Thinking Fast and Laterally: Multi-Agentic Approach for Reasoning about Uncertain Emerging Events

Stefan Dernbach
stefan.dernbach@pnnl.gov

Alejandro Michel
alejandro.michelzuniga@pnnl.gov

Khushbu Agarwal
khushbu.agarwal@pnnl.gov

Christopher Brissette
cbrissette@nvidia.com

Geetika Gupta
gegupta@nvidia.com

Sutanay Choudhury
sutanay.choudhury@pnnl.gov

Abstract

This paper introduces lateral thinking to implement System-2 reasoning capabilities in AI systems, focusing on anticipatory and causal reasoning under uncertainty. We present a framework for systematic generation and modeling of lateral thinking queries and evaluation datasets. We introduce Streaming agentic Lateral Thinking (SALT), a multi-agent framework designed to process complex, low-specificity queries in streaming data environments. SALT implements lateral thinking-inspired System-2 reasoning through a dynamic communication structure between specialized agents. Our key insight is that lateral information flow across long-distance agent interactions, combined with fine-grained belief management, yields richer information contexts and enhanced reasoning. Preliminary quantitative and qualitative evaluations indicate SALT's potential to outperform single-agent systems in handling complex lateral reasoning tasks in a streaming environment.

1 Introduction

The rapid advancement of artificial intelligence (AI) has revolutionized our ability to process and understand vast amounts of text and multi-modal content. State-of-the-art AI models now demonstrate remarkable capabilities in answering complex hypothetical questions based on user-provided context, exhibiting traits reminiscent of "System 1" thinking as described by Kahneman (2011).

A key component of human intelligence is the ability to reason under uncertainty and anticipate future events. Many critical events in personal, financial, and security domains do not occur instantaneously but are preceded by identifiable precursors. For instance, a user might be interested in monitoring risks to a specific industry supply chain arising from emerging geopolitical events (Fig. 1). Recognizing early signals and tracking their development over time enables proactive planning and appropriate responses as events unfold.

These cognitive abilities require sophisticated reasoning about: 1) possible event chains under uncertainty, 2) dynamic evaluation of event chain probabilities as new information emerges, and 3) intelligent aging and retrieval of information from long-term to short-term memory. Such traits are hallmarks of "System 2" thinking, essential for advanced cognitive processes that distinguish human-level intelligence (Kahneman (2011)).

Lateral thinking, introduced by Edward de Bono in the late 1960s (De Bono (1970)), involves solving problems through indirect, creative reasoning that may not be immediately obvious. This approach is

The 1st Workshop on System-2 Reasoning at Scale (NeurIPS 2024), Vancouver, Canada.

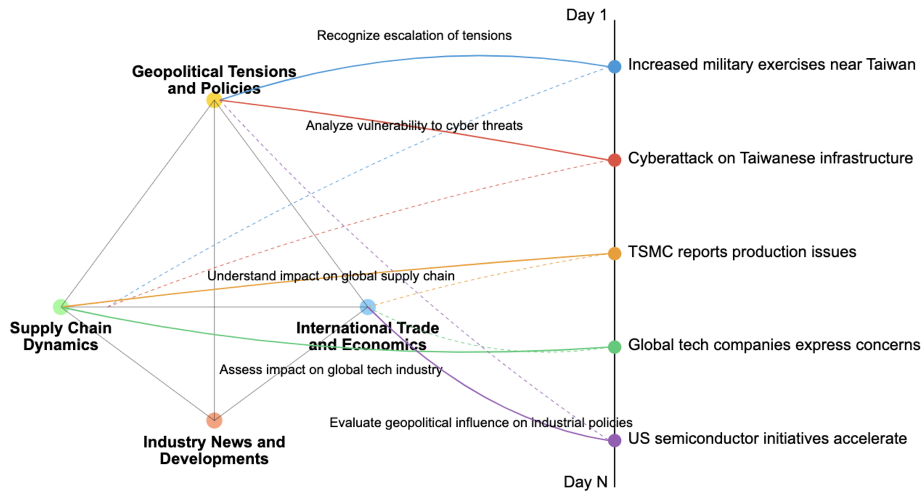


Figure 1: Conceptual overview of the SALT framework. This diagram illustrates the multi-agent structure and dynamic communication flow in processing lateral reasoning queries, such as "monitor the risk to the supply chain of American semiconductor companies". The right side shows a timeline of events that gradually unfold in a datastream over time. The left side depicts the specialized agent network, representing different topics in the problem space. This network structure is dynamically generated to address the specific lateral reasoning query. Connecting lines between agents and events indicate the reasoning criteria employed by each agent to interpret the temporal context.

particularly valuable in complex, dynamic environments where conventional methods fall short. We posit that a multi-agent architecture Wu et al. (2023) presents a promising solution for continuously processing a large volume of anticipatory questions. By leveraging specialized agents for distinct domains, this system enables parallel processing of domain-specific data and facilitates cross-domain connections. This distributed approach mirrors the non-linear pathways of lateral thinking, allowing for the discovery of novel patterns that traditional, sequential reasoning methods would miss.

In this work, we introduce a systematic framework for generating and modeling lateral thinking queries and associated evaluation datasets. This framework addresses the challenge of low-specificity queries in real-world applications, where individuals can express interest in specific outcomes but cannot describe every possible combination of event sequences leading to those outcomes. Building on this framework, we present Streaming agentic Lateral Thinking (SALT), a multi-agent system designed to simultaneously process *multiple* complex, low-specificity queries in streaming data environments. SALT implements lateral thinking-inspired System-2 reasoning through a dynamic communication structure between specialized agents.

SALT operates by monitoring evolving information around assigned topics, with agents exchanging probabilistic observations (subsequently referred to as “belief statements") to form and temporally maintain hypotheses about query-focused outcomes. Unlike static multi-agent systems, SALT's agent network's communication structure continuously adapts based on the relevance of agent expertise and the newly formed beliefs. This dynamic topology enables the system to find unconventional pathways for information flow, addressing scenarios where information is continuously changing and the immediate context rarely contains the full answer to a user query.

In summary, we make the following contributions:

1. A systematic framework for generating and modeling lateral thinking queries and evaluation datasets, providing a foundation for research in anticipatory reasoning under uncertainty.

2. Streaming agentic Lateral Thinking (SALT): A novel multi-agent architecture (Fig. 1) that implements lateral thinking-inspired System-2 reasoning for simultaneous processing of multiple anticipatory queries in streaming data environments.

2 Related Work

From an algorithmic perspective, a complex event can be decomposed into sub-events. See Fig. 4 and 3 for more illustrative examples. The process of robust prediction of such complex events can generally be reduced to three steps: 1) accurate identification of its sub-events when or before they occur, 2) the ability to connect these sub-events as they emerge over time through a single or multiple data sources, and 3) maintenance of the combined or partially materialized event information such that that memory complexity does not increase monotonically and lead to system failure. Each of these steps has been extensively studied across various fields in computer science, including databases, natural language processing, and symbolic or probabilistic reasoning methods. Drawing strong inspiration from this prior literature, SALT introduces large-language model Achiam et al. (2023); Ram et al. (2023)-based implementation of each of these steps.

Streaming query/reasoning Complex event processing has been studied extensively over the past three decades, initially motivated by the advent of online financial markets and subsequently by the growth of cyber data streams and social media. A common approach among these prior solutions Chandrasekaran et al. (2003); Choudhury et al. (2015); Della Valle et al. (2009): 1) A query decomposition approach where the complex query processing task is decomposed into a data-flow graph, with nodes representing operators that process a stream through SQL, graph, or different models, and 2) an event graph that guides how partial results are joined to produce a final answer. Maintenance of materialized partial query results and their aging over time is a key performance issue for such systems Barbieri et al. (2010); Ren and Pan (2011). As discussed in the next section, SALT derives these decomposition graphs from user-specified queries, initiates an agent network structure Wu et al. (2023), and pursues a dynamic inter-agent communication strategy for dataflow.

Induction and exploitation of Knowledge Schema Scaling up such methods on unstructured, noisy data streams extracted from text and other modalities introduces new challenges, including query event interpretation from text Lee et al. (2015), induction of unseen events Huang and Ji (2020), and evolution of underlying knowledge schema Yu et al. (2021). In the current work, we do not restrict SALT to any specific event or knowledge (graph) schema. This choice is primarily driven by the goal to support multiple user queries drawn from a wide range of diverse but related domains. Recent work Ghafarollahi and Buehler (2024) illustrates the benefits of grounding the beliefs generated in an LLM-Agent network via a knowledge graph.

Probabilistic approach Probabilistic approaches to event detection have been studied in the past for structured data streams, with “possible worlds" representing a probabilistic assignment of various events occurring in the data stream Cormode and Garofalakis (2007). Formulation of query processing operators in a Hidden Markov Model fashion and development of probabilistic stream algebra were proposed by Ré et al. (2008). In the broader context beyond event detection, belief propagation methods have been the cornerstone of probabilistic graphical models Murphy et al. (2013). More recently, loopy belief propagation methods have been combined with graph neural networks to propose neural message propagation with embeddings Satorras and Welling (2021), inspiring the implementation of the message propagation and belief synthesis steps in our proposed algorithm.

3 Methods

Streaming agentic Lateral Thinking (SALT) is designed to process streaming information based on a set of user-registered questions, continuously generating hypotheses about query matches by synthesizing the evolving reasoning outputs produced by multiple specialized agents. The key components and mechanisms of SALT are as follows:

3.1 System Architecture

The components of the SALT architecture are defined as follows.

- **Agents**: Each agent A_i in the network is associated with a set of topics $T_i = \{t_1, t_2, ..., t_n\}$. See Fig. 1 and the appendix for examples. In a multi-query setting, we compute an union of all query specific topics and use the resultant set to initilize the network.

- **Network Topology**: The agents are connected in a graph structure $G(V, E)$, where vertices V represent agents and edges E represent information flow pathways between agents. This graph dynamically evolves based on the context of each batch in the stream.
- **Belief Statements**: Agents generate and maintain a set of belief statements $B_i = \{b_1, b_2, ..., b_m\}$. Each belief statement b_j is characterized by a tuple (s, c, t, r), where s is the statement text, c is a confidence score, t is a timestamp, and r is a list of relevant reference ids. We provide an example of a belief statement in the next page.

PROBLEM DEFINITION Given the aforementioned setup, SALT aims to optimize the function $f(S, Q, A, G) \rightarrow H$, where $S = \{s_1, s_2, ..., s_t, ...\}$ is an incoming data stream, $Q = \{q_1, q_2, ..., q_m\}$ is a set of user queries, $A = \{A_1, A_2, ..., A_n\}$ is a set of specialized agents, and $G(V, E)$ is a dynamic graph representing inter-agent connections.

For each batch s_t, SALT updates G, generates and refines belief statements $B_i(t)$ for each agent A_i, and synthesizes these to produce a set of hypotheses $H(t) = \{h_1, h_2, ..., h_k\}$. The objective is to maximize $\sum_i$ relevance(h_i, Q) and $\sum_i$ accuracy(h_i) while minimizing computational complexity $C(f)$ and ensuring $|B_i(t)| \leq M$ for some memory bound M, $\forall i, t$. This formulation encapsulates

Question: What actions are American semiconductor companies likely to take in response to these vulnerabilities and risks?
Context: [Query context can be initialized from a batch of articles or prior belief statements.]
Response: American semiconductor companies are likely to take several actions, including diversifying their supply chains to reduce dependency on any single region, investing in increased cybersecurity measures to protect against cyberattacks, and advocating for diplomatic solutions to reduce geopolitical tensions. They might also increase collaboration with international partners to bolster their supply chain resilience and seek governmental support for policies that encourage domestic manufacturing capabilities.
Confidence: 0.9
Timestamp: 127
References: [article-103, article-104]

We now proceed to describe three key steps in SALT: initialization of the agent network, initiation of new belief statements and updating pre-computed beliefs, and sharing across the network topology to synthesize new belief statements. The latter two steps are executed iteratively for each batch of incoming streaming data.

Agent Initialization SALT is first initialized by translating the user-specified question set into a set of topics. In Algorithm 1, the `ExtractTopics` function uses a pre-trained topic classification model to identify relevant topics from the input questions. The `CreateAgent` function handles the construction of simple RAG agents, and the `BuildGraph` function creates initial connections between agents based on topic similarity. An example output from `ExtractTopics` is provided in section A.1.

Algorithm 1: InitializeNetwork

Input: Set of questions Q
Output: Set of agents A, Graph G
$T \leftarrow$ ExtractTopics(Q);
$A \leftarrow \{\}$;
foreach *unique topic* t_i *in* T **do**
  $A_i \leftarrow$ CreateAgent(t_i);
  $A \leftarrow A \cup \{A_i\}$;
end
$G \leftarrow$ BuildGraph(A);
return (A, G);

Stream Processing SALT processes streaming information in multiple steps. First, the input query (that can be an article, image or a batch of them) assigns a relevance score relative to each expert agent, and are processed only by agents with sufficiently high relevance. We use a cosine similarity between the query and agent representation. Each agent constructs new beliefs based on assigned articles Algorithm 2.

Belief Propagation and Synthesis Belief propagation and synthesis are key to SALT's emergent lateral thinking capabilities. Algorithm 3 shows how each agent within SALT takes in the most relevant beliefs from its neighbors, and alters its own beliefs based on this information.

Algorithm 2: ProcessInformation

Input: Information stream I, Set of agents A, Graph G
foreach *information item i in I* **do**
  $R \leftarrow$ AssignRelevance(i, A);
  foreach *agent A_j in A where $R_j >$ threshold* **do**
    $B_{new} \leftarrow A_j$.ProcessItem(i);
    $A_j.B \leftarrow A_j.B \cup B_{new}$;
    PropagateBeliefs(A_j, B_{new}, G);
  end
  $G \leftarrow$ UpdateTopology(G, R);
end

Algorithm 3: PropagateBeliefs

Input: Source agent A_s, New beliefs B_{new}, Graph G
foreach *agent A_r connected to A_s in G* **do**
  $R \leftarrow$ ComputeRelevance(A_s, A_r, B_{new});
  if $R >$ *threshold* **then**
    $B_{shared} \leftarrow A_s$.ShareBeliefs$(A_r, B_{new})$;
    $B_{synthesized} \leftarrow A_r$.SynthesizeBeliefs$(B_{shared})$;
    $A_r.B \leftarrow A_r.B \cup B_{synthesized}$;
  end
end

3.2 Dynamic Network Topology

While the multi-agent system described in prior subsections has a defined graph structure, all connected agents do not necessarily communicate with each other for every query. Consider a system which has two agents, one that specializes in the “weather events", and another specializes in “agricultural commodities market". Assume these two agents are connected via the network, and the purpose of such a multi-agent system is to predict how extreme weather events may affect the farmimg industry in the Pacific Northwest. Given a set of articles that pertains only to weather, it does not necessarily make sense to share those belief statements with our “commodities" agent.

Due to instances like the above, SALT’s network topology evolves based on the relevance and of agent connections as seen in Equation 1 which follows.

$$w_{ij} = \alpha \cdot \text{sim}(T_i, T_j) \tag{1}$$

Here w_{ij} is the connection weight between agents A_i and A_j, which handle topics T_i and T_j respectively. $\text{sim}(T_i, T_j)$ is the topic similarity and α is a coefficient which controls the relative importance of this particular connection. This construction is important for avoiding simple diffusion-like dynamics. If there were only a communication network structure without any further selective behavior, the network of agents will eventually converge on a nearly static belief set.

4 Dataset

To rigorously evaluate SALT’s performance, we developed a comprehensive dataset of lateral thinking questions and scenarios, accompanied by novel complexity metrics. This section details our data generation process and benchmarking methodology.

Dataset Creation We generated a synthetic dataset comprising 30 lateral thinking queries using a combination of state-of-the-art language models (Claude-3.5 and GPT-4o) and human expert curation. Each query was crafted to investigate trends with potential cross-industry impacts such as “Monitor extreme weather patterns for impacts on agricultural commodity prices". We also generated multiple scenarios for each query, designed to emulate real-world situations typically reported in mainstream news media. For example, the “Monitor extreme weather patterns for impacts" query can be mapped to three scenarios: a) “Severe droughts in Brazil lead to surge in coffee futures prices", b) “Unexpected frost in Florida causes orange juice futures to spike" and c) "Extended heatwave in Europe results in wheat shortage and price increase".

Article Generation For each scenario, we generated between 20 and 100 articles, each presenting unique developments related to the specific scenario. To ensure diversity and realism in the corpus, we parameterized the article generation process along several dimensions: a) perspective (e.g., conservative, liberal), b) length (ranging from 200 to 1000 words), c) temporal distribution (e.g., uniformly distributed over a 6-month period vs. clustered within a 2-week window).

4.1 Lateral Reasoning Complexity Metrics

To quantitatively model the workload complexity and ensure a balanced distribution, we propose four novel metrics for each query-scenario pair. We'll illustrate each metric using our Taiwan Strait tensions scenario from Fig. 1:

1. **Lateral Measure ($\mathcal{L}$):** Quantifies the number of intermediate causal steps between the trigger event and the outcome. Range: 1-7. *Example:* $\mathcal{L} = 3$ for our Taiwan scenario: increased military presence $\rightarrow$ disruption of global semiconductor supply chains $\rightarrow$ stock market volatility.
2. **Time Lag Complexity ($\mathcal{T}$):** Measures the temporal distance between the trigger event and its final impact, in days. Range: 0-21 (for this rapid-evolving scenario). Example: $\mathcal{T} = 14$ for our Taiwan scenario, where significant economic impacts manifest two weeks after the initial spike in tensions.
3. **Outcome Uncertainty Coefficient ($\mathcal{U}$):** Quantifies the unpredictability of the relationship between the trigger and its effects on a normalized scale. Range: 0-1. *Example:* $\mathcal{U} = 0.9$ for our Taiwan scenario, due to the highly volatile nature of geopolitical crises and their potential for rapid escalation or de-escalation.

Diversity Assurance: To ensure a wide range of lateral thinking challenges, we categorized queries into 10 broad domains (e.g., geopolitics, technology, climate change) and ensured balanced representation across these categories (Fig. 5). We also varied the complexity levels (based on our proposed metrics) across the dataset. Fig. 2(a) illustrates the diversity of the queries via complexity metrics discussed above. For illustration of a single query and it's metrics see Fig. 6 in appendix.

5 Experiments

In this preliminary study, we focus on two primary research questions:

1. **RQ1:** How does multi-agentic lateral reasoning compare with single agent approaches?
2. **RQ2:** How does the system performance change as a function of query complexity?

5.1 Experimental Setup

We explore four model variants in our experiments as listed below. As the name suggests, SingleTemporalReasoningAgent is an agent that processes the batched data stream and executes as prompt on a rolling window of historical articles. It runs each of the user-specified queries separately with no information exchange between multiple query execution. See section B for implementation details.

- **Temporal-Claude:** SingleTemporalReasoningAgent with Claude-3.5-Sonnet
- **Temporal-GPT4o:** SingleTemporalReasoningAgent with GPT-4o
- **SALT-Claude:** SALT with Claude-3.5-Sonnet
- **SALT-GPT4o:** SALT with GPT-4o

For brevity, we refer to SingleTemporalReasoningAgent with Claude (or GPT-4o) as Temporal-Claude (or Temporal-GPT4o) in subsequent discussions. We use our custom dataset of lateral thinking queries, as described in Section 3. For this preliminary study, we selected a subset of 30 queries from the full dataset to evaluate our models. This relatively small set was chosen to ensure we can manually verify the quality of generated articles and reasoning output from each target model.

5.2 Evaluation Metrics

We propose two metrics to capture key aspects of lateral thinking:

Retrieval Performance RP measures the system's ability to identify relevant information from diverse sources. A high RP indicates effective navigation through a large corpus, pinpointing pertinent articles even when connections are not immediately obvious—a crucial requirement for lateral thinking.

$$\text{RP} = \frac{\text{Number of relevant article IDs cited}}{\text{Total number of relevant articles}} \times 100\% \tag{2}$$

Hypothesis Quality HQ assesses the system's capacity to synthesize information into meaningful hypotheses. It evaluates the ability to connect disparate pieces of information, forming information rich and specific conclusions.

$$\text{HQ} = \frac{\text{Number of correctly identified sub-events}}{\text{Total number of sub-events}} \times 100\% \tag{3}$$

5.3 Results

Lateral context augmentation drives outperformance Results comparing SALT and SingleTemporalAgents are provided in Table 1. We observe that our agent network-based system achieves significantly better performance across both hypothesis generation and citation retrieval compared to the single agent based systems. SALT-Claude-Sonnet-3.5 outperforms its corresponding baseline implementation by 39.77% and 29% in citation retrieval and hypothesis quality. Salt-GPT-4o outperforms Temporal-GPT-4o by 60% and 87.63%. Overall, Claude-Sonnet-3.5 based agents perform better than GPT-4o in retaining facts and generating overall hypotheses.

Key insight While our study is preliminary and larger-scale experiments are needed to conclusively establish the superiority of lateral thinking, a consistent pattern emerges: SALT's reasoning is driven by a significantly larger and diverse context size as measured by the number of belief statements and supporting article count. This suggests that the key to it's success lies in its ability to perform lateral thinking over related contexts, sometimes multiple hops away in the agent network. The system's capacity to ensure that relevant information from distant articles is communicated to the appropriate agents appears to be crucial. However, further research is necessary to fully understand the information flow within the agent network and to validate this hypothesis. This insight, if confirmed, could represent a significant advancement in an AI system's ability to reason across diverse and seemingly unrelated information sources arriving at different points of time.

Qualitative Analysis Answers generated by SALT are more a) information-rich and b) characterized by higher specificity. Below we present a brief case study analyzing results to the query: "Monitor if any new economic activity develops as a result of changing climate patterns".

1. Comprehensiveness of assessment: SALT-GPT-4o and SALT-Claude provide more comprehensive assessments covering a broader range of economic activities. For example:
   - SALT-GPT-4o mentions "innovations in maritime technologies, including advanced ice-breaking capabilities and improved navigation systems" and "growing economic activity in climate-adaptive technologies across various sectors."
   - SALT-Claude discusses "innovation in sustainable technologies, including renewable energy, water management systems, and climate-resilient agriculture."

   In contrast, Temporal-GPT-4o and Temporal-Claude focus primarily on Arctic shipping routes and resource exploration.
2. Depth of analysis on long-term implications: SALT versions offer deeper insights into long-term economic consequences:
   - SALT-GPT-4o states: "These developments are reshaping global trade patterns and potentially shifting economic power towards nations with strong Arctic presences or advanced maritime technologies."

- SALT-Claude notes: "Overall, while climate change is creating new economic opportunities, it is also fundamentally altering global power dynamics, security concerns, and governance structures, which will have long-term implications for international trade and economic partnerships."

The Temporal-GPT-4o and Temporal-Claude outputs do not address these broader implications.

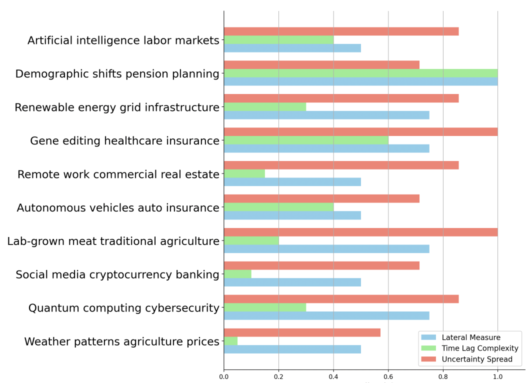
3. Balanced perspective on opportunities and challenges: SALT outputs provide a more nuanced view of both opportunities and risks:

   - SALT-GPT-4o mentions: "The exploitation of Arctic resources raises environmental concerns and potential conflicts over territorial claims. Increased maritime activity in the Arctic also brings risks of oil spills, noise pollution affecting marine life, and the introduction of invasive species."
   - SALT-Claude states: "These new economic activities come with significant challenges and potential conflicts. The competition for resources in the Arctic is intensifying, leading to increased militarization and the need for new governance structures to manage conflicting claims."

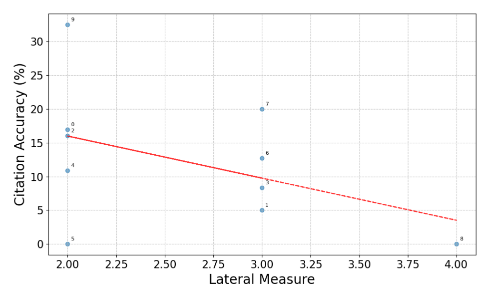
   In contrast, Temporal-GPT-4o and Temporal-Claude primarily focus on the opportunities without thoroughly addressing associated challenges.

Model	Retrieval Performance (%)	Hypothesis Quality (%)
Temporal-Claude-Sonnet-3.5	45.46	26.88
Temporal-GPT 4o	35.83	18.28
SALT-Claude-Sonnet-3.5	**63.54**	**34.68**
SALT-GPT 4o	57.33	34.30

Table 1: Performance Comparison: Reasoning Algorithms and LLM Combinations.



(a) Query Complexity Distribution



(b) Reasoning accuracy drops for queries with increased lateral measure

Figure 2: (a) Shows normalized lateral measure, time lag complexity, and uncertainty spread across various anticipatory queries. (b) Shows an evaluate of SALT over a dataset with high time-lag complexity and profiles performance as a function of query complexity.

6 Conclusions

This paper advances AI reasoning by introducing lateral thinking for agents to address low-specificity anticipatory and causal questions in streaming setting. We present a framework for generating lateral thinking queries and datasets, and introduce SALT, a multi-agent system implementing System-2 reasoning. Preliminary studies reveal SALT's superior performance over single-agent systems, suggesting a novel phenomenon: lateral information flow across long-distance agent interactions, combined with fine-grained belief management, yields richer information contexts and enhanced reasoning. Rigorous testing of this hypothesis on large-scale datasets will be subject of future work.

7 Acknowledgments

This work was supported by the Center for AI and Center for Cloud Computing at Pacific Northwest National Laboratory, a multi-program national laboratory operated by Battelle for the U.S. Department of Energy.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. Incremental reasoning on streams and rich background knowledge. In *The Semantic Web: Research and Applications: 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30–June 3, 2010, Proceedings, Part I 7*, pages 1–15. Springer, 2010.

Sirish Chandrasekaran, Owen Cooper, Amol Deshpande, Michael J Franklin, Joseph M Hellerstein, Wei Hong, Sailesh Krishnamurthy, Samuel R Madden, Fred Reiss, and Mehul A Shah. Telegraphcq: continuous dataflow processing. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 668–668, 2003.

Sutanay Choudhury, Lawrence Holder, George Chin, Khushbu Agarwal, and John Feo. A selectivity based approach to continuous pattern detection in streaming graphs. *arXiv preprint arXiv:1503.00849*, 2015.

Graham Cormode and Minos Garofalakis. Sketching probabilistic data streams. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 281–292, 2007.

E De Bono. Lateral thinking. *Blanford/England: Direct Educational Services*, 1970.

Emanuele Della Valle, Stefano Ceri, Frank Van Harmelen, and Dieter Fensel. It's a streaming world! reasoning upon rapidly changing information. *IEEE Intelligent Systems*, 24(6):83–89, 2009.

Alireza Ghafarollahi and Markus J Buehler. Sciagents: Automating scientific discovery through multi-agent intelligent graph reasoning. *arXiv preprint arXiv:2409.05556*, 2024.

Lifu Huang and Heng Ji. Semi-supervised new event type induction and event detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 718–724, 2020.

Daniel Kahneman. Thinking, fast and slow. *Farrar, Straus and Giroux*, 2011.

Kenton Lee, Yoav Artzi, Yejin Choi, and Luke Zettlemoyer. Event detection and factuality assessment with non-expert supervision. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1648, 2015.

Kevin Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. *arXiv preprint arXiv:1301.6725*, 2013.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023.

Christopher Ré, Julie Letchner, Magdalena Balazinksa, and Dan Suciu. Event queries on correlated probabilistic streams. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 715–728, 2008.

Yuan Ren and Jeff Z Pan. Optimising ontology stream reasoning with truth maintenance system. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 831–836, 2011.

Victor Garcia Satorras and Max Welling. Neural enhanced belief propagation on factor graphs. In *International Conference on Artificial Intelligence and Statistics*, pages 685–693. PMLR, 2021.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023.

Pengfei Yu, Heng Ji, and Prem Natarajan. Lifelong event detection with knowledge transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5278–5290, 2021.

A Details of SALT Architecture

A.1 Agent network initialization

User: I want to know as soon as a geopolitical event between China and Taiwan that will affect the supply chain for American semiconductor companies happens.

Prompt: Given the following query construct a numbered list of 4 areas of research that should be focussed on to answer the query.

Master Agent: To address the query effectively, the following areas of research should be focused on:

1. **Geopolitical Tensions and Policies**:

    - Monitoring diplomatic relations and potential conflicts between China and Taiwan.
    - Analysis of government policies, sanctions, and trade agreements affecting both nations.
    - Updates on military activities or significant political statements that could indicate rising tensions.

2. **Supply Chain Dynamics**:

    - Identification of critical semiconductor components and their origins in China and Taiwan.
    - Tracking of existing and potential supply chain disruptions within the semiconductor industry.
    - Assessing dependency of American semiconductor companies on Chinese and Taiwanese suppliers.

3. **International Trade and Economics**:

    - Analysis of trade flows and economic interdependencies between China, Taiwan, and the United States.
    - Monitoring changes in import/export regulations that could affect semiconductor industries.
    - Studying economic indicators that might predict or result from geopolitical disruptions.

4. **Industry News and Developments**:

    - Real-time tracking of news within the semiconductor industry, focusing on announcements from key companies.
    - Updates on corporate strategies and responses to geopolitical events by American semiconductor companies.
    - Following industry analyses and expert opinions on potential impacts of geopolitical changes.

By focusing research on these areas, one can not only anticipate but also respond effectively to any geopolitical event between China and Taiwan that might affect the supply chain for American semiconductor companies.

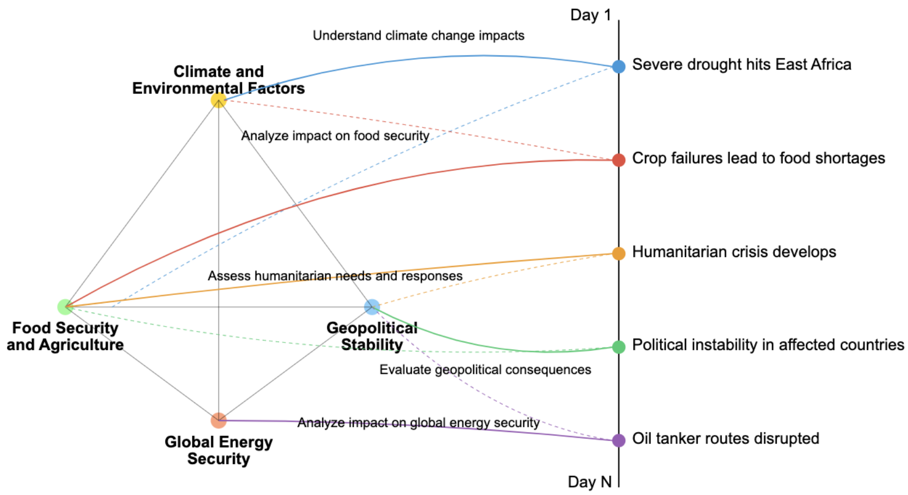


Figure 3: Illustration of a scenario where a drought in one part of the world can affect the energy security in distant places.

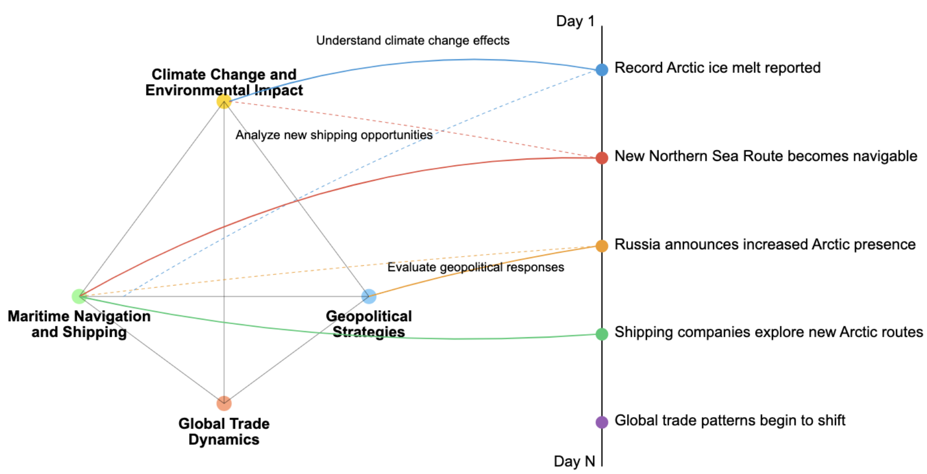


Figure 4: Illustration of a scenario where a climate change induced event can lead to upheaval in an industry.

B Benchmarking

B.1 Implementation

The SingleTemporalReasoningAgent processes streaming information and generates evolving hypotheses over time. Its core functionality involves maintaining a temporal context of past assessments while continuously integrating new information. The agent processes articles in chronological batches, capturing the progression of events and insights. For each user query, it generates hypotheses by considering both the current batch of articles and, optionally, the previous hypothesis, implementing a form of persistence in its reasoning process. This method enables the agent to build upon past knowledge, refine its assessments, and provide updated insights as new information becomes available. The agent employs a planner (or master agent) for hypothesis generation, adding adaptability to the reasoning process. As detailed in Algorithm 1, the agent manages the flow of information by sorting and filtering articles, generating hypotheses, and presenting results, creating a systematic process for continuous reasoning over time-based data streams.

Algorithm 4: SingleTemporalReasoningAgent

Input: User queries Q, Planner P, Articles A, Parameters $params$
Output: Temporal context of hypotheses H
Initialize H as empty dictionary;
Sort A chronologically;
$A \leftarrow$ FilterArticlesByDateRange$(A, params.start_date, params.end_date)$;
while A *is not empty* **do**
    $batch \leftarrow$ GetNextBatch(A);
    foreach *query* q *in* Q **do**
        $prompt \leftarrow$ GeneratePrompt$(batch, q, H[q]$.last$)$;
        $hypothesis \leftarrow P$.GenerateHypothesis$(prompt)$;
        $H[q]$.append$(hypothesis)$;
        PresentHypothesis$(hypothesis)$;
    end
    WaitForNewData$()$;
end

The algorithm maintains a temporal context of hypotheses for each user query, updating them as new batches of articles are processed. This allows the agent to incorporate past information while generating new hypotheses, resulting in a continuous stream of updated assessments based on the incoming information.

B.2 Query Configurations

Short Version	Full-Length Version
Weather patterns agriculture prices	Monitor extreme weather patterns for impacts on agricultural commodity prices.
Quantum computing cybersecurity disruption	Track developments in quantum computing for potential disruptions in cybersecurity.
Social media cryptocurrency banking	Analyze social media sentiment around cryptocurrencies for potential impacts on traditional banking.
Lab-grown meat traditional agriculture	Monitor advancements in lab-grown meat technology for effects on traditional agriculture.
Autonomous vehicles auto insurance	Track developments in autonomous vehicle technology for impacts on auto insurance industry.
Remote work commercial real estate	Analyze trends in remote work adoption for potential impacts on commercial real estate.
Gene editing healthcare insurance	Monitor developments in gene editing technology for impacts on healthcare and insurance sectors.
Renewable energy grid infrastructure	Track advancements in renewable energy storage for effects on utilities and grid infrastructure.
Demographic shifts pension planning	Analyze global demographic shifts for long-term impacts on pension funds and retirement planning.
Artificial intelligence labor markets	Monitor developments in artificial intelligence for impacts on employment and labor markets.

Figure 5: Example lateral thinking queries.


```
{
  "query": "Monitor extreme weather patterns for impacts on agricultural commodity prices.",
  "scenarios": [
    "Severe droughts in Brazil lead to surge in coffee futures prices.",
    "Unexpected frost in Florida causes orange juice futures to spike.",
    "Extended heatwave in Europe results in wheat shortage and price increase."
  ],
  "lateral_measure": 2,
  "lateral_steps": [
    "Extreme weather event",
    "Crop yield reduction",
    "Commodity price increase"
  ],
  "num_topics": 3,
  "topics": [
    "Meteorology",
    "Agriculture",
    "Commodity markets"
  ],
  "timeLagComplexity": 3,
  "uncertaintySpread": 0.4
},
```


Figure 6: Example query-scenario pairs with corresponding metrics. Note that the metrics are obtained on the basis of the “query" field.