Too Smart to Teach: The Articulability Ceiling in Language Models

Shervin Ghasemlou
Apple
shervinghasemlou@gmail.com

Abstract

Larger language models solve harder problems, but do they also produce proportionally better explanations? We study in-context explanation transfer with 40 open-weight discovery teachers (360M–72B), five receivers (360M–7B), 1,300 problems across 13 benchmarks, and a separate 1,248-problem held-out evaluation extending to frontier-scale teachers. Our central finding is an *articulability ceiling*: increases in solving capability do not produce commensurate improvements in pedagogical value. On held-out problems, mean receiver gains rise sharply from small to mid-sized teachers (2.70 to 14.51 percentage points) but barely change for frontier teachers (14.84 points); gain per 100 words shows the same descriptive plateau (5.47 vs. 5.52 points). In the discovery study, receiver score peaks descriptively in the 120–170-word bin, and a matched teacher–problem–receiver analysis associates longer explanations with lower performance (-0.0123 per 100 words). Teacher correctness is a strong predictor of transfer, while surface lexical overlap poorly characterizes explanation quality. These findings distinguish solving, teaching, and the cost of articulation under fixed receiver and token budgets.

1 Introduction

Stronger models solve harder problems, but are they proportionally better teachers? We identify an *articulability ceiling*: increasing model capability does not yield commensurate ability to convey useful reasoning through explanations.

Table 1 illustrates how two *constructed* explanations of the same problem can differ in their explicit intermediate steps. The example is pedagogical, not a quotation from model generations or an experimental comparison. Human experts similarly suffer from an "expert blind spot" [Nathan and Petrosino, 2003, Hinds, 1999]: they can omit steps needed by novices. Figure 1 previews how discovery and held-out measurements reveal unequal teaching returns per unit of capability and explanatory length.

Table 1: Constructed illustration (not observed teacher outputs): two possible explanations of an arithmetic-series task, differing in the number of intermediate steps.

Explanation style	Example explanation	Expository difference
Concise	*Use the formula* $S = n(a_1 + a_n)/2$.	Compact presentation of the solution.
Stepwise	*Pair the first and last terms, count the pairs, then multiply by the pair-sum.*	Explicit intermediate pairing steps.

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

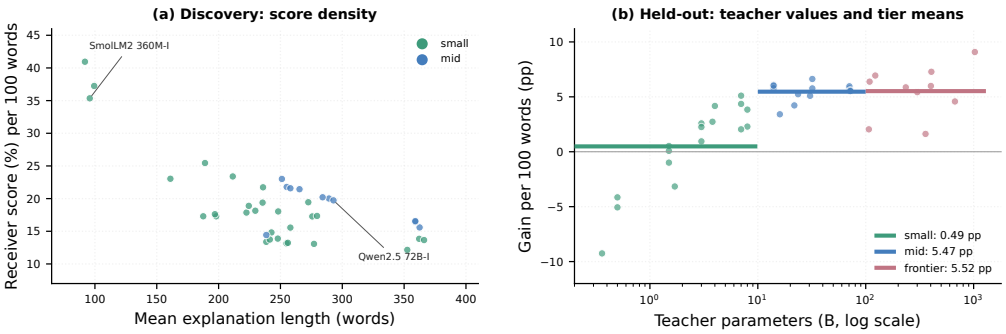


Figure 1: **The articulability ceiling at a glance.** *Left:* Discovery receiver score per 100 explanation words against teacher explanation length (40 teachers); color distinguishes parameter-count tier. This is a score-density measure, not baseline-adjusted gain. *Right:* Held-out gain per 100 words for individual teachers against parameter count (log scale), with tier means indicated. Mean efficiency rises from small to mid-sized teachers and changes little at frontier scale.

By *articulability*, we mean the degree to which competence can be externalized in explanations that benefit another model. Frontier capability need not imply frontier teaching: Polanyi [Polanyi, 1966] described tacit expertise, and Chi et al. [1981] found that experts and novices represent problems differently. We test this distinction with 40 open-weight discovery teachers (360M–72B), five receivers, and 1,300 code, math, reasoning, and knowledge problems. Our protocol:

1. Measures **skill** (pass@1) via domain-specific evaluation (code execution, numerical extraction, choice matching).
2. Elicits four **explanation types** from each model for every problem.
3. Measures **transfer**: how much each explanation improves 5 receivers, with answer-stripping and control baselines to quantify answer leakage.
4. Evaluates **explanation–solution alignment** via five metrics (ROUGE-L, unigram F1, weighted Jaccard, differential transfer, LLM-as-judge).

Three findings challenge the assumption that scaling capability scales teaching ability:

Finding 1: An articulability ceiling separates solving and teaching. In the independent held-out sample, mid-sized teachers produce 14.51 percentage points of mean receiver gain, essentially the same aggregate level as frontier teachers (14.84 points), despite the latter's much greater parameter count (Figure 8). Gain per 100 words likewise changes only from 5.47 to 5.52 points. In the *discovery-set receiver-score-per-word* measure, SmolLM2-360M-Instruct achieves 3.54×10^{-3} score units per word versus 1.97×10^{-3} for Qwen2.5-72B-Instruct ($1.8\times$); this descriptive ratio is distinct from improvement over the no-guidance baseline. Thus, teaching ability cannot be read from model size alone.

Finding 2: Verbosity has a measurable pedagogical cost. On discovery problems, the highest mean receiver *score* occurs in the 120–170-word bin (0.475), followed by lower scores for longer bins; this is a descriptive bin maximum, not a prescribed optimum. More importantly, comparing explanation styles within the same teacher, problem, and receiver yields a -0.0123 change in receiver score for each additional 100 words (95% CI $[-0.0143, -0.0104]$). Higher total gain can coexist with a verbosity penalty; token-constrained applications must balance capability and explanation cost.

Finding 3: Correctness and pedagogical form matter. Correct teachers provide about $5.6\times$ the average receiver gain of incorrect teachers, and full explanations outperform answer-only and shuffled controls. E4 (pedagogical) prompts lead on average transfer gain among the four styles. Meanwhile, instruct-tuned and specialized models receive higher explanation–solution correspondence ratings than base/code-base models from two independent LLM judges (gpt-oss-120b: 3.93 vs. 2.11; Gemma-3-27B-it: 4.23 vs. 2.22), despite a small and metric-dependent difference in lexical overlap. These results favor evaluating explanations by the benefit they provide to receivers, not solely by surface similarity or the teacher's own pass@1.

2 Related Work

Background. The "expert blind spot" [Nathan and Petrosino, 2003, Hinds, 1999] and Polanyi's tacit knowledge thesis [Polanyi, 1966] establish that human expertise often resists articulation [Chi et al., 1981, Dreyfus and Dreyfus, 1986]. Chain-of-thought prompting [Wei et al., 2022, Kojima et al., 2022] improves LLM reasoning, but growing evidence questions whether these traces are faithful [Turpin et al., 2023, Lanham et al., 2023, Ye and Durrett, 2022]. Jacovi and Goldberg [2020] formalized the plausibility–faithfulness distinction; Lyu et al. [2023] and Radhakrishnan et al. [2023] proposed methods for improving faithfulness. On the distillation side, Ho et al. [2022], Magister et al. [2023], Hsieh et al. [2023] showed LLM-generated rationales can teach smaller models, while Gudibande et al. [2023] cautioned that imitation produces superficial transfer. Sharma et al. [2023] operationalized sycophancy in RLHF-trained models [Ouyang et al., 2022, Bai et al., 2022]. Self-training and self-correction [Zelikman et al., 2022, Huang et al., 2024], self-consistency decoding [Wang et al., 2023a], and weak-to-strong supervision [Burns et al., 2023] face related challenges. Lampinen et al. [2022] studied explanations in few-shot prompts, while Saha et al. [2023] examined stronger LLM teachers explaining to weaker students at test time under a communication budget. Min et al. [2022] questioned whether demonstration content matters for in-context learning. Snell et al. [2024] studied test-time compute allocation; our experiments do not vary test-time computation and instead normalize transfer by explanatory word count. Performance scales with compute [Kaplan et al., 2020, Hoffmann et al., 2022], and capability measurement differs across protocols, including repeated sampling [Li et al., 2024] and preference judging [Zheng et al., 2023]. Additional context appears in Appendix K.

We study *in-context* explanation transfer without gradient updates. We contribute (1) per-word teaching measurements and a held-out mid-to-frontier plateau; (2) five complementary alignment measures distinguishing lexical overlap from textual correspondence and receiver benefit; and (3) cross-family transfer comparisons, with positive mean gain across receiver families and a same-family advantage.

3 Methodology

We test 40 discovery teachers and five receivers on 1,300 problems from 13 benchmarks, using task-specific scoring. Model lists, prompts and compute details are in Appendices A–H; benchmark sources in Appendix P and the held-out roster in Table 11.

3.1 Experimental Protocol

Our protocol has five steps. First, each model answers all 1,300 problems with a standardized prompt, and we define teacher skill as pass@1 under domain-specific evaluation. Second, for each problem p, each teacher generates four explanation styles at temperature $= 0.7$:

E1 (Declarative): "Describe your approach in plain English so a less experienced programmer could implement it. No code."
E2 (Annotated): "Solve this problem. Annotate each line explaining why."
E3 (Contrastive): "Do not solve. Explain: (a) key insight; (b) common mistakes; (c) why wrong."
E4 (Pedagogical): "Tutor a novice step-by-step, simple language, no jargon, no code."

This yields 208,000 explanation instances.

Third, we evaluate transfer with 5 receivers spanning 360M–7B parameters and 3 families. For each (teacher, receiver, explanation-type) triple, we:

1. **Apply heuristic answer filtering.** We remove answer-announcing lines and sentences using `strip_v4` (Appendix E). This is a lightweight preprocessing step; answers are naturally embedded in explanations, so we use controls rather than claiming perfect filtering.
2. Place the stripped explanation in the fixed receiver prompt (Appendix C), followed by the problem and answer-format line.
3. Measure receiver accuracy using the domain-specific evaluator.
4. Compute transfer gain: $\tau = \text{acc}_{\text{with explanation}} - \text{acc}_{\text{baseline}}$, using the same receiver and problem no-guidance baseline (the same receiver pass except for three discovery re-run teachers, which use the original pass baseline).

Fourth, we measure self-transfer by giving each model its own explanations and comparing accuracy to its cached baseline. Fifth, we add two controls to separate answer leakage from pedagogical content:

Answer-only: The receiver is given "The answer is: [text]" with [text] the first 100 characters of the teacher's skill response, instead of an explanation.

Shuffled: The receiver is given an explanation from the wrong problem, drawn from the same teacher.

These controls test problem-specific benefit, not faithfulness to a hidden reasoning process.

We distinguish three outcomes throughout: receiver *score* with explanation S, improvement over the matched no-guidance baseline $G = S - S_0$, and explanation economy. We report *score per word* as S/L (with L the mean stripped explanation length); it describes the density of observed success, not causal teaching benefit. *Gain per 100 words*, $100G/L$, measures baseline-adjusted improvement under the same normalization. They measure distinct aspects of explanatory economy.

3.2 Statistical Methods

We report Pearson r and Spearman ρ. Discovery controls, explanation-type comparisons and the same- versus cross-family test use problem bootstraps ($B = 2{,}000$; the smallest attainable p is 0.0005). Fixed-cell and word-matched regression intervals use problem-clustered standard errors (1,300 and 386 problem clusters). Split-half stability is summarized by its empirical percentile range over 200 random splits. Held-out tier comparisons use a two-way bootstrap (problems within source and teachers within tier; $B = 20{,}000$). We stratify descriptive results by domain.

Variance and stability. While individual binary receiver scores vary widely (row-level $\mathrm{SD} = 0.497$; pooled within-problem $\mathrm{SD} = 0.427$), teacher rankings are stable: between-teacher standard deviation is 0.094 transfer-score units across 1,300 problems, indicating that aggregate rankings are robust to single-sample explanations. This supports aggregate ranking stability despite variation across explanation–receiver interactions.

4 Experiments and Results

4.1 Skill Assessment

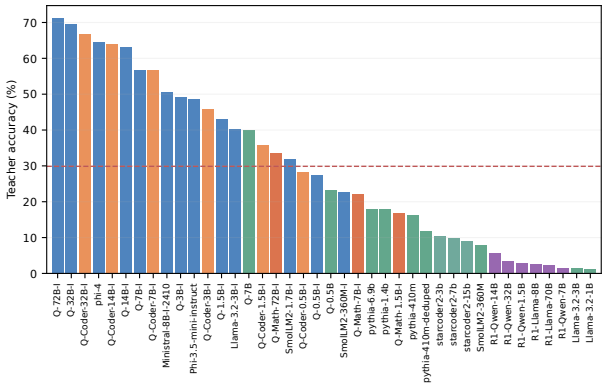


Figure 2: **Model skill scores** (pass@1) on 1,300 benchmark problems using domain-specific evaluation. Colors indicate model type: blue = instruct, orange = code-instruct, red-orange = math-instruct, purple = reasoning-distilled, green = base, and teal = code-base. The red dashed line shows the mean. Skill is assessed using the corresponding task-specific scorer for each benchmark.

Figure 2 shows pass@1 scores across all 40 teachers. Three patterns are robust:

- **Size alone does not determine skill.** Skill reflects scale and training objective (Spearman $\rho = 0.29$ for log parameters versus pass@1 across 40 teachers); parameter count alone does not determine rankings.

- **Training type creates variation at matched scale.** Models of comparable size differ in measured solving skill across training types.
- **Training type matters across families.** Instruction-tuned models generally lead on aggregate solving accuracy; domain specialists can nevertheless provide stronger transfer within their matched task domains (Section 4.5).

4.2 Transfer and Controls

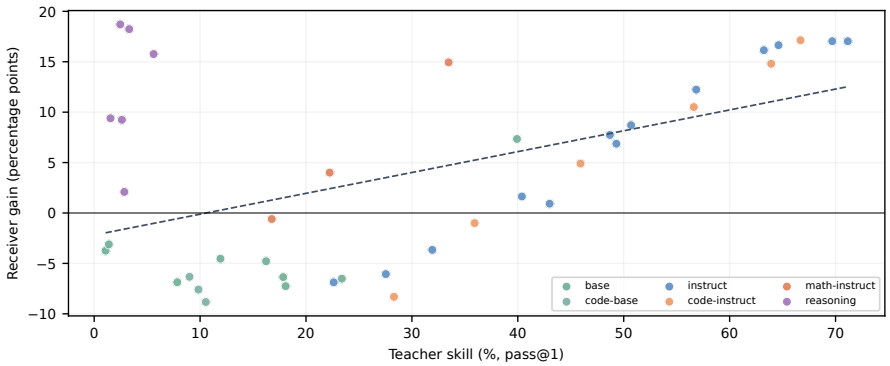


Figure 3: **Receiver gain versus teacher skill.** Each point is one discovery teacher, averaged over four explanation styles and five receivers. Colors distinguish teacher training type; the dashed line is a linear fit across teachers. Gain is measured relative to matched no-guidance receiver accuracy.

Figures 3–4 show three patterns:

- High-gain teachers span multiple sizes and training specializations; teacher skill does not uniquely determine the transfer ranking.
- Aggregate transfer correlates moderately with skill ($r = 0.50$). Explanation length increases with model size (Spearman $\rho = 0.51$ with log parameters) but not with skill across all 40 teachers ($\rho = -0.05$): reasoning-distilled teachers write the longest explanations despite relatively low pass@1. Section 4.3 separates verbosity from pedagogical efficiency.
- No teacher dominates all five receivers (Figure 4); self-transfer varies from -12.6 to $+11.6$ pp, correlating with skill ($r = 0.41$) and transfer gain ($r = 0.58$).

Figure 5 shows that E4 (pedagogical) produces the highest mean gain (5.12 percentage points), followed by E2 (4.40), E1 (4.12), and E3 (2.33). Full explanations outperform the defined answer-only and shuffled controls on average, indicating a benefit of problem-specific information beyond these controls.

With 5 receivers spanning 360M to 7B parameters across 3 model families, weaker receivers benefit more from explanations, consistent with greater headroom for improvement. Cross-family transfer (mean gain 3.4 pp) and same-family transfer (5.3 pp) are both positive; the same-family advantage is 1.8 pp (95% CI $[1.1, 2.6]$; bootstrap $p \leq 0.0005$, $B = 2{,}000$). Mean gain is positive across all three receiver families, although the Llama-3.2-3B-Instruct estimate is not distinguishable from zero ($+1.2$ pp, 95% CI $[-0.8, 3.2]$).

Table 2: Control baseline results with problem-bootstrap p-values ($B = 2{,}000$; smallest attainable $p = 0.0005$).

Condition	Avg Score	p-value vs. Full
Full explanation (E1, answer-stripped)	0.4487	—
Answer-only baseline	0.4290	≤ 0.0005
Shuffled-explanation baseline	0.3884	≤ 0.0005

Table 2 summarizes the main controls. Full explanations outperform the answer-only proxy, consistent with additional useful problem-specific content. They also outperform shuffled explanations, showing that problem-specific content matters. Teacher correctness is similarly decisive: correct teachers achieve mean receiver score 0.640 versus 0.365 for incorrect teachers, corresponding to gains of $+9.4$

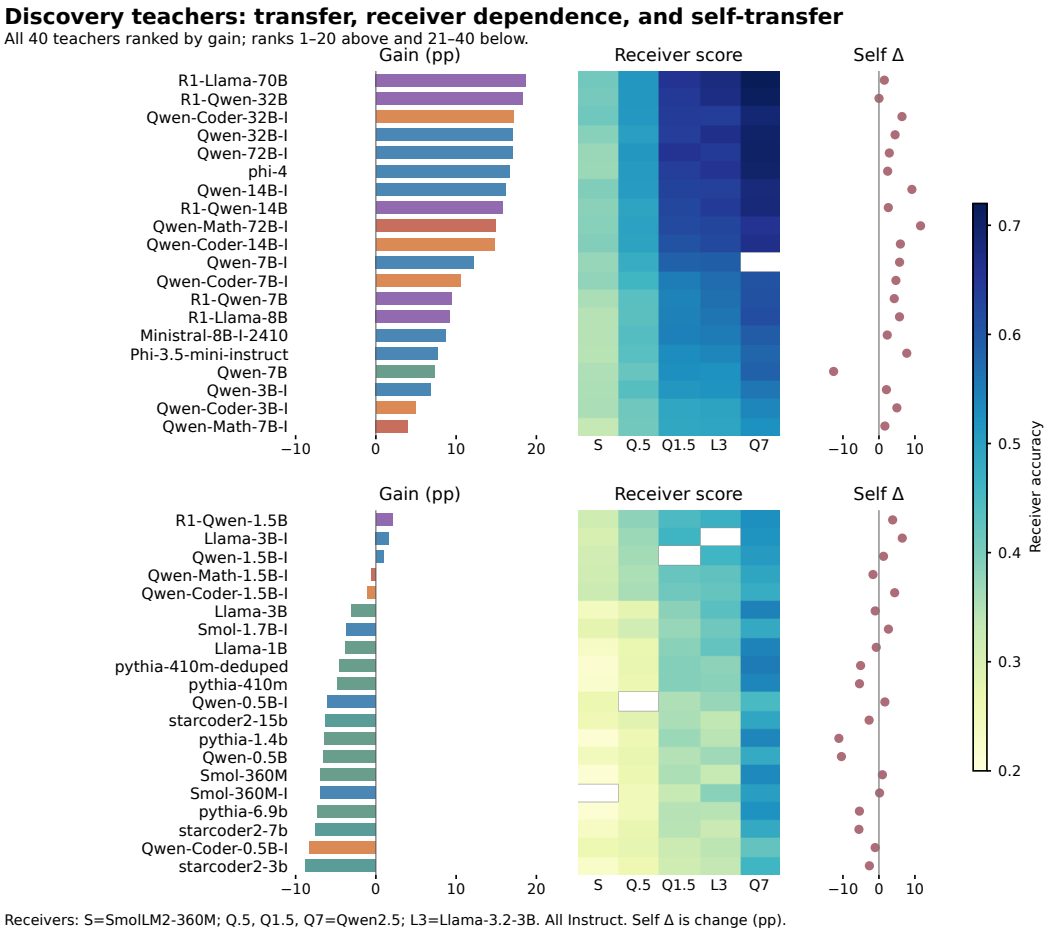
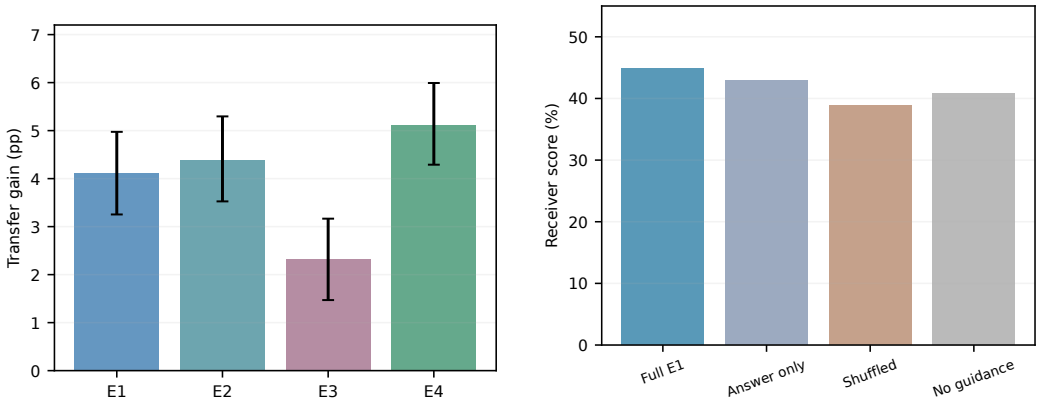


Figure 4: **Discovery teacher rankings, receiver-specific scores, and self-transfer.** All 40 teachers (ranked in two blocks) are shown by mean baseline-adjusted receiver gain (left bars, percentage points), receiver accuracy with guidance (five shaded columns), and self-transfer change (right dots, percentage points). Identity-matched cells are visually blanked in this panel; discovery aggregates include these cases, while held-out comparisons exclude them. Teacher names and types are in Appendix A and receiver definitions in Appendix B; Appendix N gives the ten highest-ranked teachers and receiver averages.



(a) Transfer effectiveness by explanation type.

(b) Control baseline comparison.

Figure 5: **Explanation type and baseline comparisons.** *Left:* E4 (pedagogical) yields the highest mean gain among E1–E4; error bars show 95% bootstrap intervals. *Right:* Receiver accuracy with full E1 explanations exceeds answer-only, shuffled, and no-guidance controls (bars show condition means).

versus $+1.7$ percentage points. The complete 40-teacher ranking appears in Figure 4; Appendix N presents the top ten and receiver means, while Appendix O examines self-transfer.

Code and scored inputs are described in Appendix I.

4.3 Articulability Ceiling

A naive reading of aggregate transfer suggests that stronger teachers produce better explanations. That pattern weakens once explanation length is taken into account. Larger models produce much longer explanations (for example, Qwen2.5-72B-Instruct averages 293 words, versus 96 words for SmolLM2-360M-Instruct), so total transfer conflates pedagogical quality with sheer verbosity.

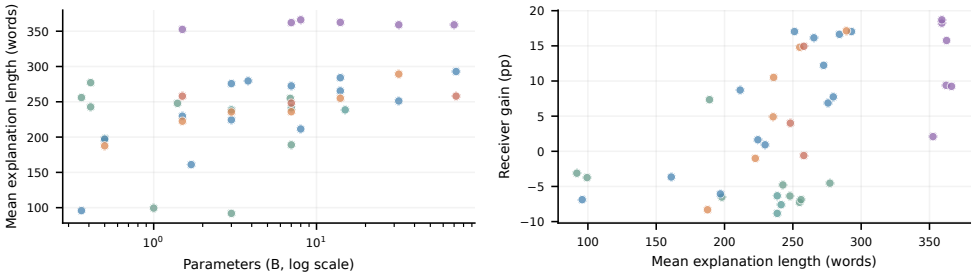


Figure 6: **Explanation length and receiver gain.** *Left:* Mean stripped explanation length versus model parameter count (log scale). *Right:* Receiver gain versus mean length across discovery teachers. Both panels show teacher-level aggregates; they do not by themselves identify the effect of adding words.

Figure 6 shows how model size, explanation length and total gain covary. Longer answers can increase total transfer while reducing score density; the within-teacher comparisons help separate these effects.

To summarize the joint teacher-level relation, we fit the polynomial interaction model

$$\tau = \beta_0 + \beta_1 \cdot \text{skill} + \beta_2 \cdot \text{length} + \beta_3 \cdot \text{length}^2 + \beta_4 \cdot (\text{skill} \times \text{length}) + \varepsilon, \tag{1}$$

which explains substantial descriptive variation ($R^2 = 0.81$, $n = 40$). Its quadratic coefficient does not independently establish a descending transfer curve in these data. The stronger length-specific result comes from the fixed teacher–problem–receiver contrast in Section 4.4, while the ceiling in returns to scale is directly visible in the held-out tier comparison.

Table 3: **Discovery receiver score by actual explanation length.** The observed maximum is in the 120–170-word bin; bins are not teacher-balanced, and the $\geq$500-word bin is small. This is receiver accuracy, not baseline-adjusted gain.

Length Bin	n entries	Mean Score	Note
$<$ 80 words	96,215	0.421	Short
80–120 words	49,020	0.461	—
120–170 words	99,025	0.475	Local high
170–220 words	139,825	0.461	Declining
220–300 words	261,045	0.430	Declining
300–500 words	394,490	0.452	Partial rebound
$\geq$ 500 words	380	0.397	Long

Table 3 reaches its highest coarse-bin receiver score at 120–170 words (0.475). At the 25-word resolution of Figure 7, the curve has two main peaks, 47.9% at 125–150 words and 49.3% at 425–450 words; the curve falls beyond 450 words where sample sizes are sparse. The $\geq$500-word subset contains only 380 observations and should not be treated as a stable optimum estimate. Length bins also differ in teacher composition. The independent fixed-cell estimate (-0.0123 per additional 100 words) shows that even within a given teacher–problem–receiver combination, verbosity can carry a measurable performance penalty. In token-constrained settings, explanatory economy matters even when larger teachers have higher absolute gains.

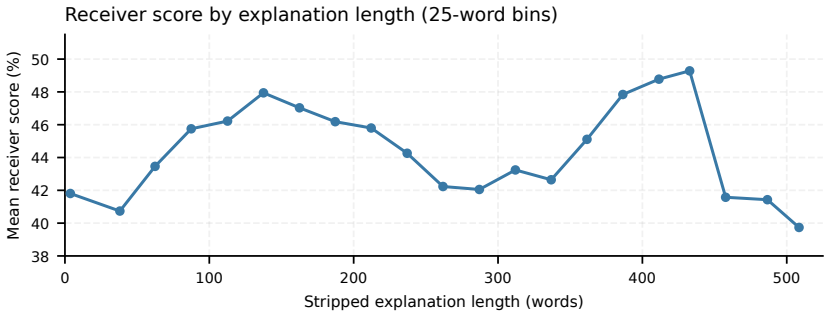


Figure 7: **Receiver performance across explanation lengths.** Mean discovery receiver score across 25-word bins of stripped explanation length. The curve shows variation across both shorter and longer explanations; observations are sparse near the upper boundary. These descriptive averages do not control for teacher or problem composition.

4.4 Held-out Evaluation and Within-Cell Length Analysis

The held-out evaluation draws problems from the same 13 benchmark sources as the discovery set, using a separate sampling seed and excluding discovery examples. The held-out set contains 1,264 problems (100 per source, except 64 for HumanEval, the available remainder after discovery sampling); 16 near-duplicates of discovery items are removed before analysis, leaving H^* with 1,248 unique items. The overlap check is at the example level rather than a guarantee against prior model-training exposure. Five receivers are evaluated with a fresh empty-guidance baseline in the same receiver pass. For receiver r, teacher t, and problem p, gain is the scored answer with an answer-stripped explanation minus the matched no-guidance score; teacher–receiver identity matches are omitted (Appendix Table 11 lists all teachers). Gains are then averaged by teacher and grouped by total parameter count: small ($<$ 10B, 17 teachers), mid (10–100B, 13 teachers), and frontier ($>$ 100B, 10 teachers). This setup measures *behavioral* transfer rather than the fidelity of the teacher's unobserved reasoning.

Held-out answers are scored with v4.1, a frozen revision of the discovery v4.0 scorer that improves answer-option, revised-final-answer, MATH unit/fraction and code-test handling; v4.0 provides a sensitivity check. Table 10 gives the held-out tier summary; Figure 8 separates total gain (accuracy percentage points) from baseline-adjusted gain per 100 words. From small to mid-sized teachers, total gain rises from 2.70 to 14.51 points and gain per 100 words from 0.49 to 5.47 points. Beyond mid-sized teachers, the measured improvements nearly stop: frontier averages are 14.84 and 5.52, respectively. The mid-minus-frontier differences are -0.33 points (95% CI $[-3.17, 2.87]$) for gain and -0.05 points (95% CI $[-1.44, 1.41]$) for gain per 100 words. Thus, the held-out results show a pronounced *empirical ceiling in incremental pedagogical return from scale*, without requiring a universal optimal size or a significant drop at the frontier.

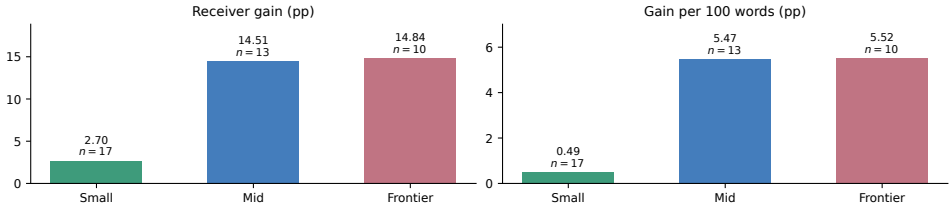


Figure 8: **Held-out transfer by teacher-size tier ($|H^*| = 1{,}248$ problems).** *Left:* average receiver gain (percentage points). *Right:* gain per 100 explanation words (percentage points). Each teacher contributes to its parameter-count tier; labels above bars give the tier mean and number of teachers. See Appendix Table 10 for values.

A complementary, within-cell analysis on the 1,300-problem discovery set holds teacher, problem, and receiver fixed when comparing the four elicited explanation styles. After controlling for explanation type, the fitted change in receiver score is -0.0123 per additional 100 words (95% CI $[-0.0143, -0.0104]$; 1.04 million transfer rows and 1,300 problem clusters). This isolates a

Table 4: **Domain-matched Qwen2.5 comparisons** on discovery problems. Receiver gain is in percentage points; the final column is specialist minus size-matched general-purpose gain (differences computed before rounding). These subset comparisons are exploratory.

Domain	Size	General gain	Specialist gain	Difference
Math	1.5B	7.65	14.08	+6.43
Math	7B	14.45	17.23	+2.78
Math	72B	12.53	16.25	+3.73
Code	1.5B	−0.99	3.10	+4.09
Code	7B	12.61	15.93	+3.31
Code	14B	14.56	19.44	+4.88
Code	32B	17.01	20.11	+3.10

within-condition length association from the across-teacher variation in explanation length. It is an observational contrast rather than a randomized deletion or compression experiment. Together, the held-out readouts and the within-cell analysis make total transfer and explanation efficiency separately observable rather than collapsing them into a single teacher ranking.

4.5 Matched Domain Specialization

Teacher scale is only one axis of specialization. To compare training orientation at approximately fixed parameter count, we pair Qwen2.5 general-purpose teachers with corresponding Math and Coder variants and evaluate them on the corresponding domain subset of the discovery sample. Table 4 reports receiver gain relative to its matched no-guidance baseline, in percentage points. Math-specialist gains exceed their general-purpose counterparts by 2.78–6.43 points in the three matched sizes; Code-specialist gains exceed the corresponding general models by 3.10–4.88 points across four sizes. These contrasts concern domain-matched subsets, not a cross-domain ranking, and remain exploratory because the sample is divided into smaller benchmark cells. They nonetheless illustrate why model family, training objective, and task domain belong alongside parameter count when describing the ability to teach.

The held-out study extends to 1,026B parameters (22 discovery teachers and 18 additional models); the specialist contrasts compare models of matched size within one family. Both measure receiver behavior, not hidden reasoning.

4.6 Explanation–Solution Alignment

Appendix Figure 9 shows a metric-dependent lexical pattern: the pooled 28 instruction-tuned and specialized teachers have slightly lower mean ROUGE-L overlap than the 12 base and code-base teachers (0.170 vs. 0.172), but higher unigram F1 (0.244 vs. 0.218) and weighted Jaccard overlap (0.139 vs. 0.107). Differential transfer provides a separate behavioral measure, and lexical alignment correlates weakly with differential transfer (Spearman $\rho = -0.12$, $p = 0.47$; $n = 40$ teachers).

Instruction-tuned models may rephrase solutions in pedagogical language, affecting lexical-overlap scores. Both LLM judges in Appendix M assign higher explanation–solution correspondence ratings to instruct and specialized models than base and code-base models (gpt-oss-120b: 3.93 vs. 2.11; Gemma-3-27B-it: 4.23 vs. 2.22). These ratings concern observable correspondence between solution text and explanations, not hidden reasoning faithfulness or a causal RLHF effect.

5 Discussion

5.1 Interpretation

Our study measures *in-context* explanation transfer, not gradient-based distillation. Its central result is an articulability ceiling: on 1,248 held-out problems, gain rises sharply from small to mid-sized teachers but nearly levels off at frontier scale (Table 10). Score density and within-teacher length contrasts separately expose the cost of explanation. Correctness and full-versus-control transfer show that content matters beyond lexical overlap. For in-context teaching and tutoring, teacher selection should reflect the receiving model and explanatory budget, not model scale alone.

5.2 Robustness Analyses

We assess four potential confounds.

Word-matched control. Among discovery explanations of identical lengths between 50–130 words, matched within problem, receiver and style, a tenfold increase in teacher parameter count is associated with $+0.121$ receiver score ($+12.1$ pp, 95% CI $[0.085, 0.157]$; 3,170 matched cells, 386 problems), controlling for teacher correctness, instruction tuning and specialization. This does not negate a ceiling in incremental returns; the separate within-teacher comparison controls teacher identity (Section 4.4).

Answer leakage stratification. After stripping, explicit final-answer statements are detected in 0.24% of explanations. On GSM8K/MATH, the gold answer value nevertheless appears in 42.7% of stripped texts.

Cluster-bootstrap sensitivity and ranking stability. Full explanations outperform answer-only and shuffled controls (both $p \leq 0.0005$), and correct teachers transfer more than incorrect teachers ($p < 0.001$). Teacher rankings are stable across problem halves (mean split-half Spearman $\rho = 0.992$ over 200 random half-splits; 2.5–97.5th percentile range $[0.986, 0.996]$).

Length-bin composition. The 120–170-word peak describes receiver score, not a universal optimum: only 8.5% of its rows come from 32B+ teachers. The $\geq$500-word bin is thin; fixed-cell length effects and held-out tiers provide complementary evidence.

Broader impacts. Better estimates of teacher–receiver transfer may help allocate explanation budgets in in-context tutoring and assistive systems. A risk is that fluent but incorrect or answer-revealing explanations may appear pedagogically effective without being faithful; reporting receiver gains with correctness and answer-only controls helps expose this limitation. Deployment requires task- and receiver-specific validation.

5.3 Limitations

The discovery and held-out sets broaden coverage but leave some benchmark subsets thin, so per-benchmark findings are exploratory. The held-out plateau is conditional on the explanation budget: receivers saw at most 512 tokens per explanation. The 22 discovery-roster teachers generated up to 512 tokens; 18 additional teachers generated up to 2,048 tokens, with only the first 512 passed to receivers. Longer explanations could change the comparison. Only five receivers (360M–7B) are studied. The held-out mid–frontier confidence intervals include zero; a universal decline is not established. Score per word must not be confused with gain per word. Neither receiver transfer nor the two textual LLM judges verifies hidden reasoning faithfulness or isolates an RLHF effect. Disjoint test examples do not rule out pretraining contamination. These boundaries qualify scope, not the importance of directly measuring teaching ability.

6 Conclusion

We identify an *articulability ceiling*: model scale and problem-solving competence do not yield proportional improvements in natural-language teaching. Across 40 discovery teachers, five receivers and 1,300 problems, correctness and explanation form predict useful transfer, while matched within-teacher comparisons link added verbosity to lower receiver performance. On 1,248 independently sampled held-out problems, mean gains nearly plateau from mid-sized to frontier teachers (14.51 versus 14.84 percentage points; gain per 100 words 5.47 versus 5.52). This ceiling in incremental returns persists despite the much larger frontier models.

Answer-only and shuffled controls demonstrate the value of problem-specific guidance; lexical overlap and LLM textual-correspondence ratings do not establish hidden reasoning faithfulness. The practical implication is to select in-context teachers by measured receiver benefit and explanation cost, rather than solving capability alone. Articulation is a distinct axis of model performance, not an automatic byproduct of scale.

Acknowledgments and Disclosure of Funding

Funding and competing interests. This work was funded by Apple, with which the author is affiliated. The author declares no competing interests.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv:2108.07732*, 2021.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Collin Burns et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv:2312.09390*, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv:2107.03374*, 2021.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. *ICLR*, 2024.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. *EMNLP*, 2023.

Michelene TH Chi, Paul J Feltovich, and Robert Glaser. Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5(2):121–152, 1981.

Aidan Clark, Diego De Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, George Bm Van Den Driessche, Eliza Rutherford, Tom Hennigan, Matthew J Johnson, Albin Cassirer, Chris Jones, Elena Buchatskaya, David Budden, Laurent Sifre, Simon Osindero, Oriol Vinyals, Marc'Aurelio Ranzato, Jack Rae, Erich Elsen, Koray Kavukcuoglu, and Karen Simonyan. Unified scaling laws for routed language models. In *Proceedings of the 39th International Conference on Machine Learning*, pages 4057–4086, 2022.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. *arXiv:2110.14168*, 2021.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv:2501.12948*, 2025.

Hubert L Dreyfus and Stuart E Dreyfus. *Mind over Machine*. Free Press, 1986.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. Specializing smaller language models towards multi-step reasoning. *ICML*, 2023.

Alex Gu, Baptiste Roziere, Hugh James Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida Wang. CRUXEval: A benchmark for code reasoning, understanding and execution. In *International Conference on Machine Learning*, 2024.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary LLMs. *arXiv:2305.15717*, 2023.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring coding challenge competence with apps. *NeurIPS*, 2021a.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *ICLR*, 2021b.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the NeurIPS 2021 Datasets and Benchmarks Track*, 2021c.

Pamela J Hinds. The curse of expertise: The effects of expertise and debiasing methods on prediction of novice performance. *Journal of Experimental Psychology: Applied*, 5(2):205–221, 1999.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. 2015.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. *arXiv:2212.10071*, 2022.

Jordan Hoffmann, Sebastian Borgeaud, et al. An empirical analysis of compute-optimal large language model training. *NeurIPS*, 2022.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, 2023.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *ICLR*, 2024.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, et al. Qwen2.5-Coder technical report. *arXiv:2409.12186*, 2024.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, 2020.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. *EMNLP*, 2023.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv:2207.05221*, 2022.

Jared Kaplan, Sam McCandlish, et al. Scaling laws for neural language models. *arXiv:2001.08361*, 2020.

Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. The CoT Collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12685–12708, 2023.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *NeurIPS*, 2022.

Andrew K Lampinen, Ishita Dasgupta, Stephanie CY Chan, Kory Mathewson, Michael Henry Tessler, Antonia Creswell, James L McClelland, Jane X Wang, and Felix Hill. Can language models learn from explanations in context? *Findings of EMNLP*, 2022.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv:2307.13702*, 2023.

Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. Common 7b language models already possess strong math capabilities. *arXiv preprint arXiv:2403.04706*, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. LogiQA 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962, 2023.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *TACL*, 2024.

Anton Lozhkov et al. StarCoder 2 and the stack v2: The next generation. *arXiv:2402.19173*, 2024.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. *IJCNLP-AACL*, 2023.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. In *ACL*, 2023.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *EMNLP*, 2022.

Subhabrata Mukherjee et al. Orca: Progressive learning from complex explanation traces of GPT-4. *arXiv:2306.02707*, 2023.

Mitchell J Nathan and Anthony Petrosino. Expert blind spot among preservice teachers. *American Educational Research Journal*, 40(4):905–928, 2003.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022.

Michael Polanyi. *The Tacit Dimension*. Doubleday, 1966.

Qwen Team. Qwen2.5-Math: The world's leading open-sourced mathematical LLMs. https://qwenlm.github.io/blog/qwen2.5-math/, 2024.

Qwen Team. Qwen2.5 technical report. *arXiv:2412.15115*, 2025.

Ansh Radhakrishnan, Karina Nguyen, Anna Chen, Carol Chen, Carson Denison, Danny Hernandez, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamile Lukosiute, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Sam McCandlish, Sheer El Showk, Tamera Lanham, Tim Maxwell, Venkatesa Chandrasekaran, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Question decomposition improves the faithfulness of model-generated reasoning. *arXiv preprint arXiv:2307.11768*, 2023.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. *KDD*, 2016.

Swarnadeep Saha, Peter Hase, and Mohit Bansal. Can language models teach? teacher explanations improve student performance via personalization. In *Advances in Neural Information Processing Systems*, 2023.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. *AAAI*, 2020.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R Johnston, et al. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*, 2023.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. Distilling reasoning capabilities into smaller language models. *ACL Findings*, 2023.

Avi Singh et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv:2312.06585*, 2024.

Dan Isaac Slobin. From "thought and language" to "thinking for speaking". *Rethinking Linguistic Relativity*, pages 70–96, 1996.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Samuel Stanton, Pavel Izmailov, et al. Does knowledge distillation really work? *NeurIPS*, 2021.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *NeurIPS*, 2023.

Mirac Suzgun, Nathan Scales, et al. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. *arXiv:2210.09261*, 2022.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Fine-tuning language models for factuality. *ICLR*, 2024.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *NeurIPS*, 2023.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *ICLR*, 2023a.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, and Hannaneh Hajishirzi. How far can camels go? exploring the state of instruction tuning on open resources. *NeurIPS*, 2023b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

Peter West, Ximing Lu, Nouha Dziri, Faeze Brahman, Linjie Li, Jena D Hwang, Liwei Jiang, Jillian Fisher, Abhilasha Ravichander, Khyathi Chandu, et al. The generative AI paradox: What it can create, it may not understand. *ICLR*, 2024.

Sarah Wiegreffe, Ana Marasović, and Noah A Smith. Measuring association between labels and free-text rationales. In *EMNLP*, 2021.

Xi Ye and Greg Durrett. The unreliability of explanations in few-shot prompting for textual reasoning. *NeurIPS*, 2022.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. In *International Conference on Machine Learning*, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D Goodman. STaR: Bootstrapping reasoning with reasoning. *NeurIPS*, 2022.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? *ACL*, 2019.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *NeurIPS*, 2023.

A Full Model List

Major teacher series are documented in the Qwen2.5, Qwen2.5-Coder, Qwen2.5-Math, DeepSeek-R1 and StarCoder2 reports [Qwen Team, 2025, Hui et al., 2024, Qwen Team, 2024, DeepSeek-AI, 2025, Lozhkov et al., 2024]. The model names, sizes and types are given in Table 5; these citations do not imply common training or evaluation protocols.

We evaluate 40 open-weight teacher models spanning 360M–72B parameters across 8 model families: Qwen2.5, DeepSeek-R1-Distill, Pythia, StarCoder2, Llama 3.2, SmolLM2, Phi, and Ministral. The set includes base, instruct-tuned, code-specialized, math-specialized, and reasoning-distilled variants so that pedagogical transfer can be compared across both scale and training methodology. Table 5 lists the 40 discovery teachers, indexed for reference.

Table 5: Complete model list with parameter counts and training type.

#	Model	Params	Type
1	Pythia-410m	410M	Base
2	Pythia-410m-deduped	410M	Base
3	Pythia-1.4b	1.4B	Base
4	Pythia-6.9b	6.9B	Base
5	SmolLM2-360M	360M	Base
6	SmolLM2-360M-Instruct	360M	Instruct
7	SmolLM2-1.7B-Instruct	1.7B	Instruct
8	Qwen2.5-0.5B	0.5B	Base
9	Qwen2.5-0.5B-Instruct	0.5B	Instruct
10	Qwen2.5-Coder-0.5B-Instruct	0.5B	Code-Instruct
11	Qwen2.5-1.5B-Instruct	1.5B	Instruct
12	Qwen2.5-Coder-1.5B-Instruct	1.5B	Code-Instruct
13	Qwen2.5-Math-1.5B-Instruct	1.5B	Math-Instruct
14	Qwen2.5-3B-Instruct	3B	Instruct
15	Qwen2.5-Coder-3B-Instruct	3B	Code-Instruct
16	Qwen2.5-7B	7B	Base
17	Qwen2.5-7B-Instruct	7B	Instruct
18	Qwen2.5-Coder-7B-Instruct	7B	Code-Instruct
19	Qwen2.5-Math-7B-Instruct	7B	Math-Instruct
20	Qwen2.5-14B-Instruct	14B	Instruct
21	Qwen2.5-Coder-14B-Instruct	14B	Code-Instruct
22	Qwen2.5-32B-Instruct	32B	Instruct
23	Qwen2.5-Coder-32B-Instruct	32B	Code-Instruct
24	Qwen2.5-72B-Instruct	72B	Instruct
25	Qwen2.5-Math-72B-Instruct	72B	Math-Instruct
26	DeepSeek-R1-Distill-Qwen-1.5B	1.5B	Reasoning-Distill
27	DeepSeek-R1-Distill-Qwen-7B	7B	Reasoning-Distill
28	DeepSeek-R1-Distill-Llama-8B	8B	Reasoning-Distill
29	DeepSeek-R1-Distill-Qwen-14B	14B	Reasoning-Distill
30	DeepSeek-R1-Distill-Qwen-32B	32B	Reasoning-Distill
31	DeepSeek-R1-Distill-Llama-70B	70B	Reasoning-Distill
32	Llama-3.2-1B	1B	Base
33	Llama-3.2-3B	3B	Base
34	Llama-3.2-3B-Instruct	3B	Instruct
35	Phi-3.5-mini-instruct	3.8B	Instruct
36	phi-4	14B	Instruct
37	Ministral-8B-Instruct-2410	8B	Instruct
38	StarCoder2-3b	3B	Code-Base
39	StarCoder2-7b	7B	Code-Base
40	StarCoder2-15b	15B	Code-Base

B Receiver Models

Table 6 lists the 5 receiver models used in transfer evaluation, spanning 3 model families and 360M–7B parameters.

Table 6: The receiver models used in transfer experiments.

Receiver	Params	Family
SmolLM2-360M-Instruct	360M	SmolLM
Qwen2.5-0.5B-Instruct	0.5B	Qwen
Qwen2.5-1.5B-Instruct	1.5B	Qwen
Llama-3.2-3B-Instruct	3B	Llama
Qwen2.5-7B-Instruct	7B	Qwen

C Explanation Type Definitions

The explanation prompts below have a blank line before "Problem: [PROBLEM]"; the placeholder is the problem body without its answer-format line. The skill and receiver prompts, by contrast, include the answer-format line. The four explanation types use these templates:

E1 (Declarative).

```
Describe your approach in plain English so a less experienced
programmer could implement it.  No code.

Problem:  [PROBLEM]
```

E2 (Annotated).

```
Solve this problem.  Annotate each line explaining why.

Problem:  [PROBLEM]
```

E3 (Contrastive).

```
Do not solve.  Explain:  (a) key insight; (b) common mistakes; (c)
why wrong.

Problem:  [PROBLEM]
```

E4 (Pedagogical).

```
Tutor a novice step-by-step, simple language, no jargon, no code.

Problem:  [PROBLEM]
```

Skill prompt. The source uses the exact line breaks and punctuation shown here; `{problem}` expands to the problem body followed by the answer-format line.

```
Solve the following problem.  Return only the final answer /
function, with no explanation.

Problem:
{problem}
```

Receiver prompt. The exact template, including its two blank lines, is:

```
A programmer provided guidance.  Use it to solve the problem.

Guidance:
{explanation}

Problem:
{problem}
```

The template is the same for self-transfer, controls and the empty-guidance baseline.

D Domain-Specific Evaluation Details

Our evaluation pipeline dispatches by problem answer type (discovery: scorer v4.0, `evaluate.py`; held-out: scorer v4.1, `evaluate_v41.py`):

- **HumanEval/MBPP**: Execute generated code against provided test cases in a subprocess with 10-second timeout. Score is 1.0 if all tests pass, 0.0 otherwise.
- **GSM8K**: The reference answer follows "####". For discovery, score the first number after the last response "####", otherwise after the last boxed expression or the last explicit answer statement, falling back to the last number. Held-out v4.1 prioritizes the latest explicit answer statement. Both use numeric tolerance 10^{-6}.
- **MATH-500**: Use the last boxed answer; absent a box, short responses (at most 60 characters) can be scored directly. Held-out v4.1 also checks the final explicit "answer is" statement and uses the last nonempty box. Normalize LaTeX and compare strings or numeric values.
- **ARC/MMLU**: Extract letter choice (A–D) from response using regex patterns handling formats like "(C)", "The answer is C", bare "C". Compare with reference.
- **BBH (eight subtasks)**: Four subtasks with lettered options (date understanding, disambiguation QA, logical deduction, and shuffled-object tracking) use the letter extractor. Boolean expressions (True/False), causal judgement, navigate, and web of lies (Yes/No) use the leading answer word or the word after "answer is".
- **HellaSwag, WinoGrande, LogiQA 2.0**: Each is presented with lettered options (four, two, and two) and scored using the multiple-choice letter extractor; a bare option text without its letter does not receive credit.
- **TruthfulQA**: Score the single lettered MC1 answer choice using the choice extractor.
- **Other tasks**: Use task-specific answer extractors (code execution, numeric/math normalization, multiple-choice, or short-string matching); unparseable answers do not receive permissive substring credit.

E Answer Filtering Details

Before presenting explanations to receiver models, we apply `strip_v4` from `pipeline/strip.py` to every explanation. This rule removes Markdown and list markers before matching three case-insensitive answer-announcement patterns, then drops lines containing `\boxed`, lines beginning with `####`, and answer-only lines for the question type. It also deletes embedded sentences announcing a final answer. The procedure has no fallback, and short or empty outputs are passed on unchanged (3.6% contain fewer than 20 characters; 3.4% are empty).

F Transfer Heatmap

Figure 4 visualizes the discovery teacher-by-receiver transfer scores for all 40 teachers, ordered by aggregate receiver gain. Each row is a teacher and each shaded cell a receiver; identity-matched cells are blanked only in this visualization. Discovery aggregate statistics, including Table 7, retain identity-matched observations, whereas held-out analyses exclude them.

G Per-Explanation-Type Transfer Results

Table 7 shows transfer scores broken down by explanation type and receiver. For the SmolLM2-360M-Instruct receiver, E2 is highest and E3 lowest. E4 has the highest mean across the five receivers.

H Computational Resources and Scale of Experiments

Hardware. Discovery and held-out runs used two 8×NVIDIA B200 GPU nodes (183,359 MiB, approximately 179 GiB per GPU), with multi-GPU execution for large checkpoints. The reported 1,300-problem counts in Table 8 describe the discovery pipeline; the held-out receiver passes are separate.

Table 7: Average transfer score by explanation type and receiver model.

Explanation Type	SmolLM2 360M (I)	Qwen2.5 0.5B	Qwen2.5 1.5B	Llama-3.2 3B	Qwen2.5 7B
E1 (Declarative)	0.313	0.389	0.478	0.491	0.572
E2 (Annotated)	0.331	0.392	0.483	0.488	0.564
E3 (Contrastive)	0.288	0.353	0.467	0.482	0.564
E4 (Pedagogical)	0.323	0.396	0.492	0.502	0.580

Scale. Table 8 summarizes the recorded model calls underlying the analyzed discovery main protocol (not judge calls, superseded re-runs, budget studies, frontier-only analyses, distillation runs or held-out experiments). Discovery and main-protocol held-out skill, self-transfer and receiver responses use greedy decoding (up to 256 new tokens; 512 for code tasks HumanEval, MBPP and APPS). Explanations are sampled at temperature $= 0.7$, top-$p = 0.95$, top-$k = 50$, with a per-request seed and at most 512 new tokens. Discovery sampling uses seed `20261027` (`pipeline/problems.py`); held-out sampling uses seed `20261028`. Generation assigns each request a reproducible 32-bit seed by interpreting the first eight hex digits of SHA-256 applied to a pipe-delimited string of model name, problem ID, explanation type and any other request-specific components (`pipeline/gen_common.py`, `seed_of`). For held-out explanations, the 22 discovery-roster teachers generated up to 512 tokens, while the 18 additional teachers generated up to 2,048 tokens; in both cases receivers were given at most 512 explanation tokens. All reported held-out results come from a separate receiver pass in which all five receivers decoded greedily for up to 1,024 tokens for all 40 teachers and the same-pass empty-guidance baseline; the main-protocol held-out receiver responses are not used. The 1,024-token allowance is the receiver output budget, not the explanation input budget. A model call here denotes one prompted generation, not one token-level forward pass.

Table 8: Model calls underlying the analyzed discovery main protocol.

Phase	Model Calls	Description
Skill (pass@1)	52,000	40 models $\times$ 1,300 problems
Explanation	208,000	Four explanation generations (E1–E4) per teacher–problem pair
Transfer	1,040,000	Transfer evaluations across five receivers and four styles
Self-transfer	208,000	Teacher self-transfer using cached skill responses
Baselines	520,000	Answer-only and shuffled control evaluations
No-guidance	6,500	Empty-guidance baseline, one per problem and receiver
Total	**2,034,500**	

Wall-clock time. The logged discovery generation run on one 8$\times$B200 node took approximately 2.1 hours (about 17 GPU-hours). Held-out generation, response scoring, analysis, and other runs were conducted separately; a single end-to-end wall-clock total was not recorded.

I Code and Data Availability

All benchmark datasets used in this study are publicly available, open-source datasets, listed and cited in Appendix P; no proprietary benchmark dataset was used. The accompanying supplementary archive provides evaluated discovery and held-out problem records, scored teacher and receiver outputs, explanation inputs, aggregate results, and analysis scripts for the reported tables and figures. Large response tables use compressed JSON Lines. The `code/README.md` gives local reproduction instructions, and the manuscript source, bibliography, and figure assets are supplied separately. Code and results will also be hosted at https://github.parsec.apple.com/sghasemlou/MLR (Apple-hosted; access may be restricted).

J Scaling and Length Sensitivities

As a check on the main articulability analyses, Table 9 reports the rank association between parameter count and baseline-adjusted gain per word in discovery subsets. The result remains positive across the listed overlapping subsets, illustrating that the baseline-adjusted metric measures a different property from the receiver-score-per-word ratio S/L. These comparisons complement the held-out scale plateau and the fixed-cell length effect; none by itself implies a universal best teacher size.

Table 9: Discovery sensitivity: Spearman correlation of log-parameter count with gain per word. Subsets overlap and are descriptive; the 322-problem row uses standalone skill-prompt scores to define the subset, not no-guidance scores.

Subset	Problems	ρ (gain/word)
All discovery	1,300	0.789
No receiver correct in skill prompt	322	0.784
Execution-only	400	0.819
Multiple-choice-only	600	0.767
Without MMLU/HellaSwag	1,100	0.796

Table 10: Held-out study (1,248 problems; 40 teachers), scorer v4.1: average receiver gains by size tier (percentage points), with matched receiver-pass baselines.

Tier	Teachers	Gain	Gain per 100 words
Small	17	2.695	0.488
Mid	13	14.513	5.469
Frontier	10	14.845	5.521

Table 11: Held-out teacher roster by total-parameter tier (17 small, 13 mid, 10 frontier); 22 models also appear in discovery (instruct and code/math-instruct variants), and 18 new models extend the roster to 1,026B; discovery base, code-base and reasoning-distilled teachers are absent.

Tier	Teachers
Small (17)	Llama-3.1-8B-Instruct, Llama-3.2-3B-Instruct, Ministral-8B-Instruct-2410, Phi-3.5-mini-instruct, Qwen2.5-0.5B-Instruct, Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, Qwen2.5-Coder-0.5B-Instruct, Qwen2.5-Coder-1.5B-Instruct, Qwen2.5-Coder-3B-Instruct, Qwen2.5-Coder-7B-Instruct, Qwen2.5-Math-1.5B-Instruct, Qwen2.5-Math-7B-Instruct, Qwen3-4B-Instruct-2507, SmolLM2-1.7B-Instruct, SmolLM2-360M-Instruct
Mid (13)	ERNIE-4.5-21B-A3B-PT, Llama-3.1-70B-Instruct, Llama-3.3-70B-Instruct, Mistral-Small-24B-Instruct-2501, Moonlight-16B-A3B-Instruct, Qwen2.5-14B-Instruct, Qwen2.5-32B-Instruct, Qwen2.5-72B-Instruct, Qwen2.5-Coder-14B-Instruct, Qwen2.5-Coder-32B-Instruct, Qwen2.5-Math-72B-Instruct, Qwen3-30B-A3B-Instruct-2507, phi-4
Frontier (10)	DeepSeek-V3-0324, ERNIE-4.5-300B-A47B-PT, GLM-4.5-Air-FP8, GLM-4.5-FP8, Kimi-K2-Instruct, Llama-3.1-405B-Instruct-FP8, Llama-4-Maverick-17B-128E-Instruct-FP8, Llama-4-Scout-17B-16E-Instruct, Mistral-Large-Instruct-2411, Qwen3-235B-A22B-Instruct-2507

K Extended Related Work

Expert blind spot and tacit knowledge. The expert blind spot was identified by Nathan and Petrosino [2003] in educational contexts: teachers with deep subject-matter expertise systematically overestimate students' readiness for formal representations. Hinds [1999] documented the "curse of expertise," showing that experts' predictions of novice performance are systematically biased. At a deeper level, Chi et al. [1981] demonstrated that experts and novices represent problems at

fundamentally different levels of abstraction, and Dreyfus and Dreyfus [1986] argued that expert performance relies on pattern recognition that resists decomposition into explicit rules. Polanyi [1966] formalized this as the *tacit knowledge* thesis: much of what experts know cannot be articulated. Slobin's "thinking for speaking" framework [Slobin, 1996] suggests that the demands of verbal expression shape which aspects of experience speakers encode.

Chain-of-thought reasoning and faithfulness. Wei et al. [2022] showed that chain-of-thought prompting dramatically improves reasoning, and Kojima et al. [2022] demonstrated that even zero-shot "let's think step by step" is effective. However, a growing body of work questions whether these reasoning chains are *faithful* to the model's actual computation. Turpin et al. [2023] found that chain-of-thought explanations can be systematically unfaithful. Lanham et al. [2023] developed metrics for measuring faithfulness. Radhakrishnan et al. [2023] showed that question decomposition can improve faithfulness, while Lyu et al. [2023] proposed methods for constructing faithful chains of thought. Ye and Durrett [2022] demonstrated the unreliability of explanations in few-shot prompting, and Wiegreffe et al. [2021] proposed methods for measuring the association between labels and free-text rationales. Kadavath et al. [2022] showed that models have partial but imperfect self-knowledge. Ribeiro et al. [2016] introduced locally faithful surrogate explanations, while Jacovi and Goldberg [2020] distinguished plausibility from faithfulness.

Sycophancy and alignment. Sharma et al. [2023] identified sycophancy as a systematic failure mode of RLHF-trained models. The RLHF training paradigm [Ouyang et al., 2022, Bai et al., 2022] optimizes for human-preferred outputs. Tian et al. [2024] proposed fine-tuning for factuality as a partial remedy.

Knowledge distillation and reasoning transfer. Classical knowledge distillation [Hinton et al., 2015] transfers soft probability distributions. More recent approaches use natural language: Ho et al. [2022] showed rationales can teach smaller models, Magister et al. [2023] demonstrated chain-of-thought distillation, Mukherjee et al. [2023] used complex explanation traces, and Hsieh et al. [2023] showed step-by-step distillation can outperform larger models. Shridhar et al. [2023] and Fu et al. [2023] explored distilling reasoning capabilities. Kim et al. [2023] collected chain-of-thought data. Gudibande et al. [2023] cautioned that imitation produces superficial transfer, and Stanton et al. [2021] questioned whether distillation transfers the right knowledge. West et al. [2024] identified the "generative AI paradox."

Self-improvement and self-consistency. Zelikman et al. [2022] introduced STaR. Wang et al. [2023a] showed self-consistency improves reasoning. Huang et al. [2024] demonstrated LLMs cannot reliably self-correct reasoning. Chen et al. [2024] showed self-debugging improves code generation. Yuan et al. [2024] and Sun et al. [2023] explored self-rewarding and self-alignment. Singh et al. [2024] scaled self-training. Burns et al. [2023] studied weak-to-strong generalization.

Scaling laws and evaluation. Kaplan et al. [2020] and Hoffmann et al. [2022] established scaling laws. Clark et al. [2022] extended to mixture-of-experts. Li et al. [2024] showed that selecting from many sampled answers reveals substantially more mathematical problem-solving capability than first-answer accuracy alone. Zheng et al. [2023] introduced LLM-as-judge evaluation. Min et al. [2022] rethought the role of demonstrations. Wang et al. [2023b] explored instruction tuning. Liu et al. [2024] showed models struggle with long contexts. Jiang et al. [2023] and Chevalier et al. [2023] developed prompt compression methods.

L Lexical Alignment by Teacher Type

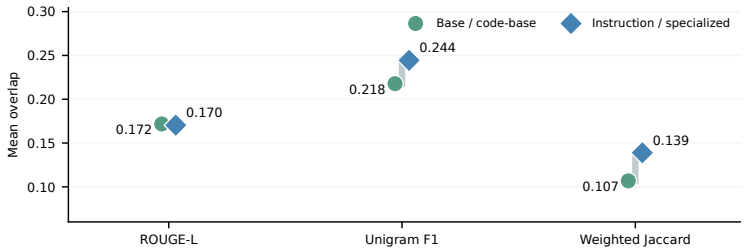


Figure 9: **Lexical explanation–solution alignment.** Mean ROUGE-L, unigram F1, and weighted Jaccard overlap between teachers' raw E1 explanations and their own skill-prompt answers, over 1,300 problems per teacher. The groups comprise 12 base/code-base teachers and 28 instruction/specialized teachers (including reasoning-distilled). Weighted Jaccard multiplies set Jaccard on stopword-filtered words by $0.5 + 0.5 \min(|A|, |B|)/\max(|A|, |B|)$; ROUGE-L F1 and unigram F1 use the first 400 lowercase alphanumeric tokens. Direction and magnitude differ by metric; lexical overlap is not an independent measure of pedagogical usefulness.

M LLM-as-Judge Evaluation Protocol

Two independently run open-weight judges, gpt-oss-120b (low reasoning effort) and Gemma-3-27B-it, greedily rated the raw E1 explanation for each correct teacher–problem pair (15,535 of 52,000; all ratings parsed). The prompt provided the problem, reference answer (or a unit-test pass note for HumanEval/MBPP/APPS), teacher solution, and explanation. Each judge returned 1–5 ratings of textual correspondence and explanation correctness. Correspondence is the measure reported here, not a test of hidden reasoning faithfulness. Across judges, Spearman $\rho = 0.837$ for correspondence and 0.823 for correctness. Pair-weighted means are 2.11/2.22 for 12 base and code-base teachers (2,172 pairs), versus 3.93/4.23 for 28 instruction-tuned, specialized and reasoning-distilled teachers (13,363 pairs), in gpt-oss/Gemma order.

N Top Teachers and Per-Receiver Breakdown

Table 12 lists the ten highest-ranked models by average transfer score. The top teacher (DeepSeek-R1-Distill-Llama-70B, mean transfer score 0.595) is followed by a mixture of 14B–72B instruction, code, math and reasoning-distilled models, showing that rankings depend on training objective as well as scale. Table 13 shows per-receiver averages across 5 receivers.

Table 12: Top 10 teacher models by average transfer score.

Model	Params	Skill	Avg Transfer
DeepSeek-R1-Distill-Llama-70B	70B	0.025	0.595
DeepSeek-R1-Distill-Qwen-32B	32B	0.033	0.590
Qwen2.5-Coder-32B-Instruct	32B	0.667	0.579
Qwen2.5-32B-Instruct	32B	0.697	0.578
Qwen2.5-72B-Instruct	72B	0.712	0.578
phi-4	14B	0.646	0.574
Qwen2.5-14B-Instruct	14B	0.632	0.569
DeepSeek-R1-Distill-Qwen-14B	14B	0.056	0.565
Qwen2.5-Math-72B-Instruct	72B	0.335	0.557
Qwen2.5-Coder-14B-Instruct	14B	0.639	0.556

The complete 40-teacher ranking is shown in main-text Figure 4.

Table 13: Per-receiver transfer averages across 5 receivers.

Receiver	Parameters	Avg Transfer
SmolLM2-360M-Instruct	360M	0.314
Qwen2.5-0.5B-Instruct	0.5B	0.383
Qwen2.5-1.5B-Instruct	1.5B	0.480
Llama-3.2-3B-Instruct	3B	0.491
Qwen2.5-7B-Instruct	7B	0.570

O Self-Transfer

Figure 4 shows each model's self-transfer delta (Δ) across all four explanation types. Self-transfer Δ ranges from -12.6 pp (Qwen2.5-7B) to $+11.6$ pp (Qwen2.5-Math-72B-Instruct), with 20 of 40 teachers within ± 4 pp and $r(\text{skill}, \Delta) = 0.41$. Five of six reasoning-distilled models obtain $+1.5$ to $+5.7$ pp from their own explanations; DeepSeek-R1-Distill-Qwen-32B is approximately unchanged.

P Benchmark Dataset Breakdown

We use a 1,300-problem sample drawn from 13 benchmark sources across four domains: code, mathematics, reasoning, and knowledge. We sample 100 problems per benchmark source, yielding equal source weighting independent of the source pool sizes. The suite draws on HumanEval [Chen et al., 2021], MBPP [Austin et al., 2021], APPS [Hendrycks et al., 2021a], CRUXEval [Gu et al., 2024], GSM8K [Cobbe et al., 2021], MATH-500 [Hendrycks et al., 2021c, Lightman et al., 2024] and MMLU [Hendrycks et al., 2021b], ARC [Clark et al., 2018], BBH [Suzgun et al., 2022], HellaSwag [Zellers et al., 2019], LogiQA 2.0 (NLI) [Liu et al., 2023], WinoGrande [Sakaguchi et al., 2020], and TruthfulQA [Lin et al., 2021]. The following table shows the full per-dataset breakdown of our 1,300-problem sample.

Table 14: Benchmark suite: 1,300 problems sampled from 13 benchmark sources across 4 domains.

Domain	Benchmark	Pool	Sampled	Evaluation
Code	HumanEval	164	100	Code execution
	MBPP	500	100	Code execution
	APPS (introductory)	1,000	100	Code execution
	CRUXEval	800	100	Code execution
Math	GSM8K	1,319	100	Number extraction
	MATH-500	500	100	LaTeX normalization
Reasoning	ARC-Challenge	1,172	100	Choice matching
	BBH (8 subtasks)	1,937	100	Letter/word matching
	HellaSwag	10,042	100	Choice matching
	LogiQA 2.0 (NLI)	3,238	100	Choice matching (2 options)
	WinoGrande	1,267	100	Choice matching
Knowledge	MMLU	14,042	100	Choice matching
	TruthfulQA (MC1)	817	100	Choice matching
	Total		**1,300**	

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately state the core empirical claims about the articulability ceiling, transfer efficiency, and observed explanation–solution correspondence, and these are supported by the reported results.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper includes a dedicated Limitations subsection in Section 5 that bounds the scope of the claims, including benchmark subset sizes, the five-receiver range, the 512-token explanation budget, and the limits of textual correspondence as evidence of mechanistic faithfulness.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper is empirical and does not present new theoretical results that require formal assumptions or proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Section 3 and the appendix provide the model lists, prompts, benchmark composition, evaluation rules, discovery and held-out sampling seeds, request-seeding rule, and compute details needed to reproduce the experimental results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The accompanying supplementary archive directly provides scored discovery and held-out inputs, analysis scripts, computed results, and figure-generation code, with commands in `code/README.md` to reproduce the reported analyses from stored responses. Re-running original large-scale model generation requires separate model access and compute; the Apple-hosted code and results repository listed in Appendix I may require authorization.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The methodology and appendix specify the teacher and receiver models, benchmark sample, prompting setup, answer filtering, evaluation procedures, and generation settings used in the experiments.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The paper reports bootstrap confidence intervals, problem-clustered regression intervals and figure error bars where quantitative comparisons support the main claims.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix H reports hardware, GPU allocation, approximate wall-clock time, and the scale of the logged model calls.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work evaluates publicly available models on benchmark problems without human subjects or private data collection, and Section 5 includes a Broader impacts paragraph describing responsible uses and risks of inaccurate or answer-revealing guidance.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: The Discussion includes a Broader impacts paragraph identifying benefits for in-context tutoring under explanation budgets and risks from misleading or answer-revealing explanations.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release a new high-risk generative model or scraped dataset; it analyzes existing public models and releases experimental artifacts rather than a new deployable model.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

Answer: [No]

Justification: The benchmark sources are publicly available and cited in Appendix P, but the paper and accompanying archive do not inventory license terms for every third-party benchmark and model. Availability of those assets does not establish permission to redistribute all of them.

Guidelines:

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

13. **New assets**

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

Answer: [No]

Justification: Appendix I and the supplementary `code/README.md` document the analysis inputs, scripts, outputs, and reproduction commands, but do not constitute complete standalone documentation of every released asset, its provenance, terms, and limitations.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: The paper does not involve human subjects, so IRB review and participant-risk disclosure are not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: LLMs are the core object of study in the experiments, and the paper explicitly describes their role throughout the methodology and results, including the LLM-as-judge protocol in Appendix M.

    Guidelines:

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