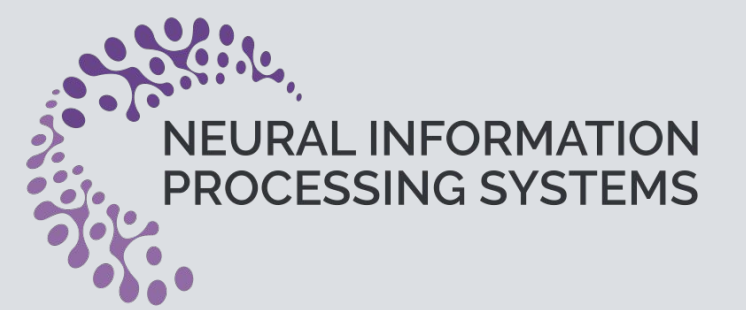
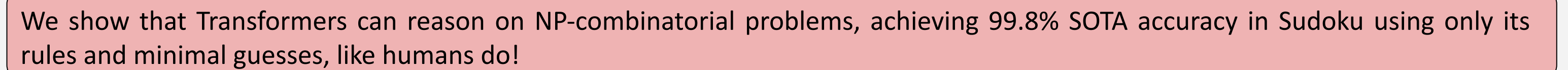


Christos Tzamos^{2,3}

³Archimedes, Athena Research Center, Greece

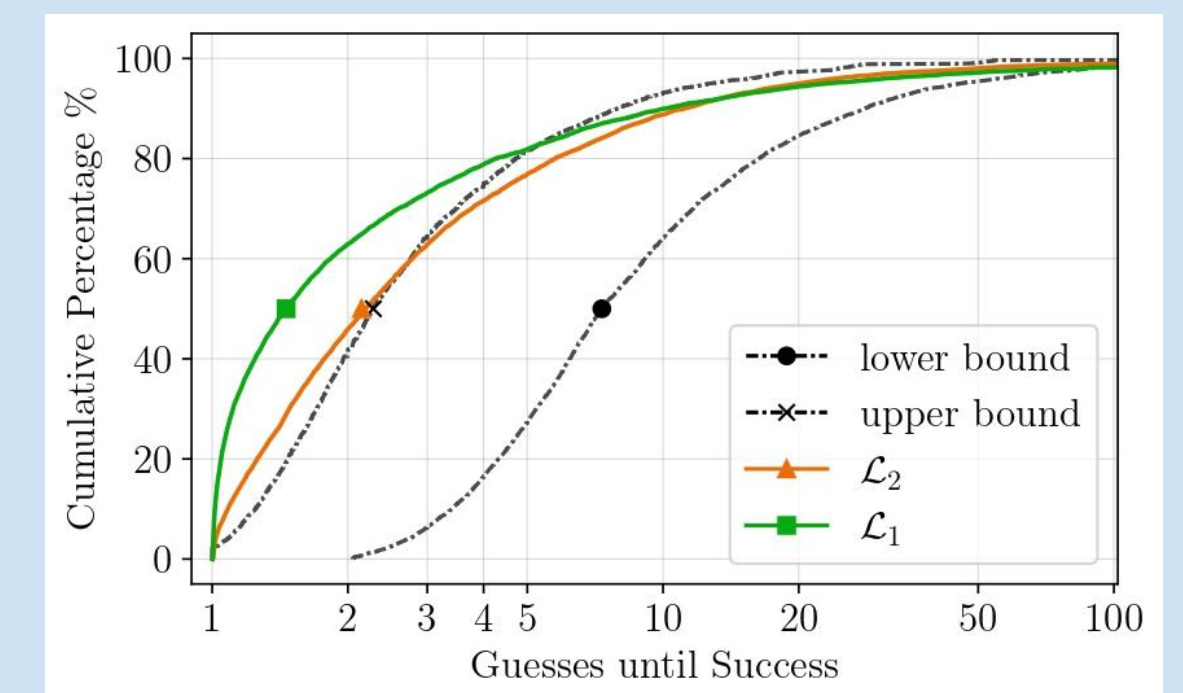


QR Code



Sudoku rules + a single guess + guess loss inspired by Min-Sum Set Cover problem

-
- The diagram illustrates the GPT-2 architecture, divided into two main phases: **training** and **decoding**.
- Legend:**
- S (start):** White circle with 'S'.
 - r (rules end):** Green circle with 'r'.
 - g (guess node):** Yellow square with 'g'.
 - d (dead end):** Red circle with 'd'.
 - e (end):** White circle with 'e'.
- Training Phase:**
- given:** A sequence of tokens (116, ..., 989) represented by white circles.
 - rules:** A sequence of tokens (135, 318, 731) represented by blue circles.
 - Backdoors:** A sequence of tokens (427, 917, 975) represented by yellow squares.
 - Process:** The 'given' tokens are processed by the 'S' node, which then feeds into the 'rules' nodes. The 'rules' nodes feed into the 'g' node. The 'g' node feeds into the 'Backdoors' nodes. The 'Backdoors' nodes feed into the 'r' node.
- Decoding Phase:**
- Process:** The 'r' node feeds into the 'g' node. The 'g' node feeds into a sequence of nodes (343, 356, 363, ..., 776, 536, 725) represented by blue circles. The 'g' node also feeds into a sequence of nodes (898, 178, 279, ..., 471, 898) represented by blue circles. The 'g' node also feeds into a sequence of nodes (975, 471, 898) represented by blue circles. The 'g' node also feeds into a sequence of nodes (427, 917, 975) represented by yellow squares. The 'g' node also feeds into a sequence of nodes (427, 917, 975) represented by yellow squares. The 'g' node also feeds into a sequence of nodes (427, 917, 975) represented by yellow squares.
 - Restart:** A red dashed arrow labeled 'Restart' points from the 'd' node back to the 'g' node.



Insights: This approach shows empirically that 99.8% of Sudokus can be solved by using only one guess (backdoor guess)!

- Assumptions: depth-1 of search and non adaptive policy
- Challenge:
 - You face n possible choices, but only a hidden subset S is valid
 - Subset S is drawn from a known distribution D
 - Each test costs 1 time unit and once it is made you only learn if it is valid
- Goal: Find a policy π that minimizes the expected time to discover a valid choice

Theorem. For any distribution \mathcal{D} over sets $S \subseteq [n]$, it holds that for any permutation τ :

$$\min_{\pi \in \Delta(n)} \mathbb{E}_{S \sim \mathcal{D}} \left[\frac{1}{\sum_{i \in S} \pi_i} \right] \leq H_n \cdot \mathbb{E}_{S \sim \mathcal{D}} \left[\arg \min_{i=1}^n \{\tau_i \in S\} \right]$$

where $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \Theta(\log n)$ is the n -th harmonic number.

Remark: Loss function (1) yields solutions with a bounded approximation to the optimal policy, whereas treating the problem as a multi-class classification task (e.g., weighted Cross-Entropy Loss) leads to much worse approximations.

- R. Palm, U. Paquet, O. Winther. Recurrent relational networks. NeurIPS'18.
- Z. Yang, A. Ishay, J. Lee. Learning to solve constraint satisfaction problems with recurrent transformer. ICLR'23.
- K. Shah, N. Dikkala, X.Wang, R.Panigrahy. Causal language modeling can elicit search and reasoning capabilities on logic puzzles. NeurIPS'24.
- J.Kim, K. Shah, V. Kontonis, S. Kakade, S. Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. ICML'25.

- A generalized framework for NP-class problems that have an efficient verifier
 - Sudoku is NP-complete problem
- No custom architecture: valina-decoder only GPT-2 with 42M parameters
- Training transcripts solely based on Sudoku rules (DFS method) and guesses
 - An approach that can solve Sudoku puzzles beyond those solvable by human-crafted strategies
 - Close related approach to human nature
- SOTA results; accuracy 99.8% on randomly generated Sudoku
- Minimization of guesses via a novel loss inspired by *Min-Sum Set Cover* problem
- SudokuPy: a Python library for generating random Sudoku puzzles

Sudoku rules + informed multiple guesses

- Encoding: Each move encoded as a 3-digit number rcv (111 \rightarrow 999); row_column_value
- Multiple Targets: Combinatorial puzzles allow multiple valid next moves
 - Instead of a single deterministic label, we support multiple next-token predictions

Cross Entropy loss: $-\log p_i \rightarrow$ Multiple target loss: $-\sum \log p_i$ over all valid next tokens
- Results: Accuracy 98.9%; SOTA compared to other prior works
Accuracy 99,10% in 1-3 SAT problem (canonical NP-complete problem)

