Prior Knowledge Makes It Possible: From Sublinear Graph Algorithms to LLM Test-Time Methods

Avrim Blum [3] Daniel Hsu [1] Cyrus Rashtchian [2] Donya Saless [3]

[1]Columbia [2]Google Research [3]TTIC

Motivation: Reliable Test-time Augmentation

Test-time augmentation critically depends on an interplay between a model's parametric knowledge and externally retrieved information.

Grand Hope: Use a small number of retrieval augmentation queries to complete missing or incorrect prior knowledge and guide reasoning, leading to reliable LLM answers.

In practice, a model's prior knowledge of models is *imperfect*: incomplete and noisy.

- Can a model with *imperfect prior* reason soundly and respond factually when using only a small number of retrieval queries?
- How can we implement efficient test-time augmentation algorithms (RAG, tool use) by considering gaps in an LLM's prior knowledge?

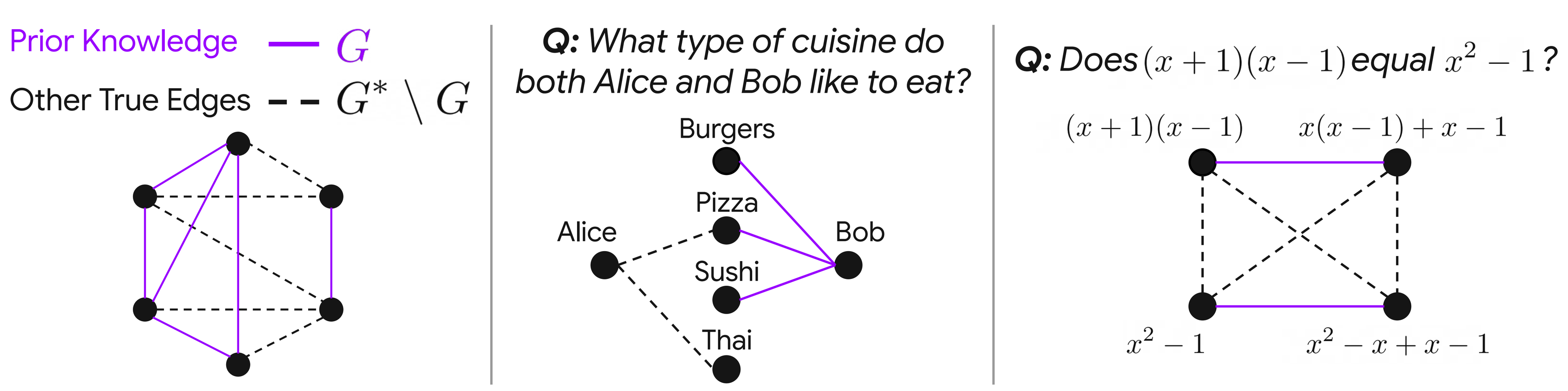

Figure 1. G^* denotes the unknown ground-truth graph and G to indicate model's pretrained knowledge graph.

We use s-t path finding on G^* as a theoretical proxy for multi-step reasoning task. The model has prior knowledge G and access to retriever oracles. In practice, we want the reasoning path to be grounded: it only consists of edges that has been explicitly observed from the oracle or the model's prior knowledge G.

We define the retrieval oracle $\mathcal{O}_{G^*} : V \to E^* \cup \{\perp\}$:

$$\mathcal{O}_{G^*} : V \to E^* \cup \{\perp\}, \qquad \mathcal{O}_{G^*}(u) = \begin{cases} (u,v), & v \sim \mathrm{Uniform}\big(N_{G^*}(u)\big) \\ \perp, & \text{if } N_{G^*}(u) = \emptyset \end{cases}$$

Definition (q-Retrieval Friendliness). The pair (G, G^*) is q-retrieval friendly for a grounded algorithm $\mathcal{A}$ if, given s, t, access to G, and retriever, the algorithm outputs:

1. **NO** when t is not reachable from s in G^*, and
2. when t is reachable, a simple s–t path P such that all edges of P are valid (i.e., they are also in G^*),

by making at most q queries in expectation.

Lower Bounds

We begin with an adversarial bridge graph to show that even a single missing piece of information can be expensive to find.

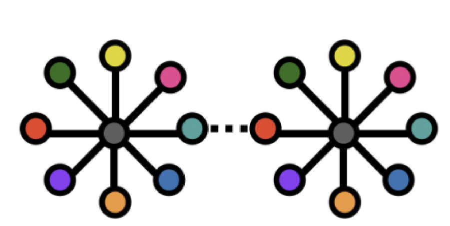

Figure 2. Double star with random bridge as prior knowledge graph G. G^* adds a single, hidden "bridge" edge between random leaves on the left and right. Any algorithm must query $\Omega(n)$ leaves on average to find this bottleneck edge.

Theorem. Let G^* be complete and G empty. For any nodes s and t, any grounded algorithm must make $\Omega(\sqrt{n})$ retrieval oracle queries to find an s-t path with success probability at least $1/2$.

Proof Idea: Birthday paradox!

Theorem above implies that without any prior knowledge, path-finding requires many queries even on dense and possibly complete graphs.

More Technical Contributions:

- We extend the lower bound to the more general setting of random graphs, even when the learner has access to a powerful retrieval oracle that given a vertex v returns *all* neighbors in G^* of the connected component in pretrained graph G that contains v.
- Proof relies on a reduction that contracts $O(\log n)$-sized components into super-nodes. Then, uses a trace based analysis by fixing the algorithm's randomness, each execution is represented as a trace.
- We provide nearly matching upper bound for Erdős-Rényi random graphs, which adapts a result of [1] for regular expander graphs to random graphs that are only approximately regular.

Bidirectional-Retrieval Augmentation Generation (BiRAG)

Require: γ-admissible pair (G^*, G), endpoints s, t, retrieval oracle $\mathcal{O}_{G^*}$

```
E_s ← ∅, E_t ← ∅
repeat
    e_s ← O_{G*}(s)
    e_t ← O_{G*}(t)
    if e_s = ⊥ or e_t = ⊥ then
        return NO
    end if
    E_s ← E_s ∪ {e_s}
    E_t ← E_t ∪ {e_t}
    Augment: G̃ ← Augment(G, E_s, E_t)
    Generate: Π ← s–t path from G̃
until Π ≠ ⊥
return Π
```

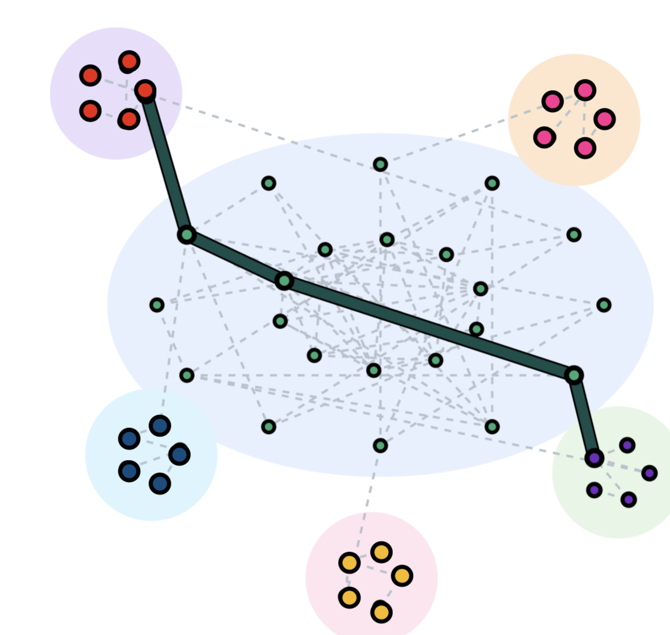

Figure 3. Illustrating γ-admissible & BiRAG. A natural extension of path-finding is multi-terminal reasoning by asking whether a learner can recover a Steiner tree connecting a set of inputs. In γ-admissible pairs, this can be done efficiently.

Efficient Test-Time Augmentation Algorithms

Definition (γ-Admissible Pair). For every vertex u, let $\mu_u^{G^*}$ be the uniform probability distribution over $N_{G^*}(u)$ given by G^*. We say that (G^*, G) is γ-*admissible* if there exists a connected component C of G such that, for every vertex u with $N_{G^*}(u) \neq \emptyset$,

$$\mu_u^{G^*}\big(N_{G^*}(u) \cap V(C)\big) \geq \gamma.$$

for some constant γ.

Claim. Any γ-admissible pair is $2/\gamma$-retrieval friendly for bidirectional-retrieval augmentation generation algorithm.

Theorem (Admissible Random Graph Example). $G^* \sim \mathcal{G}(n,p)$, $p \geq \frac{C_0 \log n}{n}$. Suppose G is formed by keeping each edge independently with probability η such that $\eta p \geq \frac{C_0}{n}$. Then, with high probability, the pair is $\frac{\gamma}{3}$-admissible where $\gamma = 1 - e^{-np\eta\gamma}$. ($C_0$ is an absolute constant)

Robustness & Reliability

To establish a robust conclusion between two entities, one should not rely on a single line of reasoning. We seek many candidate routes with evidence separated across the pretrained graph.

By finding K edge-disjoint segments in the pretrained graph even if we distrust up to $K-1$ routes, a single remaining route of a trusted suffices to certify correctness.

Definition ((K,γ)-Robust-Admissible). (G^*, G) is (K,γ)-*robust-admissible* if there exist an edge-partition $E = \biguplus_{k=1}^{K} E_k$ with $G_k := (V, E_k)$, and for each $i \in [k]$, a connected component C_k of G_k such that $\forall u \in V$ with $N_{G^*}(u) \neq \emptyset$ and all $k \in [K]$

$$\mu_u^{G^*}\big(N_{G^*}(u) \cap V(C_k)\big) \;\geq\; \gamma.$$

We show there exists an algorithm that for any (K,γ)-robust-admissible pair (G^*, G), for any two vertices $s, t \in V$

- makes $O(\frac{\log K}{\gamma})$ calls to the retrieval oracle in expectation and constructs internally K-connected subgraph between s and t.
- makes $O(1/\gamma)$ calls to the retrieval oracle in expectation and constructs internally $K/2$-connected subgraph between s and t.

Check out our paper [2] for full results :-)

Open Directions

Sub-Linear Time Algorithms with Prior Knowledge: It is natural to seek query-complexity thresholds for other graph-theoretic tasks under a partially observed prior. The relevant phenomenon is a prior sensitive phase transition. Concretely, can we find phase transitions for other graph or matrix algorithms, where prior knowledge means we can use much fewer queries? This is interesting for finding subgraphs, statistical problems (e.g., planted clique), or linear algebra tasks.

Empirical Evidence with LLMs and OOD Tasks: Our results posit that reliable pretrained knowledge and grounded retrieval oracles is necessary for efficiency and correctness of test-time methods. Can we validate this by controlling the amount of "relevant" prior knowledge in a trained model? For example, do we see an accuracy transition on some tasks when using base models vs. domain-specific models, e.g., for medical knowledge?

References

[1] Noga Alon, Allan Grønlund, Søren Fuglede Jørgensen, and Kasper Green Larsen. Sublinear Time Shortest Path in Expander Graphs. In *MFCS 2024*, volume 306 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:13, Dagstuhl, Germany, 2024. ISBN 978-3-95977-335-5. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.MFCS.2024.8.

[2] Avrim Blum, Daniel Hsu, Cyrus Rashtchian, and Donya Saless. Prior knowledge makes it possible: From sublinear graph algorithms to LLM test-time methods. In *AISTATS*, 2026. URL https://openreview.net/forum?id=fgUjxaump0.