Structure Beats Compute: A 14,580-Trial Cognitive Audit of the 1980 Mattel TMS1100 vs. Modern Frontier LLMs

Chris
Strategy Arena Research
research@strategyarena.io

May 17, 2026

Abstract

Modern AI benchmarks often conflate capability with scale: larger models, larger context windows, and larger search budgets are presumed to dominate small structured systems. We test that assumption on Dragon Labyrinth, a sparse-reward, partially observable maze-pursuit task derived from Mattel's 1980 handheld game and its TMS1100-family 4-bit controller. The public benchmark reports 14,580 fixed-seed trials, 729 configurations, an 800-game module ablation, and a live 53-game human baseline in which the TMS1100 dragon won 52 games (98

Keywords: cognitive science; embedded systems; LLM benchmarking; vintage computing; partial observability; Monte Carlo Tree Search; POMDP; cognitive load.

1 Introduction

The recent history of artificial intelligence has been dominated by scaling arguments: more parameters, more pretraining data, more inference-time search, and more compute often improve performance across broad task distributions [Brown et al., 2020, Kaplan et al., 2020, Sutton, 2019]. That lesson is important, but incomplete. In many deployed decision problems, the limiting factor is not merely raw computation. Agents must maintain state under partial observability, avoid action loops, convert sparse feedback into usable priors, and exploit the structure of the environment. These requirements are especially visible in small, closed-world games and in embedded systems where computation is scarce.

This paper turns a historical curiosity into a cognitive benchmark. The Mattel *Dungeons & Dragons Computer Labyrinth* handheld game, released around 1980, implemented a maze-pursuit dragon on the TMS1100 family of 4-bit microcontrollers. The original dragon was effective not because it possessed flexible world understanding, but because it could exploit privileged full-state access to the maze. We reproduce the task, remove the information advantage, and compare modern language-model-driven play, brute-force search, human play, and a small structured policy. The resulting benchmark is a sparse-reward partially observable Markov decision process (POMDP) in the sense of Kaelbling et al. [1998].

Our thesis is deliberately narrow: in this task, structure beats generic compute. A structured agent that implements belief tracking and loop suppression substantially outperforms both bare LLM play and high-budget Monte Carlo Tree Search, despite being orders of magnitude smaller. This does not imply that 1980-era hardware is generally more capable than frontier models. It does show that benchmark design can expose failures of generic reasoning and the value of task-specific cognitive architecture.

2 Background

2.1 The TMS1100 and the Dragon Labyrinth task

The TMS1000/TMS1100 family integrated a small CPU, ROM, RAM, and I/O control into a single low-cost microcontroller [, b]. In handheld games of the era, behavior had to be compressed into tiny rule systems. The Dragon Labyrinth game is therefore a useful historical probe: its dragon policy is not a general planner, but a compact decision mechanism tied to the maze representation and display constraints [, a].

The original game condition gave the dragon full access to the maze state. The human player observed only a local line of sight. That asymmetry matters. A policy with full state access can appear intelligent even when it is mostly exploiting information privilege. In our reproduction, all evaluated agents receive the same partial observability. We then ask whether contemporary LLMs and brute search recover the lost structure from interaction alone.

2.2 Why this is an LLM benchmark

Language-model benchmarks such as BIG-Bench, MMLU, and ARC test broad reasoning, knowledge, and question-answering capabilities [Srivastava et al., 2022, Hendrycks et al., 2021, Clark et al., 2018]. Prompting methods such as chain-of-thought and tree-of-thought improve symbolic and deliberative problem solving [Wei et al., 2022, Yao et al., 2023]. DLB differs by forcing an agent to act in a compact dynamical system with hidden state and sparse success signals. The task is simple enough to inspect but hard enough to punish agents that lack an explicit state representation.

3 Method

3.1 Benchmark design

The public DLB summary reports 14,580 fixed-seed trials across 729 configurations, with 20 games per configuration. The canonical OutilsIA game page also reports a live public human baseline of 53 completed games, a 98% TMS1100 win rate against those games, and links to the CC-BY 4.0 dataset [Tendil and OutilsIA Research, 2026b]. The OutilsIA ablation report expands the module study to eight configurations over 800 fixed-seed games [Tendil and OutilsIA Research, 2026a]. The benchmark compares five classes of agent:

1. bare frontier LLM play, represented by Claude/Grok/GPT/Gemini-style prompts;
2. brute-force Monte Carlo Tree Search with 300K simulations per decision;
3. Oracle-X1, a small structured code policy using belief state and anti-loop logic;
4. trained human play as a cohort reference;
5. the original TMS1100 dragon under its historical full-state access condition.

The full-state TMS1100 condition is intentionally separated from the fair POMDP condition. It is a historical reference and an information-asymmetry control, not a fair agent baseline.

3.2 Cognitive modules

The structured policy is decomposed into modules:

- **M1: belief state**: maintain hypotheses about target location under line-of-sight limits;
- **M2: radius filter**: constrain candidates by distance, later found redundant with M1;
- **M3: oscillation killer**: suppress repeated loops and no-progress cycles;
- **M4/M5/M6**: danger estimation, opponent next-move prediction, and planned precomputed priors.

The central ablation result is the M1+M3 combination. M1 indicates where to look, while M3 indicates how not to loop. Together they create a decision architecture rather than a mere pile of heuristics.

4 Baselines

4.1 Live human baseline

The canonical OutilsIA game page is not merely a static paper mirror; it is the live "try it" endpoint where public human attempts are collected. It reports 53 completed human games and a 98% win rate for the TMS1100 dragon against those attempts [Tendil and OutilsIA Research, 2026b]. This baseline is deliberately separated from the trained-human cohort reference in Table 3. The live figure captures ordinary public play; the cohort reference captures trained play. Together they make the benchmark harder to dismiss as an LLM-only toy task.

Table 1: Live public human baseline from the canonical OutilsIA game page. The 98% figure is the TMS1100 dragon's win rate against completed human games, not the trained-human cohort reference.

Outcome	Games	Rate
TMS1100 dragon wins	52	98.1%
Human player wins	1	1.9%
Completed games	53	100.0%

4.2 Brute-force algorithmic baseline

The second defensive baseline is brute-force search. OutilsIA reports 12 million CUDA brute-force games reaching approximately 2% win rate, while the Strategy Arena summary also reports an MCTS ceiling around 2% in the aggregate suite. This matters because the failure is not only linguistic. A non-linguistic search baseline also plateaus when it does not maintain the right hidden-state representation.

Table 2: Algorithmic brute-force baseline. OutilsIA reports 12 million CUDA brute-force games reaching approximately 2% win rate; the Strategy Arena summary reports the same MCTS ceiling in the 14,580-trial benchmark suite.

Baseline	Win rate	Interpretation
Bare frontier LLM/API agents	0–1%	No persistent spatial world model.
PPO, 16M training steps	0%	Sparse reward collapse; learns survival over treasure.
CUDA/MCTS brute force	2%	Millions of rollouts do not solve hidden state.
Oracle-X1 structured policy	15%	Belief state plus anti-loop control.
TMS1100 historical/full-state	85%	Privileged information baseline.

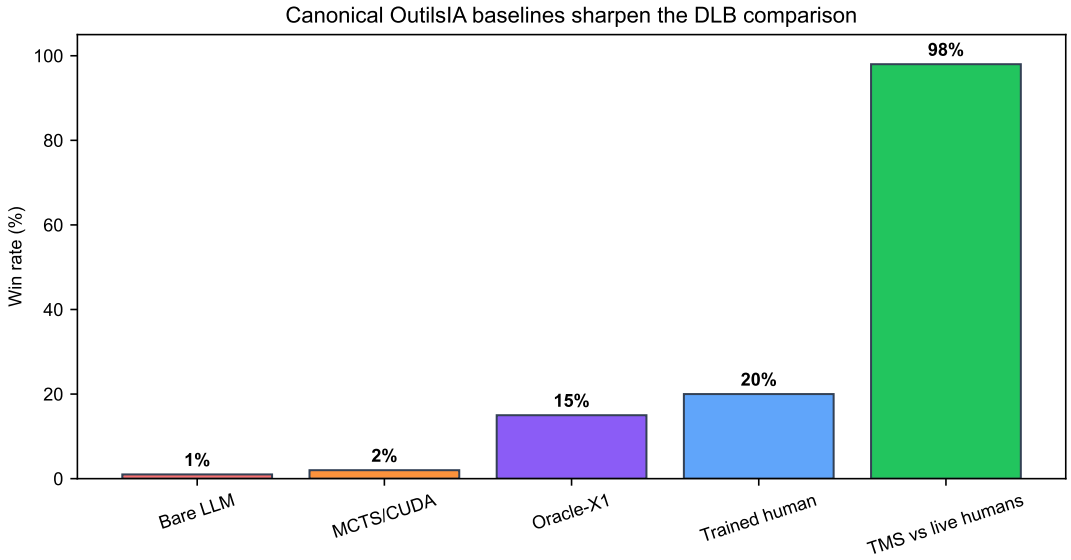

Figure 1: Canonical OutilsIA baselines. The live public human baseline makes the result concrete, while the MCTS/CUDA row shows that generic search does not remove the partial-observability bottleneck.

5 Results

Table 3: Aggregate DLB win rates. The TMS1100 row is historical/full-state and is not a fair POMDP baseline.

Approach	Win rate	Notes
TMS1100 dragon (1980, full-state)	85%	Historical information advantage.
Trained human	20%	20/80 cohort reference.
Oracle-X1 (M1+M3 structured code)	15%	Best code-only fair-condition result; 7.5x MCTS.
MCTS, 300K simulations per decision	2%	Plateaus; compute alone is insufficient.
Bare LLM (Claude/Grok/GPT/Gemini)	1%	Spatial blindness; no persistent world model.

Figure 2 visualizes the aggregate comparison. The decisive contrast is not simply between old and new hardware. It is between privileged state, generic compute, and explicit structure under fair partial observability.

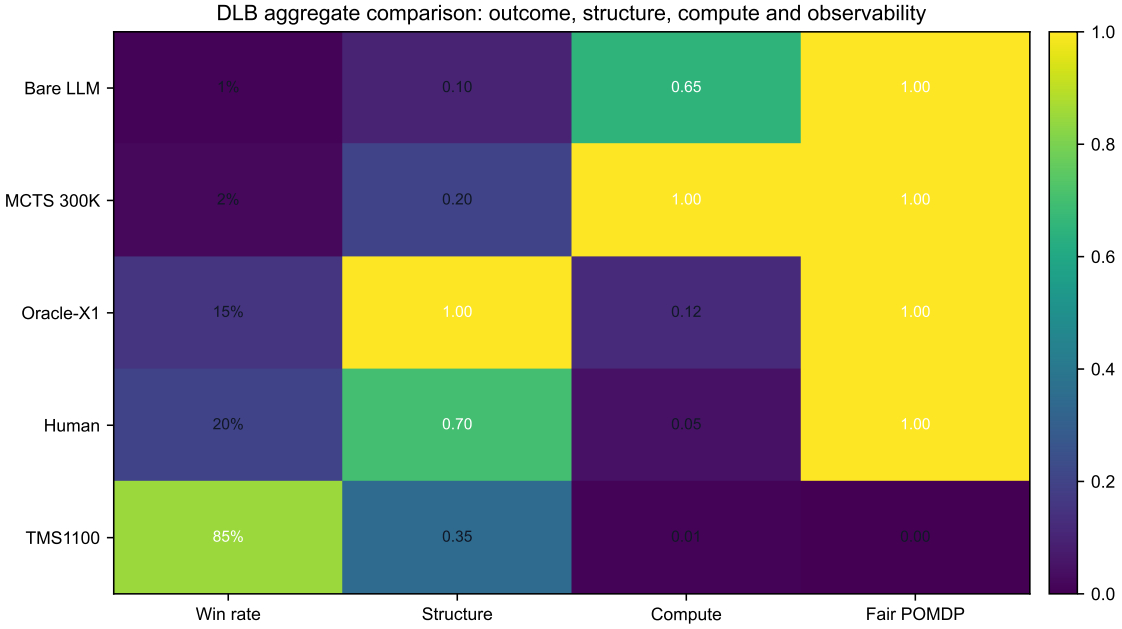

Figure 2: Aggregate DLB comparison derived from the public benchmark summary. The heatmap combines reported win rate with qualitative indices for structure, compute and fair partial observability.

The full eight-configuration module ablation is reported separately in Section efsec:ablationstudy.

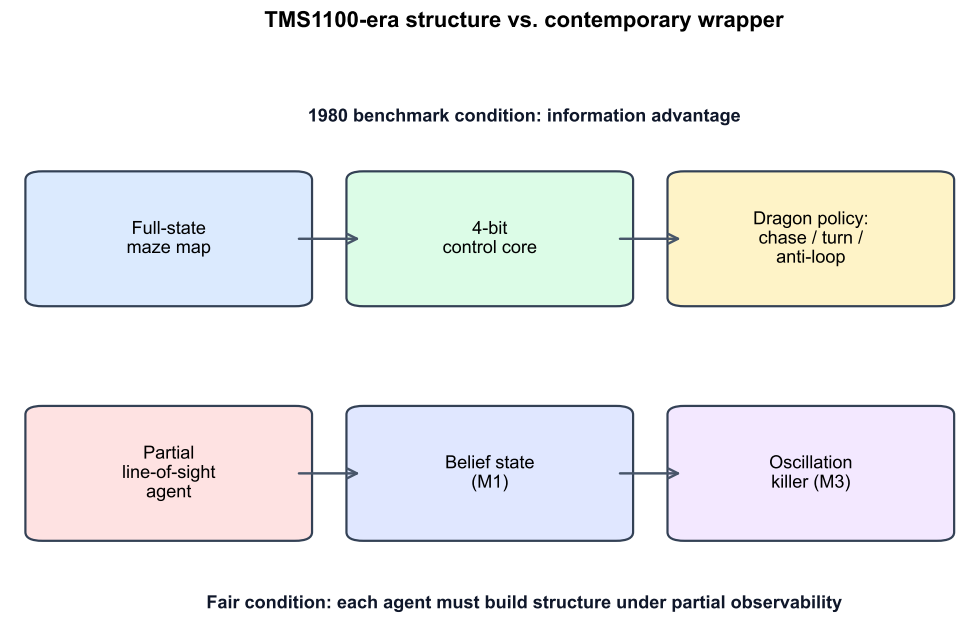

Figure 3: Conceptual architecture contrast. The historical TMS1100 condition exploits full-state access; the fair condition requires an agent to construct state under partial observability.

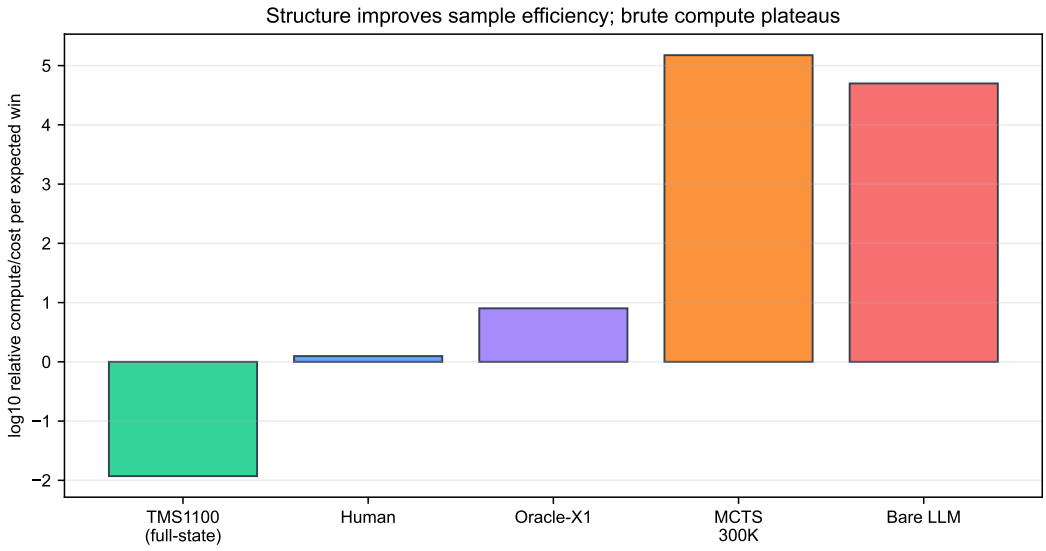

Figure 4: Relative compute/cost per expected win. Values are normalized indices, not billing estimates; the figure illustrates the plateau of brute compute and the efficiency of structure.

6 Ablation Study

The OutilsIA ablation article reports the full 2^3 design over M1, M2, and M3. Each configuration is evaluated on the same 100 maze seeds, yielding 800 controlled games in total [Tendil and OutilsIA

Research, 2026a]. The full table is stronger than the concise four-row summary because it reveals equivalence classes: NONE equals M2, M1 equals M1+M2, M3 equals M2+M3, and M1+M3 equals the full M1+M2+M3 stack.

Table 4: Eight-configuration Oracle-X1 ablation over 800 fixed-seed games from the OutilsIA ablation report.

Config	Meaning	Win	Treasure	Turns	Survival	Hits
NONE	No cognitive module	4.0%	15%	4.2	3.8	2.72
M1	Belief tracker	6.0%	13%	23.1	22.6	2.53
M2	Radius filter	4.0%	15%	4.2	3.8	2.72
M3	Oscillation killer	9.0%	26%	5.5	5.0	2.53
M1+M2	Belief + radius	6.0%	13%	23.1	22.6	2.53
M1+M3	Belief + anti-loop	15.0%	29%	29.7	29.2	1.96
M2+M3	Radius + anti-loop	9.0%	26%	5.5	5.0	2.53
M1+M2+M3	Full stack	15.0%	29%	29.7	29.2	1.96

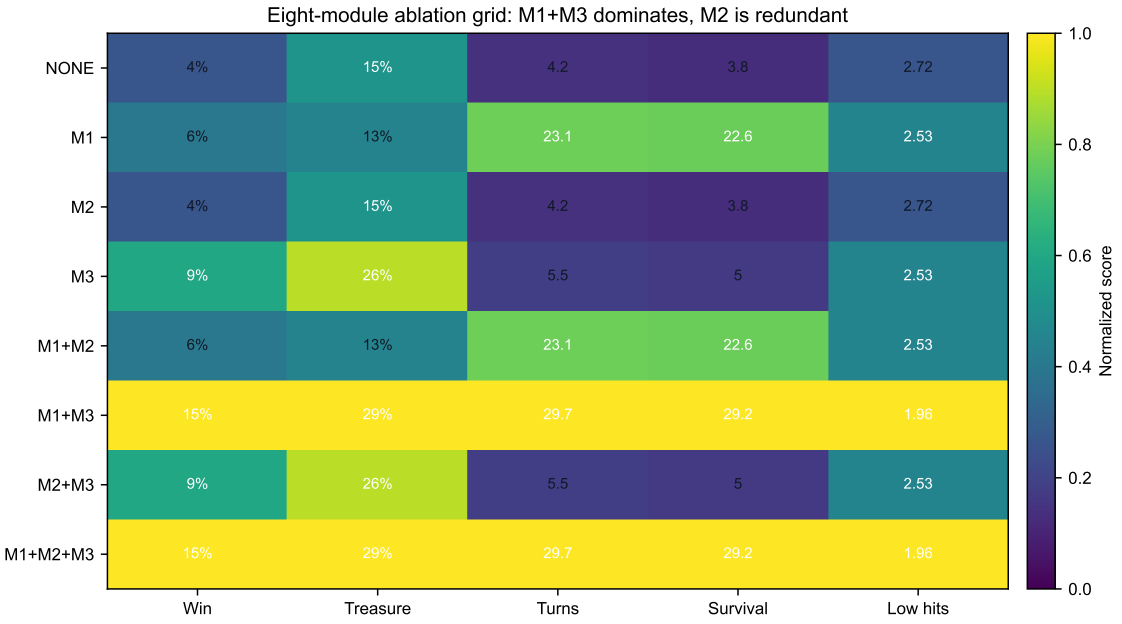

Figure 5: Full eight-configuration ablation grid. M2 is redundant across all pairings, while M1+M3 dominates both win rate and survival.

The interpretation is architectural rather than merely parametric. M1 supplies a persistent belief set over hidden treasure/dragon hypotheses; M3 prevents the agent from burning that information in local oscillations. M1 alone increases survival from 3.8 to 22.6 turns but barely raises win rate. M3 alone improves win rate to 9%, but does not construct a useful latent map. The combined M1+M3 stack reaches 15%, showing that state estimation and loop control are complementary.

7 Discussion

7.1 What the benchmark does and does not claim

The result should not be read as a broad claim that old hardware is superior to modern AI. The TMS1100 full-state row is an information-cheating baseline. Its purpose is to show how privileged state can masquerade as intelligence. The more interesting result is that when the cheat is removed, bare LLMs and brute-force MCTS do not automatically recover the required task structure. Oracle-X1, despite being a small engineered system, does better because it encodes the right state variables.

7.2 Independent qualitative confirmation

The OutilsIA ablation report includes a separate Gemini analysis that describes the same failure mode in qualitative terms: LLMs lack a persistent mental map and treat looped movement as separate local events rather than as a return to a prior state [Tendil and OutilsIA Research, 2026a]. We do not present this as an independent statistical replication. It is better understood as a cross-model diagnostic agreement: a second LLM conversation independently named the same world-model and spatial-blindness gap that the ablation quantified through M1 and M3.

7.3 POMDP and belief-state framing

The simplicity-trap development log attributes the key conceptual pivot to a Grok analysis: represent the invisible dragon/treasure configuration not as an intuition, but as a finite belief set updated by Mattel constraints and line-of-sight observations [Tendil and OutilsIA Research, 2026c]. In POMDP language, Oracle-X1 is not winning by being larger. It wins by externalizing belief state. The prompt-level LLM baselines lack that durable state across turns, so they repeatedly rediscover local facts, oscillate, or optimize for short-term survival instead of the treasure-return objective.

7.4 Implications for LLM evaluation

LLMs can generate explanations, plans, and code, but closed-loop spatial control under hidden state remains fragile when the model lacks a persistent world model. DLB stresses exactly that gap. It complements broad language benchmarks by measuring action under partial observability rather than answer selection over static prompts.

7.5 Implications for embedded and structure-first AI

TinyML and embedded AI emphasize operating under severe resource constraints [Warden and Situnayake, 2019]. DLB adds a historical lens: compact systems can be strong when they possess the right task decomposition. A practical agent need not always be larger. It may need a better state abstraction, a better loop-avoidance rule, or an explicit prior over environment structure.

8 Limitations

This bundle still uses published aggregate data rather than a full raw-trial archive inside the paper source. The OutilsIA pages state that the code, CSVs, seven ablations, and 14,580-trial dataset are available under CC-BY 4.0, and the Strategy Arena mirror exposes public summary JSON. Future versions should attach raw per-trial CSV, seed list, agent prompts, MCTS settings, and

confidence-interval scripts directly. The Gemini and Grok material should also be interpreted carefully: it is useful diagnostic provenance from collaborative AI analysis, not a peer-reviewed replication. Finally, DLB is narrow: it tests a small POMDP-like game, not general intelligence.

9 Conclusion

DLB is a small benchmark with a large warning: compute can plateau when an agent lacks the right representation. In 14,580 trials, bare LLMs reach roughly 1% win rate and high-budget MCTS/CUDA search reaches 2%, while a structured M1+M3 policy reaches 15%. The live OutilsIA game page adds a public human baseline: the TMS1100 dragon wins 52 of 53 completed human games. The scientific lesson is therefore not nostalgia. It is that benchmark design must separate information advantage, compute, and cognitive structure.

The narrative inversion is sharp: a 1980 chase rule described as roughly six Manhattan lines can still expose failure modes in a 2026 implementation stack of roughly 3300 lines, a 550x complexity ratio [Tendil and OutilsIA Research, 2026c]. More code and more rollouts did not automatically produce the missing belief state. The winning move was a smaller, more explicit cognitive architecture.

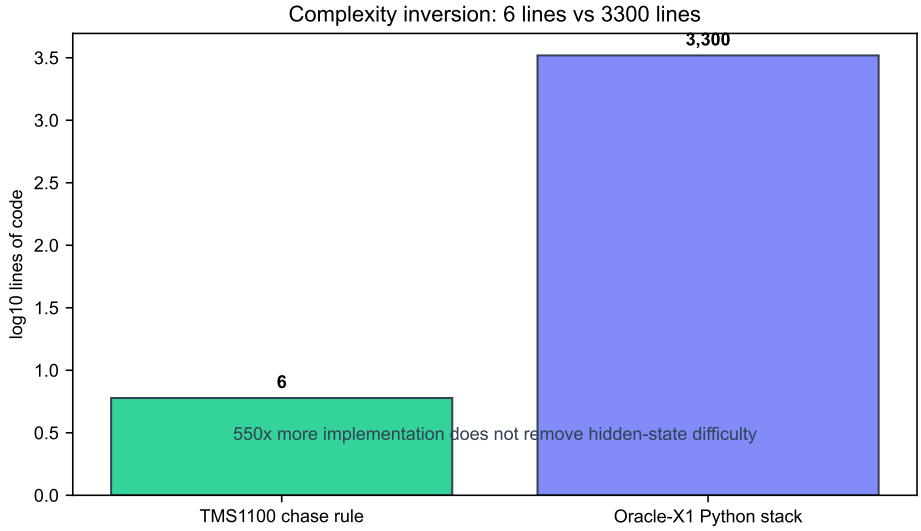

Figure 6: Narrative complexity inversion from the OutilsIA simplicity-trap log. The figure is illustrative: the core lesson is not literal line-count minimalism, but that representation dominates implementation mass in this task.

Data and Reproducibility

The canonical live game is available at `https://outilsia.fr/games/dnd-labyrinth`; the Strategy Arena research mirror is available at `https://strategyarena.io/dragon-labyrinth-benchmark`. The public game page links the dataset CSV for 14,580 trials under CC-BY 4.0, and this bundle includes the rendered Strategy Arena page, its public JSON summary, the three OutilsIA source pages, clean text extractions, derived CSV/LaTeX tables, and vector figures generated from the published aggregate data. The open challenge is live: human players and agents can play the OutilsIA endpoint and contribute further observations.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations*, 2021.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Dungeons & Dragons Computer Labyrinth Game Instructions. Mattel Electronics, 1980a. Consumer handheld game documentation.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Richard S. Sutton. The bitter lesson. http://www.incompleteideas.net/IncIdeas/BitterLesson.html, 2019.

Lysiane Tendil and OutilsIA Research. Tms1100 (1980) bat l'ia 2026: ablation study sur 800 parties. https://outilsia.fr/blog/tms1100-vs-ia-2026-ablation, 2026a. 800-game ablation study of M1/M2/M3 cognitive modules, accessed 2026-05-17.

Lysiane Tendil and OutilsIA Research. D&d labyrinth: Mattel 1980 x ia 2026. https://outilsia.fr/games/dnd-labyrinth, 2026b. Canonical live playable benchmark and public human baseline, accessed 2026-05-17.

Lysiane Tendil and OutilsIA Research. Le piege de la simplicite: 6 lignes de 1980 vs 3300 lignes de 2026. https://outilsia.fr/blog/piege-simplicite-dragon-2026, 2026c. Belief-state tracking discussion, Grok analytical contribution, and development log, accessed 2026-05-17.

TMS1000 Series MOS/LSI One-Chip Microcomputers Data Manual. Texas Instruments, 1976b. Primary documentation for the TMS1000/TMS1100 family of 4-bit microcontrollers.

Pete Warden and Daniel Situnayake. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O'Reilly Media, 2019.

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. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.

A Published Summary JSON

The file `sources/dlb_summary.json` is included in the source bundle. It reports the schema `dlb-summary-v1`, the 14,580 total trials, the 729 configurations, and the 800-game ablation study used for the tables above.

B arXiv Compilation Notes

The manuscript is designed for `pdflatex` plus `bibtex`. Compile with:

```
pdflatex main.tex
bibtex main
pdflatex main.tex
pdflatex main.tex
```