ascending order of  $S(q, a; \theta_r)$ 
5:     Let  $M_q = |\mathcal{A}(q)|$ 
6:     Write the sorted candidates as  $a_{(1)}, \dots, a_{(M_q)}$ 
7:     for  $g = 0, 1, \dots, G$  do
8:       Set  $b_g = \lfloor g M_q / G \rfloor$ 
9:     end for
10:    for  $g = 1, 2, \dots, G$  do
11:      Set  $\Pi_g(q) = \{a_{(k)} : b_{g-1} < k \leq b_g\}$ 
12:    end for
13:    Sample  $\tilde{a}(q) \sim \text{Unif}(\Pi_1(q))$ 
14:  end for
15:  Form  $\mathcal{D}_{\text{low}} = \{(q, \tilde{a}(q)) : q \in \mathcal{D}\}$ 
16:  Train the student on  $\mathcal{D}_{\text{low}}$  and obtain  $\theta_{r+1}$ 
17: end for
```

2.3 Student-Centric Answer Sampling

Directly selecting the single lowest-score answer for each question can be brittle, as the exact per-question minimum may be sensitive to score noise, outliers, or small differences among near-tied candidates. We therefore adopt SCAS, a stratified sampling procedure that preserves low learning cost while improving robustness; ablation results supporting this design are provided in Section 4.3.

At selection-training round r , let θ_r denote the current student parameters. For each question $q \in \mathcal{D}$, we compute $S(q, a; \theta_r)$ for all verified candidates $a \in \mathcal{A}(q)$ and sort the candidates in ascending order. Let $M_q = |\mathcal{A}(q)|$, and write the sorted candidates as $a_{(1)}, a_{(2)}, \dots, a_{(M_q)}$. We then partition the sorted list into G consecutive groups $\Pi_1(q), \Pi_2(q), \dots, \Pi_G(q)$ using floor-based boundaries, where $\lfloor \cdot \rfloor$ denotes the floor function:

$$\Pi_g(q) = \left\{ a_{(k)} : \left\lfloor \frac{(g-1)M_q}{G} \right\rfloor < k \leq \left\lfloor \frac{gM_q}{G} \right\rfloor \right\}. \quad (10)$$

Here, $g = 1, \dots, G$. By construction, $\Pi_1(q)$ is the lowest-score group.

We then sample one answer uniformly from the lowest-score group, $\tilde{a}(q) \sim \text{Unif}(\Pi_1(q))$, for every question q , and construct the training set

$$\mathcal{D}_{\text{low}} = \{(q, \tilde{a}(q)) : q \in \mathcal{D}\}. \quad (11)$$

The student is trained on $\mathcal{D}_{\text{low}}$ to obtain the next iterate θ_{r+1} . The full SCAS procedure is given in Algorithm 1.

3 Experiments Setup

Models & Datasets. For each math training question, we use 30 teacher models and keep

Table 1: **Overall Performance of Answer-selection Methods on Qwen2.5 Student Models.** Higher is better for all metrics. Best and second-best results within each student model are marked in **red** and **blue**, respectively.

Model	Method	DeepScaleR	GSM8K	MATH	OpenR1-Math	IFEval	LiveBench	Avg.
Qwen2.5-0.5B	RSR	7.13 $\pm$ 0.17	26.26 $\pm$ 0.35	21.62 $\pm$ 0.38	6.40 $\pm$ 0.24	22.77 $\pm$ 0.20	24.30 $\pm$ 0.21	18.08
	GRACE	5.67 $\pm$ 0.17	37.65 $\pm$ 0.53	24.13 $\pm$ 0.31	8.30 $\pm$ 0.14	23.32 $\pm$ 0.19	26.90 $\pm$ 0.21	21.00
	RSR-Item	4.53 $\pm$ 0.42	16.10 $\pm$ 0.35	11.69 $\pm$ 0.32	7.77 $\pm$ 0.46	23.47 $\pm$ 0.28	26.80 $\pm$ 0.20	15.06
	PPL	4.10 $\pm$ 0.14	6.07 $\pm$ 0.31	6.44 $\pm$ 0.22	6.23 $\pm$ 0.38	23.29 $\pm$ 0.23	26.30 $\pm$ 0.26	12.07
	IFD	5.23 $\pm$ 0.48	7.00 $\pm$ 0.34	6.59 $\pm$ 0.21	6.03 $\pm$ 0.29	24.03 $\pm$ 0.19	27.10 $\pm$ 0.14	12.66
	Router	6.63 $\pm$ 0.74	8.01 $\pm$ 0.81	9.75 $\pm$ 0.06	6.50 $\pm$ 0.50	23.84 $\pm$ 0.40	26.70 $\pm$ 0.52	13.57
	Influence	8.77 $\pm$ 0.54	7.46 $\pm$ 0.16	22.98 $\pm$ 0.26	7.63 $\pm$ 0.80	23.29 $\pm$ 0.33	26.90 $\pm$ 0.34	16.17
	SCAS	10.10 $\pm$ 0.85	45.03 $\pm$ 0.22	27.62 $\pm$ 0.07	10.77 $\pm$ 0.19	24.40 $\pm$ 0.16	27.20 $\pm$ 0.17	24.19
Qwen2.5-3B	RSR	21.53 $\pm$ 0.29	59.92 $\pm$ 0.35	45.02 $\pm$ 0.50	17.29 $\pm$ 0.45	56.56 $\pm$ 0.25	42.20 $\pm$ 0.35	40.42
	GRACE	28.40 $\pm$ 0.65	76.80 $\pm$ 0.03	57.55 $\pm$ 0.50	30.90 $\pm$ 0.78	58.75 $\pm$ 0.48	42.30 $\pm$ 0.55	49.12
	RSR-Item	20.07 $\pm$ 0.54	34.95 $\pm$ 0.59	26.91 $\pm$ 0.14	18.20 $\pm$ 0.37	59.15 $\pm$ 0.49	41.60 $\pm$ 0.30	33.48
	PPL	19.40 $\pm$ 0.71	18.88 $\pm$ 0.22	16.57 $\pm$ 0.36	15.33 $\pm$ 1.03	60.40 $\pm$ 0.47	39.90 $\pm$ 0.50	28.41
	IFD	26.80 $\pm$ 0.88	18.32 $\pm$ 0.04	16.12 $\pm$ 0.03	24.57 $\pm$ 0.82	60.48 $\pm$ 0.42	41.60 $\pm$ 0.27	31.32
	Router	22.33 $\pm$ 1.11	23.20 $\pm$ 0.45	25.79 $\pm$ 0.53	18.70 $\pm$ 0.51	59.89 $\pm$ 0.55	41.70 $\pm$ 0.34	31.94
	Influence	27.83 $\pm$ 0.73	25.22 $\pm$ 0.74	49.98 $\pm$ 0.43	28.05 $\pm$ 0.65	59.89 $\pm$ 0.64	41.60 $\pm$ 0.52	38.76
	SCAS	33.20 $\pm$ 0.21	82.82 $\pm$ 0.35	61.69 $\pm$ 0.09	33.20 $\pm$ 0.43	61.37 $\pm$ 0.33	42.30 $\pm$ 0.20	52.43
Qwen2.5-7B	RSR	30.20 $\pm$ 0.57	62.55 $\pm$ 0.31	53.89 $\pm$ 0.31	27.20 $\pm$ 0.71	60.07 $\pm$ 0.41	47.40 $\pm$ 0.37	46.89
	GRACE	38.97 $\pm$ 1.30	86.85 $\pm$ 0.47	65.71 $\pm$ 0.21	38.93 $\pm$ 0.24	62.70 $\pm$ 0.26	47.50 $\pm$ 0.18	56.78
	RSR-Item	24.37 $\pm$ 0.91	55.12 $\pm$ 0.65	37.40 $\pm$ 0.07	24.97 $\pm$ 0.54	61.00 $\pm$ 0.51	48.40 $\pm$ 0.46	41.88
	PPL	18.03 $\pm$ 0.41	20.60 $\pm$ 0.47	18.04 $\pm$ 0.03	22.17 $\pm$ 0.82	62.29 $\pm$ 0.40	47.80 $\pm$ 0.43	31.49
	IFD	17.73 $\pm$ 0.78	20.32 $\pm$ 0.24	18.25 $\pm$ 0.11	21.30 $\pm$ 0.62	61.92 $\pm$ 0.39	47.90 $\pm$ 0.26	31.24
	Router	27.47 $\pm$ 1.13	21.99 $\pm$ 0.22	22.55 $\pm$ 0.43	27.33 $\pm$ 0.45	60.26 $\pm$ 0.33	47.10 $\pm$ 0.37	34.45
	Influence	34.50 $\pm$ 0.98	19.61 $\pm$ 0.35	18.62 $\pm$ 0.31	34.50 $\pm$ 0.22	62.29 $\pm$ 0.29	47.60 $\pm$ 0.25	36.19
	SCAS	41.97 $\pm$ 1.42	89.38 $\pm$ 0.16	69.91 $\pm$ 0.19	42.13 $\pm$ 0.48	68.58 $\pm$ 0.32	49.60 $\pm$ 0.19	60.26

only verified correct answers; for each open-ended training question, we use Llama-3.3-70B-Instruct (Dubey et al., 2024) to generate 10 diverse responses. See Appendix G for more details. We evaluate six instruction-tuned student models: Qwen2.5-0.5B-Instruct, Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct (Yang et al., 2024), Llama-3.2-1B-Instruct, Llama-3.2-3B-Instruct, and Meta-Llama-3-8B-Instruct (Dubey et al., 2024). For mathematical dataset, we use MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), DeepScaleR (Luo et al., 2025a), and OpenR1-Math (Hugging Face, 2025) as both training and evaluation sources. For open-ended instruction tuning, we train on Magpie (Xu et al., 2024) and evaluate on IFEval (Zhou et al., 2023) and LiveBench (White et al., 2024). We report accuracy for mathematical tasks and official benchmark scores for open-ended tasks; when multiple runs are available, we report mean and standard deviation. Further implementation details are provided in Appendix A.

Baselines. We compare SCAS with perplexity (PPL) (Bengio et al., 2003), instruction-following difficulty (IFD) (Li et al., 2023), RSR and its item-level variant (RSR-Item) (Yang et al., 2026b), GRACE (Panigrahi et al., 2025), router-based teacher selection (Router) (Liu et al., 2026),

and influence-function-based data selection (Influence) (Humane et al., 2025). For likelihood-based scores whose selection direction is not fixed a priori, we evaluate both low-score and high-score variants. All methods are trained under the same student model, data budget, and fine-tuning configuration. Further implementation details are provided in Appendix A.

4 Results and Analysis

4.1 Overall Performance

Table 1 reports the main results on three Qwen2.5 student models. SCAS achieves the highest average score for all model sizes, improving over the strongest competing average by 3.19 points on Qwen2.5-0.5B, 3.31 points on Qwen2.5-3B, and 3.48 points on Qwen2.5-7B. The gains are especially clear on mathematical benchmarks. Compared with the strongest baseline, SCAS improves GSM8K by 7.38 points for Qwen2.5-0.5B, 6.02 points for Qwen2.5-3B, and 2.53 points for Qwen2.5-7B. Similar improvements appear on MATH, where the gains are 3.49, 4.14, and 4.20 points, respectively, and on OpenR1-Math, where the gains are 2.47, 2.30, and 3.20 points. The results also show that teacher- or difficulty-centric criteria are not uniformly reliable across student scales and tasks. GRACE is usually the strongest baseline,

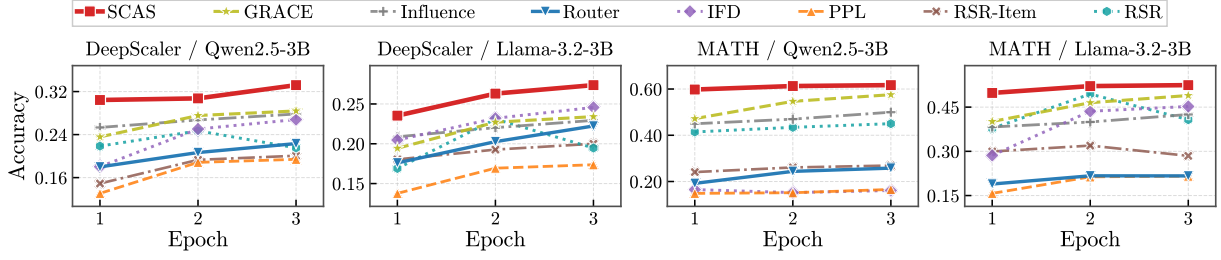


Figure 3: Data Efficiency Across Training Epochs.

but other methods occasionally provide the second-best result, such as Influence on DeepScaler for Qwen2.5-0.5B and RSR-Item on LiveBench for Qwen2.5-7B. In contrast, SCAS obtains the best or tied-best result in every model–metric column, with the only tie occurring on LiveBench for Qwen2.5-3B. SCAS also remains strong on open-ended instruction-following benchmarks, obtaining the best IFEval score for all three model sizes and the best or tied-best LiveBench score across all settings. These results support the central hypothesis of this work: among multiple correct teacher answers, selecting answers that are more learnable for the current student leads to better downstream training than relying only on teacher strength or generic data-difficulty scores.

4.2 Efficiency Analysis

Data Efficiency We next compare how different selection criteria affect learning progress over three training epochs. Figure 3 reports performance after each training epoch on DeepScaleR and MATH for Qwen2.5-3B and Llama-3.2-3B students. SCAS improves faster across datasets and student families, reaching higher accuracy after the first epoch and maintaining the lead through the final checkpoint. This pattern suggests that selecting lower-learning-cost answers helps the student absorb supervision more efficiently at each stage of fine-tuning. In contrast, difficulty-centric baselines improve more slowly or behave inconsistently across epochs, especially on MATH, where methods such as PPL and RSR-Item remain below the best-performing methods. These results show that student-centric answer selection improves data efficiency by converting the same training data and optimization schedule into higher student accuracy throughout fine-tuning.

Computing Efficiency We further evaluate the computational efficiency of SCAS. We measure this efficiency by testing the time required for dif-

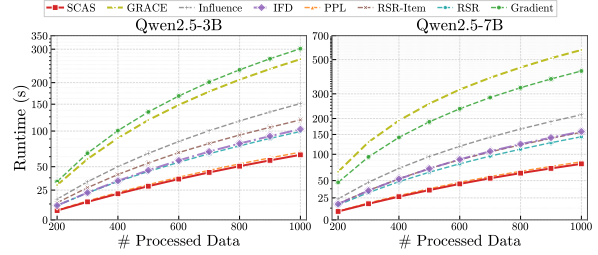


Figure 4: Computing Efficiency for Selecting 1,000 Data Points.

ferent methods to select 1,000 training data points, where each question has 30 teacher-generated candidate answers. We conduct this experiment on the MATH dataset with Qwen2.5-3B and Qwen2.5-7B. The results are shown in Figure 4. The experimental configuration follows Appendix A. Figure 4 compares SCAS with the baselines used in our experiments, including PPL, IFD, RSR, RSR-Item, Influence, and GRACE. SCAS is computationally efficient across both model sizes. At 1,000 processed data points, SCAS takes 64.9 seconds on Qwen2.5-3B and 79.9 seconds on Qwen2.5-7B, which is lower than the baselines. This efficiency follows from our forward-only design: after a single forward pass, the score only requires token-level NLL values and lightweight hidden-state aggregation. These results show that the proposed student-centric selection score is practical for big candidate pools, while retaining the training gains observed in the previous sections.

4.3 Number of Groups

The number of groups G controls the granularity of the stratified sampling step in Algorithm 1. Smaller G keeps the lowest-score group $\Pi_1(q)$ more diverse, but may introduce noise by including candidates whose learning costs are only moderately low. Larger G makes $\Pi_1(q)$ more selective, but can also make sampling more sensitive to score noise. We study this effect on DeepScaleR and MATH using two student models, Qwen2.5-3B and Llama-3.2-

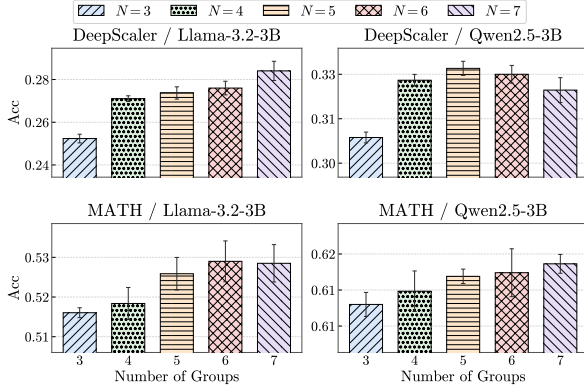


Figure 5: Effect of the Number of Groups G in SCAS.

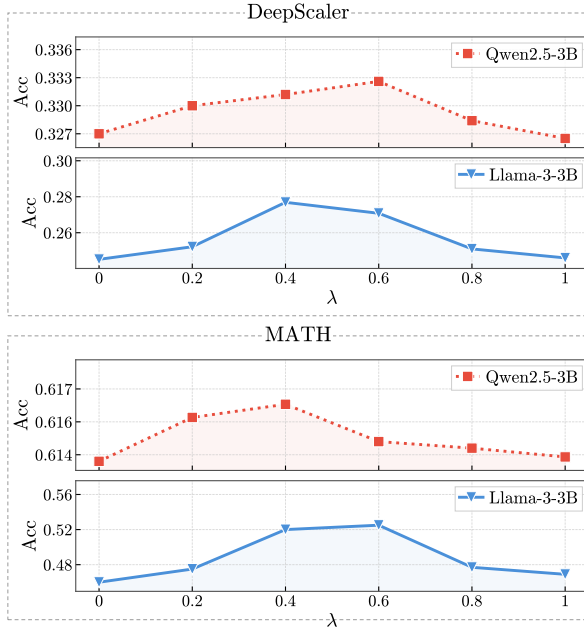


Figure 6: Effect of the Trade-off Parameter λ .

3B, by varying $G \in \{3, 4, 5, 6, 7\}$ under the same experimental setup as Section 4.1. As shown in Figure 5, all settings improve when moving from coarse partitions ($G = 3$ or 4) to a moderately finer grouping. After that, the trend depends on the model and dataset: DeepScaler with Llama-3.2-3B and MATH with Qwen2.5-3B continue to increase as G grows, whereas DeepScaler with Qwen2.5-3B and MATH with Llama-3.2-3B peak and then slightly decline. This suggests a trade-off between reducing coarse-group noise and maintaining a sufficiently robust sampling pool. We use $G = 5$ as a stable default in the main experiments.

4.4 Ablation on λ

We ablate the trade-off parameter λ in Eq. (9) to examine how the AA and AQ/QA terms contribute to student-centric answer selection. The two end-

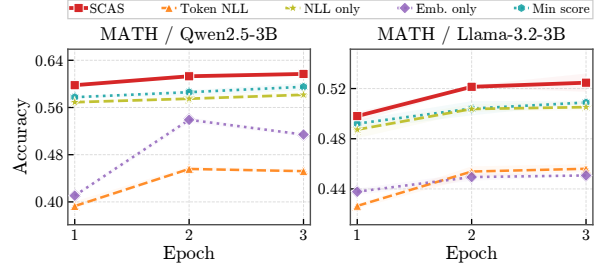


Figure 7: Ablation of the Learning-cost Score.

points correspond to using only one component of the score: the left endpoint uses the AA term alone, while the right endpoint uses only the AQ/QA term. We vary λ over $\{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ and evaluate on DeepScaler and MATH with Qwen2.5-3B and Llama-3.2-3B. As shown in Figure 6, intermediate trade-offs generally lead to better performance than either endpoint. This trend is most pronounced for Llama-3.2-3B, whose accuracy peaks around 0.4 on DeepScaler and around 0.6 on MATH. Qwen2.5-3B is less sensitive to this hyperparameter, but its best results also occur in the middle range, around 0.4–0.6. These patterns suggest that relying on either answer-internal interactions or answer-conditioning interactions alone is suboptimal. The ablation supports our blockwise formulation of the learning-cost proxy: the AA and AQ/QA terms provide complementary signals, and combining them yields a more effective criterion for selecting learnable answers. We therefore use an intermediate trade-off in the main experiments.

4.5 Ablation on the Learning Cost

We ablate the learning-cost score on MATH with Qwen2.5-3B-Instruct and Llama-3.2-3B-Instruct, training each variant for three epochs. SCAS uses the product of sequence-level NLL and the embedding-based score. We compare it with four controlled variants: Token NLL keeps the product form but replaces sequence-level NLL with token-level NLL; NLL only uses the sequence-level NLL term from the SCAS score; Emb. only uses the embedding-based term from the SCAS score; and Min score uses the full score but removes grouping, always selecting the single lowest-score answer for each question. Detailed definitions of these variants are provided in Appendix A.5. The results are shown in Figure 7. SCAS performs best for both student models throughout training. The NLL-only and embedding-only variants both fall below the full score, showing that the two terms provide

Table 2: Mean and Standard Deviation of Blockwise Scores Across Candidate Answers.

Model	Size	Stat.	DeepScaleR			MATH		
			AA	AQ/QA	QQ	AA	AQ/QA	QQ
Qwen-2.5	0.5B	Mean	0.56	0.95	4.18	1.21	1.27	4.02
		Std.	1.4465	0.8323	0.0055	2.9445	1.4772	0.0057
	3B	Mean	0.91	1.04	4.62	2.60	1.59	4.36
		Std.	2.7117	1.1836	0.0077	7.1118	2.3813	0.0077
	7B	Mean	2.59	2.01	7.57	9.96	3.62	7.22
		Std.	9.0950	2.8131	0.0067	29.1193	6.4291	0.0068
Llama-3	1B	Mean	1.32	2.93	10.31	3.11	5.73	10.47
		Std.	3.7038	4.4042	0.0073	9.0740	11.3458	0.0077
	3B	Mean	1.03	2.76	10.88	2.20	5.13	11.02
		Std.	2.7242	3.7724	0.0049	6.1921	9.5294	0.0051
	8B	Mean	1.06	3.41	14.07	2.30	6.08	14.28
		Std.	2.5867	4.2268	0.0086	6.1960	10.5918	0.0088

complementary signals for answer-level learnability. Token NLL is consistently weaker, suggesting that local token difficulty does not capture the answer-level signal needed for selection. Min score is competitive with NLL only but remains below SCAS, suggesting that always choosing the single lowest-score answer can be affected by score noise; this highlights the role of grouping in making low-cost answer selection more robust.

4.6 Mean and Std of AA, AQ/QA, and QQ

The proxy in Eq. (9) keeps the answer-involving AA and AQ/QA blocks while omitting the conditioning-only QQ block. We validate this design choice by measuring how much each block varies across candidate answers for the same question. For each of DeepScaleR and MATH, we randomly sample 2,000 examples from the training set. On these examples, we compute the mean and standard deviation of AA, AQ/QA, and QQ using six student models: Qwen2.5-0.5B, Qwen2.5-3B, Qwen2.5-7B, Llama-3.2-1B, Llama-3.2-3B, and Llama-3-8B. Table 2 shows that QQ is nearly constant across candidates: its standard deviation is at most about 0.01 in all settings. In contrast, the answer-involving blocks exhibit much larger cross-candidate variation, with AA standard deviations ranging from 1.45 to 29.12 and AQ/QA standard deviations ranging from 0.83 to 11.35. Thus, QQ behaves mostly as a question-level offset, whereas AA and AQ/QA provide the main discriminative signals during data selection.

5 Rank Correlation with Gradient Scores

We evaluate whether SCAS preserves the ranking induced by exact gradient values. For each question, we rank the candidate answers using each method and compute Spearman correlation with

Table 3: Mean Spearman Rank Correlation with Gradient Values.

Model	Method	DS	GSM8K	MATH	OR1	Mag.	Avg.
Qwen2.5-3B	GRACE	0.781	0.853	0.829	0.775	0.348	0.717
	Influence	0.604	0.676	0.662	0.356	0.261	0.512
	IFD	0.025	0.053	-0.074	0.101	0.159	0.053
	Router	0.247	-0.582	-0.288	-0.393	-0.304	-0.264
	RSR	-0.795	-0.836	-0.811	-0.796	0.221	-0.603
	RSR-Item	-0.802	-0.767	-0.791	-0.726	-0.074	-0.632
	PPL	-0.771	-0.842	-0.809	-0.791	-0.344	-0.712
	Ours	0.788	0.883	0.875	0.795	0.572	0.783
	GRACE	0.796	0.891	0.849	0.804	0.628	0.793
	Influence	0.183	0.820	0.800	-0.089	0.515	0.446
Qwen2.5-7B	IFD	-0.360	-0.303	-0.483	-0.228	-0.374	-0.349
	Router	0.285	-0.582	-0.266	-0.399	0.241	-0.144
	RSR	-0.837	-0.636	-0.816	-0.817	0.338	-0.553
	RSR-Item	-0.856	-0.722	-0.807	-0.787	0.040	-0.626
	PPL	-0.856	-0.897	-0.871	-0.844	-0.724	-0.838
	Ours	0.861	0.904	0.883	0.847	0.726	0.844

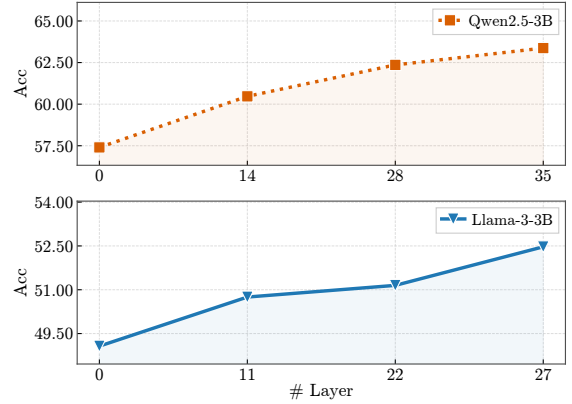


Figure 8: Effect of the Representation Layer on Math.

the ranking obtained from true gradient values; teacher-selection baselines follow the corresponding teacher-model ranking. We average over 1,000 questions from each of DeepScaleR, GSM8K, MATH, OpenR1-Math, and Magpie. Table 3 reports mean correlations on five datasets, where DS, OR1, and Mag. denote DeepScaleR, OpenR1-Math, and Magpie, respectively. SCAS achieves the best average correlation for both Qwen2.5-3B and Qwen2.5-7B, reaching 0.783 and 0.844, and is best in every dataset column. These results indicate that SCAS better preserves the gradient-value preference order while remaining forward-only.

6 Layer Analysis

In our main experiments, SCAS uses the final-layer hidden states of the student model to compute the answer-selection score. This follows our forward-only proxy: since token probabilities are computed from final-layer representations through the language modeling head, these representations are

most directly connected to the NLL-based difficulty terms in our score. To examine the effect of layer choice, we conduct a layer analysis on the Math setting with Llama-3.2-3B and Qwen2.5-3B. For each model, we select several approximately evenly spaced layers from shallow to deep, including the final layer, while keeping all other components unchanged. The results are shown in Figure 9. Performance consistently improves with deeper layers for both models, with the final layer achieving the highest average accuracy. This supports our use of final-layer hidden states, suggesting that representations closest to the language modeling head better capture the student-centric signal needed for answer selection.

7 Related Work

LLM-Generated Synthetic Data and Distillation. LLM-generated supervision has become central to modern LLM training, from synthetic responses (Wang et al., 2022; Xu et al., 2026a, 2024) to reasoning traces, agent trajectories, and tool-use demonstrations (Li et al., 2025; Xu et al., 2025c; Wang et al., 2026b). These signals are widely used for supervised fine-tuning, self-training, and distillation (Ko et al., 2025; Hu et al., 2025a; Yang et al., 2026a). Many pipelines therefore rely on the strongest available model as the teacher (Zhang et al., 2025b; Wang et al., 2025a; Zhang et al., 2025a), implicitly treating teacher task performance as a proxy for data quality (Xu et al., 2025a; Lei et al., 2026; He et al., 2026). While recent work studies better synthetic data generation, knowledge transfer (Kessler et al., 2025; Wang et al., 2026a; Shing et al., 2025), and student-aware teacher selection (Panigrahi et al., 2025), we ask a finer-grained question: given multiple correct teacher-generated answers, which answer is most suitable for the current student?

Data Selection for LLM Training. Our work also builds on data selection for model training. Classical methods select or reweight examples by quality, diversity, or estimated utility (Yoon et al., 2019; Sener and Savarese, 2017), while curriculum and self-paced learning emphasize the role of example difficulty in optimization (Bengio et al., 2009; Kumar et al., 2010; Hu et al., 2024b). Gradient- and uncertainty-based criteria further estimate instance usefulness (Gal et al., 2017; Koh and Liang, 2017), and recent LLM studies adapt these ideas through synthetic data filtering, response ranking,

adaptive sampling (Hu et al., 2023b; Xia et al., 2024; Hu et al., 2023a, 2024c,a), and student-aware teacher or trajectory selection (Yang et al., 2026b). Most of this work selects questions, examples, trajectories, or datasets (Zhang et al., 2023, 2025c; Liu et al., 2025); our setting instead selects among multiple verified correct answers to the same question. We model this answer-level choice through student-centric learning cost and instantiate it with a forward-only proxy in SCAS.

8 Conclusion

This work shows that teacher competence is not always a reliable proxy for teaching quality in teacher-generated LLM supervision. Even among correct answers, the strongest teacher may not provide the best supervision for a given student. We propose SCAS, a forward-only proxy for student-centric learning cost that selects verified teacher-generated answers matched to the current student. Across 30 teacher models, 6 student base models, and 6 tasks, SCAS consistently outperforms teacher-centric and difficulty-based selection, highlighting the importance of student-matched supervision over teacher strength alone.

9 Limitations

This work has three main limitations. First, our experiments primarily focus on 3B- and 7B-scale instruction-tuned students, and the relationship between teacher strength and student-centric supervision may differ for substantially larger students, stronger base models, or architectures with different training dynamics. Second, although our teacher pool includes 30 models, it is not exhaustive, and a larger or more diverse pool may affect our conclusions.

10 Acknowledgement

This work is partially supported by the Air Force Office of Scientific Research (AFOSR) under grant FA9550-23-1-0208, and the National Science Foundation (NSF) under grants IIS 2229876. This work is supported in part by funds provided by the National Science Foundation, by the Department of Homeland Security, and by IBM. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or its federal agency and industry partners.

References

- Marah Abdin, Jyoti Aneja, Harkirat Singh Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio Cesar Teodoro Mendes, Anh Hong Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. [Phi-4 technical report](#). *ArXiv*, abs/2412.08905.
- Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stańczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2023. [On-policy distillation of language models: Learning from self-generated mistakes](#). In *International Conference on Learning Representations*.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. [A neural probabilistic language model](#). In *Journal of machine learning research*.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. [Curriculum learning](#). In *International Conference on Machine Learning*.
- Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas R. Joglekar, Jan Leike, Ilya Sutskever, Jeff Wu, and OpenAI. 2023. [Weak-to-strong generalization: Eliciting strong capabilities with weak supervision](#). In *International Conference on Machine Learning*.
- Yihan Chen, Benfeng Xu, Xiaorui Wang, Yongdong Zhang, and Zhendong Mao. 2025a. [Training llm-based agents with synthetic self-reflected trajectories and partial masking](#). *ArXiv*, abs/2505.20023.
- Zhengyu Chen, Siqi Wang, Teng Xiao, Yudong Wang, Shiqi Chen, Xunliang Cai, Junxian He, and Jingang Wang. 2025b. [Revisiting scaling laws for language models: The role of data quality and training strategies](#). In *Annual Meeting of the Association for Computational Linguistics*.
- Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *ArXiv*, abs/2110.14168.
- Aladin Djuhera, Swanand Kadhe, Syed Zawad, Farhan Ahmed, Heiko Ludwig, and Holger Boche. 2025. [Fixing it in post: A comparative study of llm post-training data quality and model performance](#). *ArXiv*, abs/2506.06522.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 510 others. 2024. [The llama 3 herd of models](#).
- Luyang Fang, Xiaowei Yu, Jiazhang Cai, Yongkai Chen, Shushan Wu, Zheng Liu, Zhenyuan Yang, Haoran Lu, Xilin Gong, Yufang Liu, Terry Ma, Wei Ruan, Ali Abbasi, Jing Zhang, Tao Wang, Ehsan Latif, Wei Liu, Wei Zhang, Soheil Kolouri, and 5 others. 2025. [Knowledge distillation and dataset distillation of large language models: emerging trends, challenges, and future directions](#). *Artificial Intelligence Review*, 59.
- Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. [Deep bayesian active learning with image data](#). *ArXiv*, abs/1703.02910.
- Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. [Minillm: On-policy distillation of large language models](#).
- Chaoqun He, Yingfa Chen, Chaojun Xiao, Xu Han, and Lijie Wen. 2026. [Student-in-the-loop chain-of-thought distillation via generation-time selection](#).
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Xiaodong Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the math dataset](#). *ArXiv*, abs/2103.03874.
- Lijie Hu, Tianhao Huang, Huanyi Xie, Chenyang Ren, Zhengyu Hu, Lu Yu, and Di Wang. 2024a. [Semi-supervised concept bottleneck models](#). 2025 *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2110–2119.
- Lijie Hu, Chenyang Ren, Zhengyu Hu, Cheng-Long Wang, and Di Wang. 2024b. [Editable concept bottleneck models](#). *ArXiv*, abs/2405.15476.
- Zhengyu Hu, Yichuan Li, Zhengyu Chen, Jingang Wang, Han Liu, Kyumin Lee, and Kaize Ding. 2024c. [Let’s ask gnn: Empowering large language model for graph in-context learning](#). In *Conference on Empirical Methods in Natural Language Processing*.
- Zhengyu Hu, Jianxun Lian, Zheyuan Xiao, Seraphina Zhang, Tianfu Wang, Nicholas Jing Yuan, Xing Xie, and Hui Xiong. 2025a. [Unveiling the learning mind of language models: A cognitive framework and empirical study](#). *ArXiv*, abs/2506.13464.
- Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Jingang Wang, Zhenyu Chen, Jieyu Zhao, and Hui Xiong. 2024d. [Explaining length bias in llm-based preference evaluations](#). In *Conference on Empirical Methods in Natural Language Processing*.
- Zhengyu Hu, Zheyuan Xiao, MM Xiong, Yuxuan Lei, Tianfu Wang, Jianxun Lian, Kaize Ding, Ziang Xiao, Nicholas Jing Yuan, and Xing Xie. 2025b. [Population-aligned persona generation for llm-based social simulation](#). *ArXiv*, abs/2509.10127.
- Zhengyu Hu, Jieyu Zhang, Hong Wang, Siwei Liu, and Shangsong Liang. 2023a. [Leveraging relational graph neural network for transductive model ensemble](#). *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

- Zhengyu Hu, Jieyu Zhang, Yue Yu, Yuchen Zhuang, and Hui Xiong. 2023b. [How many validation labels do you need? exploring the design space of label-efficient model ranking](#). *ArXiv*, abs/2312.01619.
- Hugging Face. 2025. [Open rl: A fully open reproduction of deepseek-rl](#).
- Prateek Humane, Paolo Cudrano, Daniel Z. Kaplan, Matteo Matteucci, Supriyo Chakraborty, and Irina Rish. 2025. [Influence functions for efficient data selection in reasoning](#). *ArXiv*, abs/2510.06108.
- Samuel Kessler, Menglin Xia, Daniel Madrigal Diaz, Dongge Han, Helia Heshemi, S. Rajmohan, Victor Ruehle, and Jordan T. Ash. 2025. [Towards active synthetic data generation for finetuning language models](#). *ArXiv*, abs/2512.00884.
- Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and SeYoung Yun. 2025. [Distillm-2: A contrastive approach boosts the distillation of llms](#). *ArXiv*, abs/2503.07067.
- Pang Wei Koh and Percy Liang. 2017. [Understanding black-box predictions via influence functions](#). In *International Conference on Machine Learning*.
- M. Pawan Kumar, Ben Packer, and Daphne Koller. 2010. [Self-paced learning for latent variable models](#). In *Neural Information Processing Systems*.
- Yuxuan Lei, Tian Wang, Jianxun Lian, Zhengyu Hu, Defu Lian, and Xing Xie. 2026. [Humanllm: Towards personalized understanding and simulation of human nature](#). *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*.
- Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023. [From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning](#). In *North American Chapter of the Association for Computational Linguistics*.
- Yixia Li, Hongru Wang, Jiahao Qiu, Zhenfei Yin, Dongdong Zhang, Cheng Qian, Zeping Li, Po shan Cathy Ma, Guanhua Chen, Heng Ji, and Mengdi Wang. 2025. [From word to world: Can large language models be implicit text-based world models?](#) *ArXiv*, abs/2512.18832.
- Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong Wang, Yuxian Wang, Wu Ning, Yutai Hou, Bin Wang, Chuhan Wu, Xinzhi Wang, Yong Liu, Yasheng Wang, and 8 others. 2024. [Toolace: Winning the points of llm function calling](#). *ArXiv*, abs/2409.00920.
- Yue Liu, Hongcheng Gao, Shengfang Zhai, Jun Xia, Tianyi Wu, Zhiwei Xue, Yulin Chen, Kenji Kawaguchi, Jiaheng Zhang, and Bryan Hooi. 2025. [Guardreasoner: Towards reasoning-based llm safeguards](#). *ArXiv*, abs/2501.18492.
- Yuhang Liu, Fangyuan Xu, Vishakh Padmakumar, Daphne Ippolito, and Eunsol Choi. 2026. [No single best model for diversity: Learning a router for sample diversity](#).
- Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025a. [Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl](#). Notion Blog.
- Ne Luo, Aryo Pradipta Gema, Xuanli He, Emile van Krieken, Pietro Lesci, and Pasquale Minervini. 2025b. [Self-training large language models for tool-use without demonstrations](#). In *North American Chapter of the Association for Computational Linguistics*.
- Mihai Nadăș, Laura Dioșan, and Andreea Tomescu. 2025. [Synthetic data generation using large language models: Advances in text and code](#). *IEEE Access*, 13:134615–134633.
- Abhishek Panigrahi, Bing Liu, Sadhika Malladi, Sham M. Kakade, and Surbhi Goel. 2025. [In good graces: Principled teacher selection for knowledge distillation](#). *ArXiv*, abs/2511.02833.
- Karl Pearson. 1895. [Vii. note on regression and inheritance in the case of two parents](#). *Proceedings of the Royal Society of London*, 58:240 – 242.
- Ozan Sener and Silvio Savarese. 2017. [Active learning for convolutional neural networks: A core-set approach](#). *arXiv: Machine Learning*.
- Zhanming Shen, Hao Chen, Yu-Chien Tang, Shaolin Zhu, Wen song Ye, Xiaomeng Hu, Haobo Wang, Gang Chen, and Junbo Zhao. 2025. [Cycle-instruct: Fully seed-free instruction tuning via dual self-training and cycle consistency](#). *ArXiv*, abs/2508.16100.
- Makoto Shing, Kou Misaki, Han Bao, Sho Yokoi, and Takuya Akiba. 2025. [Taid: Temporally adaptive interpolated distillation for efficient knowledge transfer in language models](#). *ArXiv*, abs/2501.16937.
- Tianfu Wang, Max Xiong, Jianxun Lian, Hongyuan Zhu, Zhengyu Hu, Yuxuan Lei, Linxiao Gong, Xiaofan Li, Pei-Shan Tsai, Nicholas Jing Yuan, and Qi Zhang. 2026a. [Socialcoach: Personalized social skill learning with rl-based agentic tutoring and practice](#). *ArXiv*, abs/2606.04155.
- Tianfu Wang, Yi Zhan, Jianxun Lian, Zhengyu Hu, Nicholas Jing Yuan, Qi Zhang, Xing Xie, and Hui Xiong. 2025a. [Llm-powered multi-agent framework for goal-oriented learning in intelligent tutoring system](#). *Companion Proceedings of the ACM on Web Conference 2025*.
- Yike Wang, Huaisheng Zhu, Zhengyu Hu, Yige Yuan, Zhengyu Chen, Shakti Senthil, Hanna Hajishirzi, Yulia Tsvetkov, Pradeep Dasigi, and Teng Xiao. 2026b. [Rethinking the evaluation of harness evolution for agents](#).

- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. [Self-instruct: Aligning language models with self-generated instructions](#). In *Annual Meeting of the Association for Computational Linguistics*.
- Zhaoyang Wang, Yiming Liang, Xuchao Zhang, Qianhui Wu, Siwei Han, Anson Bastos, Rujia Wang, Chetan Bansal, Baolin Peng, Jianfeng Gao, Saravan Rajmohan, and Huaxiu Yao. 2025b. [Adapting web agents with synthetic supervision](#). *ArXiv*, abs/2511.06101.
- Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddhartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2024. [Livebench: A challenging, contamination-limited llm benchmark](#). In *International Conference on Learning Representations*.
- Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. [Less: Selecting influential data for targeted instruction tuning](#). *ArXiv*, abs/2402.04333.
- Ruiyao Xu, Mihir Parmar, Tiankai Yang, Zhengyu Hu, Yue Zhao, and Kaize Ding. 2026a. [Coact: Coactive llm preference learning with human-ai synergy](#). *ArXiv*, abs/2604.17501.
- Ruiyao Xu, Noelle I. Samia, and Hanbing Liu. 2026b. [Ds2-instruct: Domain-specific data synthesis for large language models instruction tuning](#). In *Conference of the European Chapter of the Association for Computational Linguistics*.
- Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024. [Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing](#). *ArXiv*, abs/2406.08464.
- Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, and Radha Poovendran. 2025a. [Stronger models are not always stronger teachers for instruction tuning](#). In *North American Chapter of the Association for Computational Linguistics*.
- Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. 2025b. [Kodcode: A diverse, challenging, and verifiable synthetic dataset for coding](#). *ArXiv*, abs/2503.02951.
- Zhangchen Xu, Adriana Meza Soria, Shawn Tan, Anurag Roy, Ashish Agrawal, Radha Poovendran, and Rameswar Panda. 2025c. [Toucan: Synthesizing 1.5m tool-agentic data from real-world mcp environments](#). *ArXiv*, abs/2510.01179.
- Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024. [Qwen2.5 technical report](#). *ArXiv*, abs/2412.15115.
- Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. 2026a. [Learning beyond teacher: Generalized on-policy distillation with reward extrapolation](#). *ArXiv*, abs/2602.12125.
- Yuming Yang, Ming Long Lai, Wanxu Zhao, Xiaoran Fan, Zhiheng Xi, Mingqi Wu, Chiyue Huang, Jun Zhao, Haijun Lv, Jian Tong, Yunhua Zhou, Yicheng Zou, Qipeng Guo, Tao Gui, Qi Zhang, and Xuanjing Huang. 2026b. [Which reasoning trajectories teach students to reason better? a simple metric of informative alignment](#). *ArXiv*, abs/2601.14249.
- Jinsung Yoon, Sercan Ö. Arik, and Tomas Pfister. 2019. [Data valuation using reinforcement learning](#). *ArXiv*, abs/1909.11671.
- Ping Yu, Jack Lanchantin, Tianlu Wang, Weizhe Yuan, Olga Golovneva, Iliia Kulikov, Sainbayar Sukhbaatar, Jason E. Weston, and Jing Xu. 2025. [Cot-self-instruct: Building high-quality synthetic prompts for reasoning and non-reasoning tasks](#). *ArXiv*, abs/2507.23751.
- Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason E. Weston. 2024. [Self-rewarding language models](#). *ArXiv*, abs/2401.10020.
- Dylan Zhang, Qirun Dai, and Hao Peng. 2025a. [The best instruction-tuning data are those that fit](#). *ArXiv*, abs/2502.04194.
- Hengyuan Zhang, Shiping Yang, Xiao Liang, Chenming Shang, Yuxuan Jiang, Chaofan Tao, Jing Xiong, Hayden Kwok-Hay So, Ruobing Xie, Angel X. Chang, and Ngai Wong. 2025b. [Find your optimal teacher: Personalized data synthesis via router-guided multi-teacher distillation](#). *ArXiv*, abs/2510.10925.
- Jieyu Zhang, Bohan Wang, Zhengyu Hu, Pang Wei Koh, and Alexander J. Ratner. 2023. [On the trade-off of intra-/inter-class diversity for supervised pre-training](#). *ArXiv*, abs/2305.12224.
- Yunxiao Zhang, Guanming Xiong, Haochen Li, and Wen Zhao. 2025c. [Edge: Efficient data selection for llm agents via guideline effectiveness](#). In *International Joint Conference on Artificial Intelligence*.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-following evaluation for large language models](#). *ArXiv*, abs/2311.07911.

A Implementation Details

A.1 Experimental Setup

All experiments were conducted on a high-performance computing server equipped with an AMD EPYC 7V12 64-Core Processor, 512 GB of system memory, 96 logical CPUs, and eight NVIDIA A100 GPUs with 40 GB of memory each. Unless otherwise specified, abbreviated student-model names refer to the following Hugging Face checkpoints: Qwen2.5-0.5B-Instruct³, Qwen2.5-3B-Instruct⁴, Qwen2.5-7B-Instruct⁵, Llama-3.2-1B-Instruct⁶, Llama-3.2-3B-Instruct⁷, and Meta-Llama-3-8B-Instruct⁸. All student models are trained using full-parameter supervised fine-tuning under the same optimization schedule. For SCAS, we use three selection–training rounds and recompute the candidate scores using the updated student between consecutive rounds. Because SCAS samples uniformly from the lowest-score candidate group, we repeat the experiments using random seeds 0, 1, and 2 and report the mean and standard deviation across runs. Table 4 summarizes the training and teacher generation settings. Table 5 reports the training-set sizes, evaluation sets and versions, and candidate-pool statistics. For mathematical datasets, the candidate count is the mean number of answers retained per question after correctness verification. For Magpie, it denotes the fixed number of responses generated per instruction. The common-correct subsets in Table 5 are used only for the controlled experiment in Figure 1, where we retain the intersection of questions answered correctly by all teachers. In the main experiments with the 30-teacher pool, we retain the full set of training questions and discard only incorrect candidate answers within each question. For the baselines, we follow prior settings and released implementations. Specifically, for perplexity (PPL) (Bengio et al., 2003) and instruction-following difficulty (IFD) (Li et al., 2023), we follow the evaluation setup of Xu et al. (2025a); for RSR and RSR-Item, we follow

Yang et al. (2026b); for GRACE, we follow the released implementation of Panigrahi et al. (2025); for router-based teacher selection, we follow Liu et al. (2026); and for influence-function-based selection, we follow Humane et al. (2025). For SCAS scoring, we use the final-layer hidden states of the current student to compute the representation aggregates μ_A and μ_Q . The final layer achieves the best answer-selection performance in our layer ablation and is closest to the LM-head residuals motivating our forward proxy. The complete layer-ablation results are reported in Section E. Our code is publicly available⁹.

A.2 Teacher Model Pool for Mathematical Tasks

For the mathematical tasks, namely MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), DeepScaleR (Luo et al., 2025a), and OpenR1-Math (Hugging Face, 2025), we collect candidate answers from a pool of 30 teacher models. The same teacher pool is used to generate candidate answers for all mathematical datasets, and all answer-selection methods operate on the same verified candidate pool after correctness filtering. The complete list of teacher models is shown in Table 6.

A.3 Candidate Answer Verification

In this work, a candidate answer is considered *verified* if it passes an automatic correctness check appropriate to the target task. The purpose of verification is to ensure that answer selection is performed among valid supervision signals rather than confounding the selection criterion with answer correctness. That is, our method does not attempt to choose between correct and incorrect answers; instead, it ranks and samples among candidate answers that have already been verified as acceptable for training.

For mathematical reasoning tasks, including MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), DeepScaleR (Luo et al., 2025a), and OpenR1-Math (Hugging Face, 2025), candidate answers are generated by the 30-model teacher pool in Appendix A.2. We verify them by comparing the predicted final answer with the reference answer and filter out candidates that are judged incorrect. Only the remaining correct candidates are retained in the candidate pool $\mathcal{A}(q)$ for each question q .

³<https://huggingface.co/Qwen/Qwen2.5-0.5B-Instruct>

⁴<https://huggingface.co/Qwen/Qwen2.5-3B-Instruct>

⁵<https://huggingface.co/Qwen/Qwen2.5-7B-Instruct>

⁶<https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct>

⁷<https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct>

⁸<https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct>

⁹<https://github.com/ppsmk388/Student-Centric-Answer-Selection>

Table 4: Training and teacher-generation settings used in the main experiments.

Setting	Value
Fine-tuning type	Full-parameter supervised fine-tuning
Training framework	LLaMA-Factory with Transformers and DeepSpeed ZeRO-3
Optimizer	AdamW using the LLaMA-Factory/Transformers defaults
Learning rate	1.0×10^{-5}
Learning-rate scheduler	Cosine decay with a warmup ratio of 0.05
Training epochs	3
SCAS selection-training rounds	3
Maximum sequence length	4,096 tokens
Random seeds	0, 1, 2
Effective batch size	64 for Qwen2.5-0.5B and Llama-3.2-1B; 48 for Qwen2.5-3B and Llama-3.2-3B; 16 for Qwen2.5-7B and Llama-3-8B
Number of candidate groups	$G = 5$
Math teacher decoding	temperature= 0.7 and max_tokens= 4096
Magpie teacher decoding	temperature= 0.7 and top_p= 0.9; 10 responses generated per instruction

Table 5: Dataset sizes, evaluation sets, and candidate-pool statistics. The last column reports the common-correct subsets used only in the controlled teacher-student comparison in Figure 1.

Dataset	Train	Evaluation set/version	Candidates/item	Fig. 1 subset
GSM8K	7,473	Official test split (1,319)	26.60	6,739 / 7,473
MATH	7,500	Official test split (5,000)	25.41	4,951 / 7,500
DeepScaleR	39,315	Held-out evaluation set (1,000)	16.85	11,258 / 39,315
OpenR1-Math	50,000	Held-out evaluation set (1,000)	18.17	–
Magpie	80,000	IFEval official set (541); LiveBench commit 3839eb0 (1,000)	10 generated	–

This filtering step ensures that all answer-selection methods operate on the same set of verified correct mathematical solutions.

For the open-ended instruction-following task Magpie (Xu et al., 2024), exact correctness verification is not well defined because there is no single reference answer. Therefore, instead of applying answer-level correctness filtering, we generate 10 candidate responses for each instruction using the same teacher model, Llama-3.3-70B-Instruct. For generating diverse responses from LLMs, we follow the configuration in Yuan et al. (2024), setting the sampling temperature to $T = 0.7$ and nucleus sampling probability to $p = 0.9$. Since all Magpie candidates are generated by the same model under the same decoding configuration, we assume that their overall response quality is comparable and that they have a similar degree of reliability for open-ended supervision. Under this setting, candidate selection mainly reflects student-centric learnability rather than differences in teacher identity or answer correctness.

A.4 Introductory Teacher-Student Comparison

Figure 1 visualizes the relation between teacher task accuracy and downstream student accu-

racy in a controlled setting. We use nine selected teacher models: gpt-oss-120b, olmo-3.1-32b-instruct, qwen3.5-27b, qwen2.5-72b, llama-3.3-70b, gemma-4-31b-it, phi-4-reasoning, gpt-5-chat_2025-10-03, and qwen3-32b. We evaluate the teachers and students on three mathematical datasets: DeepScaleR, GSM8K, and MATH. The student models in this visualization are Qwen2.5-0.5B-Instruct, Qwen2.5-3B-Instruct, and Qwen2.5-7B-Instruct.

For the x-axis, each teacher is evaluated directly on the corresponding test set, and the resulting accuracy is used as its teacher accuracy. For the y-axis, each teacher first generates answers for the student training set. To control for correctness and make the teacher comparison depend on the form of supervision rather than on different error rates in the training data, we construct the largest common correct subset of training questions: an instance is retained only if all nine selected teachers generate a correct answer for it. Each teacher then contributes its own generated answer on this same retained subset, and a Qwen2.5 student is fine-tuned on the resulting teacher-specific synthetic training set. The plotted student accuracy is the student’s test accuracy after this fine-tuning step. This protocol ensures that all teachers are compared on the same

Table 6: **Teacher Model Pool for Mathematical Candidate Generation.** We use 30 open- and closed-source teacher models to generate candidate answers for mathematical tasks.

Model	Model	Model
gemma-3-12b-it	gemma-3-27b-it	gemma-4-31b-it
gpt-4.1_2025-04-14	gpt-4o_2024-08-06	gpt-4o_2024-11-20
gpt-5-chat_2025-08-07	gpt-5-chat_2025-10-03	gpt-5-nano_2025-08-07
gpt-5.1-chat_2025-11-13	gpt-5.2-chat_2025-12-11	gpt-5_2025-08-07
gpt-oss-120b	gpt-oss-20b	grok-3_1
grok-4-fast-non-reasoning_1	grok-4_1	llama-3.3-70b-instruct
Llama-3.1-70B-Instruct	mistral-3-14b-instruct	nvidia-llama3.3-nemotron-49b-v1.5
olmo-3.1-32b-instruct	phi-4-reasoning-plus	qwen2.5-14b-instruct
qwen2.5-32b-instruct	qwen2.5-72b-instruct	qwen2.5-coder-32b
qwen2.5-math-72b	qwen3-32b	qwen3.5-27b

questions and that every plotted training answer is correct.

For each dataset–student subplot, the displayed correlation coefficient is Pearson’s (Pearson, 1895) r computed over the nine teachers. Let x_i denote the test accuracy of teacher i , and let y_i denote the test accuracy of the student fine-tuned on teacher i ’s synthetic answers. We compute

$$r = \frac{\sum_{i=1}^9 (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^9 (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^9 (y_i - \bar{y})^2}}, \quad (12)$$

where $\bar{x}$ and $\bar{y}$ are the means over the nine teachers.

A.5 Learning-Cost Ablation Variants

We define the variants used in the learning-cost ablation in Section 4.5. All variants use the same candidate-answer pool, correctness filtering, student models, and training schedule; they differ only in how candidate answers are scored or sampled. Let $E_{AA}(q, a; \theta) = \|\mu_A\|^2$ and $E_{AQ}(q, a; \theta) = \mu_A^\top \mu_Q$. The full SCAS score used before grouping is

$$S_{\text{full}}(q, a; \theta) = (1 - \lambda) d_A^2 E_{AA}(q, a; \theta) + \lambda d_A d_Q E_{AQ}(q, a; \theta). \quad (13)$$

For the token-level variant, let $\ell_i = -\log p_\theta(x_i \mid q, a_{<i})$ for $i \in I_A$ and $\ell_j = -\log p_\theta(x_j \mid x_{<j})$ for $j \in I_Q$. Instead of replacing token-level NLLs by their sequence means d_A and d_Q ,

Token NLL keeps token-level weights inside the interaction terms:

$$\bar{v}_A = \frac{1}{|I_A|} \sum_{i \in I_A} \ell_i u_i, \quad \bar{v}_Q = \frac{1}{|I_Q|} \sum_{j \in I_Q} \ell_j u_j, \quad (14)$$

$$S_{\text{tok}}(q, a; \theta) = (1 - \lambda) \|\bar{v}_A\|^2 + \lambda \bar{v}_A^\top \bar{v}_Q. \quad (15)$$

NLL only removes the embedding-side factors and uses

$$S_{\text{NLL}}(q, a; \theta) = (1 - \lambda) d_A^2 + \lambda d_A d_Q. \quad (16)$$

Emb. only removes the NLL factors and uses

$$S_{\text{emb}}(q, a; \theta) = (1 - \lambda) E_{AA}(q, a; \theta) + \lambda E_{AQ}(q, a; \theta). \quad (17)$$

For Token NLL, NLL only, and Emb. only, we sort candidates by the corresponding score and apply the same grouping-and-sampling procedure as SCAS.

Min score keeps S_{full} but removes grouping:

$$\tilde{a}_{\min}(q) = \arg \min_{a \in \mathcal{A}(q)} S_{\text{full}}(q, a; \theta_r). \quad (18)$$

Thus Min score always selects the single lowest-score candidate answer for each question.

A.6 Notation

Table 7 and Table 8 summarize the main notation used in the paper. Auxiliary symbols used only in appendix proofs are defined locally when introduced.

A.7 Datasets

MATH (Hendrycks et al., 2021). MATH is a benchmark of competition-style mathematical problems covering diverse subjects such as algebra, geometry, number theory, counting and probability, and precalculus. The problems require multi-step symbolic and quantitative reasoning, making the dataset a standard testbed for evaluating whether distilled students can learn from teacher-generated mathematical solutions. In our experiments, MATH is used as one of the mathematical reasoning sources for both answer-pool construction and evaluation.

GSM8K (Cobbe et al., 2021). GSM8K contains grade-school math word problems paired with natural-language solutions and final numerical answers. Compared with MATH, GSM8K emphasizes everyday arithmetic reasoning expressed in textual form, and therefore provides a complementary test of whether selected teacher answers help students follow concise step-by-step problem solving. We use GSM8K as a mathematical reasoning source and evaluate student accuracy by checking the final answer.

Table 7: Summary of Notation (A): Data, Sequences, Student model, and Gradient Decomposition.

Symbol	Description
Part I: Data and training sequences	
$\mathcal{D} = \{q_i\}_{i=1}^N$	Training set of questions, where N is the number of questions.
q	A question or prompt from $\mathcal{D}$.
$\mathcal{A}(q)$	Set of verified correct teacher-generated candidate answers for question q .
a	A candidate answer in $\mathcal{A}(q)$.
$x = [q; a]$	Training-templated sequence formed by concatenating the question and answer.
x_t	Token at position t in x .
$x_{<t}$	Prefix tokens before position t .
$a_{<i}$	Answer-side prefix before answer token position i .
$I = \{1, \dots, x \}$	Set of token positions in x .
I_Q	Token positions corresponding to the question or conditioning part.
I_A	Token positions corresponding to the answer part.
$ q , x , I_Q , I_A $	Sequence or block lengths.
Part II: Student model and gradient decomposition	
$p_\theta(\cdot)$	Autoregressive student language model.
θ	Student model parameters.
θ_r	Student parameters at selection-training round r .
$\mathbb{R}$	Set of real numbers.
h_t	Final-layer hidden state at token position t .
$H_\theta(x) = [h_t]_{t \in I}$	Final-layer hidden-state sequence produced by the student on x .
$W \in \mathbb{R}^{d \times m}$	Generic student weight matrix used in the gradient-decomposition analysis.
d, m	Input and output dimensions of the generic weight matrix W .
$X = [z_t^\top]_{t \in I}$	Token representations feeding into W , where z_t is the representation at position t .
$A = XW$	Pre-activation matrix corresponding to W ; this local matrix A is distinct from the answer set $\mathcal{A}(q)$.
$\mathcal{L}$	Training loss for a sequence.
$G = [g_t^\top]_{t \in I}$	Gradient of $\mathcal{L}$ with respect to A , where g_t is the token-level gradient at position t .
$\nabla_W \mathcal{L}$	Gradient of the loss with respect to W .
$\ \cdot\ _F$	Frobenius norm.
$(\cdot)^\top$	Vector or matrix transpose.
AA	Answer–answer interaction block, involving pairs $i, j \in I_A$.
AQ/QA	Answer–conditioning interaction block, involving one answer position and one question position.
QQ	Conditioning–conditioning interaction block, involving pairs $i, j \in I_Q$.

DeepScaleR (Luo et al., 2025a). DeepScaleR is a reasoning-oriented dataset associated with the DeepScaleR training recipe, which studies scaling reinforcement learning for mathematical reasoning models. The dataset provides challenging mathematical prompts suitable for generating and verifying multiple teacher solutions. We use DeepScaleR as one of the mathematical reasoning sources to test whether SCAS remains effective on harder reasoning instances beyond standard math benchmarks.

OpenR1-Math (Hugging Face, 2025). OpenR1-Math is a large-scale mathematical reasoning dataset released as part of the Open R1 effort. It aggregates and filters math problems for training reasoning models, and is intended to support open reproduction of reasoning-oriented post-training pipelines. In our experiments, OpenR1-Math provides an additional source of verifiable math questions for constructing candidate teacher answers and evaluating the resulting student models.

Magpie (Xu et al., 2024). Magpie is an alignment-data synthesis framework that constructs instruction-following data by prompting aligned language models to produce instructions and responses. Unlike the mathematical datasets above, Magpie covers open-ended instruction-following

scenarios where correctness is less easily captured by exact final-answer matching. We use Magpie as the training source for open-ended instruction tuning.

IFEval (Zhou et al., 2023). IFEval evaluates instruction-following ability using prompts with explicitly checkable constraints, such as formatting requirements, keyword constraints, and structural conditions. Because many of its criteria can be verified automatically, it provides a controlled benchmark for measuring whether a student follows user instructions after fine-tuning. We use IFEval as one of the open-ended evaluation benchmarks.

LiveBench (White et al., 2024). LiveBench is a contamination-limited benchmark designed to evaluate contemporary language models. It covers multiple ability categories and is intended to reduce the risk that benchmark items have appeared in model training data. We use LiveBench as a complementary open-ended evaluation benchmark for testing the generality of student-centric answer selection.

A.8 Baseline Adaptation and Comparison Protocol

All methods operate on the same candidate pool $\mathcal{A}(q)$ for each question q and select one candidate answer per question to construct the SFT data. The

Table 8: **Summary of Notation (B): Forward-only Learning-cost Proxy and Student-centric Answer Sampling.**

Symbol	Description
Part III: Forward-only learning-cost proxy	
$u_t = h_t / \ h_t\ $	Normalized final-layer hidden state at token position t .
μ_A	Mean normalized hidden representation over answer positions, $\mu_A = \frac{1}{ I_A } \sum_{i \in I_A} u_i$.
μ_Q	Mean normalized hidden representation over question or conditioning positions, $\mu_Q = \frac{1}{ I_Q } \sum_{j \in I_Q} u_j$.
d_A	Mean token negative log-likelihood (NLL) over answer positions.
d_Q	Mean token NLL over question or conditioning positions.
$p_\theta(x_i q, a_{<i})$	Student probability assigned to answer token x_i given the question and previous answer tokens.
$p_\theta(x_j x_{<j})$	Student probability assigned to conditioning token x_j given its prefix.
$C_{AA}(q, a; \theta)$	Answer–answer component of the proxy, $d_A^2 \ \mu_A\ ^2$.
$C_{AQ}(q, a; \theta)$	Answer–conditioning component of the proxy, $d_A d_Q (\mu_A^\top \mu_Q)$.
$S(q, a; \theta)$	Student-centric learning-cost score used to rank candidate answers.
$\lambda \in [0, 1]$	Trade-off parameter between C_{AA} and C_{AQ} in $S(q, a; \theta)$.
Part IV: Student-centric answer sampling	
SCAS	Student-Centric Answer Sampling, the proposed answer-selection and training procedure.
R	Total number of selection-training rounds.
r	Selection-training round index.
$M_q = \mathcal{A}(q) $	Number of verified candidate answers for question q .
$a_{(k)}$	The k -th candidate answer after sorting $\mathcal{A}(q)$ by increasing $S(q, a; \theta_r)$.
G	Number of consecutive groups in SCAS; distinct from the gradient matrix G in Table 7.
g	Group index, with $g = 1, \dots, G$.
$\Pi_g(q)$	The g -th group of sorted candidate answers for question q .
$\Pi_1(q)$	Lowest-score group, from which SCAS samples during training.
$\lfloor \cdot \rfloor$	Floor function used to define group boundaries.
$\tilde{a}(q)$	Answer sampled for question q from the lowest-score group.
$\text{Unif}(\Pi_1(q))$	Uniform distribution over candidates in $\Pi_1(q)$.
$\mathcal{D}_{\text{low}}$	Selected training set $\{(q, \tilde{a}(q)) : q \in \mathcal{D}\}$ constructed from the lowest-score groups.
Part V: Complexity analysis	
n_Q, n_A, n	Numbers of question tokens, answer tokens, and total tokens, with $n = n_Q + n_A$.
$F(n)$	Time cost of one forward pass on a length- n sequence.
$B(n)$	Time cost of one backward pass on a length- n sequence.

selected data are then used to train the same student model with the same number of training examples and fine-tuning configuration. We describe below how methods with different native granularities are adapted to this common candidate-answer selection setting.

RSR (Yang et al., 2026b). RSR is applied as a teacher-level selection method. It selects a teacher, and we use that teacher’s verified candidate answer as supervision for each question.

GRACE (Panigrahi et al., 2025). GRACE selects the teacher best matched to the current student. For each question, we use the verified candidate answer generated by the selected teacher.

RSR-Item (Yang et al., 2026b). RSR-Item is the item-level variant of RSR. We apply it directly at the answer level by scoring every candidate and selecting the preferred answer for each question.

PPL (Bengio et al., 2003). We compute perplexity for each question–answer pair and rank the candidate answers accordingly. Because the preferred score direction is not fixed a priori in this setting, we evaluate both low-PPL and high-PPL selection variants.

IFD (Li et al., 2023). We compute instruction-following difficulty for each question–answer pair and use it to rank the candidate answers. As with

PPL, we evaluate both low-score and high-score selection variants.

Router (Liu et al., 2026). We treat the candidate teacher answers as routing options. The router selects one candidate answer for each question, which is then used as student supervision.

Influence (Humane et al., 2025). We estimate the influence of each candidate answer on the student and select the candidate with the highest estimated utility for each question.

B Additional Results on Llama-3.2 Student Models

Table 9 reports the complete results for Llama-3.2-1B-Instruct and Llama-3.2-3B-Instruct under the same experimental protocol as Table 1. SCAS achieves the highest mean on every benchmark and for both model-level averages. Compared with the strongest competing average, SCAS improves performance by 4.18 points for Llama-3.2-1B and 3.72 points for Llama-3.2-3B. These results demonstrate that the effectiveness of student-centric answer selection generalizes beyond the Qwen2.5 model family.

Table 9: Overall Performance of Answer-selection Methods on Llama-3.2 Student Models.

Model	Method	DeepScaleR	GSM8K	MATH	OpenR1-Math	IFEval	LiveBench	Avg.
Llama-3.2-1B	RSR	10.47 $\pm$ 0.45	52.16 $\pm$ 0.85	22.44 $\pm$ 0.26	8.47 $\pm$ 1.17	50.65 $\pm$ 0.44	31.60 $\pm$ 0.27	29.30
	GRACE	12.03 $\pm$ 0.51	49.53 $\pm$ 0.80	21.19 $\pm$ 0.38	9.20 $\pm$ 0.70	47.87 $\pm$ 0.39	31.70 $\pm$ 0.23	28.59
	RSR-Item	10.90 $\pm$ 0.60	39.25 $\pm$ 1.10	25.18 $\pm$ 0.33	9.90 $\pm$ 0.20	52.60 $\pm$ 0.64	30.20 $\pm$ 0.23	28.01
	PPL	8.07 $\pm$ 1.16	8.87 $\pm$ 0.60	11.68 $\pm$ 0.07	8.93 $\pm$ 0.25	52.31 $\pm$ 0.52	29.60 $\pm$ 0.29	19.91
	IFD	11.17 $\pm$ 0.46	18.10 $\pm$ 1.03	21.14 $\pm$ 0.21	7.10 $\pm$ 0.56	50.09 $\pm$ 0.40	30.40 $\pm$ 0.16	23.00
	Router	12.30 $\pm$ 1.47	12.64 $\pm$ 0.32	15.23 $\pm$ 0.28	8.70 $\pm$ 0.35	52.50 $\pm$ 0.88	29.00 $\pm$ 0.60	21.73
	Influence	12.20 $\pm$ 0.36	35.26 $\pm$ 2.66	29.59 $\pm$ 0.15	9.26 $\pm$ 0.97	50.76 $\pm$ 0.72	31.23 $\pm$ 0.39	28.05
	SCAS	14.07 $\pm$ 0.50	55.57 $\pm$ 1.59	32.62 $\pm$ 0.72	11.83 $\pm$ 0.45	53.87 $\pm$ 0.35	32.90 $\pm$ 0.20	33.48
Llama-3.2-3B	RSR	19.70 $\pm$ 0.78	77.16 $\pm$ 0.24	40.35 $\pm$ 0.49	18.20 $\pm$ 0.26	74.86 $\pm$ 0.33	40.50 $\pm$ 0.34	45.13
	GRACE	23.80 $\pm$ 0.52	79.48 $\pm$ 1.21	44.39 $\pm$ 0.43	21.87 $\pm$ 0.51	74.27 $\pm$ 0.59	41.90 $\pm$ 0.54	47.62
	RSR-Item	21.77 $\pm$ 1.24	64.11 $\pm$ 1.48	38.89 $\pm$ 0.57	17.13 $\pm$ 0.72	73.86 $\pm$ 0.62	41.20 $\pm$ 0.31	42.83
	PPL	17.83 $\pm$ 0.51	16.65 $\pm$ 0.87	18.29 $\pm$ 0.08	14.53 $\pm$ 0.31	72.16 $\pm$ 0.59	42.20 $\pm$ 0.53	30.28
	IFD	26.80 $\pm$ 1.15	46.98 $\pm$ 1.13	29.05 $\pm$ 0.51	21.70 $\pm$ 0.87	73.38 $\pm$ 0.51	40.50 $\pm$ 0.26	39.74
	Router	20.53 $\pm$ 0.68	24.39 $\pm$ 0.73	21.10 $\pm$ 0.59	16.57 $\pm$ 0.25	72.64 $\pm$ 0.67	40.60 $\pm$ 0.33	32.64
	Influence	25.03 $\pm$ 0.65	49.84 $\pm$ 1.18	34.03 $\pm$ 0.58	21.54 $\pm$ 0.57	74.43 $\pm$ 0.80	41.30 $\pm$ 0.52	41.03
	SCAS	27.37 $\pm$ 0.64	82.45 $\pm$ 0.81	52.47 $\pm$ 0.73	25.37 $\pm$ 0.32	76.57 $\pm$ 0.40	43.80 $\pm$ 0.19	51.34

C Gradient Decomposition Details

For completeness, we derive Eqs. (1), (2), and (3). Recall that $A = XW$ with $X \in \mathbb{R}^{n \times d}$, $W \in \mathbb{R}^{d \times m}$, $A \in \mathbb{R}^{n \times m}$, and that $\mathcal{L} = \mathcal{L}(A)$ is a scalar loss depending on W only through A . We write $G = \frac{\partial \mathcal{L}}{\partial A} \in \mathbb{R}^{n \times m}$ and use the Frobenius inner product $\langle U, V \rangle_F = \text{tr}(U^\top V)$.

Since $A = XW$ and X is treated as constant when differentiating with respect to W , the differential of A is

$$dA = X dW. \quad (19)$$

By the chain rule,

$$\begin{aligned} d\mathcal{L} &= \left\langle \frac{\partial \mathcal{L}}{\partial A}, dA \right\rangle_F \\ &= \text{tr}(G^\top dA) \\ &= \text{tr}(G^\top X dW). \end{aligned} \quad (20)$$

Using the cyclic property of the trace, we obtain

$$\begin{aligned} \text{tr}(G^\top X dW) &= \text{tr}((X^\top G)^\top dW) \\ &= \langle X^\top G, dW \rangle_F. \end{aligned} \quad (21)$$

Comparing with the definition $d\mathcal{L} = \langle \nabla_W \mathcal{L}, dW \rangle_F$ yields Eq. (1):

$$\nabla_W \mathcal{L} = X^\top G. \quad (22)$$

Eq. (2) then follows immediately:

$$\|\nabla_W \mathcal{L}\|_F^2 = \|X^\top G\|_F^2. \quad (23)$$

To obtain Eq. (3), we expand the Frobenius norm in elementwise form. Let $z_i^\top$ and $g_i^\top$ denote the i -th rows of X and G , respectively. Then $XX^\top$

and $GG^\top$ have entries $(XX^\top)_{ij} = z_i^\top z_j$ and $(GG^\top)_{ij} = g_i^\top g_j$. Using $\|M\|_F^2 = \text{tr}(M^\top M)$ and the trace identity $\text{tr}(AB) = \sum_{i,j} A_{ij} B_{ji}$, we have

$$\begin{aligned} \|X^\top G\|_F^2 &= \text{tr}(G^\top X X^\top G) = \text{tr}(X X^\top G G^\top) \\ &= \sum_{i,j=1}^n (X X^\top)_{ij} (G G^\top)_{ji} \\ &= \sum_{i,j=1}^n (z_i^\top z_j) (g_i^\top g_j). \end{aligned} \quad (24)$$

which yields Eq. (3).

D Complexity Analysis

Our analysis is for a single target student weight matrix $W \in \mathbb{R}^{d \times m}$. For a candidate sequence $x = [q; a]$, let $n_Q = |I_Q|$, $n_A = |I_A|$, and $n = n_Q + n_A$. Let $F(n)$ and $B(n)$ denote the time cost of one forward pass and one backward pass of the student on a length- n sequence.

Exact blockwise gradient scoring implied by Eq. (4) requires both token representations and token-level backpropagated gradients, and therefore incurs one forward pass and one backward pass:

$$F(n) + B(n). \quad (25)$$

A direct implementation of Eq. (4) enumerates all token pairs in the AA, AQ/QA, and QQ blocks. Since these blocks contain n_A^2 , $n_A n_Q$, and n_Q^2 pairs, respectively, and each pairwise term requires $O(d + m)$ arithmetic, the additional overhead is

$$O((n_A^2 + n_A n_Q + n_Q^2)(d + m)) = O(n^2(d + m)). \quad (26)$$

Thus, under naive token-pair enumeration, the per-candidate score computation cost is

$$O(F(n) + B(n) + n^2(d + m)). \quad (27)$$

The same exact score can be computed more efficiently in matrix form. Let $X_A \in \mathbb{R}^{n_A \times d}$ and $X_Q \in \mathbb{R}^{n_Q \times d}$ denote the answer-side and question-side rows of X , and let $G_A \in \mathbb{R}^{n_A \times m}$ and $G_Q \in \mathbb{R}^{n_Q \times m}$ denote the corresponding rows of G . Then Eq. (4) can be written as

$$\begin{aligned} \|\nabla_W \mathcal{L}\|_F^2 &= \|X_A^\top G_A\|_F^2 + 2\langle X_A^\top G_A, X_Q^\top G_Q \rangle_F \\ &\quad + \|X_Q^\top G_Q\|_F^2. \end{aligned} \quad (28)$$

Computing $X_A^\top G_A$ costs $O(n_A dm)$ and computing $X_Q^\top G_Q$ costs $O(n_Q dm)$; the remaining Frobenius norms and inner product cost $O(dm)$. Hence, the additional overhead beyond forward/backward is

$$O(ndm), \quad (29)$$

and the per-candidate score computation cost under the optimized exact implementation becomes

$$O(F(n) + B(n) + ndm). \quad (30)$$

We now analyze the cost of computing our proxy in Eq. (9). This requires only one forward pass. After the forward pass, the representation aggregates in Eq. (5) are computed by linear-time accumulation over token positions, costing $O(nd)$. The difficulty terms in Eq. (6) are obtained by averaging token-level NLL values already produced by the same forward pass, which costs $O(n)$. The remaining score computation costs $O(d)$. Therefore, the per-candidate score computation cost of our proxy is

$$O(F(n) + nd). \quad (31)$$

The key difference is therefore that exact block-wise scoring requires a backward pass, and even under an optimized implementation still incurs an additional $O(ndm)$ post-backward cost, whereas our proxy is forward-only and requires only linear-time post-forward aggregation. In particular, compared with the optimized exact implementation, our score replaces

$$O(F(n) + B(n) + ndm) \quad (32)$$

with

$$O(F(n) + nd). \quad (33)$$

This is the main computational advantage of our method for large-scale answer selection.

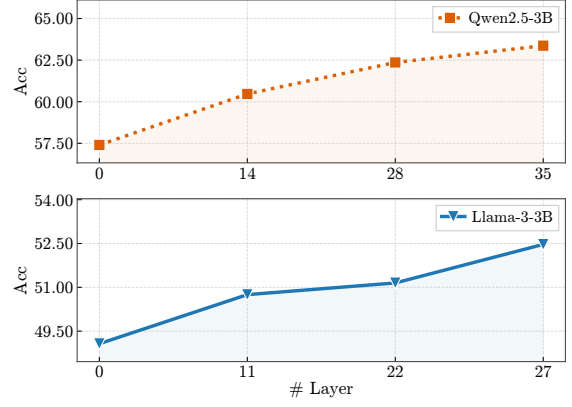


Figure 9: **Effect of the Representation Layer on Math.**

E Layer Analysis

In our main experiments, SCAS uses the final-layer hidden states of the student model to compute the answer-selection score. This choice follows our forward-only proxy: since token probabilities are computed from final-layer representations through the language modeling head, these representations provide the most direct connection to the NLL-based difficulty terms in our score. To examine the effect of layer choice, we conduct a layer analysis on the Math setting with Llama-3.2-3B and Qwen2.5-3B. For each model, we select several approximately evenly spaced layers from shallow to deep layers, and additionally include the final layer. All other components of the training and selection pipeline are kept unchanged. The results are shown in Figure 9. As the extraction layer becomes deeper, SCAS achieves consistently better performance on both student models. The final layer obtains the highest average accuracy for both Llama-3.2-3B and Qwen2.5-3B. This result supports our use of final-layer hidden states, suggesting that representations closest to the language modeling head better capture the student-centric signal needed for answer selection.

F Scaling Strong Teachers by Answer Union

We add a controlled appendix experiment to test whether simply merging answers from strong teachers can replace student-centric answer selection. We use the same nine teacher models and the same controlled teacher–student setup as Figure 1; details of the teacher set, common-correct training subset, and fine-tuning protocol are provided in Appendix A.4. Within this setup, we rank the nine

teachers by their direct test-set performance and construct cumulative teacher pools: top-1, top-1-2, top-1-3, and so on up to top-1-9. For each cumulative pool, we use the corresponding common-correct training subset and take the union of the selected teachers’ verified answers as the synthetic supervision set. The student is fine-tuned on this answer union for three epochs. There is no dynamic answer re-selection in this experiment. This setup asks whether adding more high-performing teachers, while keeping the controlled protocol of Figure 1, is sufficient to match SCAS.

Figure 10 reports the MATH results for Qwen2.5 students. Adding more high-performing teachers often improves over using only the single best teacher, but the gains are not enough to match SCAS. For Qwen2.5-0.5B, the answer-union baseline increases from 20.07 at top-1 to 24.21 at top-9, still 3.41 points below SCAS. For Qwen2.5-3B, it rises from 54.61 at top-1 to 58.55 at top-9, remaining 3.14 points below SCAS. For Qwen2.5-7B, it peaks at top-7 with 65.21 and then declines, leaving a 4.70-point gap from SCAS even at its best point. Importantly, the top- k union baseline is not a fixed-budget comparison: it trains on k verified answers for each retained question, so its answer-level training set is k times larger than the one-answer-per-question supervision used by SCAS on the same retained questions. Thus, even with up to $9\times$ answer-level training data, larger unions of strong-teacher answers do not recover the benefit of student-centric selection. The result supports our central claim: what matters is not only having more correct answers from stronger teachers, but selecting the answers that better match the current student’s learning state.

G Verification of Candidate Response Correctness and Quality

Mathematical reasoning tasks. Candidate responses are not filtered using exact string matching. For each teacher-generated response, an LLM verifier receives the problem, the reference answer, and the complete candidate solution, and evaluates both the final answer and the reasoning process. A candidate is retained only if its final answer is mathematically equivalent to the reference answer and its reasoning contains no substantive mathematical or logical error. Thus, a response with a correct final answer but flawed reasoning is rejected, while equivalent symbolic forms, simplified fractions, mi-

nor formatting differences, and omitted routine steps are accepted. To validate the model-based verification procedure, we randomly sampled 50 candidate solutions from each of GSM8K, MATH, DeepScaleR, and OpenR1-Math, yielding 200 examples in total. Four annotators independently evaluated each solution using the same final-answer and reasoning criteria, and the majority-vote label was used as the human reference. As shown in Table 11, the LLM verifier agreed with the human majority on 192 of 200 examples (96.0%), with per-dataset agreement ranging from 94.0% to 98.0%. These results support the reliability of the verifier in handling mathematically equivalent answers while rejecting solutions with substantive reasoning errors.

Open-ended instruction tuning. Open-ended instructions generally do not admit a unique reference answer. Therefore, we refer to the corresponding Magpie responses as *quality-audited candidate responses*, rather than verified-correct answers. All candidates were generated by Llama-3.3-70B-Instruct under the same decoding configuration. To assess their quality, we randomly sampled 1,000 responses and evaluated them using GPT-5 with a 1–5 rubric covering instruction following, factual or logical correctness, relevance and completeness, and clarity and coherence. Although a score of 3 denotes a usable response with minor issues, we adopt a more conservative threshold and count only scores of 4–5 as high quality. GPT-5 rated 998 of the 1,000 sampled responses (99.8%) as high quality. We further conducted a human evaluation on a random 100-response subset. Four annotators independently applied the same rubric, and the final human label was determined by majority vote. Human judgments classified 98 responses as high quality, while GPT-5 classified 99. Their binary judgments agreed on 99 of the 100 examples. These results support the overall quality of the open-ended candidate pool and the reliability of the model-based audit.

H Additional Evaluation on Logical Reasoning

To evaluate whether the method generalizes beyond mathematical reasoning, we conduct an additional experiment on logical reasoning. We fine-tune Qwen2.5-7B-Instruct on 8,000 examples from Min-

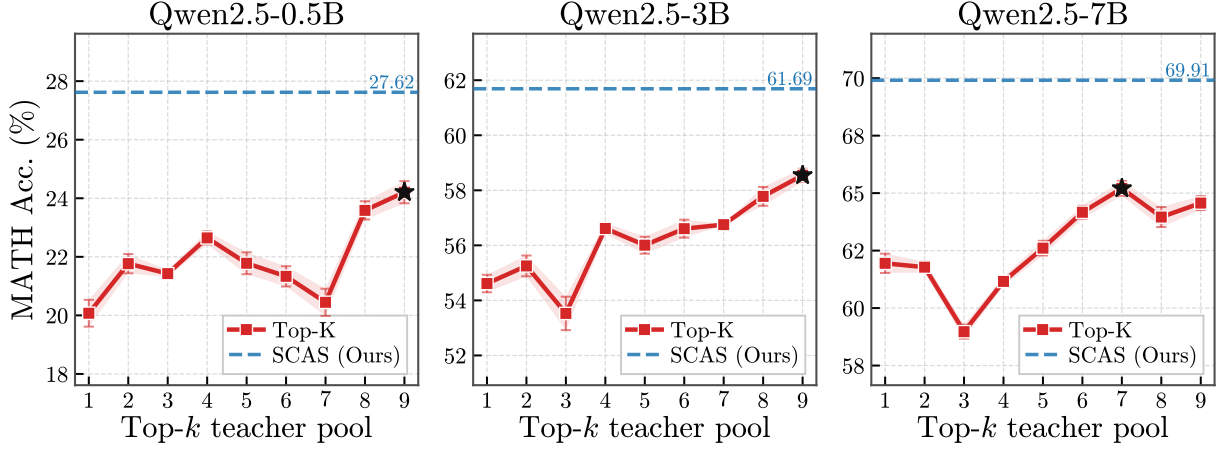


Figure 10: **Answer-Union Scaling with Strong Teachers.** Top- k merges k teachers’ verified answers, giving k training answers per retained question; SCAS uses one selected answer per question. The shaded band shows one standard deviation.

iMaxAI/SynLogic¹⁰. The candidate-answer pool is generated by Gemma-4-31B, GPT-OSS-120B, OLMo-3.1-32B, Phi-4-Reasoning, and Qwen3.5-27B. We evaluate the resulting student on the Large and Medium subsets of ZebraLogicBench¹¹. All methods use the same candidate pool, student model, training-set size, and fine-tuning configuration. As shown in Table 14, SCAS outperforms the strongest baseline by 2.1 points on the Large subset and 4.9 points on the Medium subset. This experiment provides additional evidence that student-centric answer selection generalizes from mathematical tasks to logical reasoning.

I Theoretical Analysis

In this appendix, we justify the proposed score $S(q, a; \theta)$ as a *ranking-oriented forward proxy* for the answer-involving interactions induced by a candidate answer a under the current student θ . Our goal is *not* to reconstruct the full-model gradient norm exactly. Instead, we derive a tractable answer-level quantity whose ordering is aligned with a retained last-layer interaction energy whenever the approximation error is smaller than the inter-candidate margin. Consistent with the implemented score, our analysis focuses on final-layer hidden states and final LM-head interactions.

Our analysis proceeds in three stages. First, we isolate the exact *last-layer* answer-involving interactions for a fixed question–answer sequence. Sec-

ond, we derive a *token-sensitive forward proxy* by replacing backward residual magnitudes with forward uncertainty signals. Third, we show that the implemented score in the main text arises from a *blockwise constant reduction* of this token-sensitive proxy, where the blockwise mean token NLL is the natural least-squares representative and also provides a denoising effect for answer-level ranking.

I.1 Exact last-layer answer-involving interactions

Let $x = [q; a]$ denote the training-templated sequence formed by concatenating the question q and a verified correct candidate answer a . Let $I = \{1, \dots, |x|\}$ be the token positions of x , and write

$$I = I_Q \cup I_A,$$

where I_Q indexes conditioning/question tokens and I_A indexes answer tokens.

Although the motivating decomposition in Section 2.1 is written for a generic weight matrix, the tractable forward proxy we derive here targets the *final LM head*, where the output residuals are available from a single forward pass. We use this last-layer quantity as the ranking target for answer selection.

Consider the final LM head

$$W \in \mathbb{R}^{d \times V},$$

where V is the vocabulary size. For each position $t \in I$, let $h_t \in \mathbb{R}^d$ be the final hidden state, let

$$p_t := p_\theta(\cdot \mid x_{<t}) \in \mathbb{R}^V,$$

$$y_t := x_t,$$

$$r_t := p_t - e_{y_t} \in \mathbb{R}^V.$$

¹⁰<https://huggingface.co/datasets/MiniMaxAI/SynLogic>

¹¹<https://huggingface.co/datasets/allenai/ZebraLogicBench>

Table 10: Prompt used to verify mathematical candidate solutions.

Evaluator prompt: Mathematical candidate verification	
System Prompt: You are an expert mathematics evaluator. Given a math problem, a reference answer, and a candidate solution, determine whether the candidate solution is correct. Evaluate both the final answer and the reasoning process. A candidate should be marked correct only if its final answer is mathematically equivalent to the reference answer and its reasoning contains no substantive mathematical or logical error. A candidate with a correct final answer but incorrect reasoning must be marked incorrect. Minor formatting differences or omitted routine steps should not be treated as errors. Respond with ONLY a valid JSON object: <pre>{ "correct": true, "model_answer": "extracted final answer", "final_answer_correct": true, "reasoning_correct": true, "reasoning": "brief explanation" }</pre>	
User Prompt: Problem: <problem> Reference Answer: <reference answer> Candidate Solution: <candidate solution>	
Assistant Response: <verifier JSON output>	

Table 11: Agreement between the mathematical-response verifier and majority-vote human judgments.

Dataset	Samples	LLM-human agreement
GSM8K	50	49/50 (98.0%)
MATH	50	48/50 (96.0%)
DeepScaleR	50	47/50 (94.0%)
OpenR1-Math	50	48/50 (96.0%)
Overall	200	192/200 (96.0%)

where e_{y_t} is the one-hot vector of the target token. The sequence loss is

$$\mathcal{L}(x) = \sum_{t \in I} -\log p_{\theta}(y_t | x_{<t}).$$

The gradient of the sequence loss with respect to the output head is

$$\nabla_W \mathcal{L}(x) = \sum_{t \in I} h_t r_t^{\top}. \quad (34)$$

Therefore,

$$\|\nabla_W \mathcal{L}(x)\|_F^2 = \sum_{i,j \in I} (h_i^{\top} h_j)(r_i^{\top} r_j). \quad (35)$$

Splitting token indices into conditioning and answer positions yields

$$\|\nabla_W \mathcal{L}(x)\|_F^2 = G_{AA}(x) + 2G_{AQ}(x) + G_{QQ}(x), \quad (36)$$

where

$$G_{AA}(x) = \sum_{i,j \in I_A} (h_i^{\top} h_j)(r_i^{\top} r_j), \quad (37)$$

$$G_{AQ}(x) = \sum_{i \in I_A, j \in I_Q} (h_i^{\top} h_j)(r_i^{\top} r_j), \quad (38)$$

and $G_{QQ}(x)$ is defined analogously.

Since answer selection compares multiple candidate answers for a *fixed* question q , the terms involving answer tokens are the primary objects of interest. At the last layer, the QQ block depends only on the conditioning prefix and is therefore invariant across candidate answers for the same q . Thus, for ranking purposes, it suffices to focus on the retained answer-involving interactions.

As in the main text, we work with normalized hidden states

$$u_t := \frac{h_t}{\|h_t\|_2}.$$

This yields the normalized retained blocks

$$\begin{aligned}\tilde{G}_{AA}(x) &= \frac{1}{|I_A|^2} \sum_{i,j \in I_A} (u_i^\top u_j)(r_i^\top r_j), \\ \tilde{G}_{AQ}(x) &= \frac{1}{|I_A||I_Q|} \sum_{i \in I_A, j \in I_Q} (u_i^\top u_j)(r_i^\top r_j).\end{aligned}\quad (39)$$

We then define the retained answer-involving interaction energy

$$\tilde{\mathcal{G}}_{\lambda^*}(q, a; \theta) := (1 - \lambda^*) \tilde{G}_{AA}(x) + \lambda^* \tilde{G}_{AQ}(x), \quad \lambda^* \in [0, 1]. \quad (40)$$

Our goal is to derive an efficient forward-only proxy for $\tilde{\mathcal{G}}_{\lambda^*}(q, a; \theta)$.

I.2 From exact interactions to a token-sensitive forward proxy

We first relate the residual magnitude $\|r_t\|_2$ to the token-level negative log-likelihood (NLL)

$$\ell_t := -\log p_\theta(y_t \mid x_{<t}).$$

Lemma I.1 (Residual magnitude is comparable to token NLL). *Assume*

$$p_\theta(y_t \mid x_{<t}) \geq \tau > 0.$$

Then

$$\tau \ell_t \leq \|r_t\|_2 \leq \sqrt{2} \ell_t. \quad (41)$$

Proof. Let

$$q_t := 1 - p_\theta(y_t \mid x_{<t}).$$

Since

$$r_t = p_t - e_{y_t},$$

we have

$$\begin{aligned}\|r_t\|_2^2 &= (1 - p_\theta(y_t \mid x_{<t}))^2 + \sum_{v \neq y_t} p_t(v)^2 \\ &= q_t^2 + \sum_{v \neq y_t} p_t(v)^2.\end{aligned}$$

Hence

$$q_t \leq \|r_t\|_2 \leq \sqrt{2} q_t.$$

Also,

$$\ell_t = -\log p_\theta(y_t \mid x_{<t}) = -\log(1 - q_t).$$

Using $-\log(1 - u) \geq u$ for $u \in [0, 1]$ gives

$$q_t \leq \ell_t.$$

Since $p_\theta(y_t \mid x_{<t}) \geq \tau$, we have $1 - q_t \geq \tau$, and by the mean value theorem,

$$\ell_t = -\log(1 - q_t) \leq \frac{q_t}{\tau}.$$

Therefore,

$$\tau \ell_t \leq q_t \leq \|r_t\|_2$$

and

$$\|r_t\|_2 \leq \sqrt{2} q_t \leq \sqrt{2} \ell_t,$$

which proves the claim. $\square$

The exact retained blocks in Eq. (39) depend not only on the magnitudes $\|r_t\|_2$, but also on the *pairwise alignment* between residual directions. To make this dependence explicit, define

$$\gamma_{ij} := \frac{r_i^\top r_j}{\|r_i\|_2 \|r_j\|_2} \in [-1, 1].$$

We assume that within each retained block, these pairwise alignment coefficients are concentrated around a blockwise average value.

Assumption I.2 (Blockwise residual alignment concentration). *There exist coefficients $\rho_{AA}, \rho_{AQ} \in [0, 1]$ and dispersion levels $\delta_{AA}, \delta_{AQ} \geq 0$ such that*

$$\begin{aligned}\gamma_{ij} &= \rho_{AA} + \Delta_{ij}^{AA}, \\ |\Delta_{ij}^{AA}| &\leq \delta_{AA}, \quad i, j \in I_A.\end{aligned}$$

and

$$\begin{aligned}\gamma_{ij} &= \rho_{AQ} + \Delta_{ij}^{AQ}, \\ |\Delta_{ij}^{AQ}| &\leq \delta_{AQ}, \quad i \in I_A, j \in I_Q.\end{aligned}$$

Under this assumption, the exact retained interactions admit a clean decomposition into a residual-magnitude-weighted forward proxy plus an explicit alignment error.

Define the residual-weighted summaries

$$\begin{aligned}\bar{w}_A &= \frac{1}{|I_A|} \sum_{i \in I_A} \|r_i\|_2 u_i, \\ \bar{w}_Q &= \frac{1}{|I_Q|} \sum_{j \in I_Q} \|r_j\|_2 u_j.\end{aligned}\quad (42)$$

and the corresponding residual-weighted block scores

$$\hat{T}_{AA}(x) := \|\bar{w}_A\|_2^2, \quad \hat{T}_{AQ}(x) := \bar{w}_A^\top \bar{w}_Q. \quad (43)$$

Proposition I.3 (Alignment-controlled residual-weighted proxy). *Under Assumption I.2,*

$$\tilde{G}_{AA}(x) = \rho_{AA} \hat{T}_{AA}(x) + E_{AA}^{\text{align}}(x), \quad (44)$$

and

$$\tilde{G}_{AQ}(x) = \rho_{AQ} \hat{T}_{AQ}(x) + E_{AQ}^{\text{align}}(x), \quad (45)$$

where

$$E_{AA}^{\text{align}}(x) = \frac{1}{|I_A|^2} \sum_{i,j \in I_A} (u_i^\top u_j) \|r_i\|_2 \|r_j\|_2 \Delta_{ij}^{AA}, \quad (46)$$

$$E_{AQ}^{\text{align}}(x) = \frac{1}{|I_A||I_Q|} \sum_{\substack{i \in I_A \\ j \in I_Q}} (u_i^\top u_j) \|r_i\|_2 \|r_j\|_2 \Delta_{ij}^{AQ}. \quad (47)$$

and satisfy

$$|E_{AA}^{\text{align}}(x)| \leq \delta_{AA} \hat{T}_{AA}^{\text{abs}}(x), \quad (48)$$

$$|E_{AQ}^{\text{align}}(x)| \leq \delta_{AQ} \hat{T}_{AQ}^{\text{abs}}(x), \quad (49)$$

with

$$\hat{T}_{AA}^{\text{abs}}(x) := \frac{1}{|I_A|^2} \sum_{i,j \in I_A} |u_i^\top u_j| \|r_i\|_2 \|r_j\|_2,$$

and

$$\hat{T}_{AQ}^{\text{abs}}(x) := \frac{1}{|I_A||I_Q|} \sum_{i \in I_A, j \in I_Q} |u_i^\top u_j| \|r_i\|_2 \|r_j\|_2.$$

Proof. Substituting

$$r_i^\top r_j = \|r_i\|_2 \|r_j\|_2 \gamma_{ij}$$

into Eq. (39) and using Assumption I.2 gives

$$\tilde{G}_{AA}(x) = \frac{1}{|I_A|^2} \sum_{i,j \in I_A} (u_i^\top u_j) \|r_i\|_2 \|r_j\|_2 (\rho_{AA} + \Delta_{ij}^{AA}).$$

The term involving ρ_{AA} is exactly

$$\rho_{AA} \left\| \frac{1}{|I_A|} \sum_{i \in I_A} \|r_i\|_2 u_i \right\|_2^2 = \rho_{AA} \hat{T}_{AA}(x),$$

and the remainder is $E_{AA}^{\text{align}}(x)$. The AQ case is identical. The absolute-value bounds follow from $|\Delta_{ij}^B| \leq \delta_B$ and the triangle inequality. $\square$

Proposition I.3 isolates the first source of approximation error: *alignment dispersion within the retained blocks*. The remaining step is to replace residual magnitudes $\|r_t\|_2$ by forward uncertainty signals. By Lemma I.1, token NLL provides such a signal up to bounded positive distortion.

Motivated by this, define the token-sensitive forward summaries

$$\bar{v}_A = \frac{1}{|I_A|} \sum_{i \in I_A} \ell_i u_i, \quad \bar{v}_Q = \frac{1}{|I_Q|} \sum_{j \in I_Q} \ell_j u_j, \quad (50)$$

and the corresponding token-sensitive forward proxies

$$T_{AA}(x) := \|\bar{v}_A\|_2^2, \quad T_{AQ}(x) := \bar{v}_A^\top \bar{v}_Q. \quad (51)$$

To keep the subsequent reduction explicit, define the distortion incurred by replacing residual magnitudes with token NLL by

$$\begin{aligned} \varepsilon_{AA}^{\text{nll}}(x) &:= \left| \hat{T}_{AA}(x) - \kappa_{AA} T_{AA}(x) \right|, \\ \varepsilon_{AQ}^{\text{nll}}(x) &:= \left| \hat{T}_{AQ}(x) - \kappa_{AQ} T_{AQ}(x) \right|. \end{aligned} \quad (52)$$

for some positive blockwise scale coefficients $\kappa_{AA}, \kappa_{AQ} > 0$. These coefficients absorb the bounded positive distortion between $\|r_t\|_2$ and ℓ_t , while $\varepsilon_{AA}^{\text{nll}}$ and $\varepsilon_{AQ}^{\text{nll}}$ capture the residual discrepancy due to within-block variation of that distortion.

Thus, after the alignment-controlled reduction in Proposition I.3, the natural forward-only object is the token-sensitive proxy (T_{AA}, T_{AQ}) . We next show why the implemented score in the main text uses *blockwise mean* token NLL rather than token-specific uncertainty weights.

I.3 Blockwise constant reduction and why mean token NLL

Recall the blockwise mean representations

$$\mu_A = \frac{1}{|I_A|} \sum_{i \in I_A} u_i, \quad \mu_Q = \frac{1}{|I_Q|} \sum_{j \in I_Q} u_j, \quad (53)$$

and the blockwise mean NLLs

$$d_A = \frac{1}{|I_A|} \sum_{i \in I_A} \ell_i, \quad d_Q = \frac{1}{|I_Q|} \sum_{j \in I_Q} \ell_j. \quad (54)$$

We first formalize why the blockwise mean is the natural answer-level summary of token uncertainty.

Lemma I.4 (Optimal blockwise constant approximation). *The blockwise means d_A and d_Q are the unique minimizers of the within-block squared approximation errors:*

$$\begin{aligned} d_A &= \arg \min_{c \in \mathbb{R}} \frac{1}{|I_A|} \sum_{i \in I_A} (\ell_i - c)^2, \\ d_Q &= \arg \min_{c \in \mathbb{R}} \frac{1}{|I_Q|} \sum_{j \in I_Q} (\ell_j - c)^2. \end{aligned}$$

Proof. For the answer block,

$$f_A(c) = \frac{1}{|I_A|} \sum_{i \in I_A} (\ell_i - c)^2.$$

Differentiating gives

$$f'_A(c) = -\frac{2}{|I_A|} \sum_{i \in I_A} (\ell_i - c),$$

so the unique stationary point satisfies

$$c = \frac{1}{|I_A|} \sum_{i \in I_A} \ell_i = d_A.$$

Since $f''_A(c) = 2 > 0$, this stationary point is the unique minimizer. The proof for d_Q is identical. $\square$

Lemma I.4 shows that replacing token-level uncertainty weights by d_A and d_Q is not an arbitrary heuristic. It is the least-squares optimal blockwise constant reduction of token-level uncertainty. This reduction is also natural for answer selection, which operates at the *answer level* rather than at the level of token-wise uncertainty allocation. In addition, averaging suppresses idiosyncratic token-level spikes, making the resulting answer ranking more robust to rare tokens, tokenization artifacts, and other localized effects that may not be stable across semantically equivalent candidates.

Let

$$\begin{aligned} \sigma_A^2 &= \frac{1}{|I_A|} \sum_{i \in I_A} (\ell_i - d_A)^2, \\ \sigma_Q^2 &= \frac{1}{|I_Q|} \sum_{j \in I_Q} (\ell_j - d_Q)^2. \end{aligned} \quad (55)$$

denote the within-block NLL variances.

We now define the blockwise quantities used in the main text:

$$\begin{aligned} C_{AA}(q, a; \theta) &= d_A^2 \|\mu_A\|_2^2, \\ C_{AQ}(q, a; \theta) &= d_A d_Q (\mu_A^\top \mu_Q). \end{aligned} \quad (56)$$

The next lemma formalizes the approximation error incurred by reducing the token-sensitive proxy to these blockwise quantities.

Lemma I.5 (Mean-field reduction error). *We have*

$$\begin{aligned} \|\bar{v}_A - d_A \mu_A\|_2 &\leq \sigma_A, \\ \|\bar{v}_Q - d_Q \mu_Q\|_2 &\leq \sigma_Q. \end{aligned} \quad (57)$$

Consequently,

$$|T_{AA}(x) - C_{AA}(q, a; \theta)| \leq 2d_A \sigma_A + \sigma_A^2, \quad (58)$$

and

$$|T_{AQ}(x) - C_{AQ}(q, a; \theta)| \leq d_Q \sigma_A + d_A \sigma_Q + \sigma_A \sigma_Q. \quad (59)$$

Proof. For the answer block,

$$\bar{v}_A - d_A \mu_A = \frac{1}{|I_A|} \sum_{i \in I_A} (\ell_i - d_A) u_i.$$

Using $\|u_i\|_2 = 1$ and Jensen's inequality,

$$\begin{aligned} \|\bar{v}_A - d_A \mu_A\|_2 &\leq \frac{1}{|I_A|} \sum_{i \in I_A} |\ell_i - d_A| \\ &\leq \left(\frac{1}{|I_A|} \sum_{i \in I_A} (\ell_i - d_A)^2 \right)^{1/2} = \sigma_A. \end{aligned}$$

The bound for $\bar{v}_Q$ is identical.

For Eq. (58), write

$$T_{AA}(x) = \|\bar{v}_A\|_2^2.$$

Then

$$\begin{aligned} &|\|\bar{v}_A\|_2^2 - d_A^2 \|\mu_A\|_2^2| \\ &= |(\bar{v}_A - d_A \mu_A)^\top (\bar{v}_A + d_A \mu_A)| \\ &\leq \|\bar{v}_A - d_A \mu_A\|_2 (\|\bar{v}_A\|_2 + d_A \|\mu_A\|_2). \end{aligned}$$

Since $\|\mu_A\|_2 \leq 1$ and

$$\|\bar{v}_A\|_2 \leq \|d_A \mu_A\|_2 + \|\bar{v}_A - d_A \mu_A\|_2 \leq d_A + \sigma_A,$$

we obtain

$$\begin{aligned} |T_{AA}(x) - C_{AA}(q, a; \theta)| &\leq \sigma_A (2d_A + \sigma_A) \\ &= 2d_A \sigma_A + \sigma_A^2. \end{aligned}$$

For Eq. (59), add and subtract $d_A \mu_A^\top \bar{v}_Q$:

$$\begin{aligned} |\bar{v}_A^\top \bar{v}_Q - d_A d_Q \mu_A^\top \mu_Q| &\leq |(\bar{v}_A - d_A \mu_A)^\top \bar{v}_Q| \\ &\quad + |d_A \mu_A^\top (\bar{v}_Q - d_Q \mu_Q)|. \end{aligned}$$

Using $\|\mu_A\|_2 \leq 1$, $\|\bar{v}_Q\|_2 \leq d_Q + \sigma_Q$, and Eq. (57),

$$\begin{aligned} |T_{AQ}(x) - C_{AQ}(q, a; \theta)| &\leq \sigma_A (d_Q + \sigma_Q) + d_A \sigma_Q \\ &= d_Q \sigma_A + d_A \sigma_Q + \sigma_A \sigma_Q. \end{aligned}$$

This proves the claim. $\square$

Lemma I.5 shows that the implemented score is obtained by a controlled blockwise reduction of the token-sensitive forward proxy. The error terms are governed by the within-block uncertainty dispersions σ_A and σ_Q . When token-level uncertainty is relatively homogeneous within each block, the sequence-level mean NLL provides a tight summary. When it is heterogeneous, the reduction trades some fidelity for improved answer-level robustness.

I.4 A combined proxy view and a ranking guarantee

We now combine the previous steps.

Let

$$\begin{aligned}\varepsilon_{AA}^{\text{mf}}(q, a; \theta) &:= 2d_A\sigma_A + \sigma_A^2, \\ \varepsilon_{AQ}^{\text{mf}}(q, a; \theta) &:= d_Q\sigma_A + d_A\sigma_Q + \sigma_A\sigma_Q.\end{aligned}\quad (60)$$

Theorem I.6 (Blockwise reduced forward proxy). *Under Assumption I.2, there exist nonnegative coefficients $\alpha, \beta \geq 0$ such that*

$$\left| \tilde{\mathcal{G}}_{\lambda^*}(q, a; \theta) - \left(\alpha C_{AA}(q, a; \theta) + \beta C_{AQ}(q, a; \theta) \right) \right| \leq \varepsilon_{\text{tot}}(q, a; \theta), \quad (61)$$

where

$$\begin{aligned}\varepsilon_{\text{tot}}(q, a; \theta) &= (1 - \lambda^*) \left(\varepsilon_{AA}^{\text{align}}(x) + |\rho_{AA}| \varepsilon_{AA}^{\text{nll}}(x) \right. \\ &\quad \left. + |\rho_{AA}| \kappa_{AA} \varepsilon_{AA}^{\text{mf}}(q, a; \theta) \right) \\ &\quad + \lambda^* \left(\varepsilon_{AQ}^{\text{align}}(x) + |\rho_{AQ}| \varepsilon_{AQ}^{\text{nll}}(x) \right. \\ &\quad \left. + |\rho_{AQ}| \kappa_{AQ} \varepsilon_{AQ}^{\text{mf}}(q, a; \theta) \right),\end{aligned}\quad (62)$$

with

$$\begin{aligned}\varepsilon_{AA}^{\text{align}}(x) &:= |E_{AA}^{\text{align}}(x)|, \\ \varepsilon_{AQ}^{\text{align}}(x) &:= |E_{AQ}^{\text{align}}(x)|.\end{aligned}$$

Proof. From Proposition I.3,

$$\begin{aligned}\tilde{G}_{AA}(x) &= \rho_{AA} \hat{T}_{AA}(x) + E_{AA}^{\text{align}}(x), \\ \tilde{G}_{AQ}(x) &= \rho_{AQ} \hat{T}_{AQ}(x) + E_{AQ}^{\text{align}}(x).\end{aligned}$$

By the definition of $\varepsilon_{AA}^{\text{nll}}$ and $\varepsilon_{AQ}^{\text{nll}}$ in Eq. (52),

$$\begin{aligned}\hat{T}_{AA}(x) &= \kappa_{AA} T_{AA}(x) + \Delta_{AA}^{\text{nll}}(x), \\ |\Delta_{AA}^{\text{nll}}(x)| &\leq \varepsilon_{AA}^{\text{nll}}(x).\end{aligned}$$

and analogously for the AQ block. Finally, Lemma I.5 gives

$$\begin{aligned}T_{AA}(x) &= C_{AA}(q, a; \theta) + \Delta_{AA}^{\text{mf}}(q, a; \theta), \\ |\Delta_{AA}^{\text{mf}}(q, a; \theta)| &\leq \varepsilon_{AA}^{\text{mf}}(q, a; \theta).\end{aligned}$$

and analogously for the AQ block. Substituting these relations into Eq. (40) and collecting terms yields Eq. (61), with

$$\alpha = (1 - \lambda^*) \rho_{AA} \kappa_{AA}, \quad \beta = \lambda^* \rho_{AQ} \kappa_{AQ}.$$

□

Theorem I.6 gives the precise interpretation of the main-text score: it is a *blockwise reduced forward proxy* for the retained last-layer answer-involving interaction energy. The total approximation error decomposes into three sources: (i)

alignment error, arising from residual-direction dispersion within each retained block; (ii) *NLL-scale distortion*, arising from replacing residual magnitudes by token NLL; and (iii) *mean-field reduction error*, arising from replacing token-level uncertainty allocation by its optimal blockwise constant approximation.

Because α and β are nonnegative, whenever $\alpha + \beta > 0$ the proxy in Eq. (61) can be rescaled into the same nonnegative-mixture form as the implemented score in the main text:

$$\begin{aligned}S(q, a; \theta) &= (1 - \lambda) C_{AA}(q, a; \theta) + \lambda C_{AQ}(q, a; \theta), \\ \lambda &\in [0, 1].\end{aligned}$$

In practice, λ absorbs the relative blockwise scale between the AA and AQ terms.

The relevance of this proxy to answer selection is captured by the following ranking corollary.

Corollary I.7 (Margin-based ranking preservation). *Fix a question q and two verified candidate answers $a, b \in \mathcal{A}(q)$. Suppose*

$$\begin{aligned}\tilde{\mathcal{G}}_{\lambda^*}(q, a; \theta) + \varepsilon_{\text{tot}}(q, a; \theta) \\ < \tilde{\mathcal{G}}_{\lambda^*}(q, b; \theta) - \varepsilon_{\text{tot}}(q, b; \theta).\end{aligned}\quad (63)$$

Then

$$\begin{aligned}\alpha C_{AA}(q, a; \theta) + \beta C_{AQ}(q, a; \theta) \\ < \alpha C_{AA}(q, b; \theta) + \beta C_{AQ}(q, b; \theta).\end{aligned}$$

Equivalently, whenever $\alpha + \beta > 0$, after normalizing the nonnegative coefficients α and β , the implemented blockwise proxy preserves the ordering of a and b .

Proof. Eq. (61) implies

$$\begin{aligned}\alpha C_{AA}(q, a; \theta) + \beta C_{AQ}(q, a; \theta) \\ \leq \tilde{\mathcal{G}}_{\lambda^*}(q, a; \theta) + \varepsilon_{\text{tot}}(q, a; \theta),\end{aligned}$$

and

$$\begin{aligned}\alpha C_{AA}(q, b; \theta) + \beta C_{AQ}(q, b; \theta) \\ \geq \tilde{\mathcal{G}}_{\lambda^*}(q, b; \theta) - \varepsilon_{\text{tot}}(q, b; \theta).\end{aligned}$$

The claim follows immediately from Eq. (63). □

Corollary I.7 makes explicit why the proposed score is suitable for answer selection. The score need not numerically match the exact retained interaction energy; it is sufficient that the approximation error remain smaller than the inter-candidate margin. This is precisely the regime relevant to SCAS, whose objective is to preserve the relative ordering of candidate answers rather than to estimate the exact interaction value itself.

J Teacher–Student Family Compatibility

We investigate whether the gradient-based scores capture compatibility between teacher and student model families. We conduct a controlled analysis using nine teachers: Gemma-4-31B, GPT-5-Chat, GPT-OSS-120B, Llama-3.3-70B, OLMo-3.1-32B, Phi-4-Reasoning, Qwen2.5-72B, Qwen3-32B, and Qwen3.5-27B. The students are Qwen2.5-3B and Llama-3.2-3B. For the Qwen student, the three Qwen teachers are treated as same-family and the remaining six as cross-family. For the Llama student, Llama-3.3-70B is treated as same-family and the remaining eight as cross-family. We follow the controlled setup of Figure 1. For each of DeepScaleR and MATH, we sample 1,000 questions for which all nine teachers produce verified-correct answers. We rank the nine candidate answers using either the gradient score or the student-centric learning-cost score and measure how often the selected answer is produced by a same-family teacher. We report results for direct minimum-score selection and for the grouped selection procedure used by SCAS. In the grouped setting, candidates are partitioned using the same configuration as the main experiments, and one answer is sampled uniformly from the lowest-score group. Applying this procedure to the student-centric learning-cost score corresponds to SCAS. Under family-agnostic uniform selection, the expected same-family rates are $3/9 = 33.3\%$ for Qwen2.5-3B and $1/9 = 11.1\%$ for Llama-3.2-3B. Direct minimum-gradient selection exceeds these reference rates on both datasets, indicating that teacher–student family compatibility is reflected in the gradient analysis, although the strength of this effect depends on the student. Direct minimum learning-cost selection exhibits the same tendency. Under the full SCAS grouping-and-sampling procedure, however, cross-family answers still account for 64.0%–67.3% of the selected responses. Grouping therefore reduces concentration on a single teacher family and preserves substantial cross-family diversity. These results suggest that model family influences candidate ranking but does not determine teacher selection under SCAS. Because model family is only a coarse proxy for similarity between training-data distributions, this analysis should be interpreted as evidence of compatibility rather than a causal attribution to training-data overlap.

K Prompt Templates for Teacher Generation and Evaluation

To make the data construction and evaluation protocol reproducible, we report the prompts used in our experiments. In each prompt, the dataset-specific question is inserted into the Question field, and the assistant field is reserved for model generation. The output field denotes the generated response. Tables 16, 17, 18, 19, and 20 show the prompts used to generate teacher answers for DeepScaleR, GSM8K, MATH, Magpie, and OpenR1-Math. Tables 21, 22, 23, and 24 show the prompts used for model-based evaluation. The model outputs shown in the evaluation prompt examples are responses generated by Qwen2.5-7B-Instruct.

Table 12: Prompt used for model-based quality audit of Magpie candidate responses.

Evaluator prompt: Magpie quality audit
System Prompt: You are an expert evaluator of open-ended instruction-following responses. Given a user instruction and a candidate response, evaluate the overall quality of the candidate response according to the following criteria: 1. Instruction Following: Whether the response correctly follows the user’s explicit requirements and intended goal. 2. Factual or Logical Correctness: Whether the response is factually accurate and logically consistent. For creative or subjective tasks, evaluate plausibility and internal consistency rather than requiring a unique answer. 3. Relevance and Completeness: Whether the response directly addresses the instruction and sufficiently covers its important aspects. 4. Clarity and Coherence: Whether the response is clear, understandable, and well organized. Open-ended instructions may admit multiple valid responses and may not have a unique reference answer. Do not penalize a response merely because it uses different wording, organization, or another valid approach. Use the following scoring scale: 5 = Excellent: Fully satisfies all evaluation criteria and represents exemplary instruction-tuning data. 4 = High quality: Satisfies all important evaluation criteria, provides a good response, and is suitable for instruction tuning. 3 = Good: Suitable for instruction tuning, with minor issues that do not affect its core correctness or usefulness. 2 = Weak: Contains a notable error, omission, irrelevant content, or failure to satisfy an important requirement, and is not suitable for instruction tuning. 1 = Unusable: Contains serious factual or logical errors, is largely irrelevant or incoherent, violates the main instruction, or contains unsafe content. Both scores 4 and 5 indicate high-quality responses. A score of 5 should be reserved for responses that are especially complete, precise, and well presented. Set “usable_for_training” to true for scores of 3, 4, or 5. Set “usable_for_training” to false for scores of 1 or 2. Respond with ONLY a valid JSON object in the following format: <pre>{ "score": 1, "usable_for_training": false, "issues": "none or a brief description of the identified issues", "reasoning": "a brief explanation of the evaluation" }</pre> Example: Instruction: Write a polite two-sentence email declining a meeting because of a scheduling conflict and suggest meeting next week instead. Candidate Response: Thank you for the invitation, but I am unable to attend due to a scheduling conflict. Would it be possible to reschedule the meeting for next week? Evaluation: <pre>{ "score": 5, "usable_for_training": true, "issues": "none", "reasoning": "The response follows all requested constraints and is polite, relevant, complete, and clear." }</pre> User Prompt: Evaluate the following response. Instruction: <user instruction> Candidate Response: <candidate response> Evaluation: Assistant Response: <evaluator JSON output>

Table 13: Quality audit of teacher-generated Magpie responses. High and lower denote scores of 4–5 and 1–3, respectively. The human evaluation set is a subset of the 1,000-response model-based audit set.

Judgment source	High	Lower	Human agreement
GPT-5 ($n = 1,000$)	998 (99.8%)	2 (0.2%)	–
Human majority ($n = 100$)	98 (98.0%)	2 (2.0%)	–
GPT-5 ($n = 100$)	99 (99.0%)	1 (1.0%)	99/100 (99.0%)

Table 14: Performance on ZebraLogicBench after fine-tuning Qwen2.5-7B-Instruct on SynLogic. Values are reported as mean $\pm$ standard deviation over three runs.

Method	Large	Medium
GRACE	7.2 $\pm$ 0.5	49.1 $\pm$ 0.8
RSR	5.1 $\pm$ 0.4	51.3 $\pm$ 0.7
SCAS	9.3 $\pm$ 0.6	56.2 $\pm$ 0.6

Table 15: Same-family selection rates under direct minimum-score selection and grouped sampling. Under family-agnostic uniform selection, the expected rates are 33.3% for Qwen2.5-3B and 11.1% for Llama-3.2-3B.

Dataset	Student	Min. Gradient	Min. Learning Cost	Grouped Gradient	SCAS
DeepScaleR	Qwen2.5-3B	93.1%	99.2%	57.1%	33.6%
DeepScaleR	Llama-3.2-3B	26.4%	84.5%	23.1%	32.7%
MATH	Qwen2.5-3B	93.1%	98.6%	61.9%	34.0%
MATH	Llama-3.2-3B	26.8%	92.0%	24.0%	36.0%

Table 16: Prompt used to generate teacher answers for DeepScaleR.

Teacher generation prompt: DeepScaleR	
System Prompt:	You are a helpful and precise math assistant.
User Prompt:	Please answer the math question below exactly as asked. When solving the problem, provide a detailed step-by-step solution. Clearly explain each step, including the reasoning, formulas, substitutions, simplifications, and calculations used. Do not give unnecessary unrelated explanations. Do not change the problem.
Question:	In triangle ABC with altitude AD , $\angle BAC = 45^\circ$, $DB = 3$, and $CD = 2$. Find the area of triangle ABC .
Assistant Response:	<teacher-generated answer>

Table 17: Prompt used to generate teacher answers for GSM8K.

Teacher generation prompt: GSM8K	
System Prompt:	You are a helpful and precise math assistant.
User Prompt:	Please answer the math question below exactly as asked. When solving the problem, provide a detailed step-by-step solution. Clearly explain each step, including the reasoning, formulas, substitutions, simplifications, and calculations used. Do not give unnecessary unrelated explanations. Do not change the problem.
Question:	Fred had 236 dollars to spend on 6 books. After buying them, he had 14 dollars. On average, how much did each book cost?
Assistant Response:	<teacher-generated answer>

Table 18: Prompt used to generate teacher answers for MATH.

Teacher generation prompt: MATH	
System Prompt:	You are a helpful and precise math assistant.
User Prompt:	Please answer the math question below exactly as asked. When solving the problem, provide a detailed step-by-step solution. Clearly explain each step, including the reasoning, formulas, substitutions, simplifications, and calculations used. Do not give unnecessary unrelated explanations. Do not change the problem.
Question:	In a rectangular coordinate system, what is the number of units in the distance from the origin to the point $(-15, 8)$?
Assistant Response:	<teacher-generated answer>

Table 19: Prompt used to generate candidate responses for Magpie.

Teacher generation prompt: Magpie	
System Prompt:	You are a helpful and precise assistant.
User Prompt:	Please answer the question or instruction below exactly as asked. When reasoning is needed, provide a clear step-by-step solution and explain each step. Do not give unnecessary unrelated explanations. Do not change the question or instruction.
Question or Instruction:	What is the probability that a randomly selected integer from the set $\{10, 12, 15, 18, 20\}$ is divisible by both 2 and 3?
Assistant Response:	<teacher-generated response>

Table 20: Prompt used to generate teacher answers for OpenR1-Math.

Teacher generation prompt: OpenR1-Math	
System Prompt:	You are a helpful and precise math assistant.
User Prompt:	Please answer the math question below exactly as asked. When solving the problem, provide a detailed step-by-step solution. Clearly explain each step, including the reasoning, formulas, substitutions, simplifications, and calculations used. Do not give unnecessary unrelated explanations. Do not change the problem.
Question:	I4.1 Given that x and y are real numbers such that $ x + x + y = 10$ and $ y + x - y = 10$. If $P = x + y$, find the value of P .
Assistant Response:	<teacher-generated answer>

Table 21: Evaluator prompt used for DeepScaleR.

Evaluator prompt: DeepScaleR	
System Prompt: You are an expert mathematics evaluator. Your task is to evaluate whether a model's answer to a math problem is correct. You will be given: - The problem statement - The reference answer - The model's output Carefully analyze the model's reasoning and final answer. Extract the answer from the model's output and compare it with the reference answer. Respond with ONLY a JSON object (no other text): <pre>{ "correct": true or false, "model_answer": "extracted answer", "reference_answer": "reference answer", "reasoning": "brief explanation" }</pre>	
User Prompt: Problem: $\tan 2\alpha = \frac{\cos \alpha}{2 - \sin \alpha}$, where $0 < \alpha < \frac{\pi}{2}$, find the value of $\tan \alpha$.	
Reference Answer: $\frac{\sqrt{15}}{15}$	
Model Output: We are given: $\tan 2\alpha = \frac{\cos \alpha}{2 - \sin \alpha}, \quad 0 < \alpha < \frac{\pi}{2}.$ Use the double-angle identity $\tan 2\alpha = \frac{2 \tan \alpha}{1 - \tan^2 \alpha}.$ Let $t = \tan \alpha$. Expressing the right-hand side in terms of t gives $\frac{\cos \alpha}{2 - \sin \alpha} = \frac{1}{2\sqrt{1+t^2} - t}.$ Thus, $\frac{2t}{1 - t^2} = \frac{1}{2\sqrt{1+t^2} - t}.$ Cross-multiplying and simplifying gives $4t\sqrt{1+t^2} = t^2 + 1.$ Squaring both sides, $16t^2(1+t^2) = (t^2 + 1)^2,$ so $15t^4 + 14t^2 - 1 = 0.$ Let $u = t^2$. Then $15u^2 + 14u - 1 = 0,$ which yields $u = \frac{1}{15}$ or $u = -1$. Since $u = t^2 > 0$, $u = \frac{1}{15}$. Hence $t = \tan \alpha = \frac{1}{\sqrt{15}}.$ Answer: $\tan \alpha = \frac{1}{\sqrt{15}}.$	
Evaluation instruction: Evaluate if the model's answer is correct. Respond with JSON only.	
Assistant Response: <evaluator JSON output>	

Table 22: Evaluator prompt used for GSM8K.

Evaluator prompt: GSM8K	
System Prompt: You are an expert mathematics evaluator. Your task is to evaluate whether a model’s answer to a math problem is correct. You will be given: - The problem statement - The reference answer - The model’s output Carefully analyze the model’s reasoning and final answer. Extract the answer from the model’s output and compare it with the reference answer. Respond with ONLY a JSON object (no other text):	
<pre>{ "correct": true or false, "model_answer": "extracted answer", "reference_answer": "reference answer", "reasoning": "brief explanation" }</pre>	
User Prompt: Problem: James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total meters does he run a week?	
Reference Answer: He sprints $3 * 3 = 9$ times. So he runs $9 * 60 = 540$ meters. ##### 540	
Model Output: To find out how many total meters James runs in a week, we need to calculate the distance he covers in one session and then multiply it by the number of sessions he has in a week. James runs 3 sprints each session, and each sprint is 60 meters. So, the distance he covers in one session is: $3 \text{ sprints/session} \times 60 \text{ meters/sprint} = 180 \text{ meters/session.}$ Since he runs 3 times a week, we multiply the distance covered in one session by 3 to get the total distance for the week: $180 \text{ meters/session} \times 3 \text{ sessions/week} = 540 \text{ meters/week.}$ Therefore, James runs a total of 540 meters a week.	
Evaluation instruction: Evaluate if the model’s answer is correct. Respond with JSON only.	
Assistant Response:	<evaluator JSON output>

Table 23: Evaluator prompt used for MATH.

Evaluator prompt: MATH
System Prompt: You are an expert mathematics evaluator. Your task is to evaluate whether a model's answer to a math problem is correct. You will be given: - The problem statement - The reference answer - The model's output Carefully analyze the model's reasoning and final answer. Extract the answer from the model's output and compare it with the reference answer. Respond with ONLY a JSON object (no other text): <pre data-bbox="272 801 743 958">{ "correct": true or false, "model_answer": "extracted answer", "reference_answer": "reference answer", "reasoning": "brief explanation" }</pre> User Prompt: Problem: What is the sum of the four positive factors of the positive integer value of $\sqrt{196}$? Reference Answer: Calculate $\sqrt{196} = \sqrt{2^2 \cdot 7^2} = 2 \cdot 7$. The sum of the four positive factors is $1 + 2 + 7 + 14 = \boxed{24}$. Model Output: First, find the positive integer value of $\sqrt{196}$: $\sqrt{196} = 14.$ Next, determine the positive factors of 14: The factors of 14 are 1, 2, 7, and 14. Sum these factors: $1 + 2 + 7 + 14 = 24.$ Answer: $\boxed{24}$ Evaluation instruction: Evaluate if the model's answer is correct. Respond with JSON only. Assistant Response: <evaluator JSON output>

Table 24: Evaluator prompt used for OpenR1-Math.

Evaluator prompt: OpenR1-Math	
System Prompt: You are an expert mathematics evaluator. Your task is to evaluate whether a model’s answer to a math problem is correct. You will be given: - The problem statement - The reference answer - The model’s output Carefully analyze the model’s reasoning and final answer. Extract the answer from the model’s output and compare it with the reference answer. Respond with ONLY a JSON object (no other text):	
<pre>{ "correct": true or false, "model_answer": "extracted answer", "reference_answer": "reference answer", "reasoning": "brief explanation" }</pre>	
User Prompt: Problem: Problem 2. Zoki cycled 64% of the way, and walked the remaining 9 km. How many kilometers did Zoki cycle?	
Reference Answer: 16	
Model Output: Let the total distance be D km. Zoki cycled 64% of the way, so the cycling distance is: $0.64D.$ The remaining distance is: $D - 0.64D = 0.36D.$ We are told this equals 9 km: $0.36D = 9.$ Solve for D : $D = \frac{9}{0.36} = 25.$ So the total distance is 25 km. Cycling distance: $0.64 \times 25 = 16.$ Answer: 16 km.	
Evaluation instruction: Evaluate if the model’s answer is correct. Respond with JSON only.	
Assistant Response:	<evaluator JSON output>