s2n-bignum-bench
A practical benchmark for evaluating low-level code reasoning of LLMs

NeurIPS 2026 [Evaluations and Datasets]

Balaji Rao
Soonho Kong, Ph.D. (AWS)
Juneyoung Lee, Ph.D. (AWS)
Carlo Lipizzi, Ph.D.

Motivation

- Neural theorem proving has made visible progress on mathematical benchmarks.

- But verified systems proofs often require reasoning (including but not limited to):
  - Instruction-set semantics
  - Large project-specific proof context

- In these proofs, the theorem prover reasons about actual machine state: registers, flags, memory, program counters, and fixed-width arithmetic.

- Question: can current LLMs synthesize proofs in this setting?

What is s2n-bignum?

- AWS open-source library of hand-written ARM and x86 assembly routines for big-integer cryptographic arithmetic.
- AWS s2n-bignum pairs assembly with HOL Light proofs.

```
initial machine state
     +
decoded ARM or x86 instructions
     +
formal ISA semantics
     ->
final machine state satisfying the mathematical specification
```

Formal ISA transition: `step : MachineState -> MachineState`

Benchmark Tasks

- s2n-bignum-bench is a benchmark derived from s2n-bignum.
- Each task is independent; original proof bodies are withheld.
- setup.ml: HOL Light libraries, ISA models, object code and reusable lemmas.
- query.txt: the theorem to prove. answer.txt: one candidate tactic expression.

```
prove(goal, <candidate tactic>)
```

- The candidate is accepted only when the goal closes and the global axiom list is unchanged.

Proof Corpus (v1.0)

Category	Tasks	Median proof lines
Generic	564	4
Program state	555	4
Bit vector	318	5
ARM functional correctness	439	101
x86 functional correctness	425	93

- Frozen v1.0: 2,301 tasks: 864 functional-correctness tasks and 1,437 supporting obligations.

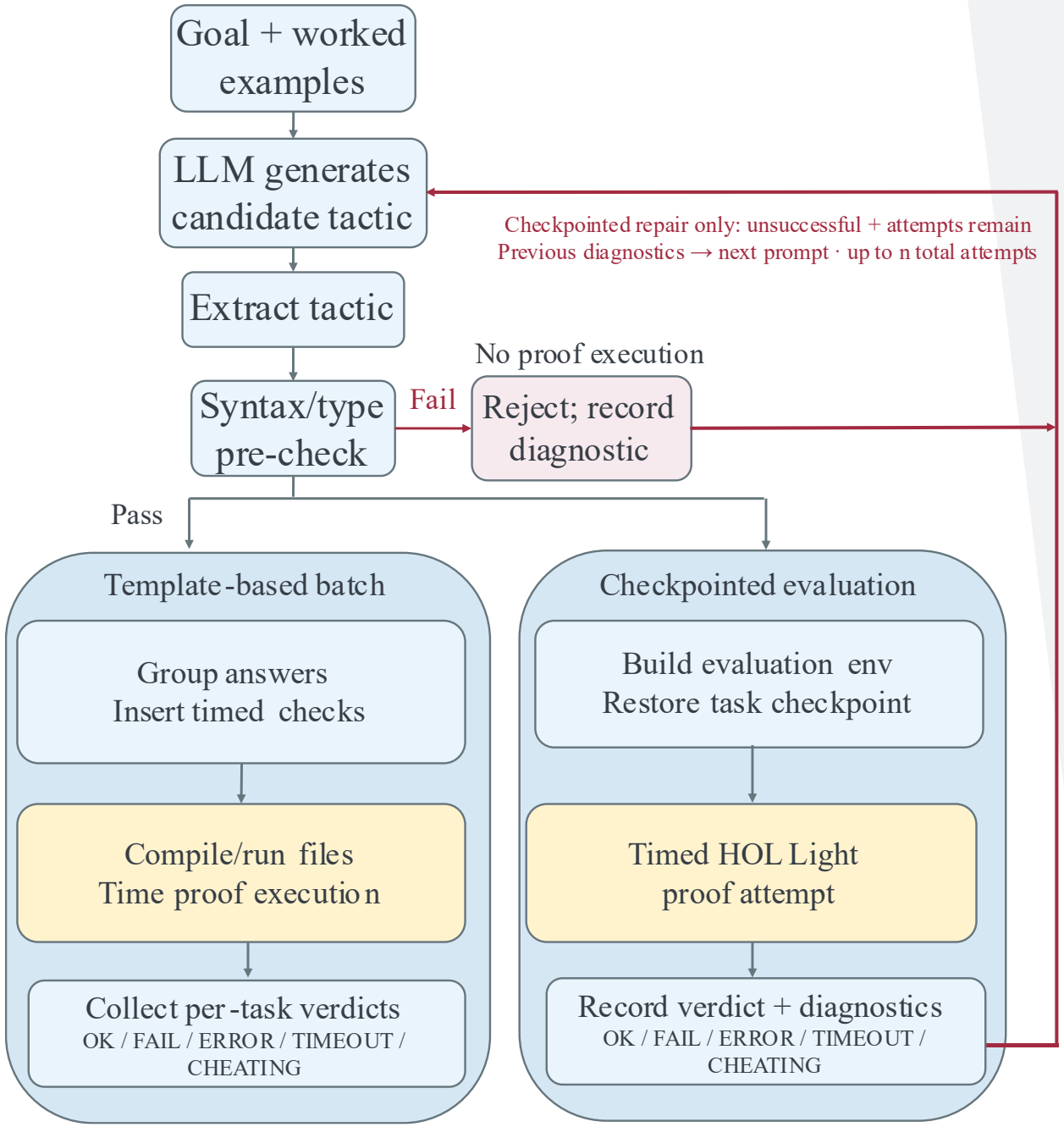
Verification Project Context

- Median query: 358 characters. Median setup.ml: 445,957 tokens, 1,362,511 characters and 36,075 lines. (measured on v1.0)
- The baseline sees the query and examples, not setup.ml.
- Selecting relevant definitions, ISA facts, object code and lemmas is a research opportunity.

Shared	Per ISA
HOL Light kernel	ISA decode/execute model
proof infrastructure	assembly implementation
bignum algebra lemmas	symbolic execution steps
high-level mathematical specs	per-routine correctness proofs

Evaluation Pipeline

- Prompt the LLM with the goal and worked HOL Light examples.
- Generate one tactic expression; run the syntax/type pre-check.
- Bound tactic construction and proof execution, including an external process timeout in checkpointed assessment.
- Accept a closed goal only when the axiom list is unchanged.

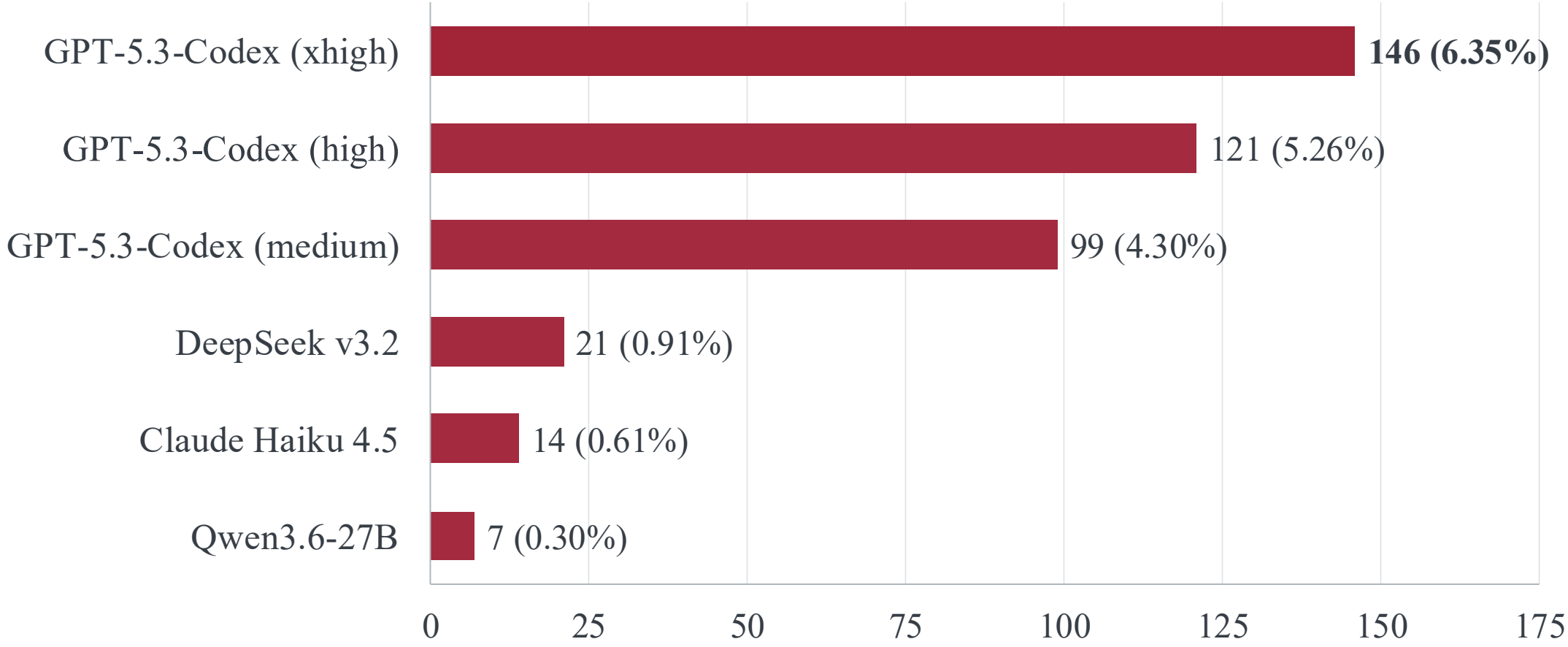


Baseline Evaluation Protocol

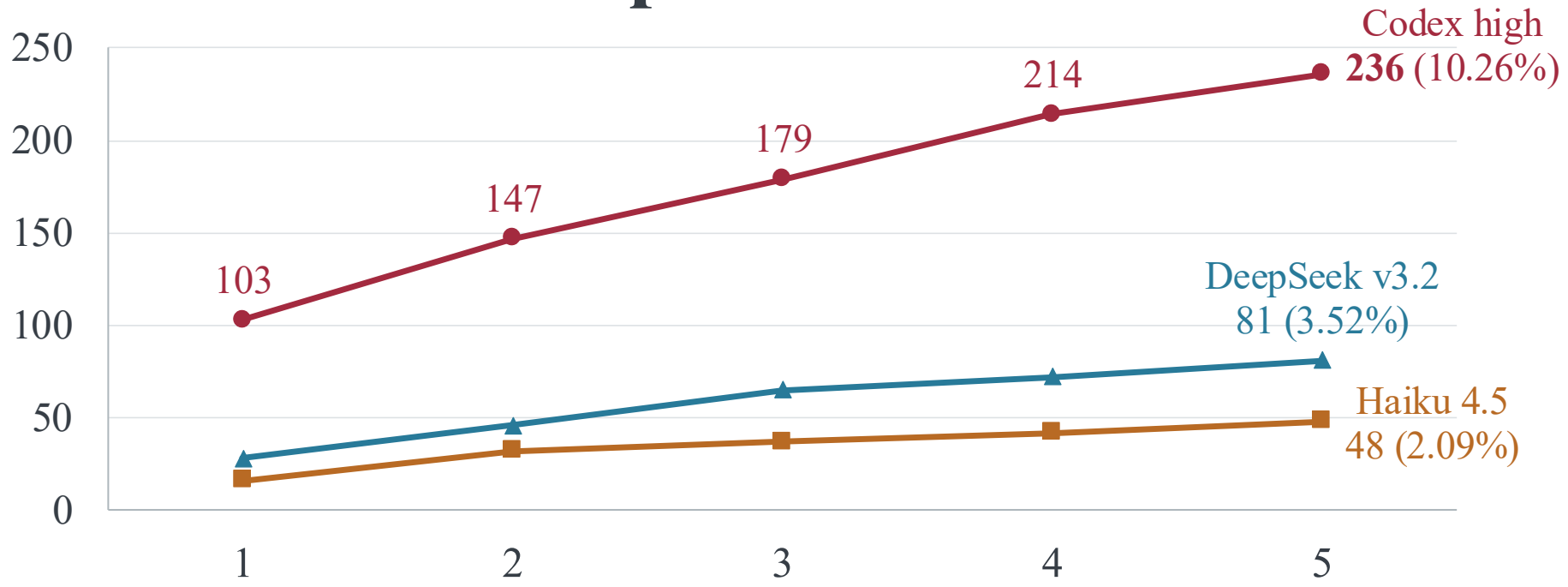
- Query-only prompting: target goal and three unrelated worked HOL Light examples; archived medium-effort answers used one example.
- Generate a complete tactic expression without setup.ml retrieval or agent scaffolding.
- Evaluate one attempt, or a separate experiment with up to five sequential repair attempts and execution feedback.
- Each repair attempt starts from the original checkpoint. The model receives the previous error feedback.

Single-Attempt Baseline Results

- All successes are supporting lemmas. None of the original baselines solves any of the 864 functional-correctness proof obligations (v1.0 - 2301 tasks).



Iterative Proof Attempts



- Cumulative success within five sequential attempts: Codex high 10.26%, DeepSeek 3.52%, Haiku 2.09%.
- The Codex run starts at 103 successes; its separate single-attempt run solves 121. [This is not independent-sample pass@5]
- Gains come from supporting lemmas: 0 / 864 functional-correctness tasks solved.

HOL Light Fluency and Proof Success

- Syntax/type fluency barrier
- Proof completion barrier
- No functional-correctness task is solved within five sequential attempts.

Model	Functional correctness (864)		
	Reached HOL, single-attempt	Reached HOL, feedback	Solved
GPT-5.3-Codex (high)	50	590	0
Claude Haiku 4.5	51	660	0
DeepSeek v3.2	15	100	0

Category	Tasks	Reached HOL	OK	FAIL	TIMEOUT	ERROR	OK / Reached
`generic`	564	386	77	271	33	5	19.9%
`program_state`	555	240	32	199	9	0	13.3%
`bit_vector`	318	172	37	131	4	0	21.5%
`functional_correctness`	864	68	0	68	0	0	0.0%
Total	2,301	866	146	669	46	5	16.9%

GPT-5.3-Codex (xhigh)

Updated Benchmark Results (v1.1)

Model (high effort)	Verified / total	Success	FC solved
Grok 4.6	219 / 2,622	8.35%	5 / 898
Kimi K3	185 / 2,622	7.06%	0 / 898
GLM 5	62 / 2,622	2.36%	0 / 898

- The refreshed August 2026 snapshot contains 2,622 tasks, including 898 functional-correctness obligations.
- Grok solves three ARM and two x86 correctness tasks, all among the smallest.
- Grok retried 87 truncated outputs with a higher output limit; none of the five correctness successes came from those retries. Corpus and budgets differ from v1.0.

Integrity and Contamination Probe (v1.0)

Verified proofs	Original	Obfuscated
Codex high	121	107
Haiku 4.5	14	11
DeepSeek v3.2	21	25

- In the reported run, 1,996 queries changed; 305 remained unchanged.
- It renames invented type variables and adds explicit type annotations while preserving logical structure.
- This probes surface-form sensitivity; structural memorization remains possible.

Conclusion

- s2n-bignum-bench evaluates proof synthesis for industrial ARM/x86 assembly inside a large HOL Light verification context.
- The original v1.0 baselines solve no functional-correctness tasks (0 / 864); the v1.1 study records the first five successes (5 / 898), all among the smallest.
- The released extraction, grading and checkpointed workflow provides infrastructure for repeated proof attempts.
- Next steps: relevant-context selection, interactive proof search, step-level interaction through hol_server and an MCP binding.



STEVENS
INSTITUTE OF TECHNOLOGY
1870

THANK **YOU**

Balaji Rao,
brao@stevens.edu

Stevens Institute of Technology
1 Castle Point Terrace, Hoboken, NJ 07030