



IEEE International Conference
on Quantum Computing
and Engineering — QCE26



Quantum-Enhanced Knowledge Graph Systems

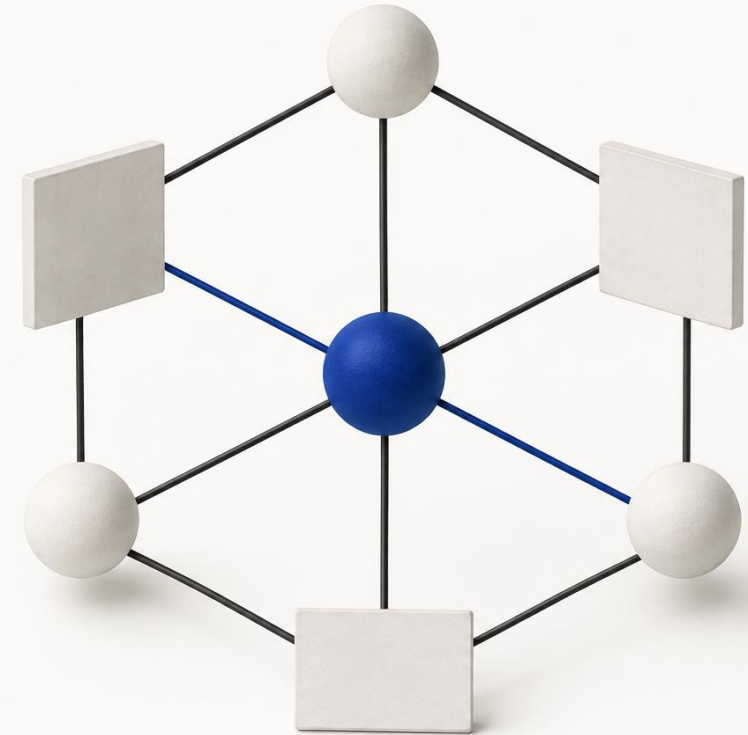
Survey, Taxonomy, and
the Road to Scalability

Wentao Huang

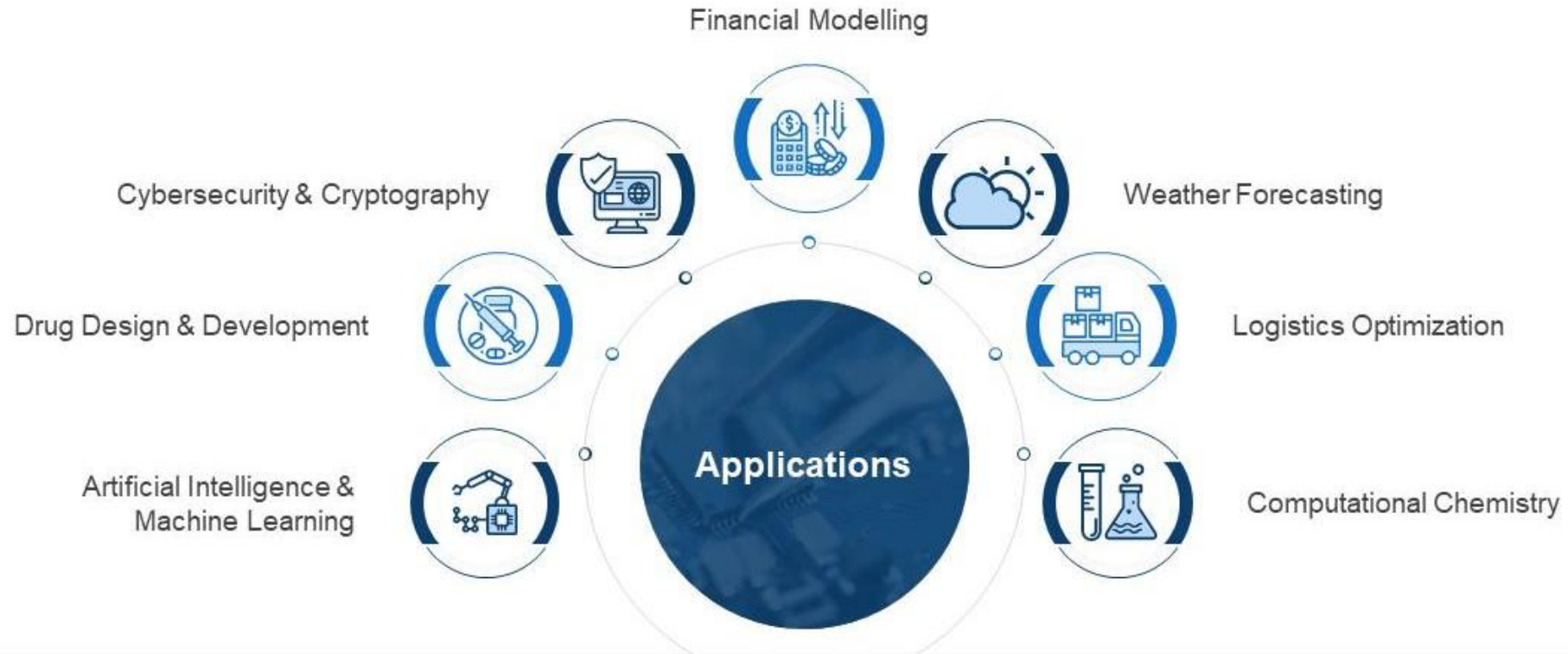
National University of Singapore

Fanfan Lin · Presenter

Zhejiang University–University of Illinois Urbana–Champaign Institute



Excitement about Quantum Computing's Application?



Quantum Methods

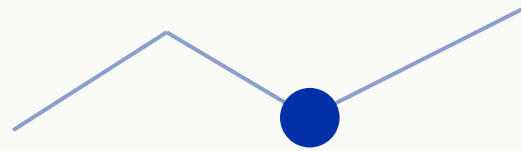
Search



Find a matching item

Grover's algorithm finds a marked element in an unstructured space of size N with $O(\sqrt{N})$ oracle calls under idealized assumptions

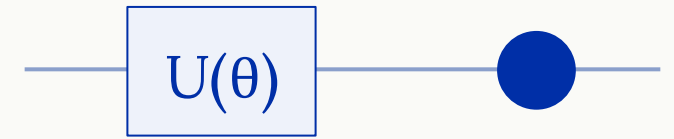
Optimization



Choose a good plan

Quantum annealing and the Quantum Approximate Optimization Algorithm (QAOA) map combinatorial optimization problems to energy landscapes or parameterized circuits

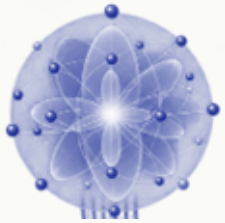
Learning



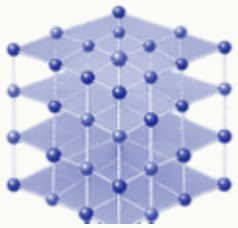
Score a possible fact

Quantum machine learning parameterizes and trains circuits for prediction; for KGs this includes embeddings and link prediction

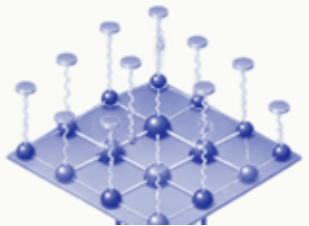
Many quantum computing tasks are inherently data-intensive.



- Large-scale Quantum States



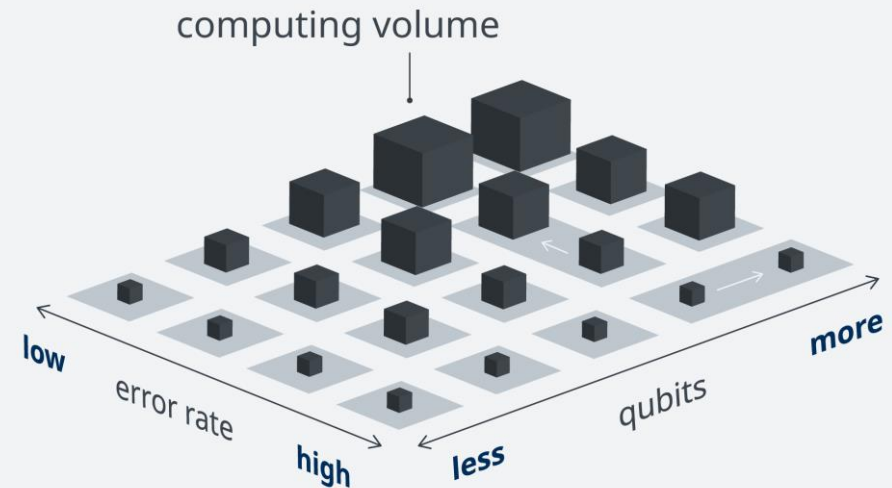
- Multidimensional Quantum Data Structures



- And Error Correction Codes in the NISQ Era

Quantum computing performance

Only at low error rates can qubits reach higher computing power



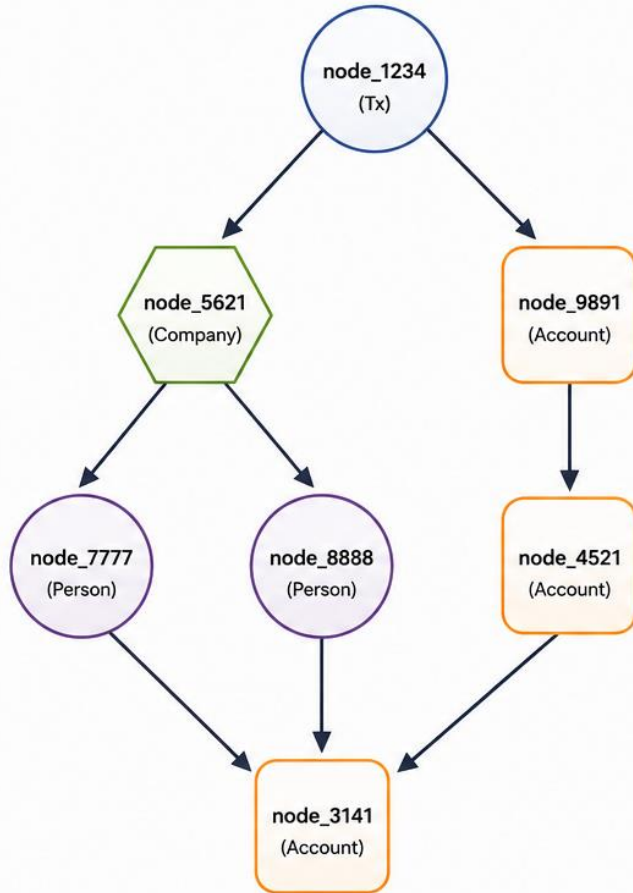
Source: IBM

©DW

What is Knowledge Graph? $\text{Data} + \text{Ontology} = \text{Knowledge Graph}$.

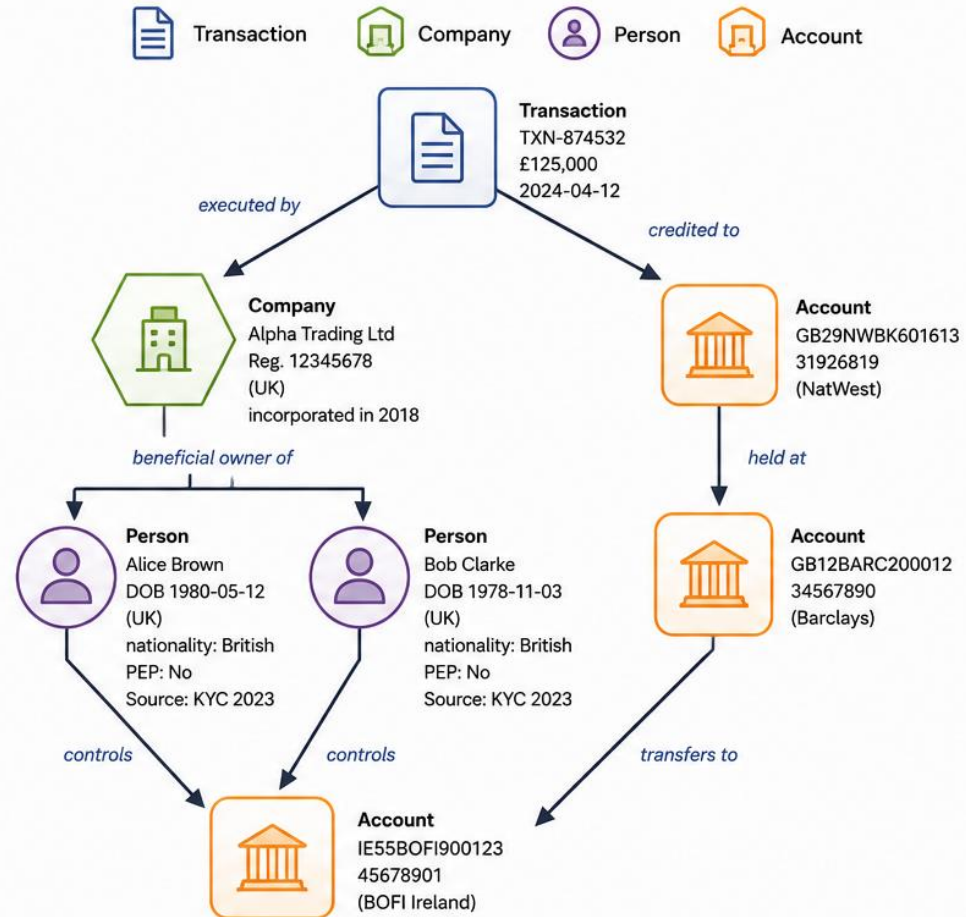
Graph database

Nodes and edges only. No meaning.



Knowledge graph

Same data. With meaning.



Same structure. Different meaning.

Knowledge Graphs at Internet Scale



Knowledge Graph

500B+

facts across 5 billion entities

Search & knowledge panels

People, places, and things



Economic Graph

1.3B+

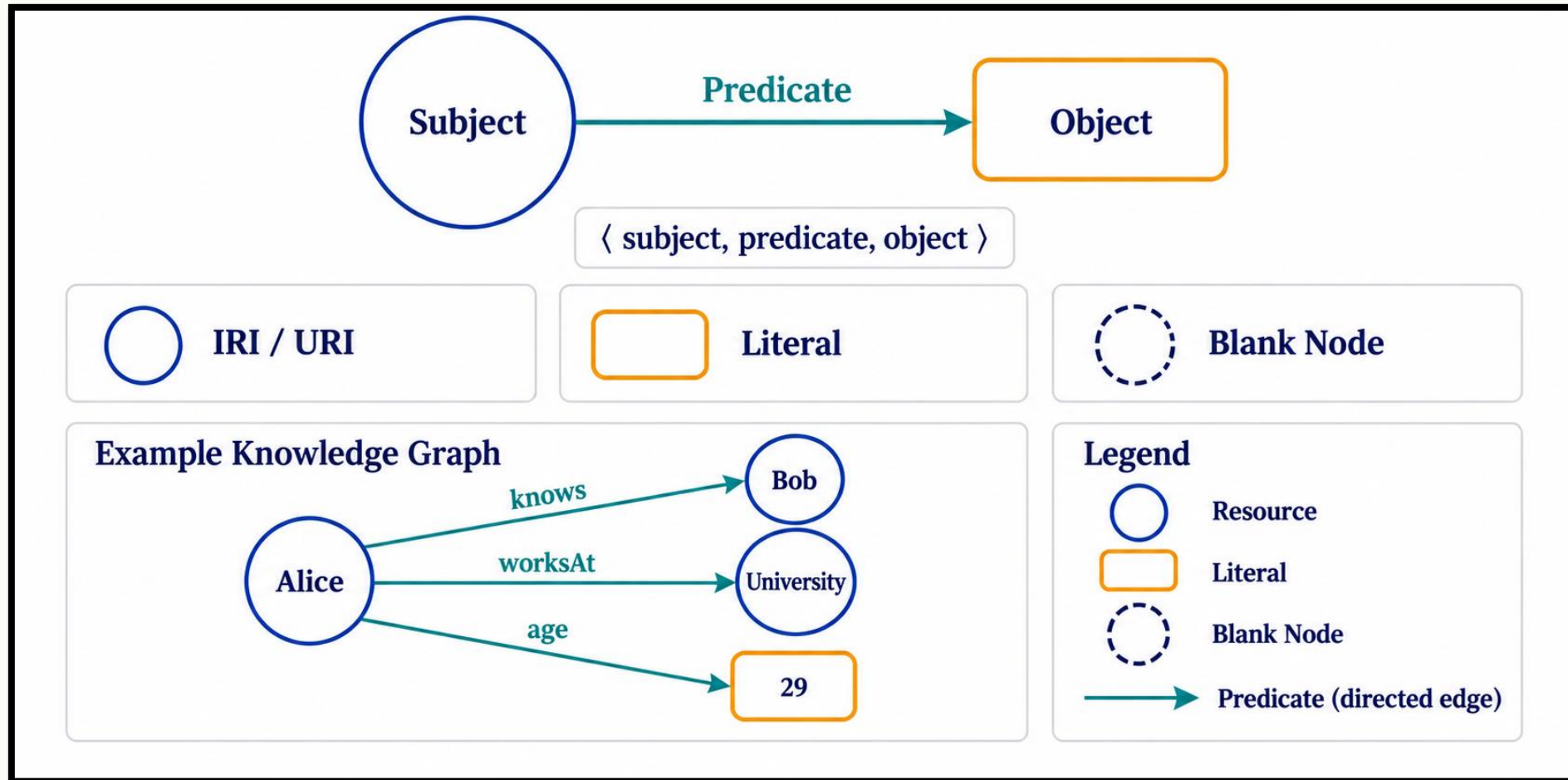
members · 71 million companies

Workforce & skills insights

Members, companies, schools, and skills

How Classical Systems Organize Connected Data: Knowledge Graphs and RDF

Building linked knowledge graph with triples



Bit vs. Qubit

BIT

*Classical
Computing*

0 

1 

QUBIT

*Quantum
Computing*



Data in the Quantum Era is Different.

Probabilistic



Quantum data is probabilistic.

Unlike a classical bit, which is 0 or 1, a quantum bit can be in superposition.

Fragile



Quantum data is fragile.

With current quantum technology in the NISQ era, quantum data is prone to noise.

Entangled



Quantum data can be entangled.

Entanglement means that multiple qubits are correlated and that measuring the state of one qubit immediately affects the others.

Research Question

- ~~• “Can Quantum Computation Accelerate a KG-based Database Task”~~
- What Data Management Infrastructure is Needed to Make Quantum Computation Useful for Large, Structured KGs?

01 Survey

02 Taxonomy

03 Roadmap

A Two-Dimensional Taxonomy

4 Data-Management Stack × 3 Quantum Paradigms



LAYER	Classical simulation	Hybrid quantum-classical	Quantum-native*
01 Representation & encoding	RDF dictionaries; basis / amplitude encoding	Predicate-separated register layouts	Graph-state encodings
02 Storage & simulation	State vectors; tensor networks	State preparation; selected execution	Fault-tolerant quantum memory
03 Access & query processing	Oracle and circuit simulation	Grover lookup; QUBO plan search	QRAM; coherent access paths
04 KG reasoning & learning	Circuit-model evaluation	Variational KGE; reasoning reductions	Quantum-native graph learning

Different Forms of Quantum Involvement

Classical simulation

Classical computer
Models quantum behavior

No physical quantum processor is required.

Hybrid

Classical orchestration
Selected quantum kernels

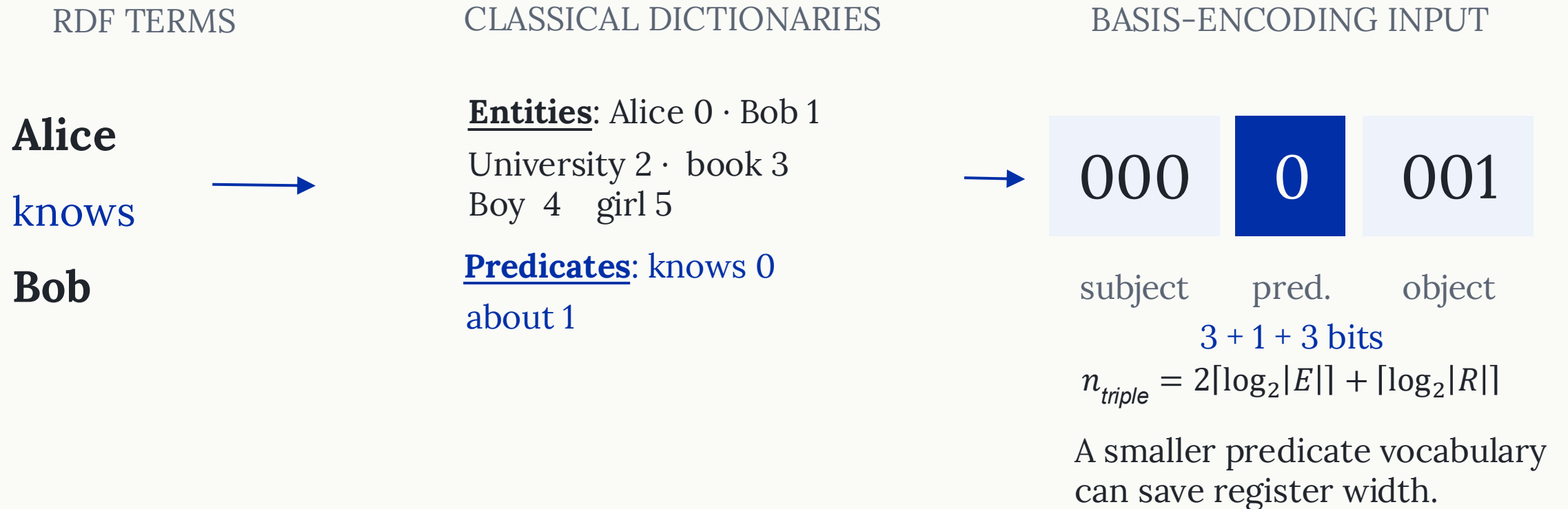
Near-term candidate;
benefit must be measured.

Quantum-native

Coherent data and access
Fault-tolerant infrastructure

Future architecture with
strong hardware assumptions.

From symbolic triples to fixed-width registers



Symbolic RDF data are mapped into the **bitstrings or amplitudes a quantum computer can act on**. The choices made fix qubit width, dictionary footprint, and state-preparation cost for later stages.

01 REPRESENTATION & ENCODING

Dictionary Construction and Two-Pass Pipelines

Stack Layer	Main question	Representative techniques	Scalability risks
Representation & encoding	Mapping RDF terms and triples to quantum states	Basis/amplitude encoding; predicate-separated coding; bit-width minimization	Dictionary memory; literal handling; state-preparation cost

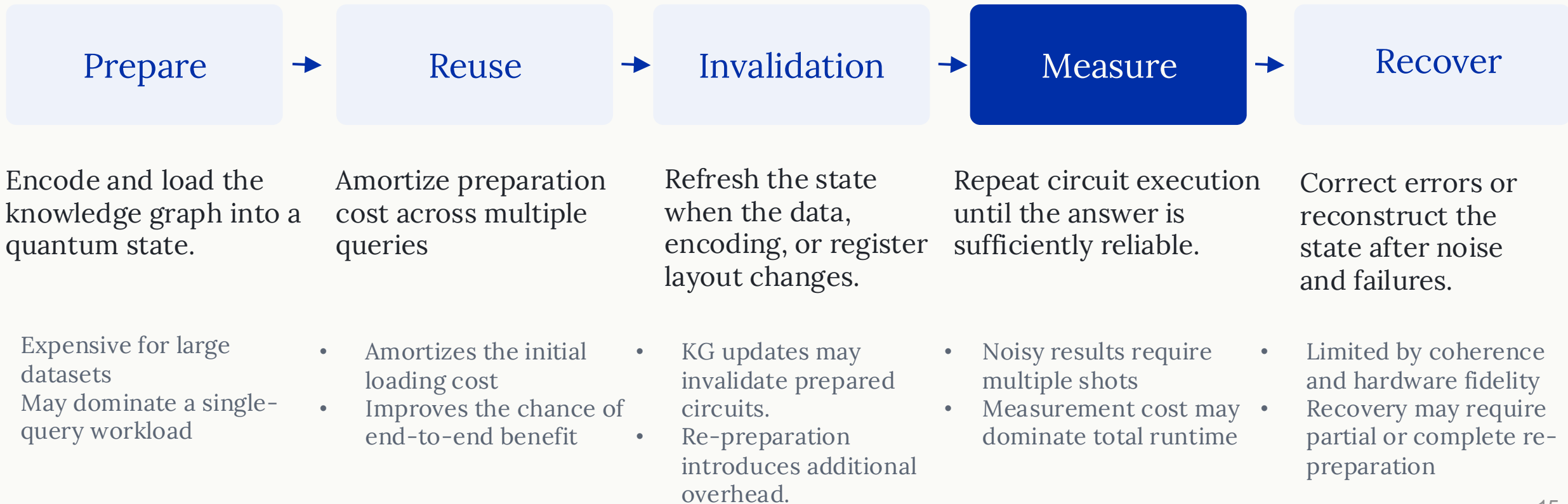
Research Directions

External-memory Construction	Build dictionaries with disk-based methods when unique terms no longer fit in main memory.
Distributed Encoding	Encode partitions in parallel while maintaining one deterministic global identifier space.
Update-aware Encoding	Add new entities and predicates without rebuilding every existing encoding and quantum circuit.
Workload-aware Encoding	Assign compact identifiers to frequently accessed graph regions to reduce common-query qubit costs.
Semantics-preserving Encoding	Retain graph meaning so that quantum oracles and access paths can exploit semantic structure.

02 STORAGE & SIMULATION

Managing the Quantum-State Lifecycle

Grover's theoretical speedup must be evaluated together with state preparation, oracle execution, and measurement costs.



02 STORAGE & SIMULATION

Classical simulation is also a data-management problem

Before running on quantum hardware, we must prepare the KG, build the circuit, store the simulated state, and manage the experiment.

READ & ENCODE

Import RDF data and convert it into quantum-ready bitstrings.

BUILD & ESTIMATE

Generate the quantum circuit and estimate whether it is practical to run.

STORE THE STATE

Choose how the classical simulator represents the quantum state.

RUN & RECORD

Execute the simulation and preserve the results for later comparison.

A quantum simulator is also a data system: it must load, store, execute, and track large experiments.

A sparse knowledge graph may still induce a difficult quantum simulation.

03 ACCESS & QUERY PROCESSING

How stored KG data are searched, joined, bound to query variables, and planned into efficient queries.

Different access tasks require different answers

- **Lookup**

Does Alice know Bob?

A keyed membership answer

→ Boolean, with a classical index baseline

- **Binding Generation**

Return all matches

Often output-heavy

- **Search**

Find an author of a Graphs paper

One matching candidate

- **Optimization**

Choose a query execution plan

A low-cost plan under a cost model

→ A decision, not the query answers

03 ACCESS & QUERY PROCESSING

Grover's advantage depends on the access model

Unstructured search · N candidates · k matches · $k > 0$

$$O(N/k)$$

Classical probes

$$O(\sqrt{N/k})$$

Quantum oracle queries

It does not establish an end-to-end speedup.

The systems questions

How is the state prepared?

What does the oracle cost?

How are answers extracted?

Can preparation be reused?

**Compare against indexed
RDF query engines.**

03 ACCESS & QUERY PROCESSING

Quantum optimization can choose a classical plan

Find authors of papers about Graphs

Plan A: filter by topic first

Graphs $\leftarrow$ about $\leftarrow$ papers $\leftarrow$ authored $\leftarrow$ people

Plan B: scan authorship first

people $\rightarrow$ authored $\rightarrow$ papers $\rightarrow$ about $\rightarrow$ Graphs

$$\min_{x \in \{0,1\}^m} x^T Q x$$

Binary choices encode plan decisions.
Q includes costs and constraint penalties.
Illustrative QUBO formulation

Quantum solver: plan selection · Classical engine: data execution

Relevant evidence: index selection [30], multiple-query optimization [31], bushy joins [33]

03 ACCESS & QUERY PROCESSING

Every quantum operator must justify its full cost

SCHEMATIC · BOX WIDTHS DO NOT REPRESENT MEASURED COSTS



$$C_{\text{class}} > C_{\text{enc}} + C_{\text{prep}} + C_{\text{circ}} + C_{\text{queue}} + C_{\text{exec}} + C_{\text{shots}} + C_{\text{dec}}$$

Compare equivalent tasks, answer quality, and operating conditions.

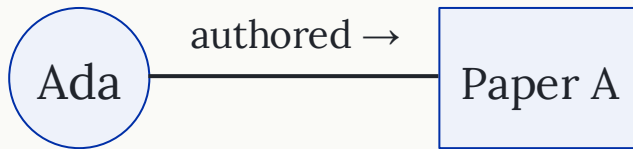
Separate one-time costs from costs repeated for every query.

Report execution and repeated-shot costs without double counting.

Retrieval, inference, and prediction differ

Retrieve

Return a stored fact



Supported by a recorded edge

Infer

Apply an explicit rule

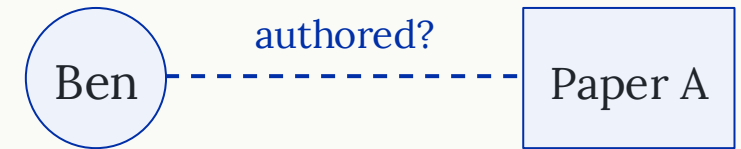
Ada : Researcher
 $\text{Researcher} \subseteq \text{Person}$

Ada : Person

Entailed under this schema

Predict

Score a candidate edge



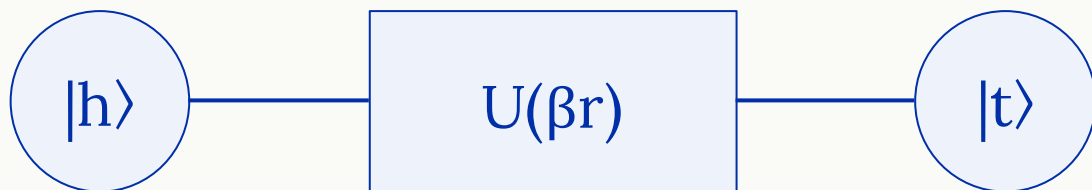
A hypothesis, not an entailment

This layer also covers graph construction and entity matching.

Candidate generation → quantum subproblem → classical validation

Quantum embeddings learn a candidate-triple score

Candidate: **Ben** — **authored?** → **Paper A** · (h, r, t)



$$\delta_{hrt} = |\langle t|U(\beta_r)|h\rangle|^2$$

Entity state → relation transformation → overlap

A score, not a calibrated truth probability

Classical training loop

Positive / negative triples → loss → parameter update → circuit re-evaluation

Different design choices explored.

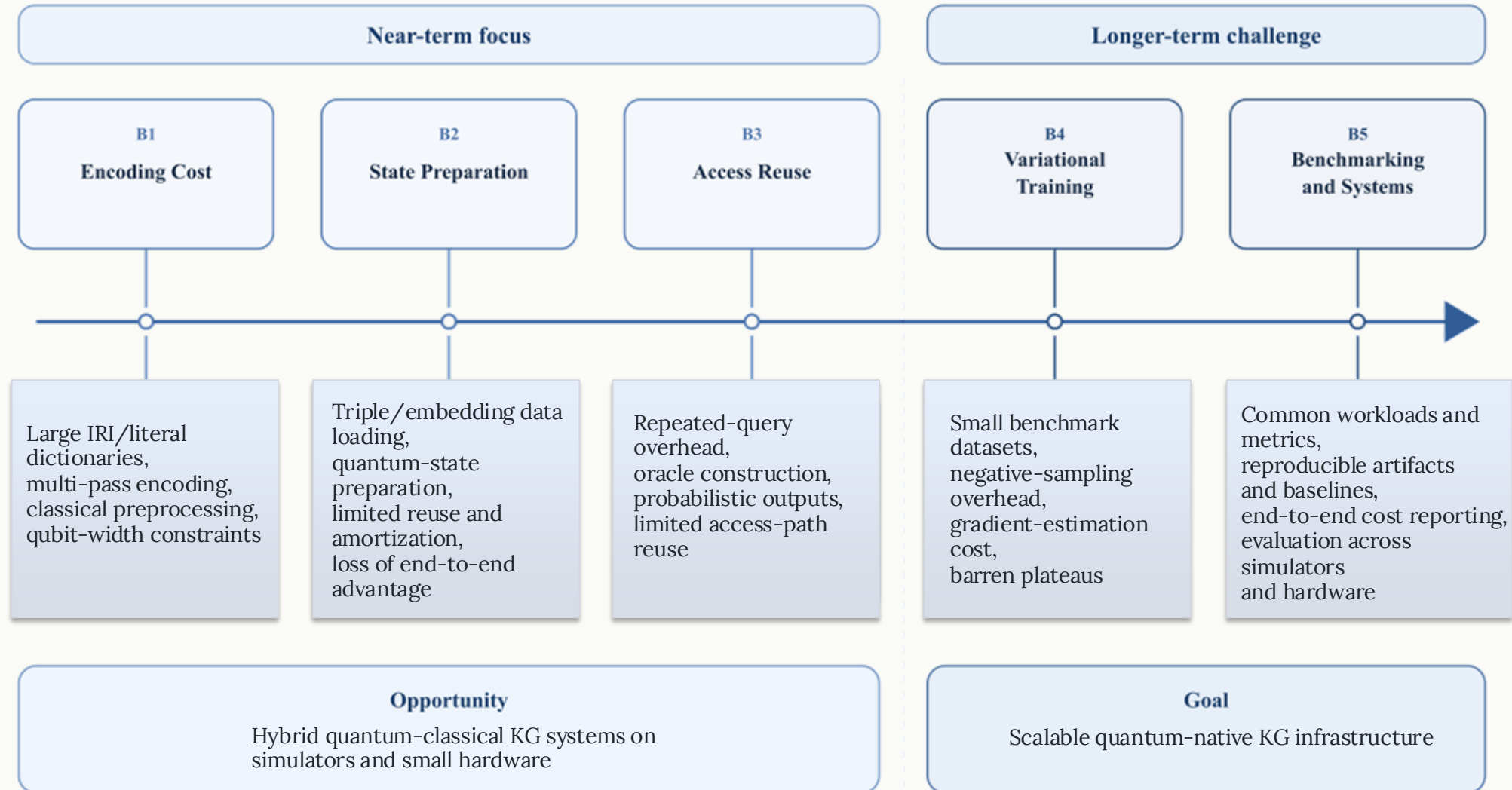
WORK	DESIGN CHOICE	DATA	QUALITATIVE LESSON
Ma et al. [15,16]	Hilbert-space KGE formulation	Small KGs	States and unitaries can parameterize triple scoring
Kurokawa et al. [49]	18 circuit templates	Kinship	Expressibility and accuracy trade off against depth/noise
Giri et al. [51]	Superposed negative samples	UMLS	Training primitives matter, beyond the representation
Giri et al. [53]	Superposition batching	UMLS	Sample-scale gains depend on coherent data loading

Five bottlenecks recur across the layers

B1	Encoding & width	Large dictionaries, literals, multipass preprocessing
B2	State preparation	Loading cost and limited amortization
B3	Access reuse	Oracles, repeated execution, output extraction
B4	Variational training	Negatives, gradients, barren plateaus, noise
B5	Benchmarking & systems	Shared workloads, baselines, reproducible traces

Benchmarking belongs in every stage of the roadmap.

Five bottlenecks recur across the layers



The Near-Term Path is a Cost-Aware Hybrid KG System

CLASSICAL CONTROL & METADATA

RDF / workload front end

Versioned dictionaries

Statistics, circuits, provenance

Hybrid optimizer

Cost · confidence · width / depth · reuse

Choose classical, simulated, or quantum execution for each candidate operator.

Classical RDF engine

Indexes, joins, reasoners, learning

Quantum simulator

Circuit and resource evaluation

Quantum hardware

Selected feasible kernels

Select quantum operators by total cost and confidence, then decode and verify the results classically.



IEEE International Conference
on Quantum Computing
and Engineering — QCE26

Thank you for your attention!

