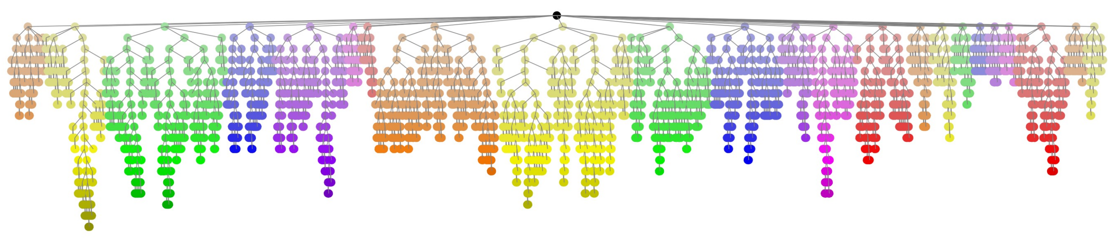
From *Isolated Conversations* to *Hierarchical Schemas*: Dynamic Tree Memory Representation for LLMs

Alireza Rezazadeh, Zichao Li, Wei Wei, Yujia Bao
{alireza.rezazadeh, zichao.li, wei.h.wei, yujia.bao}@accenture.com
Center for Advanced AI, Accenture

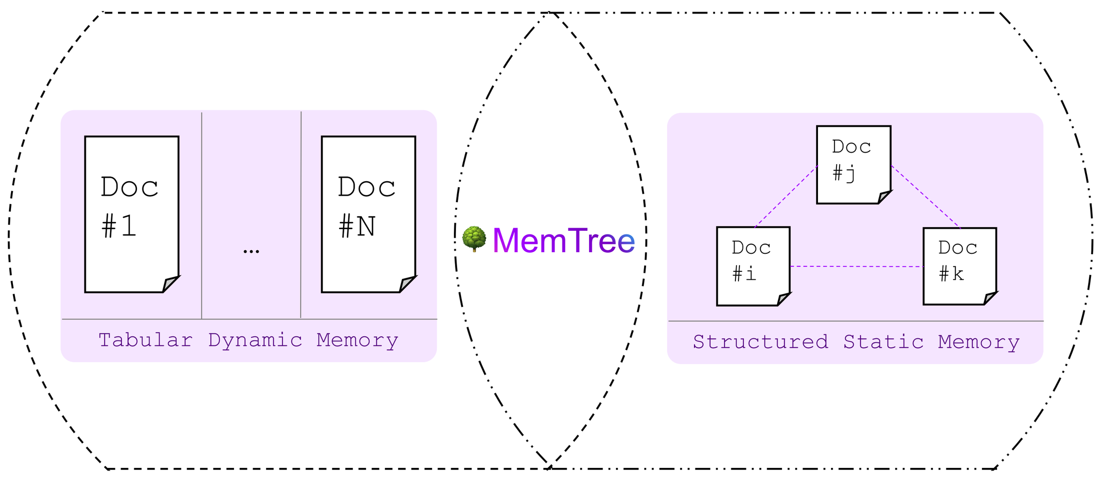
ArXiv Link

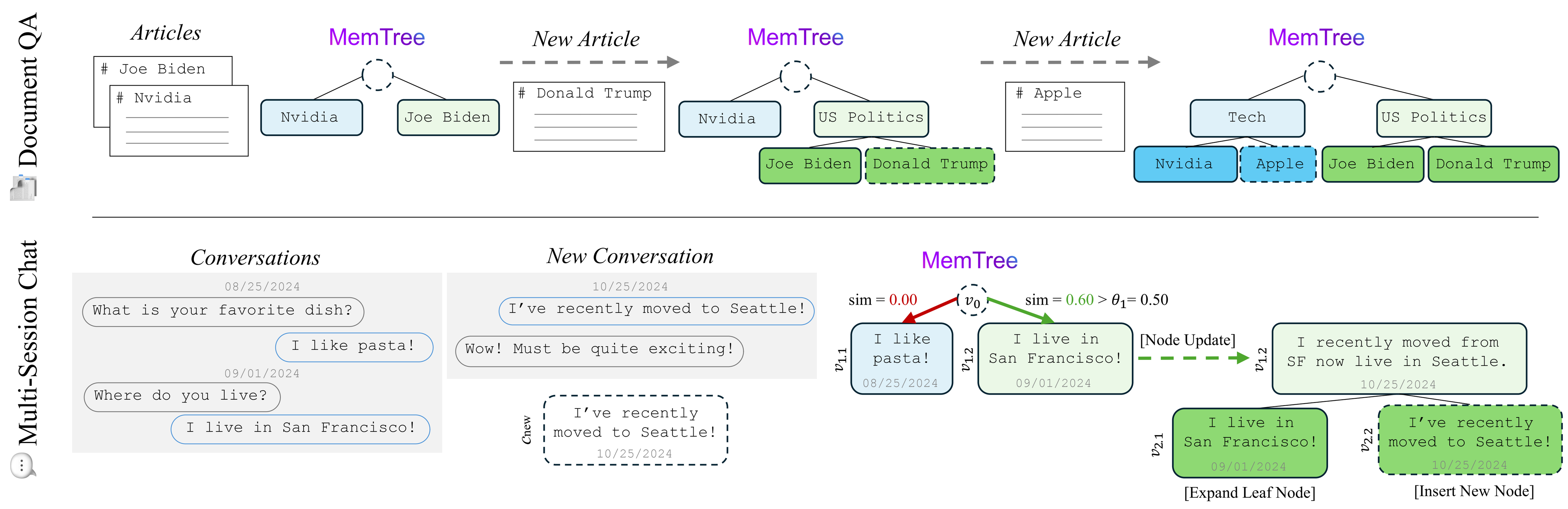
accenture



Introduction:

- Despite expanded context windows, LLMs struggle with reasoning over long-term memory.
- Existing Solutions:
  - Tabular Dynamic Memory: *MemoryStream*[1], *MemGPT*[2]
    Benefit: Enables adding information on the fly.
    Limitation: Independent rows don't scale with scattered evidence.
  - Structured Static Memory: *RAPTOR*[3], *GraphRAG*[4]
    Benefit: Provides structured knowledge with high-level representations.
    Limitation: Requires complete rebuilding to incorporate new information.





Experiments:

1. Conversational:
   - Multi-Session Chat (*MSC*):
     Context: ~14-round conversation session.
     Task: Respond to 1 follow-up question based on early conversations.
   - Multi-Session Chat - Extended (*MSC-E*)
     Context: ~200-round conversation session.
     Task: Respond to 100 follow-up questions evenly across all rounds.
2. Document QA:
   - Single-Doc QA (*QuALITY*)
   - Multi-Doc QA (*MultiHop RAG*)
     Context: 609 news articles.
     Task: Respond to 2,255 multi-hop queries by pooling evidence from multiple articles.

MemTree Algorithm

- **Tree Representation:** $T = (V, E)$
  - For each node $v \in V$:

$$v = [C_v, e_v, p_v, \mathcal{C}_v, d_v] \quad \begin{cases} \text{Textual content: } C_v \\ \text{Embedding: } e_v = f_{\text{emb}}(c_v) \in \mathbb{R}^d \\ \text{Parent node: } p_v \\ \text{Set of children: } \mathcal{C}_v \subseteq V \\ \text{Depth from root: } d_v \end{cases}$$

  - Root node (v_0) is a structural node:
    Does not hold any content: $c_{v0} = \emptyset, e_{v0} = \emptyset$

1. **Insert new info as a new node:** c_{new} , e_{new}
   - Tree Traversal: Traverse down the *most similar node* at each depth if similarity exceeds a *depth-adaptive threshold*:

$$\text{sim}(e_{\text{new}}, e_v) \geq \theta(d)$$

     - Stop Condition: If similarity is below the threshold, *insert as a child* at current depth.
     - Boundary: If traversal reaches a leaf node, *expand* it as a parent node.
   - Depth-Adaptive Threshold:

$$\theta(d) = \theta_0 e^{\lambda d}$$

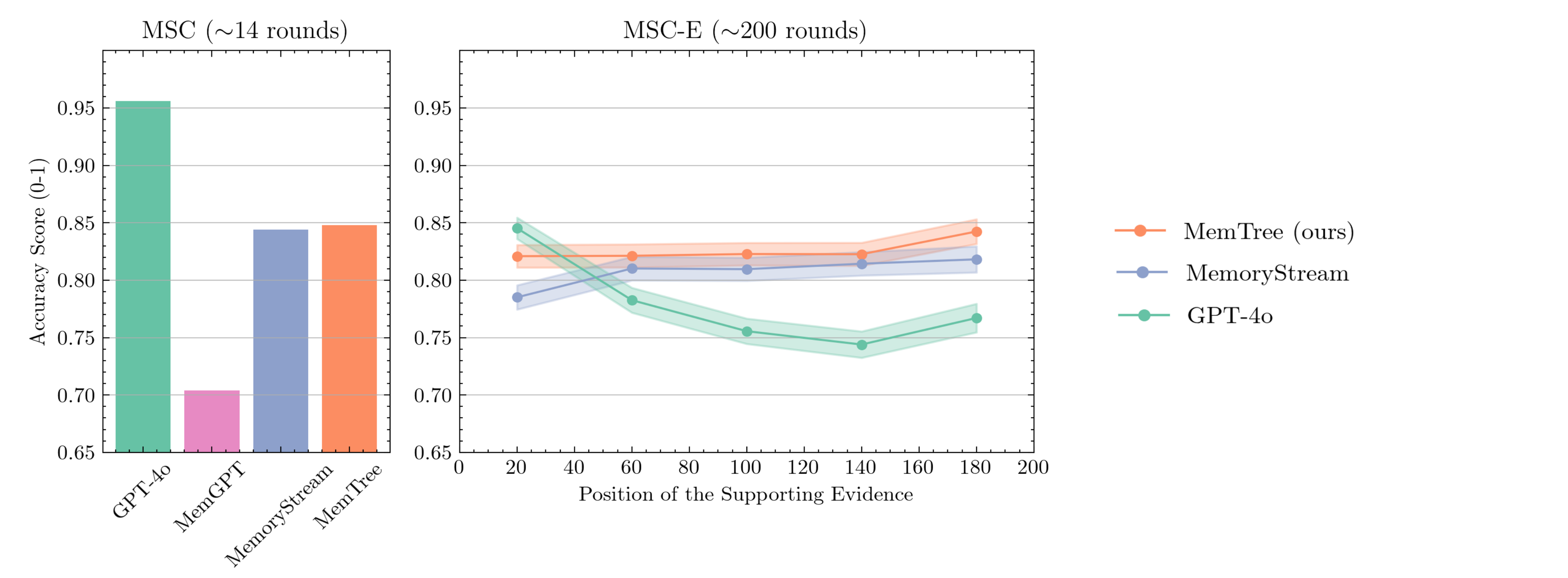
     - Threshold base (θ_0) and rate (λ)
     - Deeper nodes (more specific info) require higher similarity to integrate new info.
2. **Update parent nodes along traversal path after insertion:**
   - Conditional Content Aggregation:

$$c'_v \leftarrow \text{Aggregate}(c_v, c_{\text{new}} \mid n) \qquad e_v \leftarrow f_{\text{emb}}(c'_v)$$

   - Implemented as an LLM-based operation.

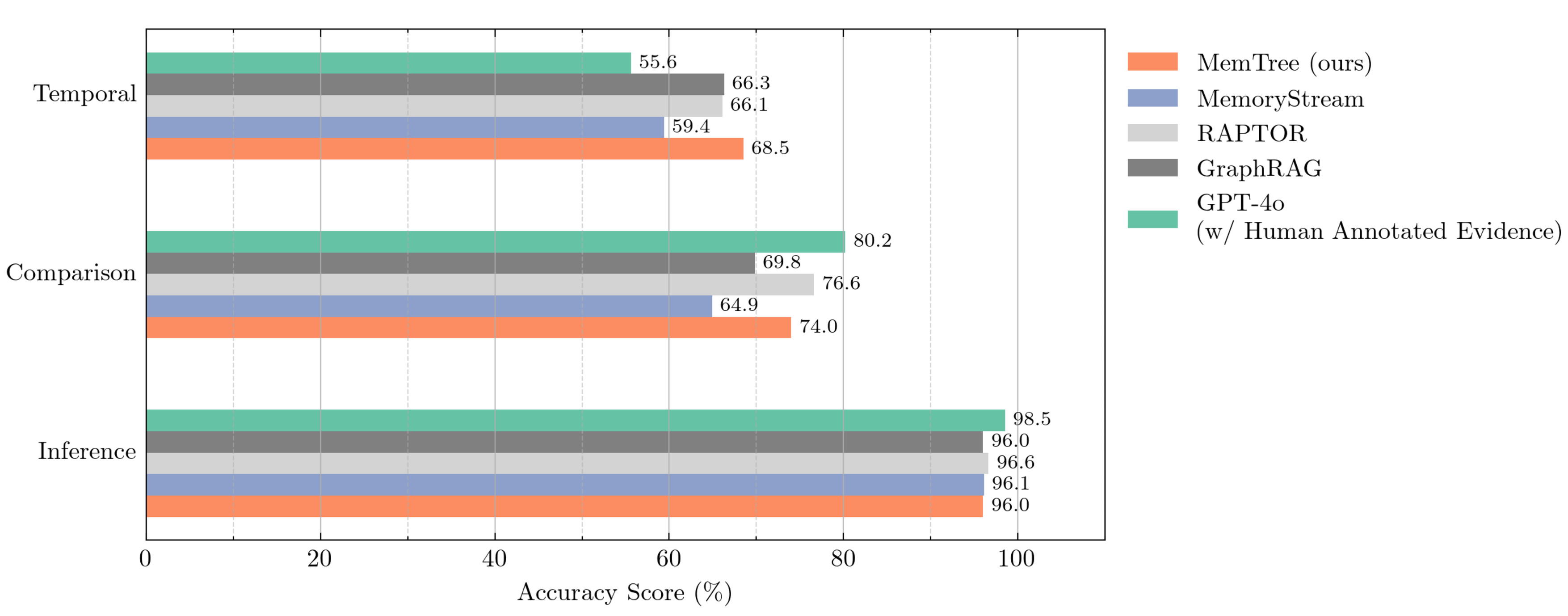
Results: Multi-Session Chat

- Long-context conversations (right figure) demand external memory, unlike short contexts (left figure)
  - Position bias[7] degrades LLM performance when utilizing full conversation history.

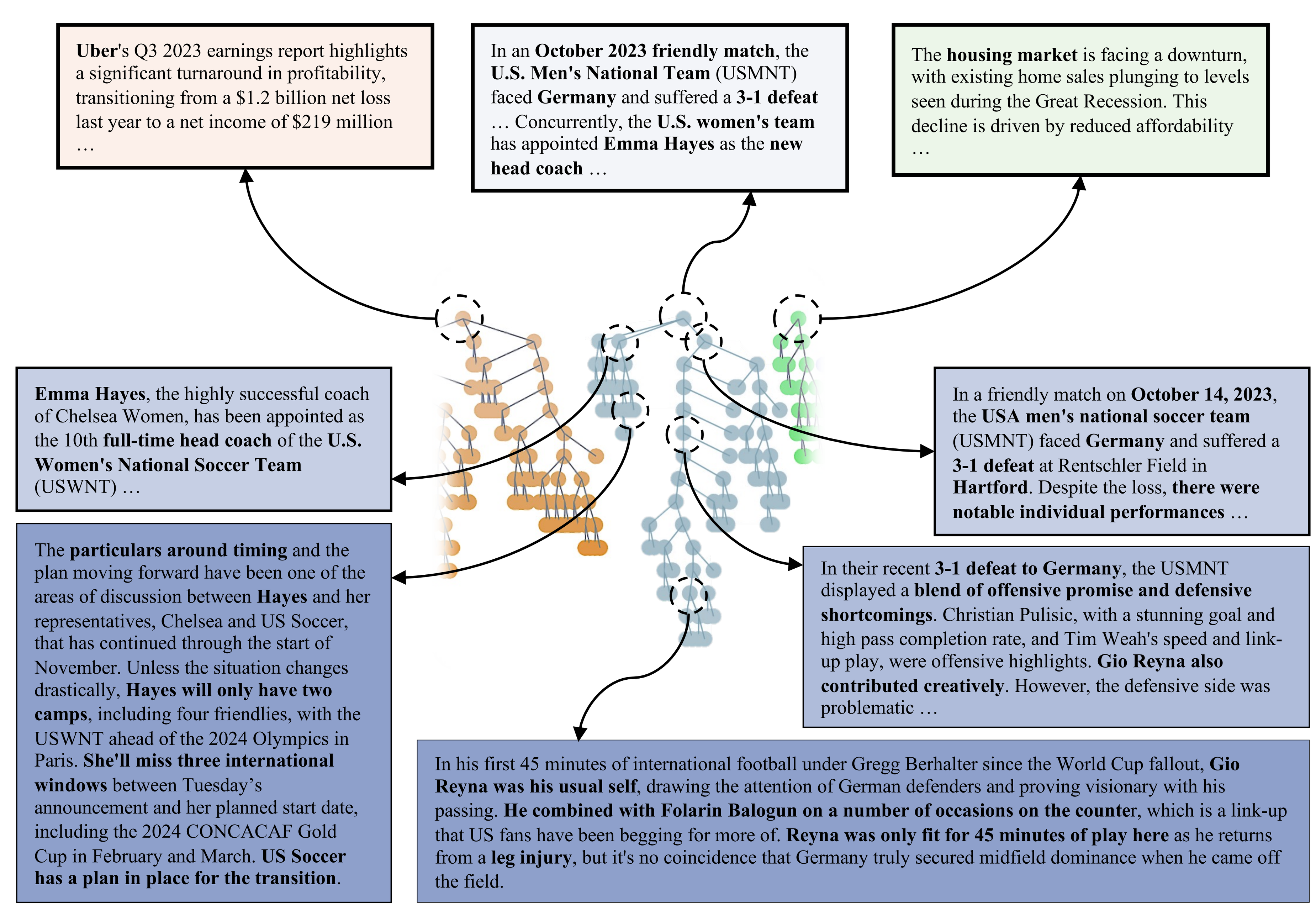


Results: Multi-Doc QA

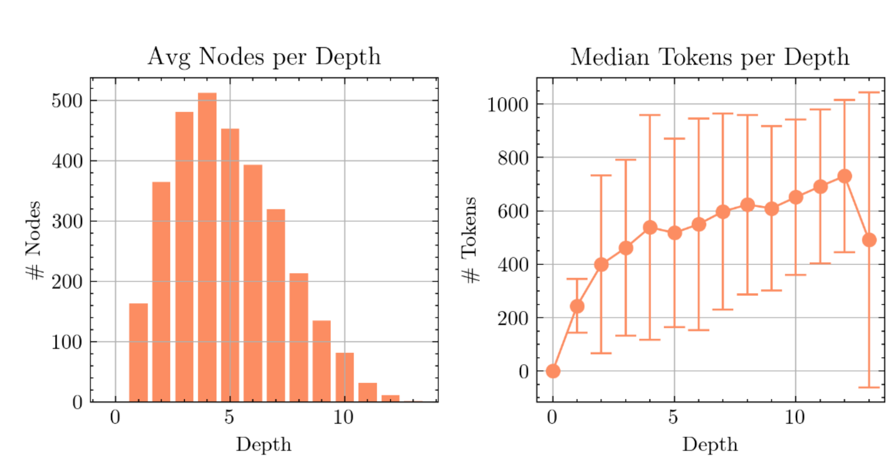
- MemTree outperforms dynamic methods, matches static methods, and surpasses all baselines, including human annotations, on temporal reasoning tasks.



- Visualization of the learned MemTree (subset):



- As the tree deepens, the information stored in the nodes becomes more detailed and increases in length.



MemTree Property	Value
#Nodes	3164
#Leaf Nodes	1706
#Branching Nodes	1458
Depth (max)	13
Depth (average)	4.9
Branching Factor	2.1
Height to Width Ratio	6.5

Theorem (Approximation Guarantee of MemTree)

MemTree aligns with Online Top-Down hierarchical clustering algorithm[5] (OTD) and inherits its theoretical properties:

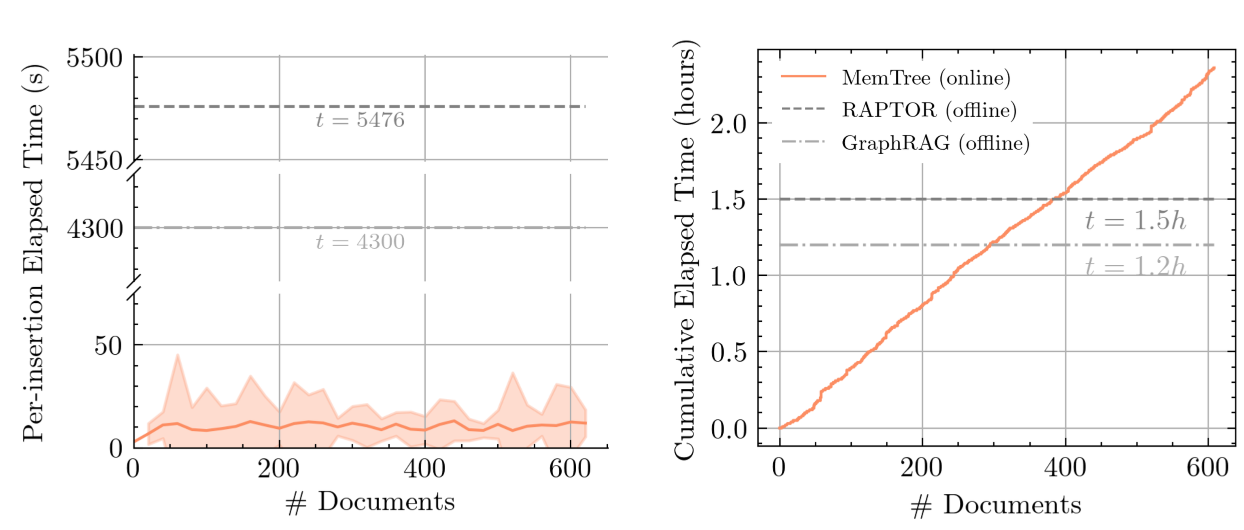
By employing a depth-adaptive threshold, MemTree processes data that is β-well-separated, and the hierarchy it maintains achieves a revenue satisfying

$$\text{Rev}(\text{MemTree}; W) \geq \frac{\beta}{3}\,\text{Rev}(T^*; W),$$

where T^* is the optimal hierarchy maximizing Moseley-Wang[6] revenue.

Efficient Memory Updates with MemTree

- MemTree's dynamic top-down insertion enables parallelized node updates, minimizing bottlenecks as memory grows. Despite a ~1.4x higher cumulative cost than static methods, it excels in real-time adaptability, ideal for dynamic use cases.



References

[1] Generative agents: Interactive simulacra of human behavior, Park et al (2023).
[2] MemGPT: Towards LLMs as operating systems, Packer et al (2023).
[3] RAPTOR: Recursive abstractive processing for tree-organized retrieval, Sarthi et al (2024).
[4] From local to global: A graph RAG approach to query-focused summarization, Edge et al (2024).
[5] Online hierarchical clustering approximations, Menon et al (2019).
[6] Approximation bounds for hierarchical clustering, Moseley et al (2017).
[7] Lost in the middle: How language models use long contexts, Liu et al 2024