



PEARL: Front-Loading Relational Chains for Multi-Hop Table Retrieval

Subeen Ho · Hyeongu Kang · Seongku Kang · Susik Yoon

Korea University DAIS LAB | EMNLP 2026 Budapest



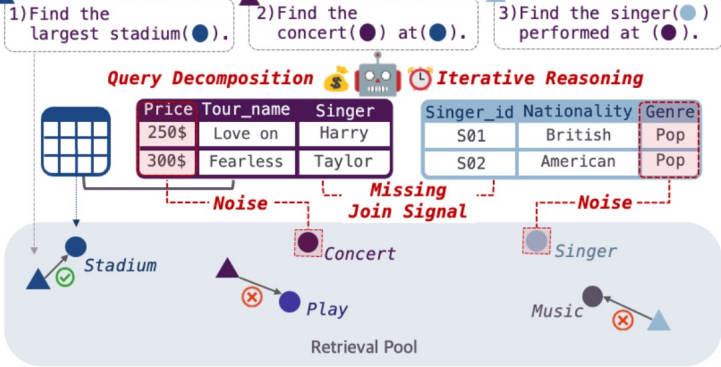
Homepage



Source Code

Motivation: Multi-hop table retrieval is harder than it should be..

What is the nationality of the singer performed at the largest stadium?



Too much noise

Whole-table embeddings intermingled within relevant and irrelevant columns, diluting the retrieval signals.

Too little join signal

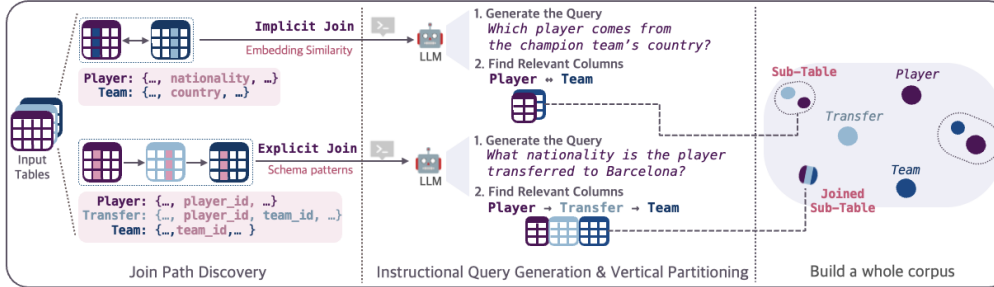
Cross-table relationships are often implicit or deeply nested, making relevant tables difficult to identify.

Too much online reasoning

Existing methods repeatedly decompose and reason over queries at inference time, increasing latency and LLM cost.

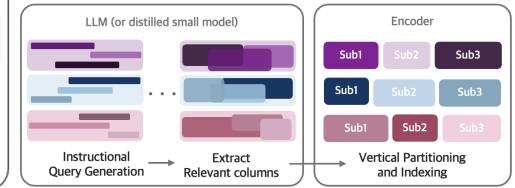
Methodology: Why reason more and index everything?

Offline: Front-load the reasoning

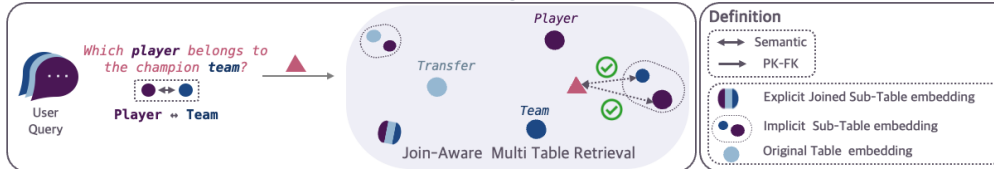


Query-Guided Vertical Partitioning

Select relevant columns to construct compact, join-aware retrieval units.



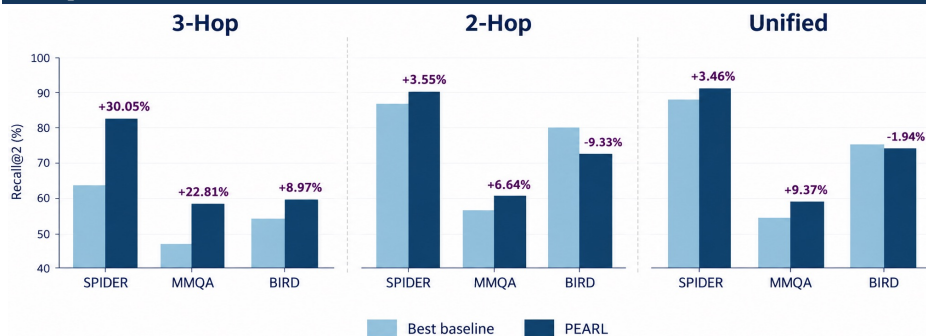
Online: Retrieve without reasoning



Offline Precomputation

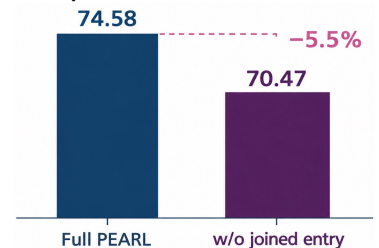
Precompute join-aware representations to eliminate query-time LLM calls

Experiments & Results: Reason less and retrieve better



Highlights

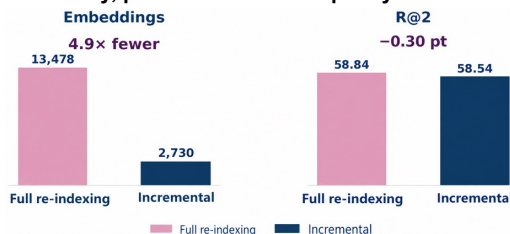
Larger gains on harder multi-hop retrieval. Join-aware representations matter most.



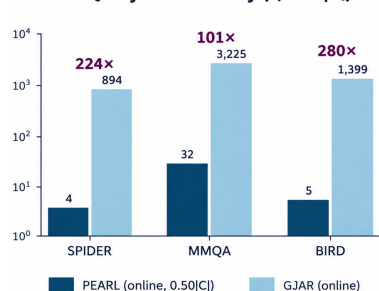
Highlights

Pay once offline. Save on every query.

Update efficiently, preserved retrieval quality.



Query-time Latency (s, 0.50|C|)



Total Token Consumption (K)

