SHARED COMPUTATION IS NOT SHARED ACCESS: DIAGNOSING AND IMPROVING MULTILINGUAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multilingual models can solve a problem in one language yet fail on the same problem in another. Final-answer accuracy leaves unclear whether the difficulty lies in executing calculations or in identifying the required quantities and operations and using intermediate results. To distinguish these difficulties, we develop a controlled framework that varies cross-language correspondence and the information supplied to the model. We measure *autonomous solving* from the question, *conditional computation* of a supplied operation, and *continued solving* from a correct partial solution. Experiments with synthetic languages, English–Chinese tasks, and two open-source model families show that accurate local computation can coexist with unreliable autonomous solving. Further diagnostics reveal difficulties in retrieving numerical inputs and completing solutions beyond a correct start. Guided by these diagnostics, we train Qwen3.5-4B first on ordered arithmetic steps extracted from natural-language solutions and then on full target-language chain-of-thought. Compared with chain-of-thought training in both stages, this route achieves higher mean autonomous accuracy under matched source exposure and training updates. These findings motivate evaluating and training multilingual models for both accurate computation and the ability to build complete solutions from a question.

1 INTRODUCTION

Large language models (LLMs) can solve a mathematical problem in English yet fail on the same problem in another language (Shi et al., 2023; Huang et al., 2023; Yang et al., 2025). Final-answer accuracy reveals this gap but leaves its source unresolved. A model may correctly perform a calculation when given the operation and inputs, yet struggle to identify them from the question or use intermediate results in later steps. Distinguishing these difficulties can guide training toward arithmetic execution or toward selecting and using the calculations needed for a complete solution.

We use *access* to describe the ability to identify the calculations required by a question and use their results to complete a solution. We examine this ability alongside arithmetic execution through three measurements (Figure 1): **autonomous solving** from the question alone, **conditional computation** with a supplied operation and its numerical inputs, and **continued solving** from a correct partial solution.

To examine how language correspondence shapes these abilities, we develop a controlled framework that renders the same arithmetic tasks in paired languages. Synthetic language pairs let us vary lexical sharing while holding the tasks and source–target word-order relationship fixed within each configuration. English–Chinese renderings introduce natural-language inputs while retaining control over the task and training history. In both settings, models are trained from scratch on source-language solutions and target-language questions only. We then evaluate Qwen3.5-4B (Qwen Team, 2026) and Phi-3.5-mini-instruct (Abdin et al., 2024) on multilingual GSM8K-derived diagnostic cases (Cobbe et al., 2021; Chen et al., 2024) to examine the same distinctions in open-source models.

Firstly, accurate local computation can coexist with unreliable autonomous solving. With full lexical sharing, synthetic-language models exceed 98.9% accuracy in conditional computation and con-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

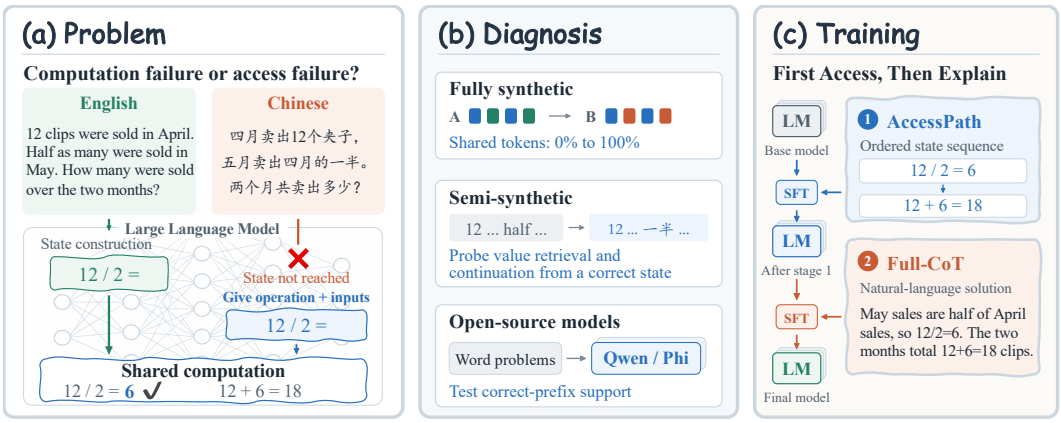

Figure 1: Study overview. (a) Language-dependent access to the computations needed to solve the same arithmetic problem. (b) Diagnosis across fully synthetic languages, English–Chinese tasks, and open-source models. (c) Two-stage training: AccessPath supervision on ordered arithmetic states, followed by Full-CoT supervision on natural-language solutions.

tinued solving, while autonomous accuracy ranges from 10.55% to 89.55% across three language-pair configurations. English–Chinese experiments and open-source models likewise show accurate computation with supplied inputs despite substantial autonomous-solving failures.

Secondly, difficulties arise in both retrieving numerical inputs and completing subsequent steps. English–Chinese models retrieve values from Chinese questions less reliably than they execute supplied operations. After a correct partial solution, first-update accuracy is much higher than final-answer accuracy. Correct prefixes also improve aggregate target-language accuracy in the open-source models, while leaving many problems unsolved.

Thirdly, guided by these diagnostics, we propose a two-stage training approach. We first train on **AccessPath** targets containing ordered arithmetic steps extracted from English solutions, and then on full target-language chain-of-thought (**Full-CoT**) (Wei et al., 2022). On Qwen3.5-4B, we compare this approach with Full-CoT training throughout, matching source-problem exposure, data order, and training updates. Across three paired seeds, our approach achieves higher mean MGSM accuracy in Swahili and Bengali. Repeating the diagnostics after training shows higher mean autonomous accuracy, while the two approaches perform similarly when given correct partial solutions.

2 Cross-lingual access to computation

2.1 Problems, solutions, and explicit task states

For a question $x^{(\ell)}$ in language ℓ with answer y, we represent a reference solution as an ordered sequence of explicit task states:

$$x^{(\ell)} \longrightarrow z_0 \longrightarrow z_1 \longrightarrow \cdots \longrightarrow z_T \longrightarrow y. \tag{1}$$

Each state $z_t = (e_t, v_t)$ records an arithmetic operation e_t, including its numerical inputs, and the resulting value v_t. Later operations can use results from earlier states. This representation separates specifying an operation and its inputs from computing its result. The states can be checked against the task rules and supplied as reference prefixes during evaluation.

2.2 Controllable synthetic reasoning data

The controlled experiments use arithmetic problems generated with GSM-Infinite (Zhou et al., 2025). A directed acyclic graph (DAG) specifies the operations and their dependencies. Instantiating the graph with numerical values determines the reference state sequence and final answer.

We render each instance in two languages while preserving its arithmetic and answer. Section 3 varies lexical sharing between constructed languages. Section 4 uses English and Chinese render-

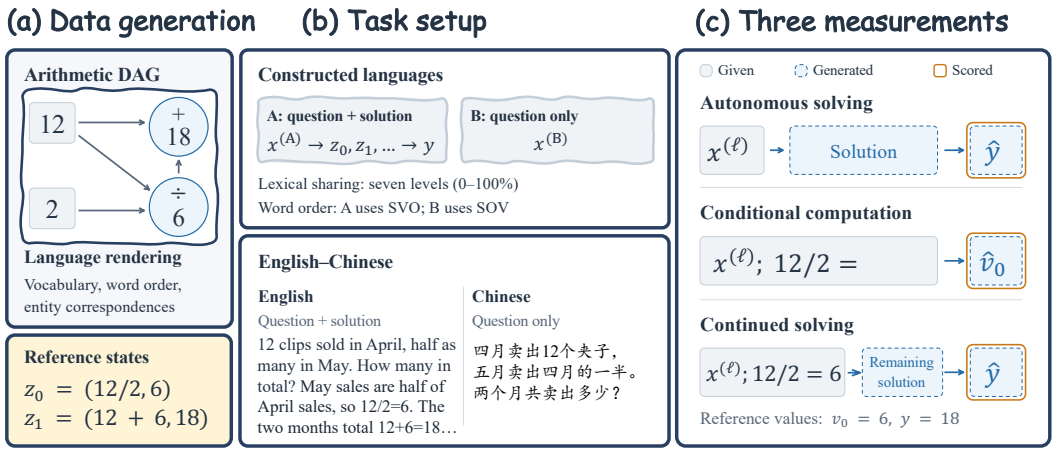

Figure 2: Controlled data and diagnostic measurements. (a) An arithmetic DAG specifies the reference computations. (b) Paired renderings preserve the task; training supplies source-language solutions and target-language questions only. (c) All measurements receive the question. Grey denotes supplied information, dashed blue boxes generated content, and orange outlines scored outputs. The illustrated prefixes end before the selected result for conditional computation and after it for continued solving. Appendix A.3 provides a textual example.

ings. The source language provides complete problem–solution pairs; the target language provides problem and question text.

2.3 Three measurements

All three measurements receive the question $x^{(\ell)}$ and a task instruction. They differ in the reference information supplied and the prediction scored.

Autonomous solving. The model generates a solution and final answer $\hat{y}$ from the question alone, choosing its own solution sequence. Autonomous-solving accuracy, $\mathbf{A}_{\text{auto}}$, scores $\hat{y}$ against y.

Conditional computation. The model receives a correct reference prefix specifying the operation e_t and its numerical inputs, with the result v_t withheld. It predicts $\hat{v}_t$, and conditional-computation accuracy, $\mathbf{A}_{\text{cond}}$, scores this value against v_t.

Continued solving. The model receives a correct partial solution and generates the remaining solution and final answer without further correction. Continued-solving accuracy, $\mathbf{A}_{\text{cont}}$, scores the generated final answer against y.

2.4 Evaluation protocol

Figure 2 illustrates a prefix ending before v_t for conditional computation and after v_t for continued solving. In the lexical-sharing study, continued solving instead starts before the first eligible result absent from the visible prefix, so the model generates both that result and the remaining solution.

Conditional computation is scored per arithmetic result; autonomous and continued solving are scored per problem. The pretrained-model comparisons use matched cases within each language and the same final-answer scorer for autonomous and continued solving. The lexical-sharing study uses different eligible cohorts for the three measurements, and autonomous scoring additionally requires valid answer-register closure. Further interface and scoring details appear in Appendices B, D, and E.

Study overview. We use diagnostic experiments to guide training and repeat the same measurements after training. We begin with synthetic languages, varying lexical sharing while holding the

arithmetic task fixed to examine whether computation and autonomous solving improve together (Section 3). We then use English and Chinese renderings of synthetic problems to examine difficulties in retrieving numerical inputs and completing solutions from correct prefixes (Section 4). To test whether these patterns extend beyond the controlled settings, we evaluate all three measurements in open-source models on multilingual word problems (Section 5).

These findings motivate a two-stage training approach: AccessPath supervision followed by full target-language CoT (Section 6). On Qwen3.5-4B, we compare this approach with Full-CoT training throughout, matching source-problem exposure, data order, and training updates. We evaluate autonomous solving on MGSM, then repeat the diagnostics to examine whether differences between the two approaches persist when operations and inputs or correct partial solutions are supplied.

3 Computation and Autonomous Solving in Synthetic Languages

We first ask whether increasing lexical sharing between two languages improves conditional computation, continued solving, and autonomous solving together. Constructed language pairs let us vary shared word forms while holding the arithmetic tasks and source–target word-order relationship fixed.

Task settings. In each language pair, the source language (A) uses subject–verb–object order and the target language (B) uses subject–object–verb order in the problem statements. Both express the same arithmetic DAGs. Training uses complete problem–solution pairs in A and questions only in B.

We vary lexical sharing over seven levels from 0% to 100%. Sharing assigns the same word form to the same meaning in both languages; numbers, variables, mathematical symbols, and solution delimiters remain shared throughout. Three independently constructed language-pair configurations give $7 \times 3 = 21$ models. Each model follows a Qwen2.5-style architecture (Qwen et al., 2025) with approximately 100M parameters and is trained from scratch on its paired-language corpus. Within each configuration, task instances, data order, token vocabulary, model initialisation, and training settings are fixed across sharing levels. Construction, training, and evaluation details appear in Appendix B.

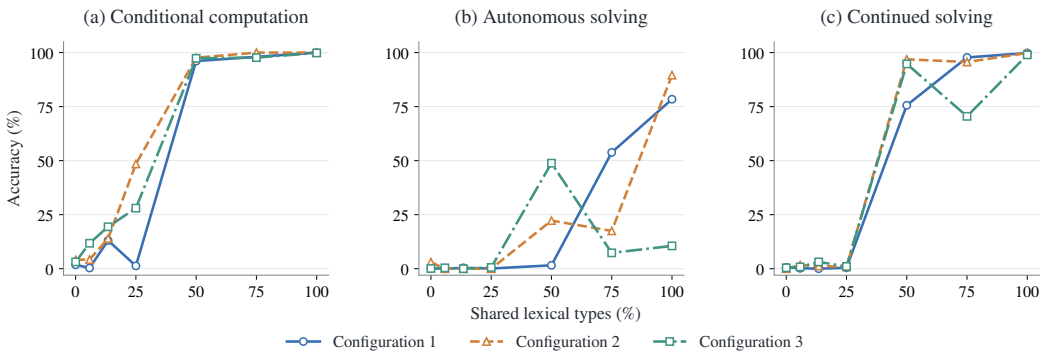

Figure 3: Target-language performance across lexical-sharing levels in synthetic language pairs. Each line shows one of three independently constructed configurations; points are measured settings. The horizontal axis gives the realised fraction of shared word types. Complete results and evaluation details appear in Appendix B.

Observations. Source-language autonomous accuracy ranges from 97.56% to 99.71% across the 21 models (Table 2). Figure 3(a,c) shows sharp improvements in target-language conditional computation and continued solving from 25% to 50% lexical sharing in all three configurations. At full sharing, their accuracies reach at least 99.83% and 98.95%, respectively.

Autonomous solving responds unevenly to lexical sharing (Figure 3(b)). At full sharing, accuracy is 78.42%, 89.55%, and 10.55% in Configurations 1, 2, and 3, respectively. In Configuration 3, full-sharing accuracy is lower than the 48.83% achieved at 50% sharing.

We further examine whether models can retrieve numerical inputs from the question at full sharing. In this setting, paired source–target problems have identical arithmetic tasks, answers, and reference solutions, while their problem statements retain different word orders. A retrieval probe asks for a numerical value stated in the question. For Configuration 3, which has the lowest target-language autonomous accuracy, Table 3 reports retrieval accuracies of 17.19% in the target language and 99.61% in the source language.

Takeaway. Full lexical sharing supports accurate computation and continued solving, while autonomous solving remains uneven across configurations. We next examine input retrieval and subsequent solution steps in English–Chinese tasks (Section 4).

4 Value retrieval and continued solving in English and Chinese

We next examine whether models can retrieve numerical inputs from a question and complete a solution from a correct prefix. English–Chinese renderings of synthetic arithmetic problems let us study these behaviors with natural-language inputs while controlling the tasks and training data.

Task settings. We render the same synthetic arithmetic problems in English and Chinese, providing complete English problem–solution pairs and Chinese questions only. Problems use modulo-23 arithmetic, with balanced counts of two to five operations.

We add auxiliary marker tokens to both languages and vary their cross-language correspondence. Within each language, the natural-language wording and marker positions remain fixed across conditions. We compare four conditions: *no bridge*, where English and Chinese use separate marker vocabularies; *instance-varying entity correspondence*, where matching entities share markers within each paired problem, but entity-to-marker assignments change across problems; *stable entity correspondence*, where each entity retains the same marker across languages and problems; and *broad lexical support*, which additionally shares markers for relations, question roles, and solution-format expressions. Non-entity markers remain language-specific in the two entity-only conditions.

We use the model architecture described in Section 3, with an English–Chinese tokeniser. Five training seeds, paired across conditions, give twenty models, each trained from scratch on 5B non-padding tokens. Evaluation uses 1,024 problems in each of three forms: the original English wording, the original Chinese wording, and held-out Chinese paraphrases. Further training and evaluation details appear in Appendix A.

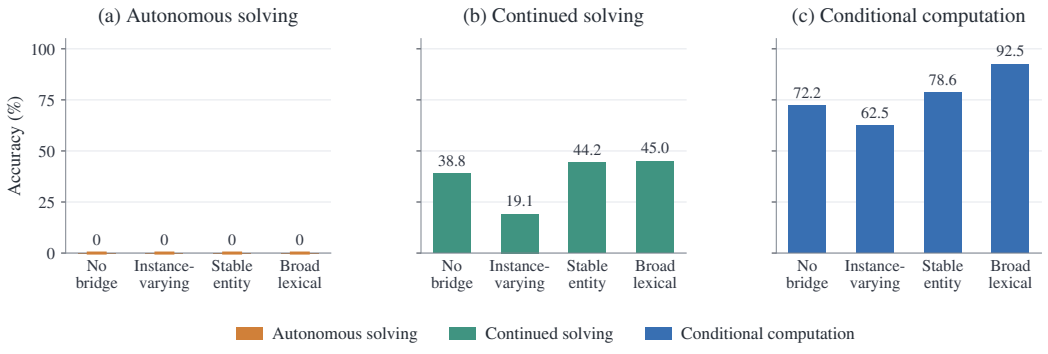

Figure 4: Three measurements on held-out Chinese paraphrases. Bars show means over five paired training seeds per condition: (a) autonomous solving, (b) continued solving, and (c) conditional computation. Exact results appear in Tables 4 and 6.

Observations. Table 4 summarises autonomous solving across the twenty models: pooled accuracy is 99.56% in English, with no correct final answers in 40,960 generations across the two Chinese forms. Five separate calibration models trained with Chinese solutions each achieve at least 98.24% autonomous accuracy on both Chinese forms (Table 5).

For the English-supervised models, we probe retrieval of a numerical value explicitly stated in the Chinese question. Figure 5(a) shows five-seed mean retrieval accuracies of 6.21–20.63% across conditions on held-out Chinese paraphrases. With the operation and its numerical inputs supplied, mean conditional-computation accuracy reaches 62.51–92.46% (Figure 4(c)). When supplied numerical states and entity markers imply different values, predictions predominantly follow the numerical states (Figure 5(c)).

With a correct partial solution, mean final-answer accuracy on held-out Chinese paraphrases rises from zero to 19.15–45.01% across conditions (Figure 4(b)). Figure 5(b) compares first-update and final-answer accuracy within these continuations. Mean first-update accuracy is higher in every condition: under broad lexical support, it reaches 92.24%, compared with 45.01% for the final answer.

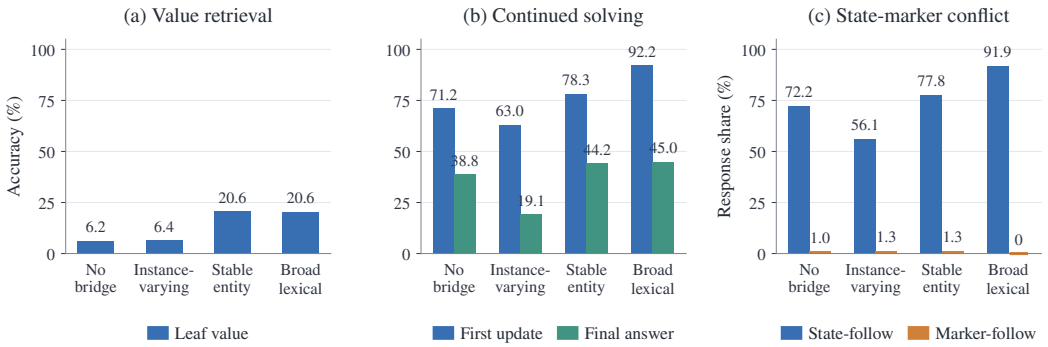

Figure 5: Additional diagnostics on held-out Chinese paraphrases. (a) Value retrieval: recovery of a numerical value stated in the question. (b) Continued solving: first-update and final-answer accuracy. (c) State–marker conflict: responses when numerical states and entity markers imply different values. Bars show means over five training seeds per condition. Table 6 reports the results for (a,b); Table 7 reports the results for (c).

Takeaway. The Chinese results reveal difficulties in both retrieving numerical inputs and completing a solution after a correct start. We next examine how correct partial solutions affect multilingual word-problem solving in open-source models (Section 5).

5 Correct-prefix effects in open-source models

How does a correct partial solution affect multilingual problem solving in open-source models? We apply the three measurements to multilingual word problems, focusing on the change in final-answer accuracy between autonomous and continued solving.

Task settings. We evaluate Qwen3.5-4B in non-thinking mode and Phi-3.5-mini-instruct in English, Swahili, Bengali, and Thai. We construct a diagnostic development split by aligning 256 English GSM8K problems (Cobbe et al., 2021) with their parallel versions in `Mathoctopus` [1] (Chen et al., 2024). Retaining at most one eligible arithmetic step per source problem and language yields 716 language-specific cases.

Within each language, both models use the same cases for all three measurements. Conditional computation and continued solving use reference prefixes in the question's language. Autonomous

[1] https://huggingface.co/datasets/Mathoctopus/GSM8KInstruct_Parallel

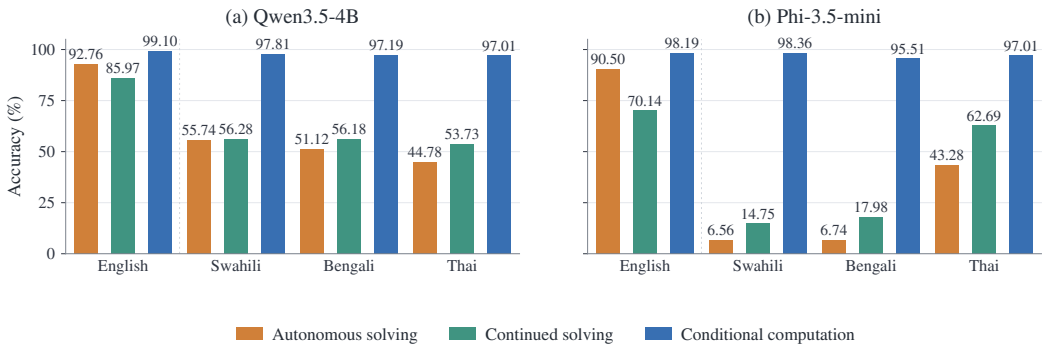

Figure 6: Three measurements on multilingual diagnostic cases. Both models use the same cases within each language. Table 9 reports the per-language accuracies.

and continued solving share the final-answer scorer and a 1,024-token generation cap. Case selection and scoring details appear in Appendices A and D.

Observations. Figure 6 shows high conditional-computation accuracy across both models and all four languages, ranging from 95.51% to 99.10%. Autonomous-solving accuracy is much lower on the target-language cases in Swahili, Bengali, and Thai, ranging from 6.56% to 55.74%.

Correct same-language prefixes improve final-answer accuracy in all three target languages for both models, with gains varying by model and language. In Figure 6(a), Qwen's accuracy changes little in Swahili, from 55.74% to 56.28%. Both models show their largest gains in Thai: from 44.78% to 53.73% for Qwen and from 43.28% to 62.69% for Phi (Figure 6(a,b)). On the same Thai cases, both models reach 97.01% in conditional computation, well above their continued-solving accuracy.

In English, where both models already solve most problems autonomously, correct prefixes instead reduce final-answer accuracy from 92.76% to 85.97% for Qwen and from 90.50% to 70.14% for Phi (Figure 6(a,b)).

Takeaway. Correct prefixes improve aggregate target-language accuracy but reduce English accuracy. We next test whether supervising complete arithmetic-state sequences improves solving from the question alone (Section 6).

6 Arithmetic-State Supervision for Autonomous Solving

The preceding diagnostics motivate supervising the calculations that connect a question to its answer. We propose a two-stage approach that first trains on ordered arithmetic states and then on full target-language chain-of-thought. We test whether this approach improves autonomous solving and repeat the three measurements after training.

Task settings. Our **AccessPath** targets contain arithmetic states extracted from English solutions in their original order, followed by the final answer. The second stage uses **Full-CoT** targets containing the target-language explanation and answer. The baseline uses Full-CoT in both stages. Both approaches receive the same target-language questions.

We fine-tune Qwen3.5-4B separately in Swahili and Bengali using LoRA (Hu et al., 2022), with three paired training seeds per language. Each stage has 320 updates with an effective batch size of 32, presenting 9,216 target-language examples drawn from 5,000 source problems and 1,024 English Full-CoT replay examples. The first stage matches source-problem exposure, data order, and training updates; token counts differ with target length. The second stage uses identical Full-CoT examples, order, and replay, with the optimizer reset for both approaches.

We compare both approaches at the end of the second stage on 250 MGSM test problems per language (Shi et al., 2023). Models generate from the question alone using greedy decoding in non-thinking mode. We use a 512-token primary cap and run a separate 1,024-token evaluation. Training and evaluation details are provided in Appendix E.

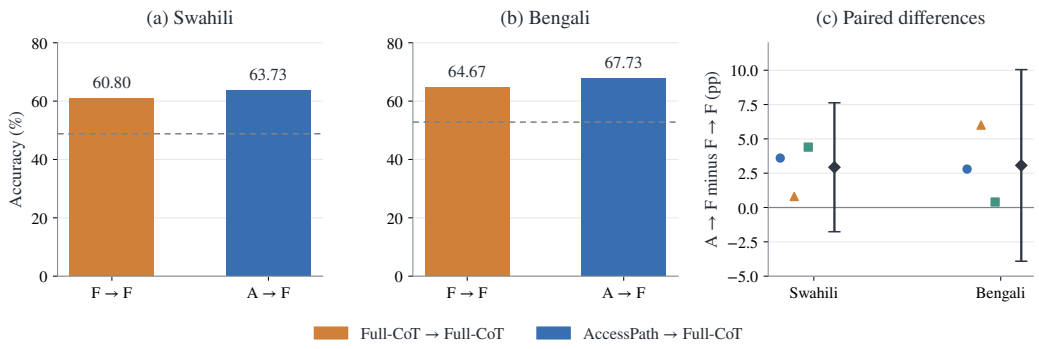

Figure 7: Qwen3.5-4B after the common Full-CoT stage. (a,b) MGSM autonomous-solving accuracy with a 512-token generation cap. Bars show three-seed means; dashed lines show the re-evaluated base model. (c) Paired training-seed differences, with means and 95% seed-level t intervals. F→F denotes Full-CoT→Full-CoT; A→F denotes AccessPath→Full-CoT. Table 13 reports the exact results.

Observations. Figure 7(a,b) shows higher mean autonomous accuracy for the AccessPath-first approach after the common Full-CoT stage: 63.73% in Swahili and 67.73% in Bengali, compared with 60.80% and 64.67% for the Full-CoT baseline. The mean paired gains are 2.93 and 3.07 percentage points, respectively. The paired difference is positive in all three training seeds per language (Figure 7(c)). At the 1,024-token cap, the mean gains are similar: 3.07 and 2.93 points, respectively (Table 13). Stage-wise training trajectories appear in Appendix E.

The AccessPath-first approach also shifts from arithmetic-state chains to prose-like solutions during the second stage. At the final checkpoint, over 98% of its outputs in each language pass an automated check for prose outside arithmetic notation, and state tags disappear (Appendix E). Mean output lengths are similar between the approaches. Table 15 reports English accuracy of 82.27–84.13% across approaches and adaptation languages, below the base model's 91.60% despite English replay.

Post-training diagnostics. We apply the three measurements to the base model and to both approaches at each stage endpoint. Separate from the MGSM test set, the diagnostic development cohort contains 165 Swahili and 164 Bengali cases drawn from 192 English source problems excluded from fine-tuning. Within each language, all checkpoints use the same cases, a common user instruction across the three measurements, and fixed reference prefixes for conditional computation and continued solving. Conditional computation supplies an explicit arithmetic expression with its result withheld; autonomous and continued solving use a 1,024-token generation cap. Cohort selection and prompt details are provided in Appendix E.

At the final checkpoint, the AccessPath-first approach achieves mean autonomous accuracy of 64.85% in Swahili and 67.48% in Bengali, compared with 60.81% and 66.06% for the baseline (Figure 8(a)). Both approaches reach 100% conditional-computation accuracy in Swahili and 99.80% in Bengali (Figure 8(b)). With correct partial solutions, mean continued-solving accuracy is 73.13% for both approaches in Swahili, and 73.58% versus 73.17% for the AccessPath-first approach and baseline in Bengali (Figure 8(c)). The 95% paired-seed intervals for the autonomous- and continued-solving differences include zero (Table 17).

Takeaway. On Qwen3.5-4B, arithmetic-state supervision followed by Full-CoT yields higher mean autonomous accuracy than Full-CoT training throughout. The post-training mean differences are larger in question-only solving than with supplied operations or correct partial solutions.

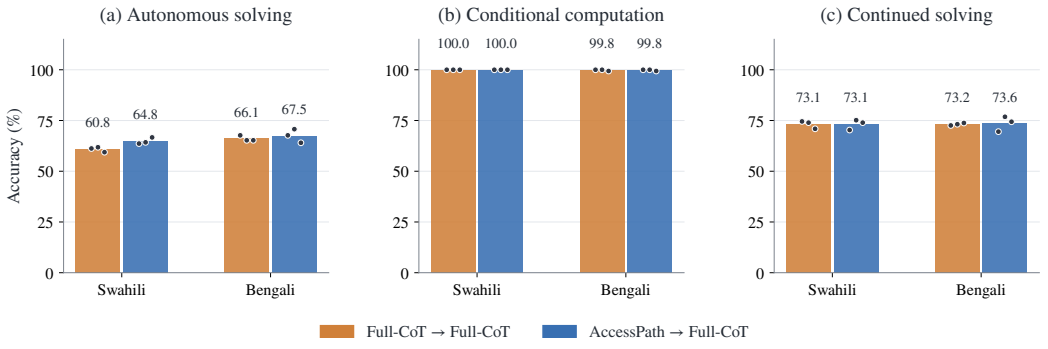

Figure 8: Qwen3.5-4B diagnostics after the common Full-CoT stage: (a) autonomous solving, (b) conditional computation, and (c) continued solving. Bars show three-seed means; dots show individual training seeds. Each language uses the same cases across approaches and measurements: 165 in Swahili and 164 in Bengali. Autonomous and continued solving use a 1,024-token cap. Table 16 reports the exact accuracies, including the base model and both training stages.

7 Related work

Multilingual reasoning and diagnosis. Multilingual reasoning benchmarks reveal performance gaps across languages (Shi et al., 2023). Translated-solution training (Chen et al., 2024) and cross-lingual prompting (Huang et al., 2023) improve performance, while recent analyses identify language understanding as a source of errors (Kang et al., 2026). In multi-hop factual reasoning, Sun et al. (2026) examine cross-lingual failures in accessing shared knowledge. Synthetic languages (Schäfer et al., 2024) and arithmetic tasks (Zhang et al., 2025) support controlled studies of language transfer and reasoning generalisation. Complementing analyses of language understanding and factual knowledge access, we distinguish executing supplied arithmetic from identifying and using the required calculations to complete a solution. We examine these behaviors by varying cross-language correspondence and the reference information supplied to the model.

Intermediate reasoning supervision. Scratchpad training and chain-of-thought prompting make intermediate computations and reasoning explicit (Nye et al., 2021; Wei et al., 2022). Symbol-LLM combines symbolic and natural-language capabilities through two-stage tuning (Xu et al., 2024), while Lin et al. (2026) develop a symbolic-to-natural-language reinforcement-learning curriculum. We use this symbolic-to-natural-language training progression to study multilingual access: whether arithmetic states extracted from English solutions improve autonomous solving from target-language questions. We compare state and Full-CoT supervision under matched source exposure and training updates, followed by a common Full-CoT stage, and repeat the diagnostics to examine computation and access.

8 Conclusion

We study why multilingual models can execute supplied calculations correctly yet fail to solve the problems that require them. Across controlled experiments and open-source models, our diagnostics distinguish arithmetic execution from autonomous solving. Further probes reveal difficulties in retrieving numerical inputs and completing solutions after a correct start. These findings motivate training on AccessPath targets before full target-language chain-of-thought. On Qwen3.5-4B, this approach yields higher mean autonomous accuracy in Swahili and Bengali than Full-CoT training throughout under matched source exposure and training updates. Post-training diagnostics show smaller mean accuracy differences between the approaches with correct partial solutions than in autonomous solving. Our results motivate evaluating and training multilingual models for the ability to select and connect calculations into complete solutions, alongside accurate arithmetic execution.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong, Dongmei Zhang, and Jia Li. Breaking language barriers in multilingual mathematical reasoning: Insights and observations. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7001–7016, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.411. URL https://aclanthology.org/2024.findings-emnlp.411/.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Xin Zhao, Ting Song, Yan Xia, and Furu Wei. Not all languages are created equal in LLMs: Improving multilingual capability by cross-lingual-thought prompting. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 12365–12394, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.826. URL https://aclanthology.org/2023.findings-emnlp.826/.

Deokhyung Kang, Seonjeong Hwang, Daehui Kim, Hyounghun Kim, and Gary Lee. Why do multilingual reasoning gaps emerge in reasoning language models? In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 31684–31716, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1586. URL https://aclanthology.org/2026.findings-acl.1586/.

Fangru Lin, Valentin Hofmann, Xingchen Wan, Weixing Wang, Zifeng Ding, Anthony G. Cohn, and Janet B. Pierrehumbert. Can large language models generalize procedures across representations?, 2026. URL https://arxiv.org/abs/2602.03542.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models, 2021. URL https://arxiv.org/abs/2112.00114.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Anton Schäfer, Shauli Ravfogel, Thomas Hofmann, Tiago Pimentel, and Imanol Schlag. The role of language imbalance in cross-lingual generalisation: Insights from cloned language experiments, 2024. URL https://arxiv.org/abs/2404.07982.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=fR3wGCk-IXp.

Chenghao Sun, Zhen Huang, Yonggang Zhang, Xinmei Tian, Xu Shen, and Jieping Ye. Bridging the language gap: Uncovering and aligning shared circuits for multi-hop reasoning in multilingual LLMs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33091–33099, 2026. doi: 10.1609/aaai.v40i39.40592. URL https://ojs.aaai.org/index.php/AAAI/article/view/40592.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Fangzhi Xu, Zhiyong Wu, Qiushi Sun, Siyu Ren, Fei Yuan, Shuai Yuan, Qika Lin, Yu Qiao, and Jun Liu. Symbol-LLM: Towards foundational symbol-centric interface for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13091–13116, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.707. URL https://aclanthology.org/2024.acl-long.707/.

Wen Yang, Junhong Wu, Chen Wang, Chengqing Zong, and Jiajun Zhang. Language imbalance driven rewarding for multilingual self-improving. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Kak2ZH5Itp.

Charlie Zhang, Graham Neubig, and Xiang Yue. On the interplay of pre-training, mid-training, and rl on reasoning language models, 2025. URL https://arxiv.org/abs/2512.07783.

Yang Zhou, Hongyi Liu, Zhuoming Chen, Yuandong Tian, and Beidi Chen. GSM-Infinite: How do your LLMs behave over infinitely increasing reasoning complexity and context length? In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 78933–78983. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zhou25m.html.

ETHICS STATEMENT

This study uses public GSM8K-derived datasets and released model weights, under their respective licences, together with programmatically generated arithmetic tasks. It involves no human subjects or newly collected personal data. Target-language accuracy is evaluated alongside English retention. The automated output-form measure assesses the presence of prose rather than the quality of an explanation, which remains important for deployment in languages with limited evaluation resources.

REPRODUCIBILITY STATEMENT

The appendices report language construction, evaluation interfaces, sample selection, training settings, and numerical results. The lexical-sharing study contains three language-pair configurations and seven sharing levels. The Qwen two-stage study uses three paired training seeds per adaptation language. The appendices report its fixed test endpoint, complete second-stage trajectories, output behavior, English retention, and pre- and post-training diagnostics. Experimental code and frozen generations are planned for release with the camera-ready version.

THE USE OF LARGE LANGUAGE MODELS

In this work, we used generative AI tools (ChatGPT and Claude) to assist with manuscript writing, including language polishing and improvements to clarity and organisation. We reviewed and revised all AI-assisted text to ensure that it accurately presents our methods, results, and conclusions. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the assistance of generative AI.

APPENDIX

A Implementation and measurement interfaces **14**
A.1 Controlled models . . . 14
A.2 Parallel data and evaluation protocols . . . 14
A.3 Supplied and generated content . . . 14

B Lexical-sharing construction and complete results **15**
B.1 Language construction and training . . . 15
B.2 Evaluation and all sharing levels . . . 15
B.3 Full-sharing source–target comparison . . . 15

C English–Chinese diagnostic results **17**
C.1 Autonomous solving and the Chinese-supervised calibration . . . 17
C.2 Value retrieval, conditional computation, and continued solving . . . 17
C.3 State–marker conflicts . . . 17
C.4 Paired training-seed variation . . . 18

D Open-source model diagnostic results **18**
D.1 Matched cases and exact accuracies . . . 18
D.2 Conditional-computation scorer sensitivity . . . 19
D.3 Termination on the matched cohort . . . 19

E Qwen two-stage training and post-training diagnostics **20**
E.1 Targets, source exposure, and optimization . . . 20
E.2 MGSM evaluation and paired uncertainty . . . 20
E.3 Performance through the common Full-CoT stage . . . 22
E.4 Output form, termination, and English retention . . . 22
E.5 Post-training diagnostic protocol and results . . . 23

A IMPLEMENTATION AND MEASUREMENT INTERFACES

This appendix describes the models, data splits, and supplied-prefix interfaces. Appendices B–D give the results for the three diagnostic settings, followed by the Qwen training study in Appendix E.

A.1 CONTROLLED MODELS

The controlled studies use randomly initialised decoder-only Transformers with approximately 100M parameters. Source-language examples contain complete solutions; target-language examples contain questions only. The lexical-sharing models use a 248-token synthetic vocabulary and the architecture and training schedule in Appendix B.

The English–Chinese models have 12 layers and hidden size 768, use an English–Chinese tokenizer without pretrained model weights, and train on 5B non-padding tokens per model across both languages. Problems use modulo-23 arithmetic, with operation counts 2–5 balanced. Four correspondence conditions and five paired training seeds give twenty main models. Five separate models trained with Chinese solutions provide the calibration in Table 5. Each evaluation form contains 1,024 problems. Detailed optimizer and decoding configurations are not retained in the archived English–Chinese summaries.

A.2 PARALLEL DATA AND EVALUATION PROTOCOLS

The open-source diagnostic study evaluates Qwen3.5-4B and Phi-3.5-mini-instruct; Qwen uses non-thinking mode. The parallel GSM8K data pool contains 5,355 aligned source problems. Removing 23 records with embedded turn markers or identical question and solution text leaves 5,332 records: 5,264 with extractable calculator states and 68 without. The original split assigns 5,000 source problems to training, 256 to diagnostic development, and 76 to a reserved set (8 with states and 68 without). Published target-language texts are otherwise unchanged.

The diagnostic development split yields 2,332 eligible arithmetic positions across four languages. Selecting at most one position per source problem and language gives the 716 matched cases used in Section 5. Appendix D specifies this selection and the scoring rules. For post-training diagnosis, source identities are checked against the actual fine-tuning pool, leaving 329 cases from 192 sources (Appendix E.5). MGSM test is a separate evaluation with 250 problems per language and is not the source of either diagnostic cohort.

For the two-stage MGSM evaluations, all checkpoints use the same 250 test questions per language, in the same order. Evaluation uses greedy decoding, batch size 8, BF16 with scaled dot-product attention, left padding, caching, no input truncation, and disabled Qwen thinking. The scorer compares the last complete generated numerical value with the normalized reference answer. The 512- and 1,024-token evaluations are generated separately; the base model and Stage-A endpoints are re-evaluated under the same execution path.

A.3 SUPPLIED AND GENERATED CONTENT

Figure 9 expands the example in Figure 2. Conditional computation receives the question and reference text through the result-producing equals sign. Continued solving additionally receives the numerical result, but none of the subsequent words. The generated result is scored in the former; the generated final answer is scored in the latter. The lexical-sharing continued-solving interface instead withholds the first eligible result, as specified in Table 1.

Problem: Natalia sold 12 clips in April and half as many in May. How many clips did she sell in total?	
Reference solution: Natalia sold $12/2 = 6$ clips in May and $12 + 6 = 18$ clips in total. Answer: 18.	
Autonomous solving	Question only. Natalia sold $12/2 = 6$ clips in May and $12 + 6 = 18$ clips in total. Answer: 18.
Conditional computation	Question + Natalia sold $12/2 = 6$
Continued solving	Question + Natalia sold $12/2 = 6$ clips in May and $12 + 6 = 18$ clips in total. Answer: 18.

Figure 9: An illustrative two-state solution. Grey denotes supplied content and blue denotes generated content. The two prefix-based conditions differ by the supplied result, 6. This example illustrates the boundary between input and output rather than an exact evaluation prompt. The original matched diagnostic cases additionally require a result absent from the question and visible prefix, a later state, and a final answer not revealed by the supplied state.

B Lexical-sharing construction and complete results

B.1 Language construction and training

Three language-pair configurations independently generate language forms, shared-word rankings, task instances, and training randomness. Within a configuration, the seven sharing levels reuse the source corpus, sample order, tokenizer, model initialisation, and optimizer settings. Configuration numbers identify independently constructed language settings, rather than repeated training on a fixed pair.

The sharing denominator contains 32 entity/content types and 20 function/relation types. Each class has a fixed hash-based ranking within a configuration; higher sharing levels use longer prefixes of that ranking. Shared-type counts are 0, 3, 7, 13, 26, 39, and 52. The nominal 6.25% and 12.5% levels therefore realise 5.77% and 13.46% sharing. Numbers, variables, operators, delimiters, and solution-register words remain shared throughout. At full sharing, problem statements retain the source SVO and target SOV orders; question templates and reference solutions use the same order in both languages.

The 248-token vocabulary retains shared and private candidate forms, including unused forms. Models have 12 layers, hidden size 768, 12 attention heads, two key/value heads, and intermediate size 3,072. Training uses one epoch of the nominal 5B-token corpus, batch size 256, a 10^{-3} peak learning rate, a cosine schedule, a 0.02 warm-up fraction, and a 1,024-token sequence limit. The 5B designation identifies the corpus series; manifest estimates from sampled document lengths are 3.660B, 3.640B, and 3.662B tokens for Configurations 1–3, fixed across sharing levels within each configuration. All results use the final checkpoint.

B.2 Evaluation and all sharing levels

Autonomous solving requires the correct answer and valid answer-register closure. Conditional computation predicts a reference arithmetic result absent from the visible prefix. Continued solving begins before the first eligible novel result and scores the final answer without the additional closure requirement. Leaf-value retrieval supplies a minimal solution scaffold and asks for a number stated in the problem. Table 1 gives the scoring units and eligible cohorts; Table 2 reports all 21 models.

B.3 Full-sharing source–target comparison

Source and target autonomous evaluations use the same 1,024 DAGs and answers, with identical reference solutions. Each auxiliary measurement uses its own eligible cohort from Table 1. Table 3 reports both languages for all three configurations.

Table 1: Lexical-sharing evaluation protocol. Variable counts are for Configurations 1/2/3. Caps count generated tokens.

Measurement	Input and scoring unit	Cases	Cap
Autonomous	Question; strict final answer, 2–5 operations	1,024	200
Conditional	Prefix before novel result; arithmetic position, 4–5 operations	613/588/603	4
Continued	Prefix before first novel result; final answer, 4–5 operations	488/482/478	256
Leaf retrieval	Minimal scaffold; one stated leaf value, 4–5 operations	512	4

Table 2: All lexical-sharing runs (%). Configuration numbers denote independently constructed language-pair settings. The first column gives nominal sharing; realised low levels are 5.77% and 13.46%.

Sharing	Config.	A autonomous	B autonomous	B conditional	B continued
0	1	99.12	0.10	1.79	0.61
0	2	99.32	3.03	4.08	0.00
0	3	99.71	0.10	3.15	0.42
6.25	1	99.41	0.00	0.33	0.20
6.25	2	99.41	0.00	4.25	1.66
6.25	3	99.51	0.39	11.77	0.84
12.5	1	99.51	0.39	13.05	0.00
12.5	2	99.51	0.00	13.95	1.24
12.5	3	99.51	0.00	19.40	3.14
25	1	99.02	0.10	1.31	0.41
25	2	99.51	0.00	48.47	0.83
25	3	99.32	0.59	28.03	1.05
50	1	98.54	1.56	96.08	75.61
50	2	98.83	22.27	97.62	96.89
50	3	98.34	48.83	97.35	94.77
75	1	99.71	53.81	98.04	97.75
75	2	97.56	17.48	100.00	95.64
75	3	98.83	7.32	97.68	70.50
100	1	99.41	78.42	100.00	99.80
100	2	99.32	89.55	100.00	99.79
100	3	99.22	10.55	99.83	98.95

Table 3: Full lexical sharing: source/target accuracy (%) within each language-pair configuration.

	Autonomous		Leaf retrieval		Conditional		Continued	
Config.	A	B	A	B	A	B	A	B
1	99.41	78.42	99.61	82.62	100.00	100.00	100.00	99.80
2	99.32	89.55	100.00	93.55	100.00	100.00	99.79	99.79
3	99.22	10.55	99.61	17.19	99.83	99.83	99.79	98.95

In Configuration 3, leaf-value retrieval returns a valid number in all 512 target-language cases but is correct in only 88. Among the 424 errors, 305 predict 1 and 86 predict 2. A conditional-computation control scores 99.83% with the correct question and 29.52% without a question. Replacing the question with another instance, random tokens, or a token-shuffled question also gives 99.83%, indicating that the supplied solution context strongly supports this local calculation.

C ENGLISH–CHINESE DIAGNOSTIC RESULTS

C.1 AUTONOMOUS SOLVING AND THE CHINESE-SUPERVISED CALIBRATION

Table 4 pools the twenty main models, each evaluated on 1,024 problems in three forms. The separate Chinese-supervised calibration tests whether the two Chinese forms can be learned with full solutions. Every calibration seed reaches at least 98.24% autonomous accuracy on both forms; the minimum across operation-count groups is 96.09%. Table 5 reports these available bounds, not per-seed point estimates.

Table 4: Autonomous solving in the English–Chinese experiment. Results are pooled over four correspondence conditions and five training seeds per condition, with 1,024 problems per evaluation form for each model. The final row combines the two Chinese forms.

Evaluation form	Generations	Accuracy (%)
Original English wording	20,480	99.56
Original Chinese wording	20,480	0.00
Held-out Chinese paraphrases	20,480	0.00
Chinese, both forms combined	40,960	0.00

Table 5: Chinese-solution calibration. The bounds hold for all five calibration seeds; they are not five-seed means.

Evaluation subset	Seeds	Reported autonomous accuracy
Canonical Chinese	5/5	At least 98.24% in every seed
Held-out Chinese paraphrases	5/5	At least 98.24% in every seed
Operation-count groups	5/5	Minimum across reported groups: 96.09%

C.2 VALUE RETRIEVAL, CONDITIONAL COMPUTATION, AND CONTINUED SOLVING

Value retrieval scores recovery of a number explicitly stated in the question. Conditional computation scores the result of a supplied operation. Within continued solving, first-update accuracy scores the first generated numerical update, and final-answer accuracy scores completion of the solution. Table 6 supplies the exact values for Figures 4 and 5(a,b).

All reported Chinese continued-solving results use a scorer that extracts the last complete numerical value. The original rule recognised an ASCII answer marker but missed Chinese answer wording. The revised rule was applied to existing generations without regenerating solutions and agrees with the original rule on all 9,556 checked English rollouts.

C.3 STATE–MARKER CONFLICTS

The conflict probe supplies a numerical state and an entity marker that imply different results. Each prediction is scored against both alternatives; predictions matching neither count toward neither rate. Table 7 reports the five-seed means in Figure 5(c).

Table 6: Held-out Chinese paraphrase accuracies (%), averaged over five training seeds per condition. Value retrieval scores recovery of a value stated in the question; first-update and final-answer accuracy score the beginning and endpoint of continued solving. Conditional computation uses a separate single-update probe.

Condition	Value retrieval	Conditional computation	Continued solving first update	Continued solving final answer
No bridge	6.21	72.18	71.22	38.79
Instance-varying entity	6.41	62.51	63.03	19.15
Stable entity	20.63	78.62	78.31	44.23
Broad lexical support	20.55	92.46	92.24	45.01

Table 7: English–Chinese state–marker conflicts on held-out Chinese paraphrases (%), averaged over five training seeds per condition. These are the values in Figure 5(c).

Correspondence condition	State-following	Marker-following
No bridge	72.25	1.02
Instance-varying entity	56.15	1.29
Stable entity	77.76	1.30
Broad lexical support	91.93	0.00

C.4 Paired training-seed variation

Table 8 compares stable and instance-varying entity correspondence on the same five paired training seeds. Rows follow the order in the recorded paired results. The intervals use training seeds as the replication unit; autonomous-solving differences are zero for every pair.

Table 8: English–Chinese paired differences (pp), stable entity minus instance-varying entity, on held-out Chinese paraphrases. Intervals use training seeds as the unit of replication. All four intervals include zero.

	Value retrieval	Conditional computation	Continued solving first update	Continued solving final answer
Pair 1	+5.08	+4.24	+4.62	+46.59
Pair 2	+0.98	+19.84	+21.24	−15.53
Pair 3	+16.99	+60.39	+56.39	+50.47
Pair 4	+49.22	+0.23	+0.84	+13.46
Pair 5	−1.17	−4.16	−6.70	+30.40
Mean	+14.22	+16.11	+15.28	+25.08
Positive pairs	4/5	4/5	4/5	4/5
95% CI	[−11.59, 40.03]	[−16.61, 48.83]	[−15.95, 46.51]	[−8.47, 58.62]

D Open-source model diagnostic results

D.1 Matched cases and exact accuracies

For each source problem and language, we retain the earliest eligible arithmetic position whose result is absent from the question and visible prefix, with at least one later state and a final answer not already exposed. The resulting cohort contains 221 English, 183 Swahili, 178 Bengali, and 134 Thai cases. Both models use the same cases within each language. Autonomous and continued solving

share a 1,024-token cap and a scorer extracting the last complete numerical value. Conditional computation reuses the frozen first-numeric scores for the selected positions. Table 9 gives the exact results in Figure 6.

Table 9: Three endpoints on the matched cohort. All accuracies are percentages of the N cases in each row. The measurements follow the order used in Section 2.3.

Model	Language	N	A_{auto}	A_{cond}	A_{cont}
Qwen3.5-4B	English	221	92.76	99.10	85.97
	Swahili	183	55.74	97.81	56.28
	Bengali	178	51.12	97.19	56.18
	Thai	134	44.78	97.01	53.73
Phi-3.5-mini-instruct	English	221	90.50	98.19	70.14
	Swahili	183	6.56	98.36	14.75
	Bengali	178	6.74	95.51	17.98
	Thai	134	43.28	97.01	62.69

D.2 Conditional-computation scorer sensitivity

The original 2,332-position inventory includes multiple steps from some problems and is distinct from the matched 716-case cohort. Table 10 compares two rules on the same frozen Qwen generations. Strict-leading scoring requires a numeric value at the start of the response; first-numeric scoring permits preceding text. The latter rule was defined after inspecting Qwen responses. Strict-leading results are unavailable for the matched cohort and for Phi in the archived summaries.

Table 10: Conditional-computation accuracy (%) on the original step inventory. Only Qwen has both scorer values in the recorded comparison. Steps sharing a source problem are not independent problem-level observations.

Language	Steps	Qwen strict-leading	Qwen first-numeric	Phi first-numeric
English	735	75.10	99.46	98.78
Swahili	599	70.95	98.83	97.83
Bengali	587	72.57	98.13	93.19
Thai	411	81.75	98.54	97.57

D.3 Termination on the matched cohort

Table 11 reports how often autonomous and continued solving reach the generation cap. For Phi, the continued-solving rates exceed 50% in Swahili and Bengali; all Qwen rates remain below 7%. These rates use the same cases and 1,024-token cap as the accuracy comparison.

Table 11: Percentage of generations reaching the token cap on the matched diagnostic cases. Both final-answer evaluations use the same 1,024-token cap and the same cases within each language.

Model	Language	Cases	Autonomous cap hit	Continued cap hit
Qwen3.5-4B	English	221	0.90	0.00
	Swahili	183	4.37	1.64
	Bengali	178	6.74	3.37
	Thai	134	4.48	6.72
Phi-3.5-mini-instruct	English	221	0.00	23.53
	Swahili	183	10.38	50.82
	Bengali	178	27.53	51.12
	Thai	134	2.24	1.49

E QWEN TWO-STAGE TRAINING AND POST-TRAINING DIAGNOSTICS

E.1 TARGETS, SOURCE EXPOSURE, AND OPTIMIZATION

The training study uses Qwen3.5-4B, two adaptation languages, two first-stage targets, and three paired training seeds (0, 1, and 2), giving twelve two-stage routes. Full-CoT targets contain the published target-language solution and answer. AccessPath targets contain all calculator states mechanically extracted from the English reference in their original order, followed by a target-language answer label and the final answer. No additional rationales or translations are generated for these targets.

Each stage has 320 optimizer updates and presents 10,240 examples: 9,216 in the adaptation language and 1,024 English Full-CoT replay examples. The target-language stream covers all 5,000 training sources; replay comes from the same pool. Every tenth example is English replay. The two first-stage routes share source exposure, order, and update count, while their target lengths and token counts differ.

Stage B continues the deterministic source stream after Stage A. Both routes receive the same Full-CoT examples, order, replay, sequence tokens, and supervised tokens in Stage B. Each loads its own Stage-A adapter and resets the optimizer and scheduler. A320, also denoted B0, is the first-stage endpoint. Table 12 gives the shared settings.

E.2 MGSM EVALUATION AND PAIRED UNCERTAINTY

The primary comparison uses the fixed B320 endpoint. This evaluation plan was specified after training and before the reported generation round; the test and development sets had already been used earlier in the project. For a paired-seed difference d, the 95% interval is $\bar{d} \pm t_{2,0.975} s_d/\sqrt{3}$. These intervals describe training variation on fixed questions, without item-sampling uncertainty or multiplicity adjustment. F→F denotes Full-CoT→Full-CoT and A→F denotes AccessPath→Full-CoT throughout this appendix.

Table 12: Two-stage training settings, for Qwen3.5-4B, fixed across routes within each language.

Setting	Value
Adapter	LoRA rank 16, alpha 32, dropout 0.05
Trainable parameters	32,464,896
Precision / attention	BF16 / scaled dot-product attention
Effective batch	32: eight GPUs, microbatch 1, accumulation 4
Optimizer	AdamW, $\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0, gradient clipping 1
Stage A schedule	16-update warm-up; cosine over 320 updates; peak 2×10^{-4}, minimum 3×10^{-5}
Stage B schedule	4-update warm-up; cosine over 320 updates; peak 5×10^{-5}, minimum 7.5×10^{-6}
Stage transition	Load Stage-A adapter; reset optimizer and scheduler
Loss	Assistant-token cross-entropy; user prompt labels masked
Sequence limit	1,152
Stage B checkpoints	8, 16, 32, 64, 128, 192, 256, 320
Replay	Every tenth example is English Full-CoT

Table 13: Qwen3.5-4B at the fixed second-stage endpoint on MGSM test, 250 problems per language. Accuracy is in percent; differences and 95% paired-seed t intervals are in percentage points. Each mean uses three training seeds.

Language	Cap	F→F	A→F	Δ	95% CI
Swahili	512	60.80	63.73	+2.93	$[-1.76, 7.63]$
Swahili	1024	60.80	63.87	+3.07	$[-1.83, 7.97]$
Bengali	512	64.67	67.73	+3.07	$[-3.91, 10.05]$
Bengali	1024	64.80	67.73	+2.93	$[-3.14, 9.00]$

Table 14: AccessPath-route minus Full-CoT-route accuracy at the second-stage endpoint, by paired training seed (pp; 512-token cap).

Language	Seed 0	Seed 1	Seed 2
Swahili	+3.60	+0.80	+4.40
Bengali	+2.80	+6.00	+0.40

E.3 Performance through the common Full-CoT stage

Figure 10 retains every evaluated second-stage checkpoint. The mean route difference changes from $+1.47\,pp$ at B0 to $+2.93\,pp$ at B320 in Swahili, and from $-2.80\,pp$ to $+3.07\,pp$ in Bengali. Both languages show an early decline after switching targets, followed by recovery. The final comparison uses B320 rather than a checkpoint selected from these curves.

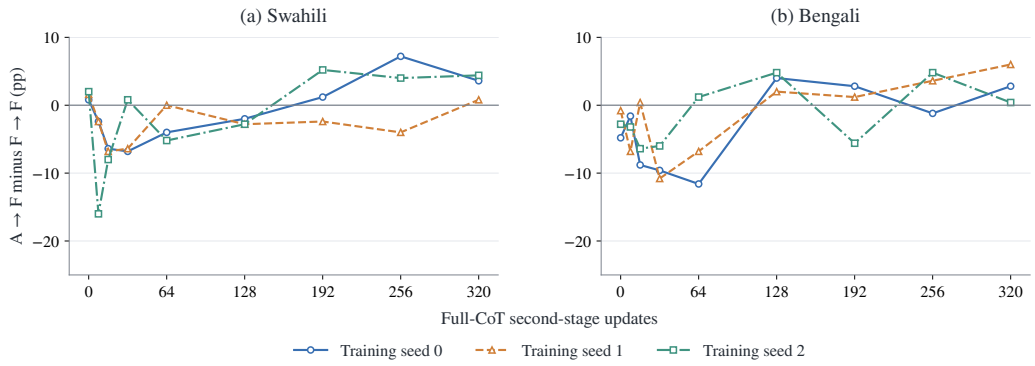

Figure 10: Qwen3.5-4B paired accuracy differences through the common Full-CoT stage on MGSM test, with a 512-token cap. Each line is one paired training seed, evaluated on the same 250 questions per language. Points include B0 and all eight saved Stage-B checkpoints; B320 is the fixed final endpoint.

E.4 Output form, termination, and English retention

The prose indicator removes state blocks, calculator markers, and fixed answer/solution labels, then checks whether at least 20 Unicode letters remain. It measures prose outside arithmetic notation, rather than language identity, fluency, or explanation correctness. At B320, AccessPath-route prose rates are 99.73% in Swahili and 98.67% in Bengali, with zero state-tag rates in both languages. Table 15 reports output lengths, cap hits, and English accuracy. The re-evaluated English base accuracy is 91.60% at the same 512-token cap.

Table 15: Qwen3.5-4B B320 output behavior and English accuracy. F and A denote Full-CoT and AccessPath origins. Values use a 512-token cap and average three training seeds.

	Output tokens		Cap hit (%)		English accuracy (%)	
Adaptation	F	A	F	A	F	A
Swahili	167.8	168.9	0.93	0.93	83.60	82.27
Bengali	219.2	208.7	2.93	2.53	83.73	84.13

A fixed example illustrates the change in output form. We use Swahili MGSM test index 0, training seed 0, and the 1,024-token evaluation, selected without conditioning on correctness. The question asks about selling the eggs remaining after eating three and using four of sixteen. The AccessPath first-stage output is:

```
Suluhisho:  <state>16-3-4=9</state>
<state>9*2=18</state>
Jibu:  18
```

After Stage B, the same route produces:

Suluhisho: Bata wa Janet hutaga mayai 16 kila siku.
Huwa anakula mayai 3 kila siku.
Huwa huokea mayai 4 kila siku.
Hivyo, huwa ana mayai 16 - 3 - 4 = <<16-3-4=9>>9 yaliyosalia kila siku.
Hivyo, huwa anauza mayai 9 kila siku.
Hivyo, huwa anapata $2 x 9 = $<<2*9=18>>18 kila siku katika soko la wakulima.
Jibu: 18

E.5 Post-training diagnostic protocol and results

Starting from the original 183 Swahili and 178 Bengali cases, source-identity checks remove 16 and 13 cases, respectively, whose source problems appear in the later fine-tuning pool. Two additional Swahili cases and one Bengali case expose the requested value in the supplied expression. These disjoint exclusions leave 165 Swahili and 164 Bengali cases from 192 English source problems, with 137 sources shared across languages. Eligibility is fixed before the generation round and applied to every checkpoint and measurement.

All measurements use the same question and step-by-step user instruction. Autonomous solving has no reference prefix. Continued solving supplies the reference explanation through the selected state's result. Conditional computation supplies the reference text followed by `<<reference_expression=` and scores the first generated number, with a 24-token cap. Strict-leading and first-number scores coincide under this interface. Final-answer evaluations score only newly generated text and use a 1,024-token cap. The base model is evaluated anew with these prompts.

Expression-format control. An alternative interface ends at the natural-language equals sign. Full-CoT-trained models often respond by rewriting a calculator expression, making first-number scoring select an operand. At B320 in Swahili, the AccessPath and Full-CoT routes score 3.43% and 10.30% with this interface, versus 100.00% for both when the reference expression is supplied explicitly. All post-training comparisons use the explicit-expression interface. Tables 16 and 17 report accuracies and paired route differences.

Table 16: Qwen diagnostic accuracies (%) on 165 Swahili and 164 Bengali instances. Trained-model values average three seeds; each base value is evaluated once.

Language	Checkpoint	A_{auto}	A_{cond}	A_{cont}
Swahili	Base	53.33	98.79	64.85
Swahili	Full-CoT A320	64.44	100.00	72.53
Swahili	AccessPath A320	65.86	99.60	71.72
Swahili	F→F B320	60.81	100.00	73.13
Swahili	A→F B320	64.85	100.00	73.13
Bengali	Base	50.00	98.17	56.10
Bengali	Full-CoT A320	64.43	99.80	73.58
Bengali	AccessPath A320	67.28	99.39	69.11
Bengali	F→F B320	66.06	99.80	73.17
Bengali	A→F B320	67.48	99.80	73.58

Table 17: Paired Qwen diagnostic differences, AccessPath origin minus Full-CoT origin (pp). Intervals describe variation over three training seeds on the fixed cohort.

Lang.	Stage	Measure	Seed 0	Seed 1	Seed 2	Mean	95% CI
sw	A320	Autonomous	+2.42	-4.85	+6.67	+1.41	$[-13.05, 15.88]$
sw	A320	Conditional	-0.61	+0.00	-0.61	-0.40	$[-1.27, 0.47]$
sw	A320	Continued	+1.82	-1.82	-2.42	-0.81	$[-6.51, 4.89]$
sw	B320	Autonomous	+2.42	+2.42	+7.27	+4.04	$[-2.91, 10.99]$
sw	B320	Conditional	+0.00	+0.00	+0.00	+0.00	$[0.00, 0.00]$
sw	B320	Continued	-4.24	+1.21	+3.03	+0.00	$[-9.40, 9.40]$
bn	A320	Autonomous	+4.88	+3.66	+0.00	+2.85	$[-3.46, 9.15]$
bn	A320	Conditional	-0.61	-0.61	+0.00	-0.41	$[-1.28, 0.47]$
bn	A320	Continued	-3.66	-6.10	-3.66	-4.47	$[-7.97, -0.97]$
bn	B320	Autonomous	+0.00	+5.49	-1.22	+1.42	$[-7.45, 10.30]$
bn	B320	Conditional	+0.00	+0.00	+0.00	+0.00	$[0.00, 0.00]$
bn	B320	Continued	-3.05	+3.66	+0.61	+0.41	$[-7.94, 8.75]$