The Karp Dataset of NP-Hardness Reductions

Mason DiCicco

Eamon Worden, Daniel Reichman, Niel Heffernan, Conner Olsen, Nikhil Gangaram
Worcester Polytechnic Institute

JMM 2026

"Can LLMs understand the structure of problems they cannot efficiently solve?"

Background: P and NP

P — **efficiently solvable:**

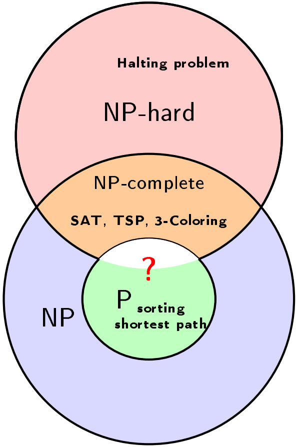
- Sorting, shortest paths, spanning trees

NP — **efficiently verifiable:**

- SAT (given an assignment, plug-in and check)
- Independent Set (given vertices, scan for any edges between them)

$$P \stackrel{?}{=} NP$$

(Can we *find* as fast as we can *verify*?)



Example: NP Problems

3SAT

- **Input:** 3-CNF formula φ
- **Question:** $\exists$ satisfying assignment?

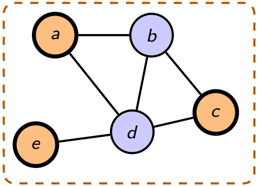
$$\underbrace{(x \vee \neg y \vee z)}_{\text{clause 1}} \wedge \underbrace{(\neg x \vee y \vee w)}_{\text{clause 2}}$$

$x{=}\mathrm{T}, y{=}\mathrm{T}, z{=}\mathrm{F}, w{=}\mathrm{T}$ ✓

Independent Set

- **Input:** Graph G, integer k
- **Question:** $\exists$ k independent vertices?

Graph G, **Target** $k = 3$



In independent set

No edges between $\{a, c, e\}$

Why This Matters

NP problems as stress tests:

- **Tunable** — difficulty scales arbitrarily
- **Verifiable** — answers checkable efficiently
- **Principled** — grounded in complexity theory

	Verifiable	**Not Verifiable**
Easy	P (sorting)	trivial (constant)
Hard	NP (SAT, IS)	undecidable (halting)

NP Benchmarks for LLMs

NPHardEval (Fan et al.)

900 problems: P $\rightarrow$ NP-hard

Monthly refresh $\Rightarrow$ no overfitting

GraphArena (Tang et al.)

10 P/NP problems on 10k *real* graphs

Social networks, molecules, flights

NPPC (Yang et al.)

25 NP-complete, infinite scaling

Difficulty $\uparrow \Rightarrow$ accuracy $<$10%

EHOP (Duchnowski et al.)

Same problem, two phrasings

"Party planning" harder than TSP

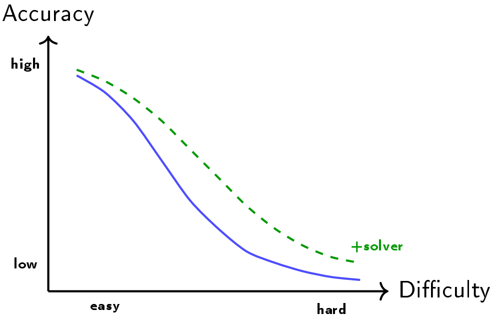
Empirical Patterns on NP Problems

LLMs vs NP:

- Good on small instances
- Performance degrades with size
- False confidence

Hybrid approaches:

- Code execution improves large instances
- *Translation* helps
  - Informal $\leftrightarrow$ formal
  - NL $\rightarrow$ SAT



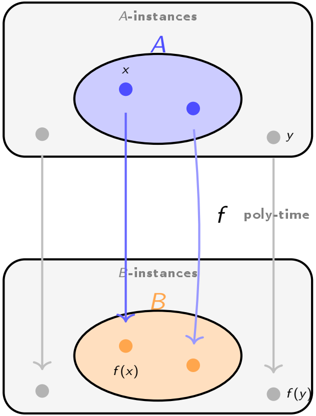
Background: NP-Hardness and Reductions

NP-hard: at least as hard as anything in NP

- E.g., SAT, INDEPENDENT SET, TSP

Reduction $A \leq_p B$:

- Efficient (polynomial-time) transform: $x \mapsto f(x)$
- $x \in A \Leftrightarrow f(x) \in B$
- "If I can solve B, then I can solve A"



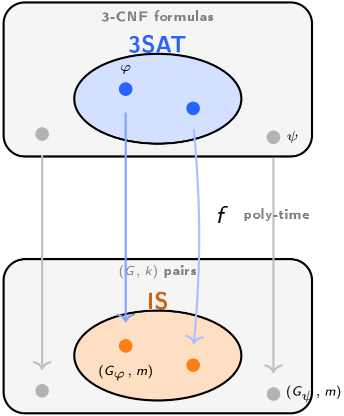
3SAT → Independent Set: The Goal

We want to show:

- Independent Set is *at least as hard* as 3SAT

Strategy:

- Transform any 3SAT formula into a graph
- Formula satisfiable ⇔ graph has IS of size k



3SAT → Independent Set: Reduction

Construction:

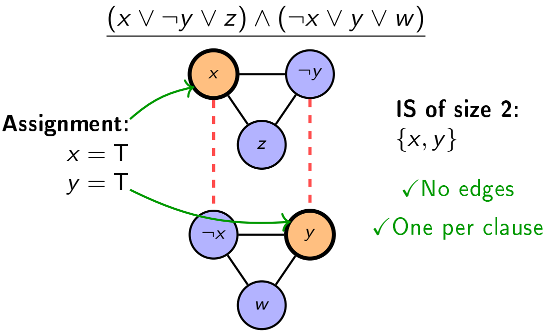
1. Literals → vertices
2. Clauses → triangles
3. Edges between x and $\neg x$
4. Set $k = \#\text{clauses}$

Why it works:

- Triangles ⇒ select ≤ 1 per clause
- Conflict edges ⇒ consistency

Conclusion

Independent Set is at least as hard as 3SAT



$k = m$ (number of clauses)

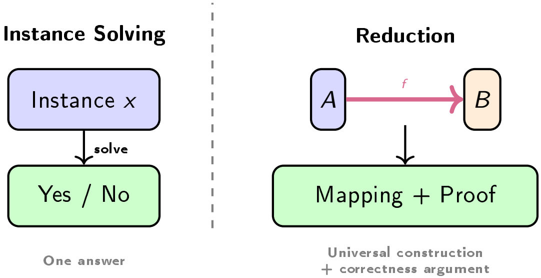
Reductions vs. Instance Solving

Instance solving:

- Input: one problem instance
- Output: yes/no (or a solution)
- Task: search or decision

Reduction construction:

- Input: two problem *definitions*
- Output: general transformation
- Task: design + proof



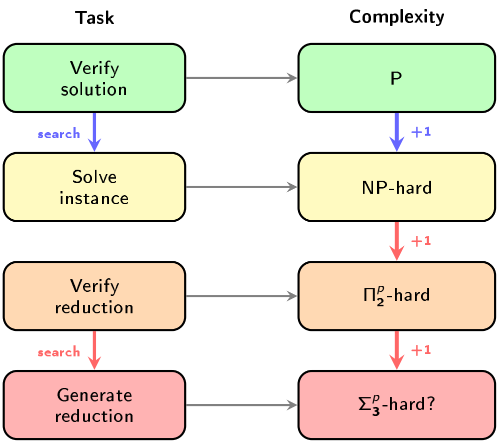
Reductions = A Meta-Hardness Test

Why reductions are harder:

- Gadget design + compositional reasoning
- Correctness proof

Verifying correctness is (likely):

- coNP-complete (circuits)
- Π_2^p-complete (NP verifiers)



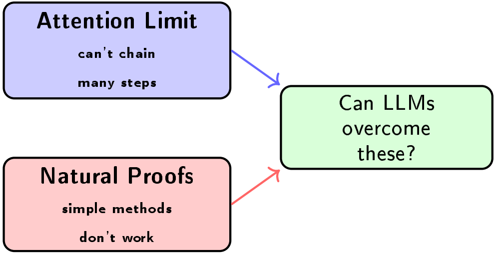
Philosophical Barriers

Why reductions are hard for LLMs:

- Reductions require chaining many local steps
- Transformers struggle here Peng et al. (2024)
  "Chopin's father is Nicolas" + "Nicolas born Apr 15"
  $\rightarrow$ "Chopin's father's birthday?" fails

Analogy from circuit complexity:

- Natural Proofs Razborov & Rudich (1997):
  "natural" techniques $\approx$ efficient algorithms
- Insufficient for proving lower bounds



Key Tension

If LLMs succeed at reductions, they're doing something nontrivial

The Karp Dataset

What:

- Curated reductions (textbook → research)
- Natural language + structured format
- Gadgets, mappings, correctness arguments

Goals:

- Evaluate & improve LLM reasoning
- Build formal verification tools
- Automatically grow/refine dataset

```json
{
  "entry_key": "3sat_to_independentset",
  "difficulty": 2,
  "reduction": {
    "source_problem": "3SAT",
    "target_problem": "Independent Set",
    "source_definition": {
      "name": "3SAT",
      "input_format": "(X,C), where X = ...",
      "yes_condition": "There exists an..."
    },
    "target_definition": {
      "name": "Independent Set",
      "input_format": "(G,k) where G = ...",
      "yes_condition": "G has an indepen..."
    },
    "reduction_steps": [
      "For each clause Ci = (ai OR bi...",
      "Add edges between every pair...",
      "Set k equal to the number of..."
    ],
    "forward_proof": "If the 3SAT...",
    "backward_proof": "Suppose the...",
  },
}
```

Preliminary Experiments

What we tried:

- Prompting off-the-shelf models
- Zero-shot reductions + self-check

Typical outcome:

- Readable, plausible-looking proofs
- Correct WHAT, missing HOW
- Unreliable self-verification

$\Rightarrow$ Motivates formal verification tools

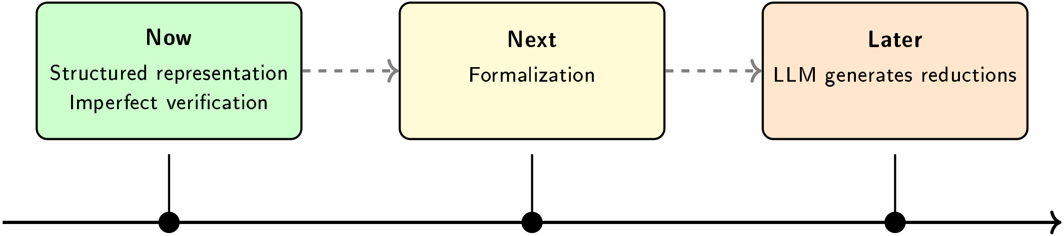
LLM output:
"Map clause ($a \vee b \vee c$) to triangle with vertices...
Add edges to avoid conflicts..."
(looks plausible)

check

Verification:
$\times$ **Conflict edge details lacking ($x \leftrightarrow \neg x$)**
$\times$ **Missing output size bound**
FAIL

Readable $\neq$ Correct
Self-verification is weak

Research Roadmap



Takeaways

This talk:

- Reductions probe compositional reasoning
- Current LLMs lack this capacity
- Even verification is difficult

Open questions:

- Auto-formalization feasible?
- Curriculum for reduction design?
- Connections to human reasoning?

Contact

Email: `mason.dicicco@gmail.com`

Questions?

References I

Wei et al., *Chain-of-Thought Prompting Elicits Reasoning in LLMs*, 2022. arXiv:2201.11903

Wang et al., *Self-Consistency Improves Chain of Thought Reasoning*, 2022. arXiv:2203.11171

Fan et al., *NPHardEval: Dynamic Benchmark on Reasoning Ability of LLMs via Complexity Classes*, ACL 2024.

Tang et al., *GraphArena: Evaluating and Exploring LLMs on Graph Computation*, 2024. arXiv:2407.00379

Yang et al., *NPPC: An Ever-Scaling Reasoning Benchmark for LLMs*, 2025. arXiv:2504.11239

Duchnowski et al., *EHOP: A Dataset of Everyday NP-Hard Optimization Problems*, 2025. arXiv:2502.13776

Peng et al., *On Limitations of the Transformer Architecture*, 2024. arXiv:2402.08164

Razborov & Rudich, *Natural Proofs*, JCSS 1997.

Franco et al., *Task-independent metrics of computational hardness predict human performance*, Sci. Rep. 2022.

Murawski & Bossaerts, *How Humans Solve Complex Problems: The Knapsack Problem*, Sci. Rep. 2016.

van Rooij, *The Tractable Cognition Thesis*, Cogn. Sci. 2008.

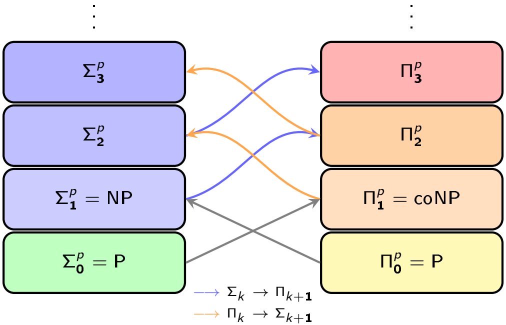
Dicicco et al., *The Karp Dataset*, 2025. arXiv:2501.14705

Karia et al., *Can LLMs translate SATisfactorily?*, AIA 2024.

Appendix: Complexity of Reduction Checking

Quantifier ladder (NP case):

- **Membership:** $\exists w \; R(x, w)$ $\boxed{\exists}$ Σ_1^p
- **Verification:** $\forall x \; [\exists w \cdots \iff \exists w' \cdots]$ $\boxed{\forall\exists}$ Π_2^p
- **Generation:** $\exists M \; \forall x \; [\exists w \cdots \iff \cdots]$ $\boxed{\exists\forall\exists}$ Σ_3^p



- The $\forall x$ (check reduction) adds one alternation.
- The $\exists M$ (guess reduction) adds another.

Appendix: LLM Reasoning

Chain-of-Thought (CoT):

- Multi-step prompting Wei et al. (2022)
- Self-consistency Wang et al. (2022)

But performance is brittle:

- Sensitive to phrasing
- Quality degrades over time