PERFECT LABELS FOR FREE? WHAT EXACT ANSWERS AND EXACT TRACES TEACH A SMALL LANGUAGE MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Exact verifiers (primality tests, SAT solvers, unit tests) label data without noise and supply the rewards of verifiable-reward post-training. Because every label is exact, a difference between two training regimes on the same inputs is a difference in the supervision format. We fine-tune Qwen2.5 models (0.5B to 7B; main size 1.5B, three seeds for the core answer and trace cells) with LoRA on exactly labeled decision tasks and compare supervision by the answer alone with a deterministic procedure trace followed by the answer. *Exact answers teach shortcuts*: on 4-digit primality the 1.5B answers (three seeds) follow "the last digit is 1, 3, 7 or 9" on 99.2–100.0% of test items, and on divisibility by 7 or 13 answer-only training stays at chance or constant at every size and configuration we ran ($n \leq 540$, including a logged sweep of four configurations, eight runs, none collapsing), while the same recipe learns divisibility by 3. *Exact answers remove computation from the answers*: at 3B and 7B the zero-shot model already divides by 7; answer-only fine-tuning drops it to chance in both seeds, and under a chain-of-thought prompt the answer-only model still answers at once at every size tested (at least 97.9% of items in three tokens or fewer) and stays at chance, while fine-tuning on its own verifier-filtered traces keeps the division (two seeds at each size); whether the capability survives is untested. *Exact traces teach steps within a task*: accuracy rises with the number of traces and with how often each table entry was supervised, a count-based prediction across tasks failed because steps differ in difficulty, and only the input's own trace, written before the answer, helps. A second model family (SmolLM2-1.7B-Instruct) shows the same direction on the three tasks run, but its timestamped prediction (S11) failed its 20-point threshold in one seed. Predictions were logged in timestamped commits before their cells ran; four of twelve failed (the cross-task dose prediction S9, the length prediction S5, the self-generated-trace prediction S10 and the second-family prediction S11), and all are reported.

1 INTRODUCTION

Verifiable-reward post-training rests on programs that decide correctness exactly: answer checkers for mathematics, unit tests for code, solvers for logic (Lambert et al., 2024; DeepSeek-AI et al., 2025). Such verifiers are often described as a source of perfect labels for free, and a verified label is never wrong. Whether a model trained on those labels learns the computation the verifier performs, or a cheaper function that agrees with it on the training distribution, is a separate question, usually confounded with label noise, task difficulty and supervision format.

With exact labels the first confound is gone. We choose small decision tasks whose structure is known in advance (which surface rules approximate the label, which steps the natural procedure takes), so that we can tell which function the model learned, and what exact supervision of a given format teaches.

Answer-only supervision (arm A) teaches a shortcut (Figure 1b). On 4-digit primality the 1.5B A answers (three seeds) coincide with "the last digit is 1, 3, 7 or 9" on 99.2–100.0% of test items, and where no surface rule exists (divisibility by 7 or 13) A stays at chance at every training size

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.

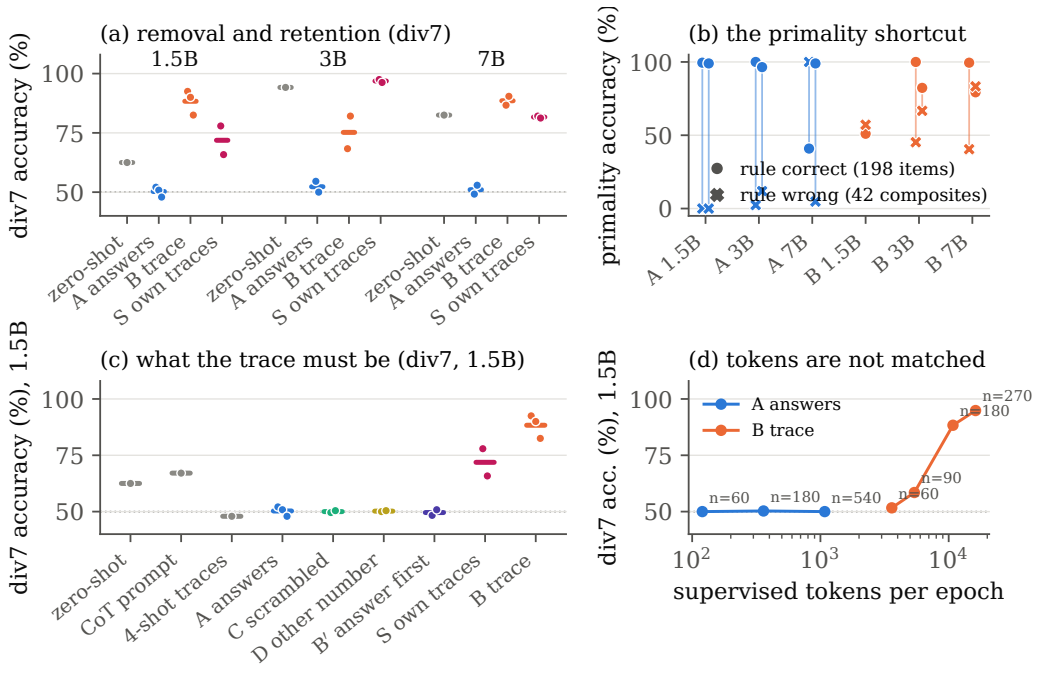

Figure 1: (a) Divisibility by 7: zero-shot accuracy, answer-only fine-tuning (A), trace fine-tuning (B) and fine-tuning on the model's own verifier-filtered traces (S) at 1.5B, 3B and 7B; points are seeds, bars means. Where the base model already divides (3B 94.2%, 7B 82.5%), A drops it to chance (3B 54.6/50.0, 7B 49.2/52.9) and S keeps it (3B 97.5/96.2, 7B 82.1/81.2). (b) Primality: accuracy of A and B on the 198 test items where the rule "last digit is 1, 3, 7 or 9" is right (circles) and on the 42 composites where it is wrong (crosses); at 1.5B and 3B, A scores 96.5–100.0% on the first group and 0.0–11.9% on the second; 7B seed 0 is the collapsed seed that answers "No". (c) Controls on div7 at 1.5B: scrambled traces (C), a correct trace of a different number (D) and the trace placed after the answer (B′) stay at chance, and a zero-shot CoT prompt (67.1%) or 4 trace demonstrations (47.9%) do not approach the model's own verifier-filtered traces (S) or the procedure trace (B). (d) The same arms against supervised completion tokens per epoch (whitespace tokens): under the shared configuration A never receives B's token budget, so B−A confounds format with tokens; C and D in (c) match B's tokens and stay at chance.

we ran. At 3B and 7B, where the zero-shot model already divides correctly, answer-only fine-tuning removes that computation from the answers in both seeds, while fine-tuning on the model's own verifier-filtered traces keeps it (Figure 1a); whether the division is lost or only suppressed is left open. *Trace* supervision (arm B: a deterministic procedure trace, then the answer) teaches the procedure once its steps are supervised often enough (Figure 2a): on divisibility by 7, accuracy rises from 51.7% at $n = 60$ training examples to 94.9% at 270 (means over three seeds), and only the input's own trace, placed before the answer, does this (Figure 1c). It is not a free improvement: at 1.5B the primality trace hurts, on argument validity answers match or beat traces (99.2% vs 88.3%), and at 3B and $n = 180$ the long-division trace underperforms the zero-shot model.

Contributions.

- Answer-only fine-tuning on exact labels learns the surface rule on primality (at 1.5B and 3B and in one of two 7B seeds), stays at chance where no surface rule fits (div7, div13) at every size and configuration we ran ($n \leq 540$; a logged sweep of four configurations, eight runs, none collapsing, S12) and, at 3B and 7B, replaces the base model's working division with a guess in both seeds; under a CoT prompt the answer-only model still answers at once and stays at chance at 1.5B, 3B and 7B (whether the capability survives is untested), while the model's own verifier-filtered traces keep the division (§4.1, Table 4).

- Trace supervision teaches the procedure within a task once its steps are supervised often enough: accuracy rises with training traces and with per-entry coverage, while a count-based prediction across tasks failed, exposing step difficulty as a confound of supervision frequency (§4.2).
- Controls and exact audits: the trace must be the input's own computation, placed before the answer, so trace semantics are load-bearing in fine-tuning where Valmeekam et al. (2026) found them dispensable when training from scratch (§4.3); trace-trained models also write shortcuts behind unfaithful steps, which the audits catch (§4.2, §5); every failed prediction is reported (Table 7).

2 Related work

Intermediate steps as supervision. Scratchpads (Nye et al., 2021) and chain-of-thought (CoT) prompting (Wei et al., 2022) improve multi-step computation, mainly on mathematical and symbolic tasks (Sprague et al., 2025). Data format changes the sample efficiency of arithmetic learning (Lee et al., 2024), fine-tuned models can learn arithmetic with decomposed steps (Liu & Low, 2023), and GCD predictions follow rules built from divisors of the number base (Charton, 2024), the kind of surface rule we probe for; procedure cloning (Yang et al., 2022) imitates the intermediate computation as our arm B does, and surface form limits arithmetic in pretrained transformers (Nogueira et al., 2021). Lee et al. (2024) also fine-tune pretrained models and find that detailed scratchpads beat plain answers; their labels are exact too. What is new here is not the labels but the instruments: a named surface rule and a fitted probe that say *what* answer-only training learned, controls that give a correct trace of the wrong number or place the trace after the answer, and the observation that answer-only fine-tuning removes a computation the base model already performs. Errors compound across steps (Dziri et al., 2023); filler tokens can replace a chain of thought on some tasks (Pfau et al., 2024), and our scrambled control keeps the tokens but removes their order. Reasoning helps when training data are local, so that intermediate variables are seen together often (Prystawski et al., 2023); our dose results are a within-task instance of that locality account, and they show where it stops predicting across tasks. Distilling rationales into small models (Hsieh et al., 2023; Ho et al., 2023; Magister et al., 2023) trains on traces before answers; placing explanations after the answer (Lampinen et al., 2022) and internalizing steps (Deng et al., 2023; 2024) are the alternatives our answer-first arm B′ probes, and on div7 it does not transfer the procedure. Transformers can also learn multi-hop composition implicitly after long training (Wang et al., 2024); our answer arm, at a few hundred examples, does not.

Theory. Parity-like targets give gradient-based learners almost no signal from end labels (Shalev-Shwartz et al., 2017). Sub-task decomposition makes otherwise unlearnable tasks learnable (Wies et al., 2023); CoT raises the expressivity of bounded-depth transformers (Feng et al., 2023; Li et al., 2024; Merrill & Sabharwal, 2024), next-token predictors trained on CoT data are universal learners (Malach, 2024), and CoT induces sparse, sample-efficient dependence (Wen et al., 2025). Transformers can represent automata through shallow shortcuts (Liu et al., 2023). The globality degree of Abbe et al. (2024) bounds what transformers learn efficiently; teacher-forced intermediate parities make parity learnable (Kim & Suzuki, 2025). Appendix F uses these results as intuition only.

Traces and verifiers in post-training. Transformers trained from scratch on verifiable traces depend less on trace semantics than usually assumed (Valmeekam et al., 2026), and CoT explanations can be unfaithful to the computation behind the answer (Turpin et al., 2023); our exact audits measure this in fine-tuned traces. Process supervision rewards steps rather than outcomes (Lightman et al., 2024); program-generated logic corpora improve reasoning (Morishita et al., 2024). STaR (Zelikman et al., 2022) and rejection-sampling fine-tuning (Yuan et al., 2023) train on self-generated rationales that reach verified answers, as our arm S does; RL with verifiable rewards drives recent reasoning models (Lambert et al., 2024; DeepSeek-AI et al., 2025). Fine-tuning on model-generated positives can amplify spurious correlations (Setlur et al., 2024); SFT and RL generalize differently (Chu et al., 2025); RLVR mostly reweights what the base model already samples (Yue et al., 2025). Our answer arm is an instance of shortcut learning (Geirhos et al., 2020); exact labels do not prevent it.

Table 1: Tasks. k: steps in the trace (prime: for primes; a composite's trace stops at its smallest factor). Tokens: mean whitespace tokens per trace. m: supervised transitions per (remainder, digit) table entry at $n = 180$.

task	trace procedure	k	tokens	m
prime	trial division by the primes $\leq \lfloor\sqrt{n}\rfloor$	11–25	71.5	–
div$d$, 4 digits	long division, $r \leftarrow (10r + x_t) \bmod d$	4	58.0	36.0 (d=2) … 5.5 (d=13)
div7, 6 digits	same, $d = 7$	6	76.0	15.4
valid	Z3 countermodel or unsatisfiable core	certificate	37.0	–

Table 2: Arms (completions trained on; the prompt is shared) and the labels of the tests. H1–H4: endpoints of a first preregistration (prior grid). S1–S12: timestamped predictions of the second file (Table 7).

arm	completion
base	none (zero-shot); also a zero-shot CoT prompt and 4 trace demonstrations
A	"Answer: Yes" or "Answer: No"
B	the procedure trace, a newline, the answer
C	B's traces permuted across examples and token-shuffled; the answer keeps the true label
D (S8)	a fluent, correct trace of a *different* number with the same label
B′	the answer first, then the trace
S (S10)	the base model's own sampled solution, kept when the exact gate accepts its final answer (STaR-style)
A-full	answers on the whole training pool

3 SETUP

Tasks. Each task has an exact decision procedure, so labels are correct by construction; arms see the same prompts and labels and differ only in the completion (Tables 1 and 2). **prime**: is a 4-digit n prime? The trace performs trial division by the primes up to $\lfloor\sqrt{n}\rfloor$, one "$n \bmod p = r$." step per prime, stopping at the first divisor. **div**d: is n divisible by d? The trace performs long division, $r \leftarrow (10r + x_t) \bmod d$ for each digit x_t, for $d \in \{2, 3, 7, 11, 13\}$ with 4 digits and $d = 7$ with 6. **valid**: does a propositional conclusion follow from its premises (Z3 (de Moura & Bjørner, 2008))? Arguments instantiate 12 forms shared between training and test, so the task is surface-learnable by design (Appendix C). Test sets have 240 items (120 per class), are problem-disjoint from training and identical across arms; training sets are balanced samples of size n drawn with the run's seed.

Training and evaluation. LoRA (Hu et al., 2022) with $r = 8$, $\alpha = 16$, dropout 0.05 on the attention projections of Qwen2.5-Instruct (Qwen et al., 2024) (0.5B, 1.5B, 3B, 7B); AdamW, learning rate 2×10^{-4}, batch size 1, 3 epochs, bf16; loss on completion tokens only; one configuration for every arm, not tuned per arm. Evaluation is greedy generation with one answer extractor for every arm; an unparsed or truncated answer counts as wrong. Details and generation caps: Appendix B.

Statistics and timestamped predictions. We lead with the difference of two arms and its 95% bootstrap interval, resampling items and seeds jointly, and give the number of seeds; with one to three seeds per cell these intervals are item-level, and the paper makes no seed-level inference; per-seed exact McNemar tests (McNemar, 1947), Holm-corrected (Holm, 1979) across tasks within a comparison, model and seed at $\alpha = 0.05$, are secondary, because they count test items, not training runs. A first preregistration, written before a prior grid of 36 runs according to the history of the repository it was written in, fixed H1–H4. A second file, committed at 20:29:47 on 2026-09-25 before any of its runs, fixes S1–S6; its append-only decision log adds S7–S12, each committed before any cell it governs had started (Appendix E). In short: H1–H4 were fixed before the prior grid; S1–S6 before any run of this study; S7–S12 were logged during the study, each before its own cells, and we call them *timestamped* rather than preregistered; S3 and S6 were effectively settled before registration. S1–S10 were committed between 20:29:47 and 21:46:45 on 2026-09-25, with other cells running concurrently; S11 and S12 at 01:35:27 and 02:41:32 on 2026-09-26, each before any of its cells ran. S7–S12 were proposed by AI after earlier results were in (S12 by an internal

Table 3: Accuracy (%) at $n = 180$, Qwen2.5-1.5B-Instruct, 240 test items per task; mean over seeds $\pm$ SD across seeds (seeds per cell: Table 6; no SD: one seed). probe: the best surface probe fitted on A's training items (§4.1). B−A: points, with a 95% bootstrap interval over items and seeds. max p: largest per-seed exact McNemar p (B vs. A); in parentheses where A > B. –: not run.

task	base	probe	A	B	C	B−A [95% CI]	max p
prime	59.6	89.7	**82.1**±0.4	55.8±3.3	52.1±2.9	-26.2 [-34.9, -17.5]	(5.7×10^{-7})
div7	62.5	46.5	50.3±2.1	**88.3**±5.2	50.0±0.6	+38.1 [+31.1, +43.9]	6.8×10^{-13}
valid	30.0	100.0	**99.2**±0.8	88.3±10.4	52.5±2.4	-10.8 [-20.8, +0.4]	(0.50)
div3	78.3	50.7	54.0±7.3	100.0±0.0	–	+46.0 [+37.9, +54.3]	1.6×10^{-27}
div11	55.0	50.3	50.1±0.9	99.4±0.5	–	+49.3 [+43.3, +55.1]	7.2×10^{-34}
div13	50.4	53.3	50.4±0.8	53.9±4.4	46.5±2.1	+3.5 [-4.4, +12.1]	0.86
div7 (6 digits)	52.5	48.1	49.4±2.1	93.1±4.4	–	+43.8 [+34.6, +52.5]	4.5×10^{-23}

AI review pass, not peer review; see the AI use statement). Eight of S1–S10 were at risk (all but S3 and S6). Of all ten, S1–S4, S6 and S8 passed, S5, S9 and S10 failed, and S7 met its 4-of-5 rule with three of four informative cells; two later entries, S11 (a second model family, §5) and S12 (an answer-only sweep, §4.1), were logged in the final hours; S11 failed on its threshold and S12 passed (Table 7, Appendix A). The surface probe and every analysis labelled post hoc belong to neither file.

4 Results

4.1 Exact answer labels teach the surface and erase computation

The primality shortcut (descriptive, post hoc). A reaches 82.1% on prime (seeds 0–2), and its answers coincide with "the last digit is 1, 3, 7 or 9" on 99.2–100.0% of test items; that rule alone scores 82.5%. A classifies 99.7% of primes and 64.4% of composites correctly, and of the 42 test composites ending in 1, 3, 7 or 9 it gets 0, 0, 0 right in seeds 0–2, including 27 with a factor 3 or 7 that any sieve would catch. On a dedicated set of 120 odd composites without a factor ≤ 7 and as many primes (S6, passed, settled in advance; the file's rule, "odd and no factor ≤ 7", coincides with the last-digit rule on that set), A reaches 49.6% and agrees with the rule on 98.8% of items. Split by the rule itself (Figure 1b), A is right on 96.5–100.0% of the 198 test items where the rule is right and on 0.0–11.9% of the 42 composites where it is wrong (4 cells at 1.5B and 3B). We named the rule after seeing the prior grid, so this analysis describes rather than tests.

Answers track surface probes. We fit logistic regression, a one-hidden-layer MLP and k-nearest neighbours on the same 180 training items as arm A, on 40 one-hot digit-by-position features alone or with engineered divisibility features (last digit even or 0/5; digit sum mod 3 and 9; alternating digit sum mod 11), or for arguments on atom-renamed word and character n-grams, and choose the probe by 5-fold cross-validation; on div2, div3 and div11 the engineered features contain the answer, so there the probe is one-hot only. The probe is a property of the data, a lower bound on what a surface learner can decode from the training items, not a property of the model, and it is informative only where it sits between chance and ceiling. A tracks the probe (Figure 2c): on prime it lies between the one-hot linear probe (79.3%) and the best probe (89.7%), on div2 both are at ceiling (99.6% and 100.0%) and on every other divisibility task both are at chance, and on valid both are near ceiling, where agreement is uninformative.

More answers do not buy the program. On prime, A reaches 59.6%, 82.1% and 75.0% at $n =$ 60, 180 and 540, and 80.8% on the full pool. On div7 it stays at chance at every n (50.0%, 50.3%, 50.0%) and on div13 likewise (50.4% and 50.0% at 180 and 360), and at $n = 540$ (div7) and $n = 360$ (div13) A collapses to a constant answer under the untuned configuration ("No", except 1.5B div13 seed 1 at $n = 360$, which answers "Yes" on every item) (Limitations); on div7 it falls below the zero-shot base (62.5%; A−base -12.2 [-19.3, -4.9] points over 3 seeds; significant after Holm correction in 2 of 3 seeds, $p_{\text{Holm}} = 0.24,\ 0.018,\ 1.9 \times 10^{-4}$). With labels permuted across items, the earlier study's answer arm sat at 48.5–52.5% on all seven of its tasks (Appendix A): correct labels drive A, and on primality what they buy is the rule.

Is A's failure on div7 the configuration? (S12, timestamped). Every A cell above uses the one shared configuration, under which A sees 14–47 times fewer supervised tokens than B (counted with the model's tokenizer) and sometimes collapses to a constant answer. A sweep logged before it ran (S12) trains A on div7 at $n = 180$ with learning rates 2×10^{-5} and 5×10^{-5} and 10 or 30 epochs, two seeds each, which gives A 3,600 to 10,800 supervised whitespace tokens in total (the unit of the run records), so that at 30 epochs A sees as many whitespace tokens as B sees per epoch on div7; the rule was that no configuration reaches 65%, at least one does not collapse, and div3 A reaches 90%. Result: the best of 8 answer-only runs (4 configurations x 2 seeds) reaches 53.3% on div7 (lr 5e-5, 30 epochs, seed 1), 4 of 4 configurations avoid a constant answer in both seeds, and the div3 control reaches 92.1%; S12 passed.

Answer-only fine-tuning removes the base model's division (3B, 7B). The 3B and 7B zero-shot models already decide divisibility: 94.2% and 92.1% on div7 and div13 at 3B, 82.5% and 80.0% at 7B. The 3B base model writes a direct division line ("$2740 \div 7 = 391.43\ldots$") on 100.0% of its div7 answers. Answer-only fine-tuning removes it (Figure 1a): div7 A falls to 52.3% at 3B (A−base -41.9 [-48.8, -34.6]) and 51.0% at 7B (-31.5 [-39.2, -23.3]), and div13 A to 49.2% and 46.7% (two seeds each; Table 6). The mechanism is not a learned rule: the 3B answer-only runs collapse to a constant or near-constant answer in three of four runs (Table 6) and the 7B runs guess without structure, and no divisibility A run agrees with any surface rule beyond $\kappa = 0.24$ (post hoc audit, Appendix A). Item by item, A keeps only 52.4% (3B) and 50.8% (7B) of the div7 items the base solved, the share a guesser would keep, while S keeps 99.8% and 92.9% (3B: two seeds; 7B: seed 0) (post hoc paired audit, Appendix A).

The model's own verifier-filtered traces keep it (arm S). Fine-tuning on the model's own verifier-filtered solutions (arm S) keeps the computation: at 3B, S reaches 96.9% on div7 (97.5, 96.2), its errors fall on the base's own failures, its few fixes of those failures are not significant after Holm correction, and its kept traces are right (the post hoc re-audit parses 97.7% and finds 96.4% fully correct; the training script's narrower audit parsed 15.0% of kept traces and counted 13.9% of them fully correct in seed 0). At 7B, S gives 81.7% (two seeds), level with the base (seed-0 discordant pairs 13:14). The foreign trace's dependence on size and the 3B dose replication are in Appendix A.

A chain-of-thought prompt does not bring it back. Removing the intermediate tokens removes serial computation by construction (Merrill & Sabharwal, 2024), so we asked whether the answer-only model still divides when asked for a chain of thought. It does not, and it does not try: at 1.5B, every answer-only adapter we tested (div7, two seeds; div13; prime) answers the CoT prompt in at most 3 tokens on every item, where the base model writes a median of 65 tokens under the same prompt. The div7 adapters score 55.0% and 53.3%, near-constant answers in opposite directions (base with the prompt 67.1%); the div13 adapter stays constant at 50.0% (the base reasons under the prompt but is near chance on div13, 54.6%); the primality adapter keeps its shortcut accuracy (82.5%). At 3B and 7B (seed 0, one run per cell) the answer-only adapters do the same (A adapters retrained inside the CoT run with the same data, seed and configuration): at least 97.9% of items answered in three tokens or fewer, div7 53.3% and 55.0%, div13 50.0% and 46.2%. Answer-only fine-tuning suppresses the reasoning behaviour itself; whether the capability underneath survives cannot be told from this test.

4.2 Exact traces teach steps seen often enough

Divisibility by 7. At $n = 180$, over three seeds, B reaches 88.3% (92.5, 90.0, 82.5) against A's 50.3% (B−A +38.1 [+31.1, +43.9]), and C stays at 50.0%. Seeds 1 and 2 are the preregistered replication (S1 passed); a deterministic re-execution of seed 0 gave 92.5%. The gain holds on both classes (82.2% on multiples, 94.4% on non-multiples).

Dose. With the trace format fixed, accuracy rises with the number of training traces while A stays flat (Figure 2a, where hollow markers are single seeds). Div7 B goes from 51.7% at $n = 60$ to 58.5% at 90 (seed-0 per-step accuracy $q = 0.736$), 88.3% at 180 and 94.9% at 270 (seed-0 $q = 0.997$), three seeds at each n (per seed: Table 6). Div13 B, with the same 4 steps, goes from 53.9% at 180

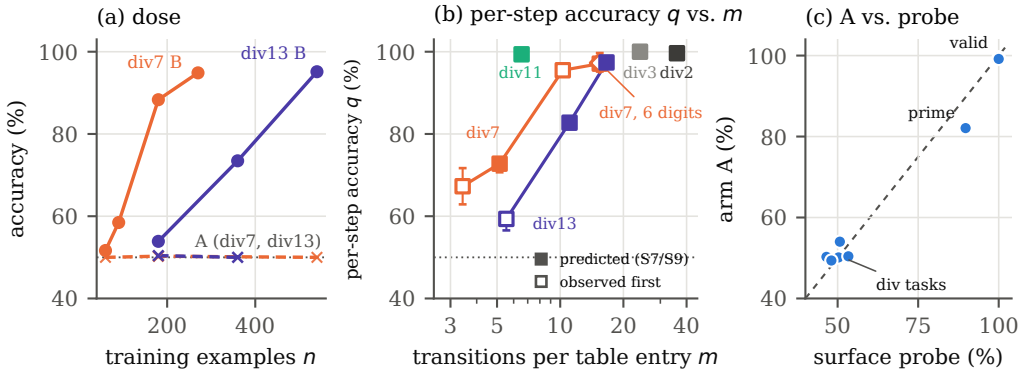

Figure 2: (a) Accuracy against training examples n for div7 and div13 at 1.5B: trace arm B (solid) and answer arm A (dashed; chance at every n); filled markers are means over two or three seeds, hollow markers single seeds. (b) Arm B per-step accuracy q (the probability that a step is right given that the previous one was; seed means, bars show the seed range) against $m = kn/(10d)$, the supervised transitions per (remainder, digit) table entry, known before training; lines join the training sizes of div7 (orange) and div13 (violet); the 6-digit div7 cell (diamond) shares m with div7 at $n = 270$. Filled: cells predicted in advance by S7 or S9; hollow: cells observed first. Answer accuracy is in panel (a). (c) Answer arm A against the surface probe fitted on the same training items (one-hot features where engineered features encode the answer); dashed line: equality; points at chance or at ceiling are uninformative, and prime is the one informative point.

(seed-0 $q = 0.566$) to 73.5% at 360 and 95.1% at 540 (seed-0 $q = 0.980$), three seeds each.[1] At $n = 180$, div13's failure is visible in its seed-0 generations: 2.1% of traces are correct and 179 of 235 wrong traces end at $r = 0$, so this seed answers "yes" 76% of the time. In div7 generations (seed 0), digits and carried remainders are copied without error (100.0%, 100.0%); every error is a wrong lookup of $(10r + x) \bmod 7$.

Timestamped dose predictions (S7, S9): held within tasks, failed across tasks. S7 predicted that q follows $m = kn/(10d)$ in five cells and passes if four hold. The three within-task cells hold: (a) div13 at $n = 540$ ($m = 16.6$), $q = 0.980$ and accuracy 94.6% (seeds 1 and 2, run later: 95.0% and 95.8%); (b) div7 at $n = 270$ ($m = 15.4$), $q = 0.997$; (c) div7 at $n = 90$ ($m = 5.1$), $q = 0.736$ and accuracy 62.1%. Cell (e), div3 and div2, holds only at ceiling ($q = 1.000$, 0.996). (d) Div11 at $m = 6.5$ reached $q = 0.994$, above the predicted interval: $10 \equiv -1 \pmod{11}$ makes its step a subtraction. S9 (failed): div13 at $n = 360$ ($m = 11.1$) reached 75.4%, 72.5% and 72.5% (seeds 0–2) against the required 80%, while div7 at the similar $m = 10.3$ reached 88.3%; the div13 step, $r' = (x - 3r) \bmod 13$, is harder. S7 thus passed by its 4-of-5 rule (4/5) only because cell (e) held at ceiling; of the four informative cells, the three within-task cells held and the cross-task cell failed, and that is how we report it. The supported reading is narrow: within a task more supervised transitions raise q; across tasks step difficulty shifts the curve, and m does not transfer across tasks or sizes (post hoc fits of q on $\ln m$: $R^2 = 0.52$ at 1.5B, 0.38 at 3B). S2 passed at 1.5B: B−A ≥ 15 points in the two preregistered seeds for div3 (+50.0, +37.5) and div11 (+48.3, +50.0), not for div13 (+5.0, -1.2).

Coverage at the entry level (post hoc). S7 counts supervised transitions per table entry on average; an audit written after the runs counts them per entry. For every step a trace-trained model wrote, let c be the number of times that exact (remainder, digit) entry appeared in its own training traces. Over 36,897 steps from every arm-B cell with saved generations (all sizes, all divisibility tasks), lookup accuracy rises with c from 21.5% at $c = 0$ to 100.0% at $c \geq 33$ (Table 5, Appendix A); within cells, the rank correlation between an entry's c and its accuracy is positive in 25 of 29 cells (median $\rho = 0.23$, Wilcoxon $p = 9 \times 10^{-6}$; the draw of training items also sets their order and the

[1] Sensitivity check, not the preregistered measure: excluding the near-trivial first step ($r_1 = x_1 \bmod d$), seed-0 q is 0.237 instead of 0.566 for div13 at $n = 180$ and 0.602 instead of 0.736 for div7 at $n = 90$.

initialization). Entries never seen in training are right in 39 of 39 div11 steps, consistent with a rule ($10 \equiv -1 \pmod{11}$), and in 1 of 74 div7 steps, consistent with lookup. At the same size, n and c, div13 entries are learned more slowly than div7 entries (1.5B, $n = 180$, $c = 5$–8: 22.5% against 92.7%; Table 5), which is the step-difficulty confound of S7 seen directly. Arm D shows the other half of what a trace teaches: its lookups are 85.9% correct as table entries but only 6.9% correct for the input's own digits, so D learned the table and not to read the number.

Length (S5): failed as preregistered. The prediction for 6-digit div7 was B $\approx q^6 = 75.8\%$, with $q = 0.955$ measured on the seed-0 4-digit generations. The primary cell, B trained and tested on 6-digit traces, reached 96.2% and 90.0% in seeds 0 and 1, outside the 10-point tolerance: S5 failed. Its own seed-0 q is 0.988: at $k = 6$ the same n gives more transitions per entry ($m = 15.4$). The secondary cell, the 4-digit model applied to 6-digit inputs, gives 77.1%, within the tolerance, but its own q is 0.906 and only 48.3% of its traces are fully correct, so the match is not evidence for compounding (post hoc forms: Appendix H).

Primality: the trace hurts at 1.5B. B reaches 55.8% against A's 82.1% (seeds 0–2; B−A -26.2 [-34.9, -17.5]), and B vs. C is not significant ($p = 0.22$, 0.33). B classifies 19.7% of primes and 91.9% of composites correctly. Class and trace length are confounded here: every prime needs the full sequence of trial divisions, and accuracy falls with the true number of divisions (100, 89, 71, 26 and 21% for 1, 2–3, 4–8, 9–16 and 17 or more). Scrambled C does not separate the two: seed 1 shows B's split, seed 0 the opposite (99.2% of primes, 0.8% of composites). At $n = 540$, B reaches 68.8% against A's 75.0% (one seed). A trial-division step is a table indexed by (n, p) with more than 10^5 entries, so $m \ll 1$ at these n. At 3B the trace arm scores higher (85.0%) without executing its trace: only 23% of its "$n \bmod p = r$" claims are correct, seed 0 agrees with "no factor ≤ 5, so prime" on 100.0% of items, and on the hard set it scores 50.4% and 53.8%: a shortcut written in the form of a trace (cf. Turpin et al., 2023).

Validity and cost. On valid, answers match or beat traces (A 99.2%, B 88.3%, C 52.5%; B−A -10.8 [-20.8, +0.4]; per seed $p = 0.50$, 7.1×10^{-15} and 4.7×10^{-10} for seeds 0–2). B's completions contain 14–47 times as many tokens as A's (Qwen tokenizer).

4.3 Controls (div7, 1.5B)

The trace must be the input's own computation. D, a fluent and correct trace of a different number with the right label, gives 50.0% and 50.4% (S8 passed): a correct trace of another number does not help. Scrambled C stays at 50.0%. C controls for token count but not for interference: it also fails where A succeeds (valid 52.5% vs. 99.2%, prime 52.1% vs. 82.1%), so C alone does not show that the trace's information is what helps.

The trace must come before the answer. B′, trained with the trace after the answer, gives 48.3% and 50.8%, like A, although it still writes a trace after answering with $q = 0.841$.

Arm S at 1.5B (S10). Sampling from the base model's CoT prompt and keeping completions the gate accepts covers 159/180 training items (keep rate 56.9% of samples). S10 predicted div7 accuracy $\leq 65\%$ in seeds 0 and 1, fewer than half of kept traces with all remainders correct, and valid $\geq 90\%$. Result: arm S reaches 77.9% and 65.8% on div7 (seeds 0 and 1) and 64.6% on valid, against 67.1% for the base model's CoT prompt; S10: failed on its accuracy clauses. The failure is informative: arm S stayed above A in both seeds, and in seed 0 above the base model's own CoT prompt, so outcome-only filtering supplied traces that helped, against the prediction. The trace audit depends on its parser: the training script's audit counts 36.5% of kept traces as fully correct, while a post hoc re-audit that also reads other division formats counts 62.9%, so the trace clause of S10 holds under the preregistered parser and fails under the post hoc one; we score S10 by its accuracy clauses, which fail either way, and report the re-audit as a check rather than a verdict.

5 Scale

Table 4 and Figure 3 (Appendix A) repeat the core cells at 0.5B, 3B and 7B. Three results replicate. Where A finds no surface rule (div7, div13, div11, 6-digit div7), answer-only training stays at chance

at every size where it was run (Table 4). The primality shortcut persists: at 3B, A reaches 82.3% and agrees with the last-digit rule; at 7B one seed does the same (82.5%) while the other collapses to "No" on 237 of 240 items, against B's 84.6% (89.2, 80.0). Trace supervision beats answers on div7 at every size (S4 passed; B−A +22.9 [+9.6, +37.5] at 3B and +37.5 [+31.2, +44.2] at 7B, two seeds each), and the div7 controls replicate: D gives 50.0% at 3B and 49.2% at 7B, B′ 49.6% and 50.0% (seeds and Yes-rates per cell in Table 6; at 3B the D runs and one B′ run are constant-output); div11 at 7B behaves as at 1.5B (Table 4).

Some cells move with size. At 3B, div3 becomes learnable from answers alone (A 99.2%, against 54.0% at 1.5B), so S2 fails at 3B, and the S9 threshold is met (B 83.3%, A 50.0%). At 7B the trace arm collapsed on valid in seed 0 (58.3%; seed 1 reached 97.5%): all 100 of that seed's 100 errors are invalid arguments answered "yes", 99 of them after a trace asserting a contradiction, the certificate's form without its check. The suppression result and its 3B dose replication are in §4.1.

A second model family (S11). A prediction logged before its cells ran required, on SmolLM2-1.7B-Instruct (the logged fallback; the first choice was gated), div7 B−A ≥ 20 points in two seeds and the grounding control D within 10 points of A. Result: on SmolLM2-1.7B-Instruct, traces beat answers on div7 by +15.0 and +22.5 points (seeds 0 and 1), significant in both, so the 20-point threshold fails in seed 0; the grounding control stays within 6.7 points of A; S11 failed. The direction replicates on every task (div7 A 45.0/47.1, B 60.0/69.6; prime A 68.3 vs B 62.5 and valid A 99.6 vs B 80.4, one seed each, the prime difference not significant with 40 vs 26 discordant items), with a smaller div7 gain at $n = 180$ than in Qwen2.5-1.5B, as the dose results would predict; a larger-n point in this family was not run. An automaton view of m is in Appendix F.

6 Discussion and limitations

What exact labels guarantee. An exact verifier guarantees that no label is wrong, not that the labels identify its function among those a model can fit. The rule A learned on primality calls every composite ending in 1, 3, 7 or 9 prime, including all 996 4-digit composites without a factor ≤ 7. Shuffled labels cannot tell that function from a surface rule; inputs where the rule fails can.

Answer-only fine-tuning suppresses the base model's test-time computation. Labels are exact in every arm and we ran no noisy-label arm, so the effect comes from the supervision format. At 3B and 7B, answer supervision replaced a working computation with a guess, in both seeds; at 1.5B, A falls below the zero-shot base on div7 in the same direction (significant after Holm correction in 2 of 3 seeds). Fine-tuning on the model's own verifier-filtered solutions kept the computation (3B, two seeds; 7B, two seeds). A pipeline that fine-tunes a capable model on verified answers alone should compare against the model's own zero-shot accuracy, not only against chance.

Supervision design: hypotheses for verifiable-reward pipelines. Our results suggest, without establishing beyond these tasks: (1) fit a cheap surface probe on the verified data before training, and evaluate where it is wrong. (2) A correct trace is not enough: budget how often each step type is supervised, expect harder steps to need more, and check per-step accuracy after a small run. (3) The trace should be the input's own computation and precede the answer (div7 only). (4) Check the base model first: where its own steps are right, its verifier-filtered traces preserved them here (two seeds at 3B and at 7B). (5) Do not read a null from one training size: our 60-example null on div7 was the bottom of a dose curve. We ran no RL; arm S is the nearest experiment (Appendix G).

Limitations. One model family for all but one test (the second family covers one size and one n), with 1.5B as the main size, and one hyperparameter configuration shared by all arms (Appendix B), not tuned per arm; answer-only failure may reflect sample inefficiency under this configuration rather than impossibility, and A was tested only up to $n = 540$. B sees 14–47 times more supervised tokens than A (tokenizer count), and the token-matched comparison (H3) was not run on div7. Binary tasks, 240 test items, one to three seeds per cell, one trace format per task. Some base cells are lower bounds (capped generations) and some A seeds collapse to a constant answer (Appendix B). Four of twelve timestamped predictions failed at the size they were logged for, and S2 and S5 also fail when rerun at 3B (Table 7), m is confounded with step difficulty, and the trace audits depend on the parser (§3, §4.3).

AI USE STATEMENT

We used AI assistants (Claude, Anthropic) to write the experiment code, including the deterministic generators of the synthetic task data; to run and log experiments under our direction; for statistics and for interpreting results in the decision log; for the intuition in Appendix F and the derivation in Appendix H; for related-work search and reference checking; for drafting and editing; and for internal review passes in which AI agents simulated reviewers and suggested methodology changes (not peer review). S7–S11 were proposed by the assistant that ran the experiments and S12 by an internal AI review pass, each committed before any of its cells ran (Appendix E). The authors reviewed all code, text and numbers and take full responsibility for this paper.

REPRODUCIBILITY STATEMENT

Code and data will be released: dataset generators with fixed seeds, the training and evaluation worker, queue runners, analysis scripts, both preregistration files with an anonymized timestamp manifest of their commits, and every run's result record with per-item correctness (local paths removed), from which each paired test can be recomputed; an anonymized copy is in the supplementary material. Section 3 and Appendices B and D give prompts, trace formats and hyperparameters. Every number in the text is a macro regenerated from the result files (`paper/exp/build_all.py`).

ETHICS STATEMENT

The study uses synthetic arithmetic and logic data only and involves no human subjects or personal data.

REFERENCES

Emmanuel Abbe, Samy Bengio, Aryo Lotfi, Colin Sandon, and Omid Saremi. How far can transformers reason? The globality barrier and inductive scratchpad. In *Advances in Neural Information Processing Systems*, 2024.

François Charton. Learning the greatest common divisor: explaining transformer predictions. In *International Conference on Learning Representations*, 2024.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *International Conference on Machine Learning*, 2025.

Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, Lecture Notes in Computer Science, pp. 337–340. Springer, 2008. doi: 10.1007/978-3-540-78800-3_24.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*, 2023.

Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit CoT to implicit CoT: Learning to internalize CoT step by step. *arXiv preprint arXiv:2405.14838*, 2024.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems*, 2023.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Advances in Neural Information Processing Systems*, 2023.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. doi: 10.1038/s42256-020-00257-z.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

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

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*, 2022.

Juno Kim and Taiji Suzuki. Transformers provably solve parity efficiently with chain of thought. In *International Conference on Learning Representations*, 2025.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, et al. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Andrew K. Lampinen, Ishita Dasgupta, Stephanie C. Y. Chan, Kory Matthewson, Michael Henry Tessler, Antonia Creswell, James L. McClelland, Jane X. Wang, and Felix Hill. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022.

Nayoung Lee, Kartik Sreenivasan, Jason D. Lee, Kangwook Lee, and Dimitris Papailiopoulos. Teaching arithmetic to small transformers. In *International Conference on Learning Representations*, 2024.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. In *International Conference on Learning Representations*, 2024.

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

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *International Conference on Learning Representations*, 2023.

Tiedong Liu and Bryan Kian Hsiang Low. Goat: Fine-tuned LLaMA outperforms GPT-4 on arithmetic tasks. *arXiv preprint arXiv:2305.14201*, 2023.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Short Papers)*, 2023.

Eran Malach. Auto-regressive next-token predictors are universal learners. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34417–34431, 2024.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. doi: 10.1007/BF02295996.

William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought. In *International Conference on Learning Representations*, 2024.

Terufumi Morishita, Gaku Morio, Atsuki Yamaguchi, and Yasuhiro Sogawa. Enhancing reasoning capabilities of LLMs via principled synthetic logic corpus. In *Advances in Neural Information Processing Systems*, 2024.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. Investigating the limitations of transformers with simple arithmetic tasks. *arXiv preprint arXiv:2102.13019*, 2021.

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. *arXiv preprint arXiv:2112.00114*, 2021.

Jacob Pfau, William Merrill, and Samuel R. Bowman. Let's think dot by dot: Hidden computation in transformer language models. *arXiv preprint arXiv:2404.15758*, 2024.

Ben Prystawski, Michael Y. Li, and Noah D. Goodman. Why think step by step? Reasoning emerges from the locality of experience. In *Advances in Neural Information Processing Systems*, 2023.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. RL on incorrect synthetic data scales the efficiency of LLM math reasoning by eight-fold. In *Advances in Neural Information Processing Systems*, 2024.

Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. In *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3067–3075, 2017.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To CoT or not to CoT? Chain-of-thought helps mainly on math and symbolic reasoning. In *International Conference on Learning Representations*, 2025.

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

Karthik Valmeekam, Kaya Stechly, Vardhan Palod, Atharva Gundawar, and Subbarao Kambhampati. Beyond semantics: The unreasonable effectiveness of reasonless intermediate tokens. *Transactions on Machine Learning Research*, 2026.

Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokked transformers are implicit reasoners: A mechanistic journey to the edge of generalization. In *Advances in Neural Information Processing Systems*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Kaiyue Wen, Huaqing Zhang, Hongzhou Lin, and Jingzhao Zhang. From sparse dependence to sparse attention: Unveiling how chain-of-thought enhances transformer sample efficiency. In *International Conference on Learning Representations*, 2025.

Noam Wies, Yoav Levine, and Amnon Shashua. Sub-task decomposition enables learning in sequence to sequence tasks. In *International Conference on Learning Representations*, 2023.

Mengjiao Yang, Dale Schuurmans, Pieter Abbeel, and Ofir Nachum. Chain of thought imitation with procedure cloning. In *Advances in Neural Information Processing Systems*, 2022.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, 2022.

A FULL RESULTS

Post hoc audits cited in §4.1. The paired audit compares each fine-tuned run with the zero-shot base item by item on div7 and div13 at 3B and 7B. Of the items the base solves, A keeps 52.4% (3B div7), 48.4% (3B div13), 50.8% (7B div7) and 48.7% (7B div13); B at $n = 180$ keeps 74.6% (3B div7) and 88.4% (7B div7); S keeps 99.8% (3B, two seeds) and 92.9% (7B, seed 0). Exact McNemar tests on lost versus fixed items and the per-run tables are in the supplementary material (`results/analysis/erasure.md`). The rule audit scores each candidate surface rule (last digit, digit sum, small factors) against every divisibility A run's predictions: no run agrees with any rule beyond Cohen's $\kappa = 0.24$; constant-output runs are listed in Appendix B.

The foreign trace and the dose at 3B and 7B (not preregistered). At 3B and 7B (not preregistered), the foreign long-division trace depends on size: at 3B it stays below the base on div7 (75.2%; B−base -19.0 [-30.4, -8.3]) and div13 (56.0%); at 7B it matches the base on div7 (88.5%; +6.0 [-0.8, +12.5]) but not on div13 (63.5%); at 1.5B, whose base scores 62.5%, it exceeds the base. More traces close the 3B gap, a within-task replication of the dose result above (not preregistered at 3B): div7 B rises from 60.0% at $n = 90$ to 97.9% at 270 in seed 0 (58.8% and 86.7% in seed 1), above the zero-shot model in one seed, and div13 B reaches 99.4% at $n = 540$ while A stays at 50.0%.

Table 4: Scale: zero-shot base / answers A / trace B at $n = 180$, accuracy in %, mean over seeds (seeds per cell: Table 6). –: not run. †Lower bound: 110 of 240 generations hit the cap. Means that include a constant-output seed (1.5B div3 A, 1.5B div13 A, 3B div7 A, 3B div13 A, 7B prime A) are listed in Appendix B.

task	0.5B	1.5B	3B	7B
div7	– / 50.4 / 79.2	62.5 / 50.3 / 88.3	**94.2** / 52.3 / 75.2	**82.5** / 51.0 / 88.5
div13	– / – / –	50.4 / 50.4 / 53.9	**92.1** / 49.2 / 56.0	**80.0** / 46.7 / 63.5
div3	– / – / –	78.3 / 54.0 / 100.0	– / 99.2 / 100.0	–
div11	–	55.0 / 50.1 / 99.4	– / 53.1 / 99.8	– / 53.5 / 99.4
div7, 6 digits	–	52.5 / 49.4 / 93.1	– / 49.2 / 82.5	–
prime	– / 68.3 / 57.5	59.6 / 82.1 / 55.8	47.1† / 82.3 / 85.0	72.1 / 66.9 / 84.6
valid	– / – / –	30.0 / 99.2 / 88.3	85.8 / 100.0 / 95.6	89.6 / 100.0 / 77.9

Table 5: Entry-level coverage (post hoc): lookup accuracy (%) of trace-trained models by c, the number of times the exact (remainder, digit) entry appeared in the cell's training traces; non-trivial entries (remainder $\neq 0$), pooled over every arm-B cell with saved generations, and div7 versus div13 at 1.5B and $n = 180$ on the same bins.

c	0	1	2	3–4	5–8	9–16	17–32	33–64
all cells, non-trivial	21.5	33.7	51.5	60.7	77.6	94.3	96.4	100.0
div7, 1.5B, $n = 180$	–	–	8.3	78.0	92.7	100.0	100.0	–
div13, 1.5B, $n = 180$	12.5	11.6	25.3	19.2	22.5	38.7	38.1	–

Table 6 lists every logged run of the arms in Table 2 and of B-ext, which uses reasoning traces sampled from a locally served 9B model and kept only when their final answer matched the verifier; only 9 traces passed, so these runs are not comparable to B and are listed for completeness. Table 9 reports the earlier answer-only study with its shuffled-label control. Per-seed exact McNemar tests (B vs. A, 1.5B): prime $p = 5.7 \times 10^{-7}$, 1.7×10^{-7}, 8×10^{-11} (A > B); div7 4.1×10^{-27}, 2×10^{-18}, 6.8×10^{-13}; valid 0.50, 7.1×10^{-15}, 4.7×10^{-10}.

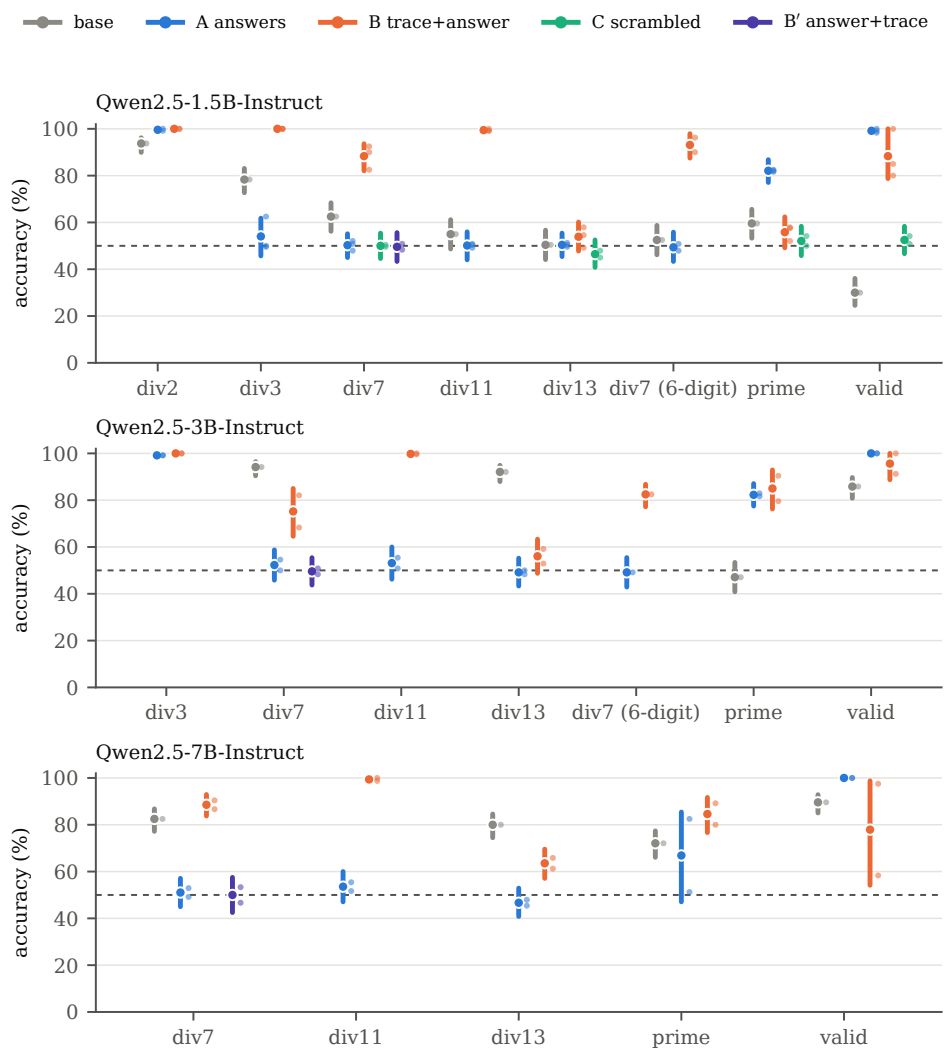

Figure 3: Accuracy by arm and task at $n = 180$; bars: 95% bootstrap interval over items and seeds where a cell has two or more seeds, Wilson interval for a single seed; small dots are single seeds; one row per model with at least three tasks (the 0.5B cells are in Table 4). Dashed line: majority class.

Table 6: All logged runs of arms base, A, B, C, D, B′, S, A-full and B-ext (240 test items per cell). Cell: the test set, prefixed by the training task when they differ; [cot] and [fewshot4] mark prompting controls on the base model. SmolLM2: SmolLM2-1.7B-Instruct (S11). Train tokens: whitespace tokens in the completions. Cap: test generations that hit the generation cap (scored wrong; – where not logged). Yes: share of test items answered Yes, from saved generations or recovered from per-item correctness; 0 or 100 marks a constant-output collapse.

model	cell	arm	n	seed	train tokens	cap	Yes (%)	acc. (%)
0.5B	prime	A	180	0	360	0	47	68.3
0.5B	prime	B	180	0	13,370	0	12	57.5
0.5B	div7	A	180	0	360	0	5	50.4
0.5B	div7	B	180	0	10,800	0	34	79.2
1.5B	prime	base	0	0	0	–	28	59.6
1.5B	prime	A	60	0	120	–	90	59.6

model	cell	arm	n	seed	train tokens	cap	Yes (%)	acc. (%)
1.5B	prime	A	180	0	360	–	67	82.1
1.5B	prime	A	180	1	360	–	68	82.5
1.5B	prime	A	180	2	360	0	68	81.7
1.5B	prime	A	540	0	1,080	–	75	75.0
1.5B	prime	B	60	0	4,355	–	12	57.1
1.5B	prime	B	180	0	13,370	–	11	57.9
1.5B	prime	B	180	1	13,260	–	12	57.5
1.5B	prime	B	180	2	13,080	13	16	52.1
1.5B	prime	B	540	0	40,095	–	28	68.8
1.5B	prime	C	180	0	12,561	–	99	50.0
1.5B	prime	C	180	1	13,120	–	12	54.2
1.5B	prime	A-full	1320	0	2,640	–	69	80.8
1.5B	prime	B-ext	9	0	896	–	20	62.1
1.5B	prime	B-ext	9	1	974	–	12	60.0
1.5B	prime	S	180	0	18,790	1	41	67.1
1.5B	prime→prime_hard	A	180	0	0	0	99	49.6
1.5B	prime[cot]	base	0	0	0	8	28	62.9
1.5B	prime[cot]	A	180	0	0	0	67	82.5
1.5B	prime[fewshot4]	base	0	0	0	104	9	39.2
1.5B	div7	base	0	0	0	–	72	62.5
1.5B	div7	A	60	0	120	–	88	50.0
1.5B	div7	A	180	0	360	–	24	52.1
1.5B	div7	A	180	1	360	–	82	50.8
1.5B	div7	A	180	2	360	–	94	47.9
1.5B	div7	A	540	0	1,080	–	0	50.0
1.5B	div7	B	60	0	3,600	–	73	50.0
1.5B	div7	B	60	1	3,600	0	32	51.7
1.5B	div7	B	60	2	3,600	0	48	53.3
1.5B	div7	B	90	0	5,400	0	52	62.1
1.5B	div7	B	90	1	5,400	0	21	56.7
1.5B	div7	B	90	2	5,400	0	34	56.7
1.5B	div7	B	180	0	10,800	–	47	92.5
1.5B	div7	B	180	1	10,800	–	44	90.0
1.5B	div7	B	180	2	10,800	–	41	82.5
1.5B	div7	B	270	0	16,200	0	50	99.6
1.5B	div7	B	270	1	16,200	0	47	92.9
1.5B	div7	B	270	2	16,200	0	45	92.1
1.5B	div7	C	180	0	10,784	–	72	49.6
1.5B	div7	C	180	1	10,824	–	30	50.4
1.5B	div7	B′	180	0	10,800	0	48	48.3
1.5B	div7	B′	180	1	10,800	0	11	50.8
1.5B	div7	D	180	0	10,800	0	100	50.0
1.5B	div7	D	180	1	10,784	0	0	50.4
1.5B	div7	S	180	0	7,473	1	57	77.9
1.5B	div7	S	180	1	7,215	0	70	65.8
1.5B	div7→div7_6d	B	180	0	0	0	32	77.1
1.5B	div7[cot]	base	0	0	0	0	70	67.1
1.5B	div7[cot]	A	180	0	360	0	12	55.0
1.5B	div7[cot]	A	180	1	360	0	76	53.3
1.5B	div7[fewshot4]	base	0	0	0	0	37	47.9
1.5B	valid	base	0	0	0	–	38	30.0
1.5B	valid	A	60	0	120	–	50	97.5
1.5B	valid	A	180	0	360	–	51	99.2
1.5B	valid	A	180	1	360	–	50	100.0
1.5B	valid	A	180	2	360	0	48	98.3
1.5B	valid	A	540	0	1,080	–	50	100.0
1.5B	valid	B	60	0	2,343	–	51	87.5
1.5B	valid	B	180	0	7,001	–	50	100.0
1.5B	valid	B	180	1	7,004	–	55	80.0
1.5B	valid	B	180	2	6,960	0	37	85.0
1.5B	valid	B	540	0	21,084	–	50	100.0
1.5B	valid	C	180	0	7,092	–	95	54.2

model	cell	arm	n	seed	train tokens	cap	Yes (%)	acc. (%)
1.5B	valid	C	180	1	7,090	–	1	50.8
1.5B	valid	A-full	1600	0	3,200	–	50	100.0
1.5B	valid	S	180	0	26,233	49	49	64.6
1.5B	valid[cot]	base	0	0	0	5	56	66.7
1.5B	valid[fewshot4]	base	0	0	0	0	30	68.8
1.5B	div2	base	0	0	0	0	47	93.8
1.5B	div2	A	180	0	360	0	50	100.0
1.5B	div2	A	180	1	360	0	50	99.2
1.5B	div2	B	180	0	10,800	0	50	100.0
1.5B	div2	B	180	1	10,800	0	50	100.0
1.5B	div3	base	0	0	0	0	29	78.3
1.5B	div3	A	180	0	360	0	0	50.0
1.5B	div3	A	180	1	360	0	77	62.5
1.5B	div3	A	180	2	360	0	15	49.6
1.5B	div3	B	180	0	10,800	0	50	100.0
1.5B	div3	B	180	1	10,800	0	50	100.0
1.5B	div3	B	180	2	10,800	0	50	100.0
1.5B	div11	base	0	0	0	32	14	55.0
1.5B	div11	A	180	0	360	0	9	50.8
1.5B	div11	A	180	1	360	0	7	49.2
1.5B	div11	A	180	2	360	0	9	50.4
1.5B	div11	B	180	0	10,800	0	50	99.2
1.5B	div11	B	180	1	10,800	0	49	99.2
1.5B	div11	B	180	2	10,800	0	50	100.0
1.5B	div13	base	0	0	0	2	23	50.4
1.5B	div13	A	180	0	360	0	0	49.6
1.5B	div13	A	180	1	360	0	58	50.4
1.5B	div13	A	180	2	360	0	8	51.2
1.5B	div13	A	360	0	720	0	0	50.0
1.5B	div13	A	360	1	720	0	100	50.0
1.5B	div13	B	180	0	10,800	0	76	54.6
1.5B	div13	B	180	1	10,800	0	18	49.2
1.5B	div13	B	180	2	10,800	0	49	57.9
1.5B	div13	B	360	0	21,600	0	27	75.4
1.5B	div13	B	360	1	21,600	0	27	72.5
1.5B	div13	B	360	2	21,600	0	28	72.5
1.5B	div13	B	540	0	32,400	0	46	94.6
1.5B	div13	B	540	1	32,400	0	45	95.0
1.5B	div13	B	540	2	32,400	0	46	95.8
1.5B	div13	C	180	0	10,824	0	49	45.0
1.5B	div13	C	180	1	10,788	5	86	47.9
1.5B	div13[cot]	base	0	0	0	0	15	54.6
1.5B	div13[cot]	A	180	0	0	0	0	50.0
1.5B	div7_6d	base	0	0	0	0	88	52.5
1.5B	div7_6d	A	180	0	360	0	7	47.9
1.5B	div7_6d	A	180	1	360	0	9	50.8
1.5B	div7_6d	B	180	0	14,040	0	48	96.2
1.5B	div7_6d	B	180	1	14,040	0	43	90.0
3B	prime	base	0	0	0	110	2	47.1
3B	prime	A	180	0	360	0	67	82.9
3B	prime	A	180	1	360	0	63	81.7
3B	prime	B	180	0	13,370	0	60	90.4
3B	prime	B	180	1	13,260	0	41	79.6
3B	prime	S	180	0	19,736	49	22	69.6
3B	prime→prime_hard	A	180	0	0	0	99	50.4
3B	prime→prime_hard	A	180	1	0	0	92	47.9
3B	prime→prime_hard	B	180	0	0	0	100	50.4
3B	prime→prime_hard	B	180	1	0	0	69	53.8
3B	div7	base	0	0	0	0	55	94.2

model	cell	arm	n	seed	train tokens	cap	Yes (%)	acc. (%)
3B	div7	A	180	0	360	0	5	54.6
3B	div7	A	180	1	360	0	100	50.0
3B	div7	B	90	0	5,400	0	45	60.0
3B	div7	B	90	1	5,400	0	67	58.8
3B	div7	B	180	0	10,800	0	27	68.3
3B	div7	B	180	1	10,800	0	41	82.1
3B	div7	B	270	0	16,200	0	48	97.9
3B	div7	B	270	1	16,200	0	47	86.7
3B	div7	B′	180	0	10,800	51	96	50.8
3B	div7	B′	180	1	10,800	0	45	48.3
3B	div7	D	180	0	10,800	0	100	50.0
3B	div7	D	180	1	10,784	0	100	50.0
3B	div7	S	180	0	11,130	0	52	97.5
3B	div7	S	180	1	11,198	0	54	96.2
3B	div7→div7_6d	A	180	0	0	0	0	50.4
3B	div7→div7_6d	B	180	0	0	0	22	54.2
3B	div7→div7_6d	B	180	1	0	0	33	65.0
3B	div7[cot]	A	180	0	360	0	3	53.3
3B	valid	base	0	0	0	10	52	85.8
3B	valid	A	180	0	360	0	50	100.0
3B	valid	A	180	1	360	0	50	100.0
3B	valid	B	180	0	7,001	0	59	91.2
3B	valid	B	180	1	7,004	0	50	100.0
3B	valid	S	180	0	30,284	0	57	92.1
3B	div3	A	180	0	360	0	50	99.2
3B	div3	A	180	1	360	0	50	99.2
3B	div3	B	180	0	10,800	0	50	100.0
3B	div3	B	180	1	10,800	0	50	100.0
3B	div11	A	180	0	360	0	6	55.4
3B	div11	A	180	1	360	0	1	50.8
3B	div11	B	180	0	10,800	0	50	99.6
3B	div11	B	180	1	10,800	0	50	100.0
3B	div13	base	0	0	0	0	53	92.1
3B	div13	A	180	0	360	0	0	50.0
3B	div13	A	180	1	360	0	28	48.3
3B	div13	A	360	0	720	0	0	50.0
3B	div13	A	540	0	1,080	0	0	50.0
3B	div13	B	180	0	10,800	0	70	52.9
3B	div13	B	180	1	10,800	0	42	59.2
3B	div13	B	360	0	21,600	0	33	83.3
3B	div13	B	360	1	21,600	0	45	92.9
3B	div13	B	540	0	32,400	0	50	99.6
3B	div13	B	540	1	32,400	0	49	99.2
3B	div13[cot]	A	180	0	360	0	0	50.0
3B	div7_6d	A	180	0	360	0	1	49.2
3B	div7_6d	B	180	0	14,040	0	35	82.5
7B	prime	base	0	0	0	4	30	72.1
7B	prime	A	180	0	360	0	1	51.2
7B	prime	A	180	1	360	0	67	82.5
7B	prime	B	180	0	13,370	1	60	89.2
7B	prime	B	180	1	13,260	0	36	80.0
7B	prime	S	180	0	26,737	25	32	72.1
7B	div7	base	0	0	0	1	45	82.5
7B	div7	A	180	0	360	0	35	49.2
7B	div7	A	180	1	360	0	75	52.9
7B	div7	B	180	0	10,800	0	43	86.7
7B	div7	B	180	1	10,800	0	46	90.4
7B	div7	B′	180	0	10,800	3	16	53.3
7B	div7	B′	180	1	10,800	0	54	46.7

model	cell	arm	n	seed	train tokens	cap	Yes (%)	acc. (%)
7B	div7	D	180	0	10,800	0	39	47.9
7B	div7	D	180	1	10,784	0	61	50.4
7B	div7	S	180	0	10,803	0	41	82.1
7B	div7	S	180	1	10,728	0	42	81.2
7B	div7[cot]	A	180	0	360	0	60	55.0
7B	valid	base	0	0	0	3	42	89.6
7B	valid	A	180	0	360	0	50	100.0
7B	valid	A	180	1	360	0	50	100.0
7B	valid	B	180	0	7,001	0	92	58.3
7B	valid	B	180	1	7,004	0	52	97.5
7B	div11	A	180	0	360	0	26	55.4
7B	div11	A	180	1	360	0	16	51.7
7B	div11	B	180	0	10,800	0	49	98.8
7B	div11	B	180	1	10,800	0	50	100.0
7B	div13	base	0	0	0	0	47	80.0
7B	div13	A	180	0	360	0	25	45.4
7B	div13	A	180	1	360	0	6	47.9
7B	div13	B	180	0	10,800	0	26	61.3
7B	div13	B	180	1	10,800	0	35	65.8
7B	div13[cot]	A	180	0	360	0	68	46.2
SmolLM2	prime	A	180	0	360	0	36	68.3
SmolLM2	prime	B	180	0	13,370	0	23	62.5
SmolLM2	div7	base	0	0	0	0	4	49.6
SmolLM2	div7	A	180	0	360	0	29	45.0
SmolLM2	div7	A	180	1	360	0	91	47.1
SmolLM2	div7	B	180	0	10,800	0	41	60.0
SmolLM2	div7	B	180	1	10,800	0	51	69.6
SmolLM2	div7	D	180	0	10,800	0	87	51.7
SmolLM2	valid	A	180	0	360	0	50	99.6
SmolLM2	valid	B	180	0	7,001	0	30	80.4

Table 7: Timestamped predictions (S1–S6 before any of their runs; S7–S12 appended during the study, each before its cells) and their status. 3B and 7B were not the preregistered model; their columns report the same tests at those sizes; [k]: number of seeds. –: not run.

	prediction	result (1.5B)	status	3B	7B
S1	div7 B $\geq$ 80, B$-$A $\geq$ 20 pp (seeds 1, 2)	B 90.0/82.5, A 50.8/47.9	passed		
S2	traces help $\geq$ 2 of div3/11/13 (2 seeds)	B$-$A div3 +50.0/+37.5, div11 +48.3/+50.0, div13 +5.0/-1.2	passed	failed (div3 +0.8)	div3 –
S3	div2 A $\geq$ 95	99.6	passed (settled)		
S4	div7 B $>$ A at 3B (2 seeds)	(3B) A 54.6/50.0, B 68.3/82.1	passed		
S5	6-digit div7 B within 10 pp of $q^6 = 75.8$	96.2/90.0	**failed**	failed	–
S6	prime A on hard set $<$ 60, follows rule	49.6, rule agreement 98.8	passed (settled)	passed	–
S7	q follows m; $\geq$ 4 of 5 cells	(a)–(c) hold, (e) at ceiling, (d) fails: 3/4 informative	passed (4/5)	post hoc only	
S8	arm D $\leq$ 60 (2 seeds)	50.0/50.4	passed	D 50.0 [2]	D 49.2 [2]
S9	div13, $n = 360$: B $\geq$ 80, A $\leq$ 60	B 75.4, A 50.0	**failed**	passed (83.3)	–
S10	arm S div7 $\leq$ 65 (2 seeds); valid $\geq$ 90	div7 77.9/65.8 (seeds 0/1); valid 64.6	**failed**	contradicted (div7 97.5/96.2)	div7 81.7 [2] (base 82.5)
S11	second family: div7 B$-$A $\geq$ 20 pp (2 seeds), D within 10 pp of A	SmolLM2-1.7B-Instruct: B$-$A +15.0/+22.5, D 51.7 vs A 45.0	**failed**		
S12	div7 A sweep (lr, epochs; 2 seeds): no cell $\geq$ 65, one non-constant, div3 A $\geq$ 90	max 53.3 (lr 5e-5, 30 epochs, seed 1); non-constant 4/4; div3 92.1	passed		

Table 8: Timestamped cells in detail. Qwen2.5-1.5B-Instruct, 4 digits and $n = 180$ unless stated; accuracy in %. S2–S4 and S10 rows are means over the seeds in Table 6; other rows seed 0 unless stated. For S7 the B column is the per-step accuracy q (written p in the preregistration text); for S8 and S10 it is the accuracy of arms D and S. –: not run.

	cell	probe	A	B	criterion	met
S1	div7, s1 / s2	46.5	50.8 / 47.9	90.0 / 82.5	B $\geq$ 80, B−A $\geq$ 20 pp, $p <$ 0.01 in each seed	passed
S2	div3	50.7	54.0	100.0	$\geq$ 2 of the 3 with B−A $\geq$ 15 pp in both seeds	passed
	div11	50.3	50.1	99.4		
	div13	53.3	50.4	53.9		
S3	div2	100.0	99.6	100.0	A $\geq$ 95	passed
S4	div7, 3B model		52.3	75.2	B > A	passed
S5	div7 (6 digits), s0 / s1		49.4	96.2 / 90.0	$\lvert B - q^6 \rvert \leq 10$ pp; $q^6 = 75.8$	failed
	secondary: 4-digit B			77.1	same	
S6	prime→hard set		49.6	–	A < 60; agrees more with the rule than with the label	passed
S7	(a) div13, $n = 540$			0.980	$q \geq 0.95$, acc. $\geq$ 85 (94.6)	holds
	(b) div7, $n = 270$			0.997	$q \geq 0.97$	holds
	(c) div7, $n = 90$			0.736	$q \leq 0.80$, acc. $\leq$ 75 (62.1)	holds
	(d) div11			0.994	$0.566 < q < 0.955$	fails
	(e) div3 / div2			1.000 / 0.996	$q \geq 0.97$	holds
S8	div7, arm D, s0 / s1			50.0 / 50.4	D $\leq$ 60 in both seeds	passed
S9	div13, $n = 360$		50.0	75.4	B $\geq$ 80, A $\leq$ 60	failed
S10	div7, arm S			71.9	S $\leq$ 65 in seeds 0 and 1; valid S $\geq$ 90	failed

Table 9: Earlier answer-only study (same model and LoRA configuration; 240 training items, answer-only prompts, 3–4-digit numbers for arithmetic; mean of two seeds). Correct labels vs. labels permuted across training items.

task	base	correct labels	shuffled labels
divisible by 3	55.0	67.3	49.6
divisible by 7	53.3	51.0	52.5
divisible by 13	54.2	50.0	50.0
perfect square	51.2	83.3	51.5
prime	50.0	82.9	50.0
argument validity	66.7	99.4	50.0
joint consistency (SAT)	50.0	99.8	48.5

B SETUP DETAILS

Base-model cells whose generations often hit the cap are lower bounds: 3B prime base 47.1% (110 of 240 capped), 1.5B prime with 4 demonstrations 39.2% (104 capped) and 1.5B div11 base 55.0% (32 capped). The 1.5B valid base cell (30.0%) has no saved generations and is unaudited. Arm A collapses to a constant answer in single seeds (Yes rate near 0 or 100% in Table 6): 1.5B div3 and div13 seed 0, 3B div7 seed 1 and div13 seed 0; at larger n, 1.5B div7 at $n = 540$, 1.5B div13 at $n = 360$ (seed 0 "No", seed 1 "Yes") and 3B div13 at $n = 360$ and 540 ("No"); 3B div7→div7_6d ("No") and 3B div7_6d (Yes rate 1%); and 7B prime seed 0, which answers "No" on 237 of 240 items. A single untuned configuration is used throughout, so these collapses are part of what answer-only training did here, not a tuned optimum.

C THE LOGIC TASK

Arguments instantiate 6 valid forms (modus ponens, modus tollens, hypothetical syllogism, disjunctive syllogism, constructive dilemma, contraposition) and 6 invalid ones (affirming the consequent, denying the antecedent, affirming a disjunct, a broken chain, the converse, and "or" to "and") with distinct random atom letters, rendered in English ("if A then B", "not A", "A or B"). The schema

Table 10: Training and evaluation setup, shared by every arm and model.

models	Qwen2.5-0.5B/1.5B/3B/7B-Instruct, bf16; 0.5B and 1.5B on a laptop GPU, 3B and 7B on one RTX 5090
adapter	LoRA $r = 8$, $\alpha = 16$, dropout 0.05, on q/k/v/o projections only
optimization	AdamW, learning rate 2×10^{-4}, batch size 1, 3 epochs, loss on completion tokens only
prompt	the model's chat template; one user turn ending "End your reply with 'Answer: Yes' or 'Answer: No'."
seed	selects the balanced training sample and seeds PyTorch (adapter initialization, data order)
decoding	greedy; generation cap 176 tokens (div2/3/7), 192 (div11/13), 240 (6-digit div7), 512 (prime), 320 (valid), 512 for the CoT prompt
extractor	last "Answer: Yes/No" in the completion, else the first standalone Yes/No; otherwise unparsed; unparsed and truncated answers count as wrong
pools	valid: 1,600 training arguments; prime: 1,320 numbers (arm A-full); test items never in a pool
arm D	for each training item, the correct trace of another pool number with the same label
arm S	$K = 4$ samples per item at temperature 0.7 from the base model's CoT prompt; one accepted completion per item

proposes an argument and Z3 decides it: the argument is labeled valid iff the premises together with the negated conclusion are unsatisfiable. For a valid argument the trace names the unsatisfiable core; for an invalid one it gives the satisfying assignment as a countermodel. Training and test share the 12 forms and never an instance, so a model can reach high accuracy by recognizing the form, which the atom-renamed n-gram probe does (100.0%). The task therefore checks that answers work when the surface determines the label; it is not a test of logical reasoning. Its results: base 30.0%, A 99.2% (99.2 and 100.0), B 88.3% (100.0 and 80.0), C 52.5% (54.2 and 50.8); at $n = 60$, A 97.5% and B 87.5%; at $n = 540$, both 100.0%.

D TRACE FORMATS

All arms share the prompt, which ends "End your reply with 'Answer: Yes' or 'Answer: No'." Examples of training completions, verbatim:

div7, B (prompt: "Is 1617 divisible by 7?"):

```
Process the digits of 1617 keeping a running remainder mod
7.  r = (10*0 + 1) mod 7 = 1.  r = (10*1 + 6) mod 7 = 2.  r
= (10*2 + 1) mod 7 = 0.  r = (10*0 + 7) mod 7 = 0.  Final
remainder 0, so 1617 is divisible by 7.
Answer:  Yes
```

div7, C (a scrambled trace from another example; the answer line keeps this example's label):

```
running mod r = so = 4067 Process by 7.  7 a = 0.  0) 0.  7
mod (10*5 = r 7 the = = keeping + r is mod mod 4) (10*0 + of
4067 r + 7) 5.  = + digits remainder 7.  (10*0 = 7 0,
divisible Final 6) 4.  (10*4 mod remainder
Answer:  Yes
```

prime, B, a composite ("Is 4143 a prime number?"):

```
Check prime divisors up to isqrt(4143) = 64.  4143 mod 2 =
1.  4143 mod 3 = 0.  3 divides 4143, so 4143 is composite.
Answer:  No
```

prime, B, a prime ("Is 1493 a prime number?"):

```
Check prime divisors up to isqrt(1493) = 38.  1493 mod 2 =
1.  1493 mod 3 = 2.  1493 mod 5 = 3.  1493 mod 7 = 2.  1493
```

```
mod 11 = 8.  1493 mod 13 = 11.  1493 mod 17 = 14.  1493 mod
19 = 11.  1493 mod 23 = 21.  1493 mod 29 = 14.  1493 mod 31
= 5.  1493 mod 37 = 13.  No prime divisor up to 38, so 1493
is prime.
Answer:  Yes
```

valid, B, invalid ("Premises: (1) if H then D; (2) D. Conclusion: H."):

```
Assume every premise is true and the conclusion false.  That
is satisfiable:  set D=true, H=false.  Under this assignment
all premises hold while the conclusion is false, a
countermodel, so the argument is invalid.
Answer:  No
```

valid, B, valid ("Premises: (1) if S then R; (2) S. Conclusion: R."):

```
Assume every premise is true and the conclusion false, i.e.
~(R) holds.  Together these force a contradiction:  {S -> R,
S, ~(R)} cannot all hold at once.  Since assuming the
conclusion false is impossible, the conclusion follows.
Answer:  Yes
```

A (every task): `Answer:  Yes` or `Answer:  No`.

E PREREGISTRATION TEXT AND DECISION LOG

The decision log below is reproduced verbatim; its entries use the word "preregistered" for S7–S12, which the paper calls *timestamped* because they were written during the study after earlier results were in (§3).

Provenance of the prior div7 result. A verdict written on 2026-08-04 from the 60-example trace run alone (seed 0, 50.0%; the later three-seed mean is 51.7%) stated that traces do not unlock divisibility by 7, while the preregistered 180-example cell was still running; that cell finished at 92.5% with no code change. The dose curve shows why both were right about their own n. The replication of that cell was the go/no-go condition S1, and it passed (90.0% and 82.5% in seeds 1 and 2).

First preregistration (prior grid). Written before any run of the prior grid, according to the history of the repository in which it was written, and never edited afterwards; results were appended below it. The file enters the present repository with the first commit of the manifest (P1), so its earlier date rests on the original repository. It fixed: arms A, B, C, an answers-only arm at larger scale, arms using model-generated traces, and a conditional wrong-trace arm; Qwen2.5-1.5B-Instruct with the LoRA configuration of §3; tasks prime, valid and div7, all reported; 240 problem-disjoint test items per task (120 per class), identical across arms, graded by one parser on the final answer. Endpoints: H1, B $>$ A at $n = 180$ per task, paired McNemar, Holm across the three tasks; H2, B $>$ C at $n = 180$; H3, accuracy at matched training tokens; H4, on div7 B reaches about 60% with A at chance. The full text is in the supplementary material. Outcomes: H1 held on div7 only, H2 on div7 and valid, H3 failed where answers work (not tested on div7), and H4 held at $n = 180$ only.

Second preregistration and its decision log. Reproduced from `PREDICTIONS.md` up to the S10 entry; the six later entries (the S10 primary outcome, a wording note, S11, a second-model-family test none of whose cells had run at submission, the post hoc correction of the S10 trace-audit numbers, and the two entries recording the remaining S10 cells) are summarized in §4.3 and reproduced in the supplementary material. Markdown markup converted, commit hashes replaced by the anonymized identifiers P1–P6 of the timestamp manifest and the location in each time stamp removed, text otherwise unchanged. The file's p (per-step accuracy) is q in the paper. The clock times in the header and the first entries were first written as estimates (the header as "~20:35") and later replaced by commit timestamps; the decision log notes this. After the runs, a dated note was appended recording that the seed-0 "re-run" was a deterministic re-execution.

> **Preregistered predictions and decision rules (frozen before any run listed here).** Committed 2026-09-25 20:29:47 (commit P1), before any of the runs below were

started. The commit hash of this file is the preregistration. Earlier results (results/THINKING_VS_DATA.md, frozen 2026-08-04, and its later staged results block) are prior data, reported as such.

Provenance of the prior div7 result (stated in the paper). The 2026-08-04 verdict "H4 (a procedure trace unlocks div7): NO" was written from the n = 60 trace run alone (B@60 = 50.0%). The preregistered main cell, B@180 seed 0, finished afterwards at 92.5% with the same worker (no code change). The paper reports the premature verdict and the later result.

Claim under test. Exact labels teach shortcuts; exact traces teach short programs. With labels from an exact verifier: answer-only supervision (arm A) recovers roughly what a simple probe on the answer surface recovers; trace supervision (arm B: procedure trace, then answer) succeeds when the per-step trace accuracy p, compounded over k steps, stays high (predicted accuracy about p^k), and fails when the procedure is long.

Decision rules. S1 (go/no-go, before any other new run): div7 B@180 seeds 1 and 2 each reach >= 80% accuracy, with B - A >= 20 percentage points and McNemar p < 0.01 against A of the same seed. If S1 fails, the paper is not submitted in this form. **S2:** at least 2 of {div3, div11, div13} (4-digit numbers) give B - A >= 15 pp with the same sign in both seeds. **S3:** div2 A >= 95% (the answer is surface-learnable from the last digit). **S4:** the div7 sign (B > A) holds for Qwen2.5-3B-Instruct. **S5 (falsifiable length test):** on 6-digit div7, B's accuracy is within 10 pp of p^6, where p is B's per-step accuracy measured on 4-digit div7 generations. **S6 (shortcut):** A on prime agrees more with the rule "odd and no factor <= 7" than with the true label on hard negatives (odd composites with no factor <= 7), where A's accuracy drops below 60%.

Definitions: a "win" is a difference of >= 10 pp with both seeds agreeing in sign; "neither" is both arms at or below majority class + 8 pp. McNemar tests per seed, Holm correction across tasks within each hypothesis family, alpha = 0.05. Every cell is reported, including those that go against the claim.

Arms. base (zero-shot), A (answers), B (trace then answer), C (length-matched scrambled trace), B' (answer then trace, div7 only), plus eval-only controls: zero-shot chain-of-thought prompt and 4-shot trace demonstrations on the base model. Model: Qwen2.5-1.5B-Instruct (laptop), Qwen2.5-3B-Instruct and others where the RTX 5090 allows. LoRA r = 8, alpha 16, 3 epochs, lr 2e-4, bf16, greedy decoding, n = 180 training examples, 240 problem-disjoint test items.

Decision log.

2026-09-25 21:09:47 (commit P2) — **S1 PASS.** div7 B@180 seed 1 = 90.0% vs A 50.8% (+39.2 pp, McNemar p = 2e-18); seed 2 = 82.5% vs A 47.9% (+34.6 pp, p = 6.8e-13). Runs in results/thinking_vs_data/runs_L1.jsonl, produced by the unchanged worker experiments/exp_thinking_ft_worker.py. The paper proceeds; the remaining tests run on experiments/exp_worker_v2.py.

2026-09-25 21:26:10 (commit P3; the div7_6d B seed-0 generation file was written at 21:27:30) — **S5 clarification, recorded before either S5 cell finished** (div7_6d B seed 0 was mid-training; the 4-digit-adapter transfer eval had not started). Primary S5 cell = arm B trained and tested on the 6-digit task (div7_6d), seeds 0 and 1, compared with p^6 where p = per-step accuracy of 4-digit div7 B generations (seed 0: p = 0.955, p^6 = 0.758). Secondary = the 4-digit B adapter evaluated on 6-digit inputs. Also reported, labelled post hoc: the fraction of fully correct 6-digit traces vs p^6, and answer accuracy vs p^6 + (1 - p^6) * g, where g is the answer-correct rate when the trace is wrong (4-digit: g = 0.561), since a wrong trace still yields the right yes/no answer about half the time.

2026-09-25 21:29:52 (commit P4) — **S5 (primary) FAILED as preregistered**: 6-digit div7 B seed 0 = 96.2% vs p^6 = 75.8% (+20.4 pp > 10 pp tolerance). Its own per-step accuracy is 98.8% (vs 95.5% for the 4-digit model), so p is not a fixed property that transfers between training sets.

2026-09-25 21:29:52 (commit P4; div11 B seed 0 started after 21:30:58) — **S7 (new hypothesis, preregistered before any of its cells ran):** per-step trace accuracy p is set by the number of supervised transitions per entry of the (remainder, digit) table, m = k * n / (10 * d) (k steps per trace, n training examples, 10d table entries). Observed so far: div7 n=180 m=10.3 -> p=0.955; div7_6d n=180 m=15.4 -> p=0.988; div13 n=180 m=5.5 -> p=0.566. Predictions: (a) div13 B n=540 seed 0 (m=16.6): p >= 0.95 and answer accuracy >= 85%; (b) div7 B n=270 seed 0 (4-digit, m=15.4): p >= 0.97; (c) div7 B n=90 seed 0 (m=5.1): p <= 0.80 and answer accuracy <= 75%; (d) div11 B n=180 (m=6.5, not yet run): 0.566 < p < 0.955; (e) div3 B n=180 (m=24) and div2 B n=180 (m=36): p >= 0.97. S7 passes if at least 4 of (a)-(e) hold; every cell is reported either way.

Note (21:35): the clock times in the entries above were first written as estimates; they have been replaced by the commit timestamps, which are the authoritative record.

2026-09-25 21:38:05 (commit P5) **S8 and S9, preregistered before their cells ran** (div11 B seed 0 = 99.2% had finished; none of the cells below had started): S8 (grounding control): div7 arm D (a fluent, correct long-division trace of a DIFFERENT number with the same label) at n = 180, seeds 0 and 1: accuracy <= 60% in both seeds, i.e. within 10 pp of A, far below B. S9 (dose, step difficulty fixed): div13 B at n = 360, seed 0 (m = 11.1 transitions per table entry, close to div7's 10.3 at n = 180): accuracy >= 80%; div13 A at n = 360 seed 0 stays <= 60%.

2026-09-25 21:46:44 (commit P6) **S10, preregistered before any arm-S cell ran.** Arm S = self-generated traces filtered by the exact verifier on the final answer only (STaR / rejection-sampling fine-tuning, K = 4 samples per item at temperature 0.7 from the base model's chain-of-thought prompt; one kept completion per item; same 180 items and LoRA as A/B). Prediction: outcome-only verification does not supply correct steps, so on div7 arm S does not unlock the task: div7 S accuracy <= 65% in seeds 0 and 1, and fewer than half of the kept div7 traces have all remainders correct. On valid, where answers are surface-learnable, S >= 90%. Prime S is reported without a prediction.

F AN AUTOMATON VIEW (INTUITION)

Remark 1 (intuition, not a theorem)**.** Let $x_1 \cdots x_k$ be the digits of n and $\delta_d(r, a) = (10r + a) \bmod d$. Then $d \mid n$ iff $\delta_d^*(0, x_1 \cdots x_k) = 0$: divisibility by d is recognized by a d-state automaton reading digits, whose states the arm-B trace writes. Under teacher forcing each supervised state is a function of two tokens already in the context, so n traces are kn samples of one table with $10d$ entries, $m = kn/(10d)$ per entry. Answer-only supervision gives n samples of the k-fold composition, and for d coprime to 10 the label is nearly independent of every proper subset of the digits: for $d = 7$ and $k = 4$, fixing any three digits leaves $\Pr(7 \mid n)$ between 0.10 and 0.22 (marginal 0.14).

The remark resembles the globality barrier of Abbe et al. (2024) and the teacher-forced CoT of Kim & Suzuki (2025), but those concern growing input length and do not apply literally to fixed 4-digit inputs. It counts how often each transition is supervised, not how hard it is: for $d = 3$ and $d = 11$ the transition is an addition or subtraction mod d, for $d = 7$ and $d = 13$ it involves a multiplication, and the results follow difficulty as well as m. Its independence argument for answer-only supervision applies to $d = 3$ as much as to $d = 7$, yet at 3B answers alone teach div3 (99.2%). The count is the wrong measure there: divisibility by 3 depends on the digits only through their sum, a symmetric, low-degree function, while divisibility by 7 or 13 does not, and the locality and degree accounts of Abbe et al. (2024) and Prystawski et al. (2023) separate the two where the independence count does not. The remark is intuition, not a theorem.

G RLVR: A HYPOTHESIS, NOT A RESULT

With a binary verifier the RLVR reward depends on the final answer only, so we hypothesize, without testing, that it behaves like answer supervision with on-policy sampling: a surface rule already earns most of the reward on a surface-learnable task (the last-digit rule is right on 82.5% of our prime test items), and on div7 the policy finds the procedure only if its own samples sometimes execute it (Yue et al., 2025). We ran no RL; arm S is the nearest experiment.

H THE COMPOUNDING APPROXIMATION (POST HOC)

This derivation is post hoc: it was not part of the preregistered S5 rule, and it does not change S5's verdict. Suppose each emitted state is correct with probability q, independently, and that after the first error the emitted final state is uniform on $\mathbb{Z}_d$. With probability q^k the trace is error-free and the answer is right. Otherwise the answer is right with probability $1/d$ on a multiple of d and $(d-1)/d$ otherwise, so on a balanced test set the accuracy is $q^k + \frac{1}{2}(1 - q^k)\,[\frac{1}{d} + \frac{d-1}{d}] = (1 + q^k)/2$. Both assumptions can fail: errors are correlated across steps (a model that miscomputes one table entry does so whenever the entry recurs), and a wrong state need not be uniform (on div13, wrong traces end at $r = 0$ in 179 of 235 cases). The positive-class accuracy, $q^k + (1 - q^k)/d$, is the cleaner

check, because a model that answers "no" by default scores well on non-multiples: on 6-digit div7 it is 88.8%.