Small-Model Linguistic Reasoning under Compute Constraints

Da-Chen Lian[1]* **Chunki Lim**[1] **Kai-Yang Peng**[1]
Nai-Wei Lu[2] **Pei-Yang Lin**[3] **Shu-Kai Hsieh**[1]
[1]National Taiwan University, Taiwan
[2]Independent Researcher, Taiwan
[3]Kang Chiao International School Linkou Campus, Taiwan
{d08944019,r14142001,r14142015,shukaihsieh}@ntu.edu.tw
weiweiaakid@gmail.com, peiyanglin2027@gmail.com

Abstract

Linguistics Olympiad problems require inferring linguistic rules from a few examples. We study which training and inference methods help a 4B open-weights model solve such problems under the compute constraints of the IOL-AI 2026 challenge, which allows one T4 GPU and 30 minutes. We generated solutions to 175 problems with a frontier LLM; former national-team contestants reviewed 150 and flagged flawed reasoning far more often than wrong answers, revealing a gap between answer correctness and acceptable linguistic reasoning. Trained on full-length solutions, the model could not finish its answers within the generation limit, while a model trained on shortened solutions scored similarly to full-length reasoning at a fraction of the length. We also tested multilingual and in-domain training data, voting and three variants of it, and grammar retrieval; none improved consistently across evaluations. Our 4B system placed fourth of 48 teams, behind three teams that all used a 14B model.

Keywords: Linguistics Olympiad, small language models, reasoning distillation, test-time compute, low-resource languages, benchmark transfer

1 Introduction

Linguistics Olympiad (LO) problems[1] present a small set of translated words or sentences from an unfamiliar language and ask solvers to infer the underlying rules well enough to translate, complete, or match new items. Because many problems involve low-resource or endangered languages and because answer keys are not widely represented in ordinary training data, LO benchmarks have become useful probes of linguistic reasoning in language models (Şahin et al., 2020; Bean et al., 2024; Sánchez et al., 2025; Goyal and Dan, 2025; Lian et al., 2025a). This advantage may not last. As training corpora grow, prior exposure to benchmark material or the underlying languages becomes more plausible, and measurable exposure effects have already been reported (Khouja et al., 2026).

Most previous work evaluates frontier models with generous inference budgets. Methods that help these models may not have the same effect on smaller models. We examine whether detailed solutions generated by frontier models provide useful training examples for small models, and whether voting or linguistic reference material helps them solve new puzzles. We ask: **Which training and inference methods help a small model solve linguistic puzzles under compute constraints?** We study this question in the IOL-AI 2026 setting,[2] which provides an offline sandbox, one T4 GPU, and a 30-minute limit (§3).

We tested four types of methods, each aimed at a different problem. Fine-tuning trains the model on step-by-step solutions (traces) or instruction-following examples; shortening the traces tests whether this training still helps when the model's answers must fit the generation limit. Test-time voting and its refinements try to obtain more reliable answers from repeated samples. Grammar retrieval supplies linguistic reference material that might help the model infer linguistic rules. Output handling and time-budget changes, such as finishing incomplete answers and stopping generation before the deadline, aim to produce answers in the required format before time runs out.

The study makes three contributions.

Shortened training solutions scored similarly to full-length ones while fitting the generation

* Corresponding author.

[1]https://ioling.org/problems/

[2]https://iolai.org/

Proceedings of the 38th Conference on Computational Linguistics and Speech Processing (ROCLING 2026), pages 214–224
November 6-7, 2026 ©2026 Association for Computational Linguistics

limit. Trained on full-length solutions, the 4B model answered none of 27 local inputs within 2,048 tokens, and answered 24 only when allowed 14,336, far more than the 30-minute limit permits. A single model trained on three lengths of the same reviewed solutions, each with its own length instruction, scored 0.1317 at the shortest length within 2,048 tokens and 0.1503 at full length within 14,336, a difference within the noise of 27 inputs. This model was not scored in the competition, and we did not test whether the reviewers' corrections improved it (§4, §6.2).

We generated detailed solutions to 175 LO problems with a frontier LLM and obtained expert review for 150 of them. Former national-team contestants flagged flawed reasoning in 48 of the reviewed problems, 45 of which had no answer marked wrong, revealing a gap between answer correctness and acceptable linguistic reasoning that matters when such solutions are used as training material (§4).

A 4B model placed fourth of 48 teams, ahead of teams using larger models. All three higher-placed teams submitted Qwen2.5-14B-Instruct, the organizers' baseline model, and most teams below us whose submission repositories we could locate used 7B to 14B models, like the organizers' 14B baseline. Our Qwen3.5-4B system,[3] fine-tuned on task-specific answers and multilingual instruction data, scored 0.1604 on the private split against 0.1767 for third place, a gap well within the movement between the public and private leaderboards (§6, §7). We also report comparisons of voting, multilingual data, and grammar retrieval, none of which improved consistently, and how much leaderboard scores moved between the public and private leaderboards. The accompanying code is maintained at `https://github.com/lopentu/small-model-linguistic-reasoning`. Links to the other archived submissions appear in Appendix A.

2 Related Work

LO benchmarks. PuzzLing Machines (Şahin et al., 2020) introduced Rosetta-stone puzzles for machine reasoning. LingOly (Bean et al., 2024) scales to 1,133 problems over more than 90 mostly low-resource languages; Linguini (Sánchez et al., 2025), whose format IOL-AI 2026 adopts, contains 894 questions over 75 languages; IOL-BENCH (Goyal and Dan, 2025) digitizes the IOL archive; and modeLing (Chi et al., 2024) introduces novel problems to reduce contamination. LOBSTER (Lian et al., 2025a) adds structured rule-level evaluation. LLMs still perform poorly on puzzles that high-school contestants largely solve, with failures associated with morphological complexity and with linguistic features absent from English (Choudhary et al., 2025).

Reasoning methods. Methods reported to improve performance include analogical prompting (Ramji and Ramji, 2025), induction-then-application with self-consistency (Garnham and Shareghi, 2026), and iterative refinement (Lian et al., 2025a,b); multi-agent variants help mainly when a weaker aggregator is paired with a stronger solver (Lian et al., 2025a). Tree-of-thought prompting underperformed standard prompting with GPT-3.5 (Lin et al., 2023). These studies use frontier or mid-size models and do not test whether those improvements persist in independent evaluations.

Small-model fine-tuning and linguistic augmentation. Small curated sets of long reasoning traces can unlock reasoning in 32B models (Muennighoff et al., 2025; Ye et al., 2025), but distillation can hurt smaller models when the training data are poorly matched to their capacity (Xu et al., 2025). Other work shortens the reasoning used for training. TokenSkip (Xia et al., 2025) deletes low-importance tokens at several compression ratios and conditions the model on the ratio, and C3oT (Kang et al., 2025) trains on long and GPT-4-compressed reasoning under separate condition markers. Our length tiers follow the same pattern, with deletion guided by a written specification and applied to expert-reviewed olympiad solutions under a fixed time budget. Evidence on grammar augmentation is also mixed. On MTOB, a benchmark for translating a language from a grammar book (Tanzer et al., 2024), follow-up work attributes most of the improvement to the book's parallel examples, with grammatical explanations contributing little (Aycock et al., 2025). Typological resources make related languages easy to find (Littell et al., 2017), so grammar retrieval is simple to build, but LOBSTER found no significant improvement from re-

[3] `https://huggingface.co/richardlian/iolai-2026-4b-mist-oldv2`

trieval of grammars matched to the language by Glottocode (Lian et al., 2025a); we test a related setup on a 26B model locally, mostly using grammars of related languages when the exact language was unavailable. We find no evidence of benefit, but inconsistent outputs for identical prompts limit the comparison.

3 Task and Constraints

Linguistics Olympiad problems are self-contained inference puzzles, in which solvers use the supplied examples to infer the relevant rules without needing to know the language beforehand. For example, a Terêna puzzle in Linguini (Sánchez et al., 2025) gives *ndûti/tiûti* (my/your head), *mbôro/peôro* (my/your pants), and *ndâki/teâki* (my/your arm), among other pairs. It asks for the "your" form of *ndôko* (my nape). The answer, *teôko*, combines a consonant change (*nd* to *t*) with a vowel change (*ô* to *eô*). Solvers must infer both patterns from the examples and apply them together.

Each Linguini (Sánchez et al., 2025) evaluation input contains a self-contained context in an unfamiliar language and a query of one of five types: `translation`, `fill_blanks`, `match_letters`, `text_to_num`, or `num_to_text`. Predictions are JSON lists whose answers follow the query order. Answer completion means producing a final answer that the scorer can read within the token or time limit, but a completed answer can still be wrong. The official score is the geometric mean of exact-match accuracy (EM) and chrF, a character n-gram overlap measure (Popović, 2015), with both components weighted by IOL point values: $G = \sqrt{\mathrm{EM_w} \cdot \mathrm{chrF_w}}$.

The deployment sandbox provides one T4 GPU (16 GB; no bf16 support), no network access, a frozen software image, and 30 minutes for the full evaluation. Teams could make up to ten public submissions per day and later freeze two submissions for the private split; we made 22 public submissions. Public labels were hidden, but repeated leaderboard feedback makes those scores exploratory; they are not independent test results. For local development, we split the Linguini test set (160 inputs, 85 problems, 894 scored items) at problem level into development (114 inputs) and validation (46 inputs) folds and computed bootstrap intervals by resampling whole problems. Local scoring uses the same EM/chrF combination but weights answer items equally because the official point weights were unavailable.

The competition's 90 test items come from the IOL 2026 individual contest, presented in the Linguini format and unpublished during the competition, so they do not overlap the Linguini test set we used for development. The public leaderboard scored a subset of the 90 items, and the private leaderboard scored the rest.

4 An Expert-Reviewed Solution Corpus

4.1 Construction and Review

We built the corpus in three steps. First, Claude Opus 4.6, Claude Opus 4.7, and Claude Sonnet 4.6 generated detailed English solutions for 175 problems: 110 IOL individual problems (2003–2025), 40 APLO problems (2019–2026),[4] and 25 Taiwan Linguistics Olympiad problems (2023–2026).[5] Second, ChatGPT compared each solution with the problem's official solution and reported errors; we kept the reports the official solution supported, and Claude revised the solutions. This step corrected 26 final answers and exposed circular derivations, in which an official answer had been used to justify the rule meant to derive it; we re-derived such rules from the puzzle data or marked the step as unsupported. On the 20 most recent IOL problems (2022–2025), ChatGPT judged all final answers correct but still found over-generalization, internal inconsistency, circularity, and arithmetic errors in the derivations. Third, former national-team contestants reviewed 150 of the 175 solutions; time did not allow review of the remaining 25, and only reviewed solutions were used for training. The reviewers marked span-level edits by issue type, mostly in the reasoning, and we applied them. The 150 problems received 159 reviews with 347 edits (Table 1).

Of the 150 problems, 141 received one review; the revisions were not checked in a separate adjudication step, and we did not measure inter-annotator agreement. Because one edit marks one span and a problem can receive several, the counts describe issues that reviewers flagged and cannot be interpreted as error rates.

[4] https://aplo.asia/problems/

[5] https://ioltw.github.io/zh/student/previous-problems

Issue type	n
wrong_reasoning	77
unclear	67
(unlabelled)	68
other	53
formatting	45
too_verbose	24
wrong_answer	**8**
missing_visual_check	5
total	347

Table 1: Expert span-level edits across the 159 reviews, by issue type. One problem can receive several edits.

4.2 Answers and Derivations

The review annotations document corrections to derivations as well as answers. `wrong_reasoning` was assigned to 77 edits, compared with 8 `wrong_answer` edits. Because the answers had already been checked before expert review, the edit ratio cannot establish how often reasoning errors occur compared with answer errors. The edits nonetheless flag faulty derivations in solutions whose final answers were kept. This matters for distillation, because solutions with correct final answers may still teach faulty linguistic reasoning, and it motivates rule-level evaluation such as LOBSTER's checklists (Lian et al., 2025a).

4.3 Trace Compression

The reviewed traces have a median length of 7.3k tokens (Qwen3.5 tokenizer). To test whether shorter training solutions help the model complete answers within the generation limit, we compressed them by deletion only. The longest tier keeps the trace except for meta-sections such as the problem restatement and summaries, which a script removed. Two shorter tiers follow the terse style of Caveman:[6] at most 55% of the words, and at most 20% with a 900-word cap. For these, GPT-5.6-sol deleted words from each trace following a written specification (Appendix C), and a script rejected any output whose words did not all appear in the original trace in the same order. Each tier has its own length instruction, giving 444 training examples from 148 problems, and each example ends with a short list of the induced rules and the final answers, identical across the three tiers. We use this set in the training comparisons below.

[6] https://github.com/juliusbrussee/caveman

5 System

We prepared the submitted 4B model's training mixture from task-specific direct answers and multilingual instruction examples selected from existing datasets; it did not use the expert-reviewed reasoning traces, although its direct answers include those of the reviewed problems. Our main system uses supervised fine-tuning (SFT) for Qwen3.5-4B[7] with 16-bit LoRA (Hu et al., 2022) (rsLoRA, Kalajdzievski, 2023; $\alpha = 2r$, rank 32, completion-only loss, sequences of up to 12,288 tokens). Merged weights are exported to an 8-bit GGUF (5.0 GB VRAM, 40 tok/s on a T4). At deployment, the model is prompted to output only the JSON answer list. The first answer is a greedy decode, up to four more are sampled at temperature 1.0 (top-p 0.95), and an item-level majority vote combines them, with ties going to the greedy answer. The 26B mixture-of-experts (MoE) comparison system (Gemma-4-26B-A4B; 14.7 GB resident) answers each problem once with greedy decoding, using a prompt that asks for the linguistic rules before the answers. If automatic checks find a formatting problem, such as a missing or repeated answer, the model is asked once to revise. Appendix B lists the other base models we evaluated.

The 4B system writes its answers to the submission file after each batch, filling any unanswered item with a word from the problem text, and stops generating at 85% of the 30-minute limit.

6 Results

6.1 Leaderboard Outcome

We selected two submissions for the private leaderboard: the 4B model trained on direct answers and multilingual data, and the 26B system (§5). The 4B system scored 0.1604 on the private split and placed fourth among 48 teams.[8] On the public leaderboard, the 4B scored 0.1451 and the 26B 0.1616; the 26B scored 0.1507 privately. The 4B final used temperature 1.0; §6.2 compares it with a model trained on direct answers alone under the same settings. The winning system scored 0.2245 publicly and 0.1979 privately. The 4B–26B private gap is only 0.0097, much smaller than the observed between-split movement (§7); we

[7] https://huggingface.co/Qwen/Qwen3.5-4B

[8] Counting individual submissions, we rank 5th, because one team has two submissions above ours.

Training data	Local	Public
none (untuned)	0.0351	–
direct answers only	–	0.1206
full-length traces	no answers	–
compressed traces	0.1317	none valid
direct + multilingual	–	**0.1440**
direct + Linguini	–	0.1188
direct + Linguini + multilingual	–	0.0797

Table 2: Fine-tuning results for the 4B model. Local scores come from an evaluation set of 27 inputs whose problems were excluded from training. They use greedy decoding with a 2,048-token limit; we report the best of six saved models, chosen on this same set. The model trained on compressed traces was submitted without its trained weights, so it has no valid leaderboard score. Public scores use temperature 0.7.

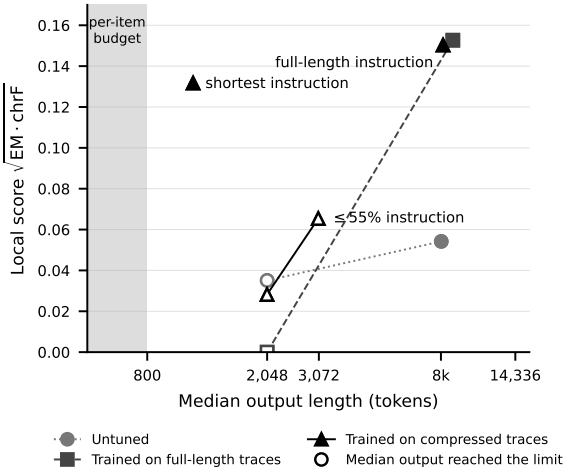

Figure 1: Local score of the 4B against median output length on the 27-input evaluation set. Hollow markers show runs whose median output reached the generation limit, so most answers were cut off. The shaded region marks the approximately 800 tokens per item that the 30-minute limit allows. Labels give the length instruction used with the model trained on compressed traces.

therefore do not interpret it as evidence that the 4B system is superior. The comparison is also not compute-matched, since the 4B combines up to five answers per problem, while the 26B uses one plus an occasional revision.

6.2 Effects of Fine-Tuning on a 4B Model

First, shortened training solutions improved the 4B local result (Table 2). The 4B model fine-tuned on full-length traces produced no scorable answers for any of the 27 evaluation inputs at a generation limit of 2,048 tokens. This limit is already about 2.5 times the deployment budget, because at the measured speed of 40 tokens per second the 30-minute limit permits approximately 800 tokens for each of the 90 test items. With delete-only compression, fine-tuning on the compressed traces raised the 4B score from 0.0351 to 0.1317 and answer completion from 10/27 to 24/27 on an evaluation set of 27 inputs from 11 problems excluded from training. We report the best of six saved models (two LoRA ranks, three epochs each), chosen on this same set; the other five scored 0.057–0.101, all above the untuned model. Because each length tier had its own instruction, the same model can be asked for each length. With the full-length instruction and a limit of 14,336 tokens it scored 0.1503, but its answers had a median of 8.1k tokens, far beyond the time budget, and the $\leq 55\%$ instruction scored lower at both 2,048 and 3,072 tokens (Figure 1). The 26B model trained on the same set scored 0.0968 and 0.0399 in two configurations, against 0.049–0.084 in three runs of the untuned 26B. Figure 1 plots the 4B scores against output length. For 26B checkpoints trained on full-length solutions, increasing the generation limit to 14,336 tokens still yielded no scorable answers. Their outputs included headings naming problems from the training data and repeatedly generated the same text without giving final answers. Being cut off at the generation limit before reaching the answers therefore does not explain their failure. The outputs suggest memorization of the training format; with one model of each size, we cannot tell whether this reflects model size or model family. The full-length-trained 4B did answer at this limit, completing 24/27 with a score of 0.1527 against 0.0542 for the untuned 4B, so its failure at 2,048 tokens came from being cut off. Outputs of this length, however, do not fit the 30-minute budget. The compressed-trace 4B never received a valid leaderboard score, because the learned fine-tuning updates, stored in a LoRA adapter, were not merged into the base model before submission, leaving the submitted weights untuned. The adapter was trained and merged with different software libraries, which name the model's layers differently, so the merge found no matching layers and completed with only a warning. A correctly merged variant trained to write reasoning before its answers scored 0.0 publicly, with no exact matches and a chrF of 0.1718, because its reasoning-length outputs did not fit within the 30-minute limit.

Inference settings	Score G Direct	+Multi.	Exact match Direct	+Multi.
$T = 1.0$	0.1453	0.1451	0.0935	0.0968
$T = 0.7$	0.1206	0.1440	0.0644	0.0968

Table 3: Adding multilingual data on the public split. Within each row, both models share the same settings. The rows differ in sampling temperature; runs were unseeded. The 4B final used the multilingual model at $T = 1.0$.

Second, multilingual data scores higher at only one of two sampling temperatures. We compared a 4B model fine-tuned on task-specific direct answers alone with one fine-tuned on the same answers mixed with 956 multilingual instruction examples. The direct-only weights came from a separate, earlier training run, merged after the layer names were corrected. That merge still skipped the model's linear-attention layers, so only 128 of the 248 weight matrices updated in training changed in the submitted model. We selected the multilingual examples from MIST, retaining Belebele, Aya, and TyDiQA sources, and formatted their answers as JSON lists. The public scores are shown in Table 3.

On the public split at $T = 0.7$, adding multilingual data raises G from 0.1206 to 0.1440 and EM from 0.0644 to 0.0968, but at $T = 1.0$ the two scores are almost identical: 0.1453 and 0.1451. The multilingual model's EM is 0.0968 in both rows, while the direct-only model's EM falls from 0.0935 to 0.0644. Adding multilingual data showed no public-score improvement under the fourth-place system's settings. Its effect on the private split remains unknown because the direct-only model has no private score. Multilingual data may make the output less sensitive to the sampling temperature, but this remains a hypothesis, because the pattern depends on the low direct-only score at $T = 0.7$, and we have no confidence interval for the score difference between the paired public submissions.

We also trained separate models from the untuned 4B on the same direct answers plus the 114 Linguini development inputs, and on direct answers, Linguini, and multilingual data together. The Linguini model's local score increased sharply (Table 4), but its public score did not improve (0.1188 vs. 0.1206 for the direct-only model), and the model trained on all three sources scored 0.0797. Both models had already been trained on many of the local problems through the direct-answer data they share, and the local runs combined twelve answers per input for the direct-only model and five for the Linguini model. The local increase therefore does not show that Linguini training helped, and we compare the training conditions on public scores.

6.3 Voting When Answer Accuracy Is Low

Self-consistency voting (Wang et al., 2023) produced the largest apparent public improvement: 0.0739 at $N = 1$, 0.1154 at $N = 3$, and 0.1453 at $N = 5$, before declining at larger N. Here N is the number of answers combined by voting; the first is a greedy decode and the rest are sampled at temperature 1.0. Submissions up to $N = 7$ share weights and differ only in N, so the $N = 1$ point is greedy decoding alone; the $N = 12$ submission generated its answers in batches. Each of these scores comes from one public submission scored once, and because we chose later settings based on earlier public scores, they are exploratory.

Locally, we generated twelve answers per development-fold input once (one greedy, eleven sampled) with the same weights and prompt, in batches, and scored each vote size by voting over subsets of these answers (Figure 2). Greedy decoding scores 0.2234; voting over sampled answers scores 0.1732 at one answer (median difference from greedy -0.0553, 95% CI $[-0.1006, -0.0186]$) and 0.2109 at eleven (-0.0160, $[-0.0510, +0.0144]$, within noise). The scores average eight random selections from these answers; the paired bootstrap intervals use only the first selection.

When the vote starts with the greedy answer and adds sampled answers in their saved order, the score stays within 0.009 of greedy at every N and is 0.2211 at $N = 11$ (bootstrap median paired $\Delta G = -0.0029$, 95% CI $[-0.0206, +0.0200]$). In this local comparison, voting improves on a single sampled answer but shows no clear difference from greedy decoding, so it does not reproduce the public increase. The comparisons differ in evaluation data and in how answers were generated, and our measurements do not identify which explains the difference. These comparisons measure accuracy only; we did not measure the runtime cost of larger votes.

None of the voting variants improved consistently. Confidence-weighted voting weights each

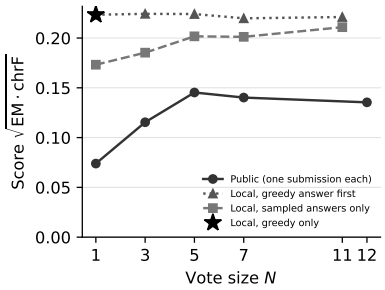

Figure 2: Score by vote size. The public curve (one submission per point; $N = 1$ is greedy decoding; $N = 12$ generated answers in batches) rises to $N = 5$. On saved local outputs from the same weights, voting over sampled answers improves with more samples but remains below greedy. Starting with a greedy answer and adding samples leaves the score essentially unchanged.

Local and public comparisons		
Intervention	Local ΔG	Public ΔG
In-domain SFT[a]	$+0.245$	-0.002
Confidence-weighted vote	$+0.020$	-0.009
Temperature $1.0 \to 0.7$ [b]	$+0.024$	-0.025
Adding votes[d]	-0.003	$+0.071$
Same public split; different sampling temperature		
	$T = 0.7$	$T = 1.0$
Multilingual SFT[c]	$+0.0234$	-0.0002
Local measurement only		
	Local ΔG	Public
Prompt-diversity voting	$+0.015$	not submitted
Grammar retrieval (26B)[e]	-0.057	not submitted

Table 4: Every method evaluated twice, grouped by evaluation setting, and two evaluated once. A positive ΔG favors the intervention over its control. [a]Both models trained on problems that appear in the local inputs, so the in-domain local score increase ($0.2410 \to 0.4855$) may reflect answers seen in training and cannot establish a training effect. [b]Same weights at both temperatures, majority@12 ($0.2206 \to 0.2447$). [c]Multilingual minus direct-only within each row of Table 3. [d]Local: greedy answer first, $N = 1 \to 11$ on saved outputs generated in batches; public: $N = 1 \to 5$ with the same weights. [e]Inconsistent outputs for identical prompts limit what we can conclude about retrieval. All public comparisons are exploratory because the split was queried repeatedly.

sampled answer by the model's confidence in it, computed from its token probabilities, and selects the answer with the largest total weight. It improved locally ($\Delta G = +0.0197$, 95% CI $[+0.0042, +0.0341]$) but scored lower on the public split than unweighted voting at equal N (0.1266 vs. 0.1354). Lowering temperature from 1.0 to 0.7 improved two local runs but scored lower in a public pair whose only active difference was the temperature (0.1206 vs. 0.1453). Voting over prompts with different query-item orders showed no clear local difference ($\Delta G = +0.0154$, 95% CI $[-0.0000, +0.0329]$) and was not submitted.

At roughly 10% item accuracy, voting can help only when the correct answer recurs across samples. On the development fold, 63% of items received no correct answer in any of the twelve samples. Of the remaining items, 69% received it at most three times, and the vote selected it for 38%.

6.4 Answer Completion and Retrieval

Although a system must finish its answers within the time limit and use the required output format, final scores alone do not identify why an answer failed to match. On the private split, nine submissions had EM=0 but non-zero chrF (0.086–0.324); one achieved chrF 0.3244, the highest of any single private submission, yet received a final score of zero. Our own revisions also show large score changes that we cannot cleanly attribute. For the 26B, two submissions with identical weights scored 0.0883 and 0.1616 on the public split, a difference of 0.073, after we changed the prompt, answer parsing, and time management together. Removing the time-management change alone, a second deadline that reserved time to finish incomplete answers, had no detectable local effect, so we cannot say which change produced the difference.

Grammar retrieval likewise showed no evidence of benefit. We indexed 35.7k reference-grammar chunks for 1,098 languages, selected grammars for the puzzle language or, when unavailable, a language from the same family, and injected two excerpts rich in glossed examples. On the 26B system, retrieval reduced score by $\Delta G = -0.057$ (95% CI $[-0.106, -0.014]$). Only 2 of 118 retrievals over the 160 inputs matched the exact language, and outputs often degenerated into meta-commentary. However, the local inference system produced different outputs for identical prompts, and its control score was below a previously recorded baseline. These problems prevent us from isolating retrieval's effect, so we find no evidence of benefit and only suggestive evidence of harm.

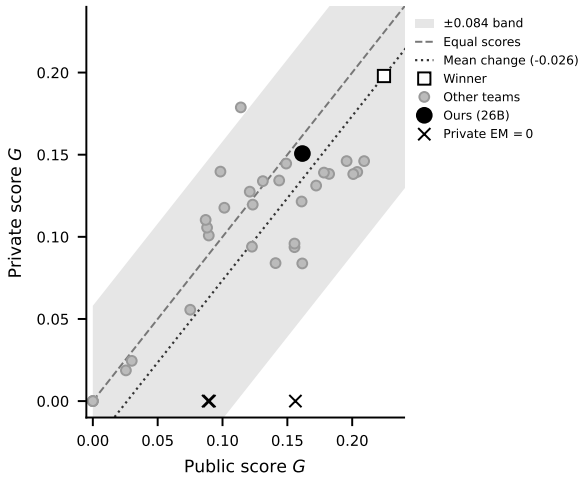

Figure 3: The 35 frozen submissions matched across the public and private leaderboards. The dashed line marks equal scores on both splits, the dotted line the mean score change $\bar{\Delta} = -0.0265$, and the shaded band $\pm 1.96\,\mathrm{SD} = \pm 0.084$. Three systems lost all exact matches on the private split (EM $= 0$, crosses; two overlap near public 0.09) and are excluded from the ± 0.067 band. Scores generally fall on the private split, and rank is only partly preserved ($\rho = +0.668$). Our marked point is the 26B system; the public board lists only each team's best submission, so our 4B final has no matched public row.

7 Discussion

How stable is a small leaderboard? After removing organizer entries, 35 frozen systems could be matched across the public and private leaderboards. Their mean score change is -0.0265, with SD 0.0431, giving an empirical 1.96-SD band of ± 0.084; excluding the three matched systems with EM$\rightarrow$ 0 narrows the band to ± 0.067. The changes differ by metric: EM falls 28% relative while chrF rises 4%. Rankings also change substantially (Spearman's $\rho = 0.668$), with half of the public top 10 leaving the private top 10. Our team moved from 11th publicly (the 26B row) to 4th privately (the 4B); we do not attribute this movement to any identified intervention.

Which improvements were consistent? Table 4 lists every method evaluated twice; none improved both times. Confidence-weighted voting improved the local score by 0.0197, but lowered the public score by 0.0088. Adding multilingual data improved the public score by $+0.0234$ at temperature 0.7, but changed it by -0.0002 at temperature 1.0. The other comparisons answer narrower questions. The public score difference after training on Linguini was -0.0018, and its local increase does not show a training effect because both models had seen many of the local problems. The large public voting difference came from submissions sharing weights, but the local comparison found no clear difference from greedy decoding; the two differ in evaluation data and in how answers were generated. The compressed-trace model has no valid leaderboard score, and retrieval was measured only locally. These comparisons do not establish which methods generally improve performance on new evaluation data.

The fourth-place result does not identify which components helped. The between-split spread in Figure 3 describes movement across the field; it is not a confidence interval for a fine-tuning effect. Local score increases from different training conditions should not be compared directly because they were scored on different local inputs against different controls.

8 Conclusion

We studied which training and inference methods help a 4B open-weights model solve Linguistics Olympiad problems with one T4 GPU and 30 minutes. Expert review of frontier-model solutions found flawed reasoning far more often than wrong answers, so a correct answer is a weak check on a solution used for training. A model trained on full-length solutions could not finish its answers within the generation limit, while a model trained on shortened solutions scored similarly at a fraction of the length; this local result was not tested in the competition. Our 4B system placed fourth of 48 teams, behind three teams that all used a 14B model. No other method we evaluated in two settings, locally and publicly or publicly at two sampling temperatures, improved in both, and scores of the same systems moved within an empirical ± 0.084 band between the public and private leaderboards, so small differences on one leaderboard say little about a method. Future work could test whether the expert corrections themselves improve training, evaluate shortened solutions under the competition's time limit, and train small models to reason concisely, as recent work reports for larger models.[9]

[9] https://bottlecapai.com/post/thinkingcap-qwen3-6-27b/

Limitations

All our results come from one competition, whose 90 test items were divided between the public and private leaderboards. Only our two final submissions received private scores, so no individual method was tested on the private leaderboard. The compressed-trace model had no valid leaderboard submission. Each public voting score comes from one submission, and we chose later settings based on earlier public scores. The local voting comparison generated its answers differently from the submissions. In the in-domain local comparison, the local inputs include problems from the direct-answer training data that both models share, so its score difference does not show improved reasoning.

Expert review has its own limits: 141 of the 150 problems received one review, there was no separate adjudication step to check the revisions, and no inter-annotator agreement study was conducted. We also do not compare pre-review and post-review traces as training data, so the edits document expert-identified defects and corrections without establishing corpus-wide reliability or a causal training benefit. Results concern one small-model family and one task format.

The experiments were run against the competition deadline. We did not run systematic hyperparameter sweeps, and successive submissions often changed more than one setting at a time, such as sampling temperature and vote size, so several comparisons rest on pairs that differ in more than the factor being compared. Implementation errors also affected several runs. Some fine-tuned models were submitted with untuned or partly merged weights, and the reasoning-first model did not finish within the time limit. We report these runs together with their errors, and Appendix A links each submitted configuration.

Ethics Statement

The study uses AI-generated solutions to published olympiad problems and expert review by consenting annotators. Neither the solution corpus nor the annotations are released. Because LO problems draw on real, often endangered languages, model-generated solutions may contain incorrect claims about them. The paper reports only counts of the errors experts found and does not present model-generated claims as descriptions of these languages.

References

Seth Aycock, David Stap, Di Wu, Christof Monz, and Khalil Sima'an. 2025. Can LLMs Really Learn to Translate a Low-Resource Language from One Grammar Book? In *The Thirteenth International Conference on Learning Representations*.

Andrew M. Bean, Simi Hellsten, Harry Mayne, Jabez Magomere, Ethan A. Chi, Ryan Chi, Scott A. Hale, and Hannah Rose Kirk. 2024. LINGOLY: A benchmark of olympiad-level linguistic reasoning puzzles in low resource and extinct languages. In *The Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Nathan Chi, Teodor Malchev, Riley Kong, Ryan Chi, Lucas Huang, Ethan Chi, R. McCoy, and Dragomir Radev. 2024. ModeLing: A novel dataset for testing linguistic reasoning in language models. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 113–119, St. Julian's, Malta. Association for Computational Linguistics.

Mukund Choudhary, KV Aditya Srivatsa, Gaurja Aeron, Antara Raaghavi Bhattacharya, Dang Khoa Dang Dinh, Ikhlasul Akmal Hanif, Daria Kotova, Ekaterina Kochmar, and Monojit Choudhury. 2025. UNVEILING: What makes linguistics olympiad puzzles tricky for LLMs? In *Proceedings of the Second Conference on Language Modeling*.

Jamie Garnham and Ehsan Shareghi. 2026. Could language models win the International Linguistics Olympiad? In *Proceedings of the 30th Conference on Computational Natural Language Learning*, pages 481–500. Association for Computational Linguistics.

Satyam Goyal and Soham Dan. 2025. IOL-BENCH: Benchmarking LLMs on linguistic reasoning. ArXiv:2501.04249.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*.

Damjan Kalajdzievski. 2023. A rank stabilization scaling factor for fine-tuning with LoRA.

Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2025. C3oT: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24312–24320.

Jude Khouja, Lingyi Yang, Karolina Korgul, Simeon Hellsten, Vlad A. Neacsu, Harry Mayne, Ryan Othniel Kearns, Andrew M. Bean, and Adam Mahdi. 2026. LINGOLY-TOO: Disentangling reasoning from knowledge with templatised orthographic obfuscation. In *Proceedings of the International*

Conference on Learning Representations (ICLR). ArXiv:2503.02972.

Da-Chen Lian, Ri-Sheng Huang, Pin-Er Chen, Chunki Lim, You-Kuan Lin, Guan-Yu Tseng, Zhen-Yu Lin, Pin-Cheng Chen, and Shu-Kai Hsieh. 2025a. LOBSTER: Linguistics olympiad benchmark for structured evaluation on reasoning. In *Proceedings of the 37th Conference on Computational Linguistics and Speech Processing (ROCLING 2025)*, pages 193–229, National Taiwan University, Taipei City, Taiwan. Association for Computational Linguistics.

Da-Chen Lian, Ri-Sheng Huang, Pin-Er Chen, Chunki Lim, You-Kuan Lin, Guan-Yu Tseng, Zi-Cheng Yang, Zhen-Yu Lin, Pin-Cheng Chen, and Shu-Kai Hsieh. 2025b. Lingbench++: A linguistically-informed benchmark and reasoning framework for multi-step and cross-cultural inference with llms. ArXiv:2507.16809.

Zheng-Lin Lin, Chiao-Han Yen, Jia-Cheng Xu, Deborah Watty, and Shu-Kai Hsieh. 2023. Solving linguistic olympiad problems with tree-of-thought prompting. In *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*, pages 262–269. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14. Association for Computational Linguistics.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20275–20321. Association for Computational Linguistics.

Maja Popović. 2015. chrF: Character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395. Association for Computational Linguistics.

Raghav Ramji and Keshav Ramji. 2025. Inductive linguistic reasoning with large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22783–22810. Association for Computational Linguistics.

Gözde Gül Şahin, Yova Kementchedjhieva, Phillip Rust, and Iryna Gurevych. 2020. PuzzLing Machines: A Challenge on Learning From Small Data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1241–1254, Online. Association for Computational Linguistics.

Eduardo Sánchez, Belen Alastruey, Christophe Ropers, Arina Turkatenko, Pontus Stenetorp, Mikel Artetxe, and Marta R. Costa-jussà. 2025. Linguini: A benchmark for language-agnostic linguistic reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Garrett Tanzer, Mirac Suzgun, Eline Visser, Dan Jurafsky, and Luke Melas-Kyriazi. 2024. A Benchmark for Learning to Translate a New Language from One Grammar Book. In *The Twelfth International Conference on Learning Representations*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. 2025. TokenSkip: Controllable chain-of-thought compression in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3351–3363, Suzhou, China. Association for Computational Linguistics.

Haoran Xu, Baolin Peng, Hany Awadalla, Dongdong Chen, Yen-Chun Chen, Mei Gao, Young Jin Kim, Yunsheng Li, Liliang Ren, Yelong Shen, Shuohang Wang, Weijian Xu, Jianfeng Gao, and Weizhu Chen. 2025. Phi-4-mini-reasoning: Exploring the limits of small reasoning language models in math. ArXiv:2504.21233.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. LIMO: Less is more for reasoning. In *Proceedings of the Second Conference on Language Modeling*.

A Archived Models and Submissions

Table 5 links to archived model weights and submission code on Hugging Face. Each link points to the submitted revision. The training mixtures labeled multilingual and Linguini include task-specific direct-answer examples (§5).

Archive	Configuration	Public G
26B	Comparison system	0.1616
vote5	Direct only; original code	0.1453
4B final	Direct + multilingual; original code	0.1451
e36mist	Direct + multilingual; revised code	0.1440
vote7	Direct only; original code; $N = 7$	0.1402
vote12	Direct only; batched; $N = 12$	0.1354
wvote	Direct only; weighted vote; $N = 12$	0.1266
t07n5b	Direct only; revised code	0.1206
e39ling	Direct + Linguini; revised code	0.1188
pvote5	Direct only; batched	0.1062
e40all	Direct + Linguini + multilingual; revised code	0.0797

Table 5: Archived models and submission variants, ordered by public score. All scores are public-split submissions; the private split scored only the two final systems. Links select the submitted revisions. Unless noted, submissions combine $N = 5$ answers; original and revised code differ in effect only in sampling temperature (1.0 and 0.7). "Direct only" weights were trained on direct answers alone.

These submissions differ in training data, voting settings, and code version. Their public scores record the attempted configurations; the comparisons that test individual methods are discussed in Results.

B Candidate Models

Table 6 lists the base models we evaluated before choosing the 4B and 26B systems. Speed and memory on the T4 separated the candidates more clearly than development scores, which were within the noise of our small development sets for the leading candidates.

C Trace Compression Specification

GPT-5.6-sol (medium reasoning effort) received the following instructions for each trace, condensed from the specification used.

- Compress one expert-reviewed solution trace by *deleting* tokens. Words, sentences, sections, headers, and punctuation may be removed; nothing may be added, replaced, reordered, or paraphrased.

Model	Format	tok/s	GB	Outcome
Qwen2.5-14B	AWQ	11.8	14.6	organizers' baseline
Gemma-4-12B (QAT)	q4_0	23.7	–	replaced by 26B
Gemma-4-26B-A4B	IQ3_XXS	41.3	12.4	faster than 12B
Gemma-4-26B-A4B	IQ4_XS	44.6	14.7	26B system
Qwen3.6-35B-A3B	IQ3_XXS	44.7	13.8	scored below 26B
Qwen3.6-27B	IQ3_XXS	9.9	12.1	too slow
Qwen3.5-4B	fp16	16.0	10.7	scored below 12B
Qwen3.5-4B	Q8_0	40.5	5.0	4B system
Qwen3-4B-Thinking	fp16	18.1	9.9	speed test only
VibeThinker-3B	fp16	23.6	7.1	no parseable answers
VibeThinker-1.5B	fp16	30.3	3.7	speed test only
Granite-4.0-h-tiny	Q4_K_M	–	–	scored below 26B
Nanbeige4.2-3B	Q4_K_M	–	–	scored below 26B
Bonsai-27B	ternary	–	–	scored below 26B

Table 6: Candidate base models. Single-stream decoding speed (tok/s) and peak GPU memory (GB) were measured on the T4, fp16 models with transformers and the others with llama.cpp, so speeds are comparable only within a format family. The last three were tested only on a local GPU. "Scored below 26B" refers to development scores.

- **Middle tier** (at most 55% of the trace's words): delete meta-sections (problem information, restatement, overview, alternative approaches, common pitfalls, extended knowledge, the answer and summary sections), headers, connectives, hedges, restatements, and verification back-references. Keep every data observation, correspondence, derivation step, and rule statement.
- **Shortest tier** (at most 20% of the trace's words and at most 900 words): keep only the decisive observations, the induced rules, and the derivations of the asked items, as telegraphic fragments. "Grammar may crumble; content may not."
- Do not include the rules or final-answer blocks, which are appended afterwards. Keep linguistic forms (non-ASCII text, glosses, IPA) byte-exact. In tables, pipes and columns may be deleted, but cells that carry evidence are kept.