LLMs for Game Theory: Entropy-Guided In-Context Learning and Adaptive CoT Reasoning

Tommaso Felice Banfi[1], **Sashenka Gamage**[2]
[1]Politecnico di Milano, Dept. of Electronics, Information and Bioengineering, Milan, Italy.
[2]The Hong Kong Polytechnic University, Dept. of Electrical and Electronics Engineering, Hong Kong, China.
tommasofelice.banfi@mail.polimi.it, sashenka.gamage@connect.polyu.hk

Abstract

We propose a novel LLM-based framework for reasoning in discrete, game-theoretic tasks, illustrated with *Tic-Tac-Toe*. The method integrates in-context learning with entropy-guided chain-of-thought (CoT) reasoning and adaptive context retrieval. The model dynamically adjusts both the number of retrieved examples and reasoning paths according to token-level uncertainty: concise reasoning with minimal context is used when uncertainty is low, whereas higher uncertainty triggers expanded multi-path CoT exploration. Experimental evaluation against a sub-optimal algorithmic opponent shows that entropy-aware adaptive reasoning substantially improves decision quality, increasing the average game outcome from -11.6% with the baseline LLM to $+9.5\%$ with entropy-guided adaptive reasoning over 100 games (win = +1, tie = 0, loss = -1), while maintaining a relatively low number of LLM queries per game. Statistical validation confirms that the improvement is significant, and correlation analysis reveals a negative association between token-level entropy and move optimality. These findings demonstrate that uncertainty-guided adaptive reasoning effectively enhances LLM performance in sequential decision-making environments.

1 Introduction

Large Language Models (LLMs) have recently shown impressive capabilities across a wide range of reasoning and knowledge-intensive tasks. They perform well in single-step inference, language understanding, and few-shot generalization. However, they still face notable challenges in structured, sequential decision-making problems, where each move affects all future states and outcomes. In such settings, effective reasoning requires not only local consistency but also long-term planning and adaptive use of prior knowledge.

In this work, we explore how to improve the performance of LLMs in sequential reasoning environments by combining two complementary mechanisms. First, we present *entropy-guided context retrieval*, a method that dynamically adjusts the number of retrieved examples based on the model's predictive uncertainty. When the model shows low entropy (high confidence), it works with a compact and focused context; when uncertainty increases, it automatically retrieves semantically similar past states and their corresponding optimal actions, extending its in-context knowledge. Second, we introduce an *adaptive chain-of-thought (CoT)* reasoning framework, where the model uses concise, single-path reasoning when confident and branches into multiple reasoning paths when token-level entropy indicates uncertainty. Together, these mechanisms enable efficient, uncertainty-aware reasoning, focusing computational effort only when necessary.

We use *Tic-Tac-Toe* as a controlled and fully observable testbed to evaluate our approach. Despite its simplicity, the game captures key aspects of sequential decision-making (state transitions, opponent modeling, and outcome optimization) while allowing objective evaluation of move quality through the minimax algorithm. This setup supports a rigorous and quantitative assessment of LLM reasoning and provides statistical validation of the proposed methods.

The rest of this paper is organized as follows. Section 2 reviews related research on reasoning with LLMs, in-context learning, and uncertainty estimation. Section 3 defines the game-theoretic setting and evaluation framework. Our proposed approach, including entropy-guided context retrieval and adaptive CoT reasoning, is presented in Section 4. Section 5 describes the experimental setup, baselines, and metrics, followed by a detailed analysis of the results in Section 6. Section 7 outlines current limitations and potential directions for future research. Finally, Section 8 concludes the paper, highlighting the broader impact of uncertainty-aware reasoning for LLMs in sequential decision-making tasks.

2 Related Work

Prior works in the NLP research community has shown that chain-of-thought prompting improves LLM performance on complex reasoning tasks by exposing intermediate reasoning steps (Wei et al. 2023). In addition, frameworks such as Tree-of-Thoughts implement branching over "thought" states (i.e., multi-path reasoning (Zhou et al. 2024) and tree-based CoT (Bi et al. 2025)) to perform deliberative search with LLMs, demonstrating substantial performance gains in non-trivial tasks (Yao et al. 2023).

However, evaluating CoT and related reasoning frameworks remains an open challenge. Most of the methods focus on getting the answer, rather than evaluating the quality of the reasoning method. Benchmarking studies (Golovneva

---

et al. 2023; Prasad et al. 2023; Jacovi et al. 2024) have shown that CoT reasoning performance can vary widely across tasks and domains, and that standardized evaluation metrics are often insufficient to capture nuanced reasoning quality. A recent survey (Lee and Hockenmaier 2025) emphasize that assessing CoT effectiveness in open-ended reasoning tasks is inconsistent due to the fragmented progress over evaluation design and benchmark development. In our setting, this limitation is mitigated, as each game state in *Tic-Tac-Toe* has a well-defined optimal move due to the underlying game theory. This allows us to quantitatively evaluate step-by-step reasoning quality by comparing model-generated moves against the algorithmic optimal strategy obtained algorithmically.

In line with that, retrieval-augmented generation methods (RAG) (Lewis et al. 2021) have shown that giving examples at inference time improves knowledge-intensive tasks and reduces the burden on the model's internal parameters. Finally, uncertainty-aware or entropy-guided reasoning approaches (Yao et al. 2023; Wang et al. 2023; Zhang et al. 2025; Jia, Cai, and Liu 2025), adaptively decide when to expand the LLM reasoning depth or breadth (based on the model's uncertainty). Recent work also highlights the synergy between RAG and in-context learning (ICL), where the strong inductive feature bias of ICL can be reduced by RAG, injecting external information at inference time, providing more relevant examples or facts (Min et al. 2022). As we discussed before, evaluating only final answers in multi-step reasoning often obscures the nuanced reasoning errors that models make. For instance, GridPuzzle (Tyagi et al. 2024) develops a dataset and evaluation framework to analyze reasoning chains in grid-based puzzles, identifying where models like GPT-4, Claude-3, and LLaMA-2 falter. Similarly, the EIC (Error Identification and Correction) benchmark (Li et al. 2024) evaluates mathematical reasoning by highlighting error types and enabling models to correct mistakes with targeted prompts. These works emphasize the importance of truly measuring reasoning ability beyond overall accuracy.

Other methods, such as RCOT (Reversing Chain-of-Thought) (Xue et al. 2023) and Verify-and-Edit (Zhao et al. 2023), focus on test-time correction by detecting inconsistencies in the generated reasoning chains and guiding LLMs to revise them. RCOT performs a backtracking approach by reconstructing problems from solutions to find factual inconsistencies, while Verify-and-Edit introduces knowledge-aware editing to improve CoT reliability.

Classical game AI research has long relied on search algorithms such as minimax (Russell and Norvig 2020) and alpha-beta pruning (Knuth and Moore 1975). With the advent of deep learning, systems like AlphaZero have shown that integrating learned policy networks with planning can outperform pure search-based engines in complex games like chess and Go (Silver et al. 2017). The Alympics ("Olympics for Agents"), a simulation framework that uses LLM agents to study game-theoretic interactions, offers a methodology for constructing strategic games, defining evaluation criteria (Mao et al. 2024). It highlights issues of strategic reasoning, adaptation, and how LLMs can simulate human-like strategic interactions. By incorporating entropy-based measures of uncertainty (Wang et al. 2023; Zhou et al. 2023), one can identify decisions where LLM agents are less confident and adaptively expand reasoning paths, reduce hallucination, and improve efficiency by focusing computation where uncertainty is high. Building on these insights, our approach integrates retrieval-augmented prompting with entropy-guided chain-of-thought reasoning to simulate agentic decision-making in fully observable game environments, enabling both rigorous evaluation and controlled sub-optimality testing.

In our case, the game of *Tic-Tac-Toe* is simple enough that a full minimax solution is feasible and provides the optimal moves for all possible board states. Our work lies at the intersection, we combined a learned vector representation of game states with retrieval-augmented prompting and an entropy-guided chain-of-thought strategy.

3 Problem Definition

We consider discrete, two-player, turn-based games, a standard framework in game theory for sequential decision-making under perfect information. A game is defined by the tuple (S, A, T, R), where:

- S denotes the set of possible game states (board configurations),
- $A(s)$ is the set of legal actions available in state $s \in S$,
- $T : S \times A \to S$ represents the deterministic transition function, and
- $R : S \times A \to \mathbb{R}$ is the reward function.

We focus on *Tic-Tac-Toe*, a finite, zero-sum, perfect-information game played on a 3×3 grid. Players alternate placing their marker (X or O) on an empty cell. A player wins by completing a horizontal, vertical, or diagonal line of three markers, while a draw occurs if all cells are filled without a winning configuration.

Formally, the state space is $S = \{s_1, s_2, \ldots, s_{3^9}\}$, and for each state s, the action space is defined as $A(s) = \{(i, j) \mid \text{cell } (i, j) \text{ is empty}\}$. The game evolves according to the transition function $s_{t+1} = T(s_t, a_t)$ at each turn t.

In our setting, the LLM acts as one of the players, competing against a fixed suboptimal algorithmic opponent. Its performance is evaluated based on the distribution of outcomes (win, tie, and loss) across multiple games. During play, the model can retrieve contextual information from a limited database of board states and corresponding optimal actions to inform its decisions.

4 Method

In this section, we present the proposed framework for LLM-based decision-making in *Tic-Tac-Toe*. The approach integrates two complementary components: *entropy-guided context retrieval* and *adaptive chain-of-thought reasoning*, designed to improve efficiency and robustness in sequential reasoning tasks.

The overall architecture follows a game loop where the LLM observes the current board state, generates a reasoning trace, and selects the next action based on both its internal confidence and retrieved contextual information. Although developed and evaluated in the context of turn-based games, the framework is general and can be extended to broader sequential decision-making scenarios involving LLMs.

4.1 Game Loop

The match is initialized with an empty board B, where the LLM plays as "X" and the algorithmic agent as "O". The starting player is selected randomly, and the game proceeds as follows:

1. The algorithmic agent selects a move according to its policy and updates the board B.
2. A structured prompt is constructed for the LLM, including the current board configuration, available moves, the identity of the active player, and a set of retrieved examples with their corresponding optimal actions (see Section 4.2).
3. The prompt is passed to $\text{LLM}(x_q, p, \mathcal{R}_q)$, and the model's textual output is obtained.
4. The output is parsed to extract move coordinates. If the move is invalid, the LLM is prompted to regenerate. If it remains invalid, a random valid move is selected.
5. If a terminal state (win, loss, or tie) is reached, the game ends; otherwise, the loop continues from step 1.

4.2 Board Representation and Context Retrieval

Following Tyagi et al. (Tyagi et al. 2024), we represent the *Tic-Tac-Toe* board as a 3×3 tensor

$$B \in \{0, 1, 2\}^{3\times 3}, \tag{1}$$

where 0 denotes an empty cell, 1 represents the marker "X", and 2 represents "O". To facilitate LLM-based retrieval, we flatten B into a vector

$$x = \text{vec}(B) \in \mathbb{R}^9 \tag{2}$$

and encode it into a lower-dimensional latent representation $z \in \mathbb{R}^d$ using an autoencoder:

$$z = f_\theta(x), \tag{3}$$

$$\hat{x} = g_\phi(z), \tag{4}$$

where $f_\theta : \mathbb{R}^9 \to \mathbb{R}^d$ is the encoder and $g_\phi : \mathbb{R}^d \to \mathbb{R}^9$ is the decoder. The reconstruction loss ensures that the latent space preserves the essential information of the board:

$$\mathcal{L}_{\text{rec}}(\theta, \phi) = \|x - g_\phi(f_\theta(x))\|^2. \tag{5}$$

To enable effective context retrieval, we further structure the latent space using a contrastive learning objective. Given a pair of boards (x_i, x_j) with associated optimal moves (m_i, m_j), the contrastive loss encourages latent vectors of boards with the same optimal move to be close, while pushing apart boards with different optimal moves:

$$\mathcal{L}_{\text{con}} = \begin{cases} \|f_\theta(x_i) - f_\theta(x_j)\|^2, & \text{if } m_i = m_j, \\ \max\left(0, \tau - \|f_\theta(x_i) - f_\theta(x_j)\|\right)^2, & \text{if } m_i \neq m_j, \end{cases} \tag{6}$$

where $\tau > 0$ is a margin hyperparameter controlling the minimum distance between latent representations of boards with different optimal moves. This formulation ensures that similar board states are grouped in the latent space according to the optimal action, enabling efficient retrieval of relevant past states for in-context reasoning.

The final training objective combines the reconstruction and contrastive components:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{con}}, \tag{7}$$

where $\lambda > 0$ balances the contribution of the contrastive loss relative to the reconstruction loss. Minimizing $\mathcal{L}$ results in a latent space that simultaneously preserves board structure and organizes states according to optimal moves, supporting uncertainty-aware context retrieval for LLM reasoning.

Vector Database. A vector database is constructed as

$$\mathcal{D} = \{(z_i, m_i)\}, \tag{8}$$

where each entry consists of a latent embedding $z_i = f_\theta(x_i)$ paired with its corresponding optimal move m_i. In practice, approximately 20% of all possible board states are stored in $\mathcal{D}$, selected uniformly at random, together with their minimax-optimal moves.

At inference time, given a query board x_q, we first compute its embedding

$$z_q = f_\theta(x_q). \tag{9}$$

The retrieval set $\mathcal{R}_q$ is then constructed by selecting the k most similar examples from $\mathcal{D}$ according to cosine similarity:

$$\text{sim}(z_q, z_i) = \frac{z_q \cdot z_i}{\|z_q\| \, \|z_i\|}, \tag{10}$$

$$\mathcal{R}_q = \text{Top-}k\{(x_i, m_i) \in \mathcal{D} \mid \text{sim}(z_q, z_i) \text{ largest}\}. \tag{11}$$

To adapt retrieval to the model's uncertainty, we employ an *entropy-aware* mechanism detailed in Section 4.3.

The retrieval size k is dynamically adjusted according to

$$k = \min\left(k_{\max}, \lceil k_0 + \alpha \cdot H_q \rceil\right), \tag{12}$$

where k_0 is the base number of examples, $\alpha > 0$ is a scaling factor, and $k_{\max}$ is a hard cap determined by the maximum available context window of the LLM. Consequently, high-entropy outputs trigger the retrieval of additional examples to enrich the context, while low-entropy outputs limit the retrieval to a small set of nearest neighbors, preserving context space for reasoning. In all cases, the size of the final LLM context $\mathcal{C}_q$ is bounded by the token budget $|\mathcal{C}_q| \leq L_{\max}$, where $L_{\max}$ denotes the maximum number of tokens the model can condition on.

Context Construction. The final context $\mathcal{C}_q$ provided to the LLM consists of three components:

1. The current board configuration x_q,
2. The identity of the active player $p \in \{\text{X}, \text{O}\}$, and
3. The retrieved set $\mathcal{R}_q = \{(x_{i_j}, m_{i_j})\}_{j=1}^{k}$, containing the k most similar boards and their corresponding optimal moves.

These elements are concatenated and formatted into a structured prompt, which is then passed as input to the LLM, see Appendix A for an example of the prompt. This structured context ensures that the model can leverage both the current board state and relevant past experiences to make informed decisions.

4.3 Adaptive Chain-of-Thought Reasoning

In our framework, *chain-of-thought* (CoT) reasoning is employed to model sequential decision-making in *Tic-Tac-Toe*. Each CoT step corresponds to a game turn, and intermediate reasoning involves simulating both the current player's move and the opponent's response. Formally, let $s_t \in S$ denote the board state at turn t, and let the CoT generate a sequence of state-action pairs:

$$\text{CoT} = \big((s_0, a_0), (s_1, a_1), \dots, (s_T, a_T)\big), \tag{13}$$

where $a_t \in A(s_t)$ is the move selected by the active player or the opponent at step t, and T is the horizon until a terminal state is reached.

Reasoning Modes. We define several CoT strategies with increasing complexity:

- **Direct output**: the LLM outputs a single move a_t for the current state s_t without simulating future steps. Mathematically, the selected action is

  $$a_t = \arg\max_{a \in A(s_t)} P_{\text{LLM}}(a \mid s_t, \mathcal{C}_t), \tag{14}$$

  where $\mathcal{C}_t$ is the context provided to the LLM.
- **Multi-CoT**: the LLM generates n independent single-path sequences $\text{CoT}^{(j)}$ from the current state until a terminal state is reached. The final move is selected by a majority vote over the first actions $a_0^{(j)}$ of each sequence:

  $$a_t = \text{mode}\{a_0^{(j)}\}_{j=1}^{n}. \tag{15}$$
- **Tree-based CoT**: at each step, the LLM generates n candidate actions for the active player, and for each candidate, all possible responses of the opponent are simulated, forming a tree of depth d:

  $$\mathcal{T}_t = \{(s_{t+1}^{(i)}, a_{t+1}^{(i)})\}_{i=1}^{n \cdot |A(s_{t+1})|}. \tag{16}$$

  The evaluation function $V : S \to \mathbb{R}$ (entropy based selection) is used to select the top-k branches at each level:

  $$\mathcal{B}_t = \text{Top-}k\{\mathcal{T}_t \mid V(s_{t+1}^{(i)}) \text{ is maximal}\}. \tag{17}$$
- **Entropy-guided CoT**: a hybrid approach in which multiple reasoning branches are generated only when the LLM exhibits high uncertainty. Uncertainty is quantified by the token-level entropy of the CoT output (see below). Formally, for a reasoning step t, if $H_t^{\text{step}} > H_{\text{th}}$, the model generates n_t parallel branches:

  $$n_t = \min\big(n, |A(s_t)|\big), \tag{18}$$

  where $|A(s_t)|$ is the number of legal actions at s_t.

Token-level Entropy and Adaptive Branching. At each reasoning step t, the LLM generates a sequence of tokens $\{v_{t,k}\}_{k=1}^{L_t}$, each associated with a probability distribution over the vocabulary $\{p_{t,k}^{(i)}\}_{i=1}^{|V|}$. The token-level entropy is computed as

$$H_{t,k}^{\text{token}} = -\sum_{i=1}^{|V|} p_{t,k}^{(i)} \log p_{t,k}^{(i)}, \tag{19}$$

and the average token entropy defines the step-level entropy:

$$H_t^{\text{step}} = \frac{1}{L_t} \sum_{k=1}^{L_t} H_{t,k}^{\text{token}}. \tag{20}$$

To adaptively determine the number of reasoning branches, we define a set of ordered entropy thresholds

$$0 = H_0 < H_1 < \cdots < H_m, \tag{21}$$

where m is the number of threshold levels. Each threshold H_j corresponds to a predefined number of branches n_j, forming a mapping

$$H_t^{\text{step}} \in [H_j, H_{j+1}) \;\Rightarrow\; n_t = n_j, \quad j = 0, \dots, m-1. \tag{22}$$

In this way, higher step-level entropy triggers the generation of more reasoning branches, while low entropy results in fewer branches, optimizing computational resources.

The actual number of branches is selected as

$$n_t = \min\big(n_j, |A(s_t)|\big), \tag{23}$$

where $|A(s_t)|$ is the number of legal actions at state s_t.

Once generated, each branch represents a candidate CoT trajectory that is recursively expanded. To control the computational cost, only the top-k branches are retained at each step according to a scoring function $S(\text{CoT})$ based on the average entropy of the tokens in the reasoning path:

$$\mathcal{B}_t = \text{Top-}k\{\text{CoT branches at step } t \mid S(\text{CoT})\}. \tag{24}$$

This adaptive threshold mechanism allows the model to selectively explore multiple reasoning paths only when the uncertainty is high, providing a principled balance between reasoning quality and computational efficiency. Future extensions may consider dynamically updating thresholds H_j based on the distribution of entropies observed across previous steps, rather than using fixed values.

5 Experimental Setup

We evaluate the proposed LLM agent against a suboptimal algorithmic opponent in *Tic-Tac-Toe*. Each game continues until a win or a tie. The opponent employs a precomputed Minimax table to rank all legal moves according to their optimality.

Let n denote the number of legal moves, and let $r_i \in \{1, \dots, n\}$ be the rank of move i (with $r_i = 1$ corresponding to the most optimal move, and $r_i = n$ to the least optimal). Given a skill level $\alpha \in [0, 1]$ (set to $\alpha = 0.95$ in our experiments), the probability of selecting move i is defined as

$$P(i) = \frac{\max\big(0, 1 - |r_i - \alpha n|/(n-1)\big)}{\sum_{j=1}^{n} \max\big(0, 1 - |r_j - \alpha n|/(n-1)\big)}. \tag{25}$$

Table 1: Performance of the LLM agent against an algorithmic opponent using the LLaMA-7B model. Each cell shows the average outcome per game S over 100 games and the rounded average number of LLM queries per game. Positive S values indicate that the LLM agent won more games than it lost (e.g., +10 means 10% more wins than losses), and vice versa for negative values.

	No Additional Context		Fixed-Size Context		**Entropy-Guided Context**	
	S [%]	Queries	S [%]	Queries	S [%]	Queries
No CoT	-11.6	3	-5.2	4	-2.8	4
Single CoT	-8.2	13	-2.6	13	-0.1	15
Multi CoT	-7.5	24	-1.2	26	+4.8	28
Tree-based CoT	-2.7	165	+4.5	178	+9.8	188
Entropy-Guided CoT	-4.1	48	+3.8	56	**+9.5**	**48**

This distribution peaks at the move corresponding to the agent's skill level and decays linearly toward both less optimal and overly optimal moves, ensuring that the agent plays moves consistent with its skill on average while introducing controlled randomness. Since *Tic-Tac-Toe* is a solved game, statistical validation of the strategy is unnecessary, as the Minimax table provides a complete ranking of valid moves for any board state.

The LLM employs parallel and tree-based chain-of-thought (CoT) reasoning. In our experiments, we configured $n = 3$ parallel CoT sequences, each allowing up to 3 branches per turn, and retain at most $k = 10$ top paths at each step to control computational cost.

All experiments utilize the LLaMA-7B model (Touvron et al. 2023a,b) to establish baseline performance. This model was chosen to balance inference capability and computational resource requirements, as further discussed in Section 7.

We report two primary metrics per configuration:

1. The average game outcome S over 100 games, expressed as a percentage, with scores assigned as win = +1, tie = 0, and loss = -1.
2. The average number of LLM queries per game, including additional calls required for multi-path reasoning, recomputation due to invalid outputs, and context expansion under high uncertainty.

For the first move of the game, due to board symmetry and multiple equally optimal moves, entropy-based branching is not applied and all equally optimal moves are considered valid without adjustment.

Hyperparameters and Implementation Details. All experiments employed the LLaMA-7B model (Touvron et al. 2023a) to establish baseline performance. For initial testing, we used the smaller Gemma 3 270M model (Team et al. 2025) on a single NVIDIA GeForce RTX 4050 with 6 GB VRAM and an Intel i7-12700H CPU with 16 GB RAM. Final evaluations with LLaMA-7B (Touvron et al. 2023a) were conducted on an NVIDIA GeForce RTX 4060 Ti with 16 GB VRAM, sufficient for inference-only tasks. The operating system used was Ubuntu 24.04 LTS.

For next-move generation, LLM hyperparameters were set to ensure deterministic and precise outputs: temperature = 0.1, top-p and top-k sampling disabled (top_p=1.0, top_k=0), and beam search set to 2. The maximum number of generated tokens per move was 10, covering all possible coordinate formats. Padding and end-of-sequence tokens were assigned to the tokenizer's `eos_token_id`.

A fixed random seed of 42 was used to ensure reproducibility in selecting and ordering retrieval examples (see Section 4.2). Each combination of context type and CoT reasoning (3 context types × 5 CoT strategies = 15 configurations) was initially evaluated using the Gemma 3 model. Final evaluations employed LLaMA-7B with 100 games per configuration against the algorithmic opponent, measuring both average game outcome S and the number of LLM queries per game. Hyperparameters were selected based on preliminary experiments to guarantee consistent generation of valid move coordinates.

The implementation was developed in Python using the `transformers` library for model and tokenizer management, `bitsandbytes` for quantization, `torch` as the main framework, and `langchain` to handle context retrieval and CoT reasoning.

6 Results and Discussion

Our results in Table 1 and Figure 1 show that both context retrieval and chain-of-thought (CoT) reasoning substantially affect the LLM agent's performance. Without additional context, the agent performs poorly, with negative average outcomes across all CoT strategies. Introducing a fixed-size context improves performance for all strategies by providing relevant examples that guide move selection. Entropy-guided context retrieval further enhances outcomes by dynamically adjusting the number of retrieved examples based on predictive uncertainty, allowing the model to access more relevant information when needed. For example, using entropy-guided context without CoT improves the outcome from -11.6% to -2.8%. Multi CoT generates multiple candidate reasoning paths, and tree-based CoT explores a broader set of future states and opponent responses, achieving the highest performance (+9.8%) but with a substantially higher number of queries (188 on average). Combining entropy-guided CoT with entropy-aware context retrieval achieves nearly equivalent performance (+9.5%) while requiring only 48 queries, roughly one-fourth of the tree-based CoT cost. These results demonstrate that adaptive context retrieval and selective multi-path reasoning enable efficient, high-quality decision-making with significantly reduced computational overhead.

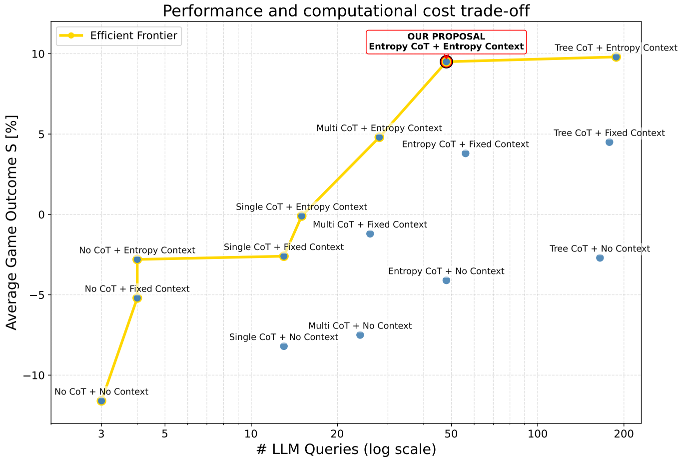

Figure 1: Comparison of retrieval-based in-context learning methods with different CoT strategies, evaluated in terms of average game outcome S and computational cost (number of LLM queries). The plot highlights the performance–cost trade-off and the resulting efficient frontier across methods.

Entropy as a Proxy for Uncertainty. Token-level entropy has been recently used as a measure of model uncertainty in discrete multi-step reasoning tasks (Zhang et al. 2025; Jia, Cai, and Liu 2025; Zhou et al. 2023). It quantifies the uncertainty of the predicted probability distribution over the next token: low entropy indicates a concentrated distribution on a few tokens, while high entropy corresponds to a more uniform distribution over multiple plausible tokens.

We evaluated the LLM on a random subset of 500 valid board states, generating the next move without CoT or additional context and recording the token-level entropy of each output, as defined in Equation 20. For each move, we computed its ranking percentile relative to all valid moves in the same state. Figure 2 shows the relationship between token-level entropy and move optimality percentile.

High entropy arises primarily in two cases: (i) genuine uncertainty about the next move (true positive), or (ii) multiple moves with similar optimality (false positive). False positives are mostly observed in early-game states and are mitigated by avoiding entropy-based branching on the first move. In general, high entropy consistently indicates potentially suboptimal choices.

To quantify this noisy relationship, we computed rank-based correlations between entropy and move optimality percentiles. Spearman correlation is $\rho = -0.471, p < 10^{-3}$, and Kendall Tau correlation is $\tau = -0.346, p < 10^{-3}$, both statistically significant. These results confirm a consistent negative trend: moves with higher entropy are more likely to be suboptimal.

This analysis supports the use of token-level entropy as a proxy for uncertainty in sequential decision-making tasks.

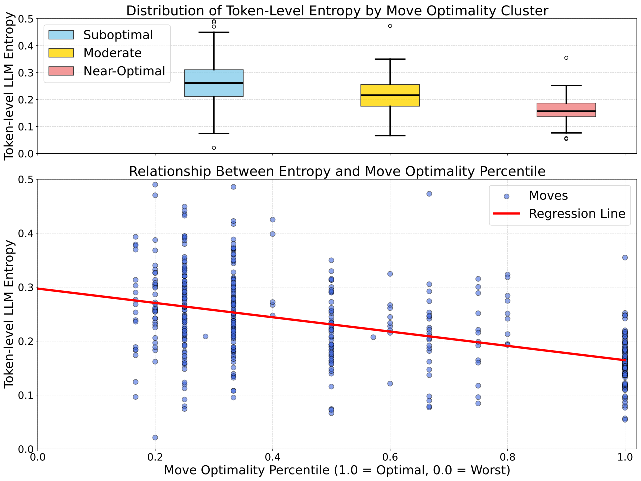

Figure 2: Token-level entropy versus move optimality percentile. Top: boxplots grouped into three clusters (*Suboptimal*, *Moderate*, *Near-Optimal*), showing higher entropy for less optimal moves. Bottom: scatter plot of individual moves with a linear regression line, illustrating the negative relationship between entropy and move optimality. Data are based on 500 randomly selected board states.

Discussion on Zero-Shot Claim. The results support the hypothesis that retrieval-augmented in-context learning (RA-ICL) enables effective zero-shot reasoning in structured decision-making tasks such as *Tic-Tac-Toe*. Although the retrieval encoder described in Section 4.2 was trained in a self-supervised manner on domain data, the objective of learning semantically consistent representations of board states is task-agnostic and does not involve supervision on optimal actions. Thus, the LLM is never fine-tuned for the downstream task, and reasoning emerges purely from contextual conditioning provided by retrieval.

Unlike fine-tuning approaches, which modify model parameters, retrieval-augmented inference operates externally: relevant examples are dynamically retrieved and injected into the prompt based on latent similarity. This mechanism aligns with large-scale architectures such as GPT-5 (OpenAI 2025) and Gemini 2.5 (Gemini Team, Google 2024), where retrieval modules enable adaptive context composition accessing vast external knowledge bases (e.g., web documents, code repositories, etc.) without model fine-tuning.

Empirically, the model exhibits context-driven reasoning that scales with the adaptivity of the retrieval process. The entropy-guided mechanism allows selective context expansion only under high uncertainty, reducing unnecessary computation while maintaining decision quality. Overall, the proposed architecture satisfies the key zero-shot conditions: no task-specific fine-tuning, self-supervised retrieval embedding construction, and dynamic in-context adaptation driven by model uncertainty.

These results indicate that retrieval-augmented chain-of-thought reasoning constitutes a viable and scalable zero-shot framework for structured problem-solving.

7 Limitations and Future Work

The proposed framework has several limitations that define directions for future work.

First, all experiments were conducted on *Tic-Tac-Toe*, a deterministic and fully observable game with a small state and action space. This simplified setup allows controlled evaluation but does not capture the challenges of larger or stochastic environments. Extending the framework to more complex games such as Connect Four or Go would test scalability and robustness under higher-dimensional state representations and longer reasoning sequences.

Second, the model used (LLaMA-7B (Touvron et al. 2023a)) has limited reasoning capacity. This helps isolate the effects of adaptive chain-of-thought and context retrieval, but stronger models (e.g., GPT-5 (OpenAI 2025), Gemini 2.5 (Gemini Team, Google 2024)) may already handle part of this reasoning intrinsically. Future work should test whether entropy-guided mechanisms still offer benefits when the base model is more capable.

Third, the use of token-level entropy assumes a direct link between linguistic uncertainty and decision uncertainty. While supported by correlation results, this assumption may not hold in settings where uncertainty arises from ambiguity rather than lack of knowledge. Alternative uncertainty estimators, such as ensemble variance or mutual information, could provide a more reliable signal.

Fourth, the retrieval database was generated from the full game tree and may not generalize to domains with incomplete or biased data. Retrieval redundancy or conflict between examples could also affect the model's reasoning consistency.

Finally, the evaluation focused on quantitative performance (win/loss and query count) without assessing logical soundness or interpretability of the reasoning traces. Extending analysis to partially observable tasks and studying reasoning quality under uncertainty remain open directions.

Future work will address these limitations by scaling the approach to larger games and more general decision-making tasks, integrating improved uncertainty estimation, and evaluating reasoning behavior beyond accuracy-based metrics.

8 Conclusion

We introduced a framework for sequential decision-making with large language models that combines entropy-guided context retrieval with adaptive chain-of-thought reasoning. By selectively expanding reasoning depth and contextual information under uncertainty, the method improves decision quality while maintaining a low number of LLM queries. In our experiments on *Tic-Tac-Toe*, this approach increased the average game outcome from -11.6% to +9.5%, demonstrating that uncertainty-aware adaptation effectively enhances model consistency and efficiency.

Although tested on a simple, fully observable game, the framework is model-agnostic and generalizable to larger or partially observable domains. Its core principle of dynamically allocating reasoning effort based on uncertainty offers a scalable method for integrating retrieval-augmented reasoning and adaptive CoT in more complex decision-making or reasoning tasks where zero-shot performance remains limited.

Future extensions will explore applying this mechanism to domains with non-deterministic dynamics and incomplete information, where uncertainty-guided reasoning may provide significant benefits in both performance and interpretability.

Acknowledgments

The author would like to thank the anonymous reviewers for their valuable feedback. This work was developed by the author T.F. Banfi while also affiliated with Huawei Milan/-Maxwell Research Center, but it is independent and not related to his work there. This research did not receive any specific grant.

References

Bi, Z.; Han, K.; Liu, C.; Tang, Y.; and Wang, Y. 2025. Forest-of-Thought: Scaling Test-Time Compute for Enhancing LLM Reasoning. arXiv:2412.09078.

Gemini Team, Google. 2024. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. Technical report, Google DeepMind. Technical report on Gemini 2.X model family.

Golovneva, O.; Chen, M.; Poff, S.; Corredor, M.; Zettlemoyer, L.; Fazel-Zarandi, M.; and Celikyilmaz, A. 2023. ROSCOE: A Suite of Metrics for Scoring Step-by-Step Reasoning. arXiv:2212.07919.

Jacovi, A.; Bitton, Y.; Bohnet, B.; Herzig, J.; Honovich, O.; Tseng, M.; Collins, M.; Aharoni, R.; and Geva, M. 2024. A Chain-of-Thought Is as Strong as Its Weakest Link: A Benchmark for Verifiers of Reasoning Chains. arXiv:2402.00559.

Jia, Y.; Cai, J.; and Liu, X. 2025. Entropy-Guided Tree of Thoughts: A Dynamic Approach to Diverse Path Generation in LLM Reasoning. In *2025 3rd International Conference on Big Data and Privacy Computing (BDPC)*, 132–136.

Knuth, D. E.; and Moore, R. W. 1975. An analysis of alpha-beta pruning. *Artificial Intelligence*, 6(4): 293–326.

Lee, J.; and Hockenmaier, J. 2025. Evaluating Step-by-step Reasoning Traces: A Survey. arXiv:2502.12289.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; tau Yih, W.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401.

Li, X.; Wang, W.; Li, M.; Guo, J.; Zhang, Y.; and Feng, F. 2024. Evaluating Mathematical Reasoning of Large Language Models: A Focus on Error Identification and Correction. arXiv:2406.00755.

Mao, S.; Cai, Y.; Xia, Y.; Wu, W.; Wang, X.; Wang, F.; Ge, T.; and Wei, F. 2024. ALYMPICS: LLM Agents Meet Game Theory – Exploring Strategic Decision-Making with AI Agents. arXiv:2311.03220.

Min, S.; Lyu, X.; Holtzman, A.; Artetxe, M.; Lewis, M.; Hajishirzi, H.; and Zettlemoyer, L. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? arXiv:2202.12837.

OpenAI. 2025. GPT-5 System Card. Technical report, OpenAI. Version with system card, August 13, 2025.

Prasad, A.; Saha, S.; Zhou, X.; and Bansal, M. 2023. ReCEval: Evaluating Reasoning Chains via Correctness and Informativeness. arXiv:2304.10703.

Russell, S.; and Norvig, P. 2020. *Artificial Intelligence: A Modern Approach*. Hoboken, NJ, USA: Pearson, 4th edition. ISBN 978-0134610993.

Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; Lillicrap, T.; Simonyan, K.; and Hassabis, D. 2017. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. arXiv:1712.01815.

Team, G.; Kamath, A.; Ferret, J.; Pathak, S.; Vieillard, N.; Merhej, R.; Perrin, S.; Matejovicova, T.; Ramé, A.; Rivière, M.; Rouillard, L.; Mesnard, T.; Cideron, G.; bastien Grill, J.; Ramos, S.; Yvinec, E.; Casbon, M.; Pot, E.; Penchev, I.; Liu, G.; Visin, F.; Kenealy, K.; Beyer, L.; Zhai, X.; Tsitsulin, A.; Busa-Fekete, R.; Feng, A.; Sachdeva, N.; Coleman, B.; Gao, Y.; Mustafa, B.; Barr, I.; Parisotto, E.; Tian, D.; Eyal, M.; Cherry, C.; Peter, J.-T.; Sinopalnikov, D.; Bhupatiraju, S.; Agarwal, R.; Kazemi, M.; Malkin, D.; Kumar, R.; Vilar, D.; Brusilovsky, I.; Luo, J.; Steiner, A.; Friesen, A.; Sharma, A.; Sharma, A.; Gilady, A. M.; Goedeckemeyer, A.; Saade, A.; Feng, A.; Kolesnikov, A.; Bendebury, A.; Abdagic, A.; Vadi, A.; György, A.; Pinto, A. S.; Das, A.; Bapna, A.; Miech, A.; Yang, A.; Paterson, A.; Shenoy, A.; Chakrabarti, A.; Piot, B.; Wu, B.; Shahriari, B.; Petrini, B.; Chen, C.; Lan, C. L.; Choquette-Choo, C. A.; Carey, C.; Brick, C.; Deutsch, D.; Eisenbud, D.; Cattle, D.; Cheng, D.; Paparas, D.; Sreepathihalli, D. S.; Reid, D.; Tran, D.; Zelle, D.; Noland, E.; Huizenga, E.; Kharitonov, E.; Liu, F.; Amirkhanyan, G.; Cameron, G.; Hashemi, H.; Klimczak-Plucińska, H.; Singh, H.; Mehta, H.; Lehri, H. T.; Hazimeh, H.; Ballantyne, I.; Szpektor, I.; Nardini, I.; Pouget-Abadie, J.; Chan, J.; Stanton, J.; Wieting, J.; Lai, J.; Orbay, J.; Fernandez, J.; Newlan, J.; yeong Ji, J.; Singh, J.; Black, K.; Yu, K.; Hui, K.; Vodrahalli, K.; Greff, K.; Qiu, L.; Valentine, M.; Coelho, M.; Ritter, M.; Hoffman, M.; Watson, M.; Chaturvedi, M.; Moynihan, M.; Ma, M.; Babar, N.; Noy, N.; Byrd, N.; Roy, N.; Momchev, N.; Chauhan, N.; Sachdeva, N.; Bunyan, O.; Botarda, P.; Caron, P.; Rubenstein, P. K.; Culliton, P.; Schmid, P.; Sessa, P. G.; Xu, P.; Stanczyk, P.; Tafti, P.; Shivanna, R.; Wu, R.; Pan, R.; Rokni, R.; Willoughby, R.; Vallu, R.; Mullins, R.; Jerome, S.; Smoot, S.; Girgin, S.; Iqbal, S.; Reddy, S.; Sheth, S.; Põder, S.; Bhatnagar, S.; Panyam, S. R.; Eiger, S.; Zhang, S.; Liu, T.; Yacovone, T.; Liechty, T.; Kalra, U.; Evci, U.; Misra, V.; Roseberry, V.; Feinberg, V.; Kolesnikov, V.; Han, W.; Kwon, W.; Chen, X.; Chow, Y.; Zhu, Y.; Wei, Z.; Egyed, Z.; Cotruta, V.; Giang, M.; Kirk, P.; Rao, A.; Black, K.; Babar, N.; Lo, J.; Moreira, E.; Martins, L. G.; Sanseviero, O.; Gonzalez, L.; Gleicher, Z.; Warkentin, T.; Mirrokni, V.; Senter, E.; Collins, E.; Barral, J.; Ghahramani, Z.; Hadsell, R.; Matias, Y.; Sculley, D.; Petrov, S.; Fiedel, N.; Shazeer, N.; Vinyals, O.; Dean, J.; Hassabis, D.; Kavukcuoglu, K.; Farabet, C.; Buchatskaya, E.; Alayrac, J.-B.; Anil, R.; Dmitry; Lepikhin; Borgeaud, S.; Bachem, O.; Joulin, A.; Andreev, A.; Hardin, C.; Dadashi, R.; and Hussenot, L. 2025. Gemma 3 Technical Report. arXiv:2503.19786.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023a. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023b. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Tyagi, N.; Parmar, M.; Kulkarni, M.; RRV, A.; Patel, N.; Nakamura, M.; Mitra, A.; and Baral, C. 2024. Step-by-Step Reasoning to Solve Grid Puzzles: Where do LLMs Falter? arXiv:2407.14790.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv:2203.11171.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.

Xue, T.; Wang, Z.; Wang, Z.; Han, C.; Yu, P.; and Ji, H. 2023. RCOT: Detecting and Rectifying Factual Inconsistency in Reasoning by Reversing Chain-of-Thought. *arXiv preprint arXiv:2305.11499*.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T. L.; Cao, Y.; and Narasimhan, K. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. arXiv:2305.10601.

Zhang, J.; Wang, X.; Mo, F.; Zhou, Y.; Gao, W.; and Liu, K. 2025. Entropy-based Exploration Conduction for Multi-step Reasoning. arXiv:2503.15848.

Zhao, R.; Li, X.; Joty, S.; Qin, C.; and Bing, L. 2023. Verify-and-Edit: A Knowledge-Enhanced Chain-of-Thought Framework. *arXiv preprint arXiv:2305.03268*.

Zhou, C.; You, W.; Li, J.; Ye, J.; Chen, K.; and Zhang, M. 2023. INFORM : Information eNtropy based multi-step reasoning FOR large language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on*

Empirical Methods in Natural Language Processing, 3565–3576. Singapore: Association for Computational Linguistics.

Zhou, Q.; Zhang, Z.; Xiang, X.; Wang, K.; Wu, Y.; and Li, Y. 2024. Enhancing the General Agent Capabilities of Low-Parameter LLMs through Tuning and Multi-Branch Reasoning. arXiv:2403.19962.

A LLM Input Prompt Example

The following example in Box A illustrates the structured prompt provided to the LLM during gameplay. This prompt includes retrieved examples with corresponding optimal moves, the current board state, and the list of available positions, ensuring consistent input formatting and supporting in-context reasoning.

We note that numerous variations in prompt design and formatting are possible, and more extensive prompt engineering could potentially improve performance. In our experiments, we found that the simple coordinate format (x, y) worked reliably, both for facilitating the LLM's output and for programmatically extracting moves from the textual response. An alternative would have been a structured JSON format with separate fields for x and y, but this was discarded after initial testing as the simpler format provided sufficient accuracy and consistency for our objectives. Prompt optimization, while potentially beneficial, is outside the scope of this work.

LLM Prompt Example

```
EXAMPLES OF BOARDS AND
CORRESPONDING OPTIMAL MOVES:

Example 1:
Board:
X | O | X
O | X | _
_ | _ | O
Optimal move: (2,2)

...

CURRENT GAME STATE:
You are playing Tic-Tac-Toe as X.
The opponent is O.
Current board (3x3, _ for empty):
O | _ | X
_ | X | O
_ | _ | _

Empty positions (x,y):
(0,1), (1,0), (2,0), (2,1), (2,2)

You are X.
Your next move in (x,y) format is:
```

Reproducibility Checklist

---

1. General Paper Structure

1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) Yes, see Section 4 Method.

1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) Yes, see Section 6 Results and Discussion.

1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) Yes, see Section 1 Introduction and Section 2 Related Work.

2. Theoretical Contributions

2.1. Does this paper make theoretical contributions? (yes/no) Yes, but the paper does not introduce a completly new formal theoretical method. The work relies on the theoretical assumption of entropy as a proxy for uncertainty, which is empirically validated in Section 6 Results and Discussion.

If yes, please address the following points:

2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) Yes, see Section 7 Limitations and Future Work.

2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) NA.

2.4. Proofs of all novel claims are included (yes/partial/no) NA.

2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) NA.

2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) Yes, see Section 2 Related Work and Section 3 Method.

2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) Yes, see Section 6 Results and Discussion where we empirically validate and discuss the use of entropy as a proxy for uncertainty.

2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) No, evaluation code will be publicly released upon publication.

3. Dataset Usage

3.1. Does this paper rely on one or more datasets? (yes/no) No, we do not rely on any datasets. Optimal moves are precomputed using the Minimax algorithm for Tic-Tac-Toe.

If yes, please address the following points:

3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) NA

3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) NA

3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) NA

3.5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations (yes/no/NA) NA

3.6. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available (yes/partial/no/NA) NA

3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing (yes/partial/no/NA) NA

4. Computational Experiments

4.1. Does this paper include computational experiments? (yes/no) Yes. For all subsequent questions in this subsection, if not otherwise specified, refer to Section 5 Experimental Setup.

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) Yes, final hyperparameters were selected based on preliminary experiments.

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) NA

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) No, evaluation code will be publicly released upon publication.

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) Partial, code will be publicly released upon publication.

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) Yes.

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) Yes.

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) Yes.

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) Yes.

4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) Yes.

4.11. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) Yes, see Section 6 Results and Discussion and Table 1.

4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) Yes, see Section 6 Results and Discussion with Spearman and Kendall Tau correlation tests.

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) Yes.