ConRAD: Conformal Risk-Aware Neural Databases

Sonia Horchidan
KTH, Stockholm
sfhor@kth.se

Fabian Zeiher
KTH, Stockholm
zeiher@kth.se

Xiangyu Shi
KTH, Stockholm
xiangyus@kth.se

Vasiliki Kalavri
Boston University
vkalavri@bu.edu

Henrik Boström
KTH, Stockholm
bostromh@kth.se

Ioannis Kontoyiannis
University of Cambridge
ik355@cam.ac.uk

Paris Carbone
KTH, Stockholm
parisc@kth.se

ABSTRACT

Querying incomplete knowledge graphs with neural predictors is powerful but dangerous. Errors compound across multi-hop pipelines with no formal bound on the completeness of results. We introduce ConRAD, the first framework to enforce declarative marginal recall guarantees natively within a neural graph database query engine. Given a user-specified risk budget, ConRAD automatically derives per-operator prediction thresholds that satisfy the recall target in expectation over the query distribution, with finite-sample, distribution-free statistical validity via Conformal Risk Control, while maximizing end-to-end precision. To scale calibration across multi-operator query topologies, we introduce a *quantile-space scalarization* that reduces intractable high-dimensional threshold searches to a single parameter. We further design *the conformal gate*, a novel physical operator that dynamically bypasses neural inference when local graph evidence suffices, eliminating unnecessary model inferences in dense graph regions. Evaluated across three benchmarks and eight query topologies, ConRAD satisfies all risk budgets, with empirical recall falling below the target by at most 0.0547 across all settings. It reduces neural invocations to zero in near-complete graph regions, and achieves precision that matches or exceeds best-case static baselines that offer no guarantees and require manual threshold search.

PVLDB Reference Format:
Sonia Horchidan, Fabian Zeiher, Xiangyu Shi, Vasiliki Kalavri, Henrik Boström, Ioannis Kontoyiannis, and Paris Carbone. ConRAD: Conformal Risk-Aware Neural Databases. PVLDB, 19(11): 3511 - 3524, 2026.
doi:10.14778/3836663.3836705

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/soniahorchidan/conrad.

1 INTRODUCTION

Knowledge graphs (KG) operate under the Open World Assumption (OWA), where missing edges represent unobserved facts rather than true negatives [1, 23]. Neural Graph Databases (NGDBs) aim to address this incompleteness by composing exact graph retrievals with probabilistic neural predictors [10, 24, 25, 45]. Recent foundation models [18, 19] excel at answering complex queries over incomplete KGs. However, these models are trained to optimize pointwise accuracy on individual predictions [52], and while post-hoc calibration techniques can improve score reliability at the single-operator level [57], no existing method provides formal correctness guarantees when such predictors are composed into multi-hop query plans. When inference-based operators form multi-hop execution pipelines, statistical errors compound non-linearly. An early false negative irreparably prunes valid subtrees, while a false positive floods downstream operators with cascading noise. In standard machine learning practice, practitioners mitigate uncertainty by manually tuning static prediction thresholds or top-k limits. While this heuristic approach may suffice for isolated predictions, it breaks down entirely under the compositionality of database queries. A globally tuned threshold cannot dynamically adapt to operator heterogeneity or local graph sparsity. By relying on brittle hyperparameter tuning rather than system-managed bounds, this approach effectively forces the application developer to act as the query optimizer, manually managing risk across complex query topologies without any mathematical assurances.

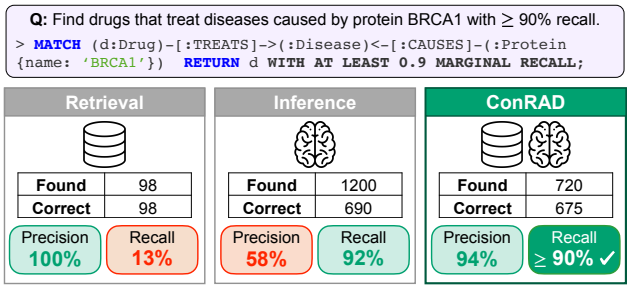

Figure 1: Illustrative precision-recall trade-off on one query (750 ground truth answers). Pure retrieval fails the recall target (13% vs. ≥ 90%) due to graph incompleteness; uncalibrated inference meets it (92%) but precision collapses to 58%. ConRAD achieves 94% precision while enforcing the recall target in expectation over the query distribution, with finite-sample validity for the specified risk budget.

The consequences are severe in safety-critical deployments. Consider a drug safety monitoring application over a biomedical KG. A query such as *find all approved drugs that interact with proteins involved in pathways associated with cardiac events* requires multi-hop conjunctive reasoning across drug → protein, protein → pathway, and pathway → adverse effect relations. The underlying graph is inherently incomplete, as thousands of new biological interactions

Proceedings of the VLDB Endowment, Vol. 19, No. 11 ISSN 2150-8097.
doi:10.14778/3836663.3836705

Table 1: Summary of main notations.

Symbol	Description
$\hat{\mathcal{G}} = (\mathcal{V}, \mathcal{L}, \hat{\mathcal{E}})$	Observed (incomplete) KG with factual triples $\hat{\mathcal{E}}$.
$\mathcal{G} = (\mathcal{V}, \mathcal{L}, \mathcal{E})$	Complete ground-truth KG under OWA ($\hat{\mathcal{E}} \subseteq \mathcal{E}$).
$q = \omega_1 \circ \cdots \circ \omega_k$	Query q as a composition of k operators.
$\mathcal{T}$	Query topology (e.g., 3p, 2u, ip).
ω_i	The i-th operator in the query plan.
$f^{(i)}$	Retrieval over observed triples $\hat{\mathcal{E}}$.
$\tilde{f}^{(i)}_{\lambda_i}$	Stochastic inference with threshold λ_i.
$\tilde{f}^{(i)}_{\lambda_i}$	Conformal gate with threshold λ_i.
$\hat{Y}^{(i)}$	Set of output entities produced by operator ω_i.
$Y^{(i)}$	Ground truth set (entities reachable in $\mathcal{G}$).
$\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_k]$	Vector of thresholds for all operators in query q.
$\mathcal{R}(\boldsymbol{\lambda})$	Expected end-to-end risk (e.g., FNR) under $\boldsymbol{\lambda}$.
$C(\boldsymbol{\lambda})$	Expected execution cost (e.g., cardinality) under $\boldsymbol{\lambda}$.
$\boldsymbol{\gamma} \in \Gamma$	Scalarization strategy.
α	User-specified risk budget (e.g., maximum FNR).
$\eta \in [0, 1]$	Global scalar parameter for pipeline strictness.
$\delta \in (0, 1)$	Routing threshold in the conformal gate.

are published in the literature weekly [13]. However, if uncalibrated thresholds cause the system to silently miss valid drug candidates, the entire screening pipeline becomes unreliable. Without formal statistical bounds on expected result completeness, adopting probabilistic neural predictors in such workflows is not viable [16].

We argue that correctness must transition from developer-tuned hyperparameters to a declarative constraint managed natively by the query engine. In a traditional database system, the user specifies a logical query and the optimizer selects the fastest valid physical plan. Similarly, a neural database should allow users to specify a risk budget, delegating the selection of the most precise execution plan to the system. To realize this vision, we introduce ConRAD (Conformal Risk-Aware Neural Databases). To our knowledge, ConRAD is the first framework to provide formal, finite-sample marginal recall guarantees over composed stochastic query plans in an NGDB. Given a recall target (e.g., risk budget $\alpha = 0.1$, requiring 90% recall), ConRAD automatically derives per-operator prediction thresholds that jointly satisfy the global risk budget while maximizing end-to-end precision. The guarantee is marginal rather than per-query: individual queries may fall above or below the target, but the average over queries drawn from the calibration distribution is provably bounded with finite-sample validity. This is the strongest distribution-free guarantee achievable without restrictive assumptions or infinite calibration data [17]. Our approach is grounded in Conformal Risk Control (CRC) [6], a framework that bounds arbitrary loss functions over black-box models.

To illustrate this tension, consider the precision–recall tradeoff depicted in Figure 1. We execute a query over an incomplete KG where the ground truth contains 750 answers. Pure database retrieval yields perfect precision but recovers only 13% of answers, because graph incompleteness prevents most valid paths from being explored. Replacing deterministic retrieval with uncalibrated link prediction recovers more answers but floods the pipeline with false positives. There, while recall incidentally reaches 92%, the user must accept an unpredictable deterioration in result quality, as precision drops to 58%. ConRAD bridges this gap by calibrating a hybrid approach that merges retrieval and inference to maximize precision (94%) while satisfying a user-defined marginal recall target ($\geq$ 90%).

Realizing this vision poses three technical challenges. First, standard CRC calibrates a scalar threshold for a single predictor, whereas NGDB query plans require coordinated calibration across multiple operators. Naively partitioning the global risk budget via the union bound ignores inter-operator correlations and is wasteful, as we demonstrate in Sec. 4. Second, the joint threshold space grows exponentially in the number of operators and is non-decomposable: each operator's threshold shifts the input distribution of every downstream operator, so independent tuning cannot guarantee the joint constraint. Per-query guarantees are unattainable with finite calibration data [17]. Third, neural inference is both computationally expensive and fundamentally unnecessary when the observed graph is locally complete. Invoking a neural model on dense, well-connected regions wastes resources and introduces false positives that degrade downstream precision. These challenges motivate the design of ConRAD. Our contributions are as follows:

(1) We formalize the mapping of user-declared recall targets to per-operator threshold vectors as a constrained optimization problem, replacing ad-hoc threshold tuning with a system-managed calibration process grounded in finite-sample statistical guarantees. The formulation is agnostic to both the neural scoring model and the underlying graph store (Sec. 3).
(2) We extend Conformal Risk Control from scalar to vector calibration over multi-operator query topologies via a *quantile-space scalarization* that reduces the k-dimensional threshold space to a single monotonic parameter, while preserving the nested sets required by CRC (Sec. 4.1).
(3) We introduce *the conformal gate*, a novel physical database operator that dynamically routes between retrieval-based execution and neural inference at each logical operator. By placing exact and neural evidence on a unified calibrated scale, the gate bypasses inference entirely in dense graph regions (Sec. 4.2).
(4) We implement what is, to our knowledge, the first fully queryable neural graph database with formal correctness guarantees, integrating UltraQuery [19] with Neo4j [1]. We evaluate on three benchmarks, eight standard query topologies, and data incompleteness levels ranging from 5% to 40%. ConRAD satisfies recall targets across all settings (max. downward deviation: 0.0547) and achieves precision matching or exceeding best-case static baselines that offer no guarantees. Furthermore, ConRAD reduces neural invocations by up to 100% in well-connected regions under generous risk budgets (Sec. 6.2).

Although our evaluation centers on KGs, ConRAD's formal guarantees extend to any predictive pipeline that composes probabilistic operators into a directed acyclic graph, including retrieval-augmented generation (RAG) [33], multi-stage retrieval [31, 43], and learned database components [30, 32, 41]. By transitioning correctness from a manually-tuned parameter to a declarative system constraint, ConRAD provides a principled blueprint for integrating machine learning inference into the broader data processing stack.

[1] https://neo4j.com/

2 SETTING AND FOUNDATIONS

ConRAD targets the execution of complex logical queries over incomplete KGs under OWA. Given a user query and a declarative recall target, ConRAD automatically derives calibrated per-operator thresholds that satisfy the guarantee while maximizing end-to-end precision. Below, we describe ConRAD's data model, supported queries, execution model, and the statistical foundations it relies on. Table 1 summarizes the notation used throughout the paper.

2.1 Data Model and Supported Queries

We represent an observed KG as a directed, labeled graph $\hat{\mathcal{G}} = (\mathcal{V}, \mathcal{L}, \hat{\mathcal{E}})$, where $\mathcal{V}$ is the set of entities, $\mathcal{L}$ the set of relation types (edge labels), and $\hat{\mathcal{E}} \subseteq \mathcal{V} \times \mathcal{L} \times \mathcal{V}$ the set of observed triples. We adopt the standard KG triple model used by existing neural query answering frameworks [46, 47]. Node and edge properties, as found in property graph models, can be represented by encoding property values as entities connected via dedicated relation types. Under OWA, we assume a complete ground-truth graph $\mathcal{G} = (\mathcal{V}, \mathcal{L}, \mathcal{E})$ with $\hat{\mathcal{E}} \subseteq \mathcal{E}$, with $\mathcal{E} \setminus \hat{\mathcal{E}}$ representing true but unobserved facts.

ConRAD supports the Existential Positive First-Order (EPFO) fragment of first-order logic, supporting existential quantification ($\exists$), conjunction ($\wedge$), and disjunction ($\vee$) [46]. Queries are DAGs of operators where leaf nodes are anchor entities and the sink produces the answer set. Three operators map directly to EPFO primitives: *projection* navigates a relation $r \in \mathcal{L}$ from a source set $S \subseteq \mathcal{V}$ to retrieve reachable entities; *intersection* and *union* perform set conjunction and disjunction over intermediate results.

We evaluate over eight of the nine standard query topologies introduced by the Query2Box benchmark [46]: projection chains (`2p`, `3p`), intersections (`2i`, `3i`), unions (`2u`), and mixed compositions (`ip`, `pi`, `up`). These cover all EPFO composition primitives (i.e., cascading joins, conjunctive merges, and parallel unions) at varying depths. We exclude the single-hop topology (`1p`) as it involves no operator composition and is therefore trivially handled by standard CRC on a single predictor.

What ConRAD does not support. ConRAD excludes negation and iterative (recursive) queries. Negation is semantically ill-defined under OWA [1] and breaks the monotonicity of operator composition (Theorem 4.1), invalidating the nestedness property on which our recall guarantees depend (Sec. 4.1). Iterative queries are excluded because the threshold vector $\boldsymbol{\lambda}$ we propose in Sec. 4.1 has fixed dimensionality determined by the query topology. Unbounded recursion would, therefore, require a variable-length threshold vector, which is incompatible with our offline calibration procedure.

2.2 Execution Model

A key design principle of ConRAD is that each logical operator is backed by two execution primitives: a deterministic *retrieval* operator that traverses observed edges in $\hat{\mathcal{G}}$, and a stochastic *inference* operator that scores candidate triples using a neural model. The conformal gate (Sec. 4.2) dynamically composes these primitives.

Retrieval. The retrieval operator $f^{(i)}$ executes an exact traversal over $\hat{\mathcal{E}}$. Given an input set $\hat{Y}^{(i-1)} \subseteq \mathcal{V}$ and a relation $r \in \mathcal{L}$:

$$f^{(i)}(\hat{Y}^{(i-1)}, r) = \{t \in \mathcal{V} \mid \exists h \in \hat{Y}^{(i-1)},\ (h, r, t) \in \hat{\mathcal{E}}\} \tag{1}$$

By construction, every returned triple exists in $\hat{\mathcal{G}}$, guaranteeing zero false positives. However, recall is bounded by graph completeness.

Inference. The inference operator $\tilde{f}^{(i)}_{\lambda_i}$ replaces exact traversal with a learned scoring function $\phi : \mathcal{V} \times \mathcal{L} \times \mathcal{V} \to [0, 1]$ provided by a neural model. Given an input set $\hat{Y}^{(i-1)}$ and a relation r, the operator admits all candidate entities whose score exceeds a threshold λ_i:

$$\tilde{f}^{(i)}_{\lambda_i}(\hat{Y}^{(i-1)}, r) = \{t \in \mathcal{V} \mid \exists h \in \hat{Y}^{(i-1)},\ \phi(h, r, t) \geq \lambda_i\} \tag{2}$$

The threshold λ_i governs the precision–recall trade-off at each operator: lowering λ_i recovers missing answers but potentially floods downstream operators with false positives.

The threshold selection problem. The resulting optimization problem is inherently coupled: each operator's threshold affects the input distribution of every downstream operator, making independent per-operator tuning insufficient (Sec. 1). ConRAD eliminates this burden. Given a query $q = \omega_1 \circ \cdots \circ \omega_k$ represented as a DAG of k operators, ConRAD automatically derives a threshold vector $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_k]$ that satisfies a user-declared recall target while maximizing result-set cardinality as a proxy for precision.

2.3 Statistical Foundations

ConRAD's calibration procedure builds on Conformal Prediction methods, which we briefly review in this section.

2.3.1 Standard Split Conformal Prediction. The Conformal Prediction (CP) framework calibrates a model by measuring how unusual new data points are compared to a held-out set [4, 50, 55]. Given n exchangeable tuples, $\mathcal{D}_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^{n}$, CP requires a non-conformity score $s(x, y) \in \mathbb{R}$, quantifying the disagreement between a tuple's features x and a proposed label y (e.g., $1 - \phi(x, y)$ for neural scoring). It then computes the empirical quantile $\hat{q}$ of these non-conformity scores on $\mathcal{D}_{\text{cal}}$ at the level $\lceil (n+1)(1-\alpha) \rceil / n$ and constructs a prediction set $C(x_{n+1}) = \{y \in \mathcal{Y} \mid s(x_{n+1}, y) \leq \hat{q}\}$ for any new exchangeable tuple (x_{n+1}, y_{n+1}), where α is the user-defined error budget. This procedure provides a *marginal coverage guarantee*: the true outcome is guaranteed to be in the prediction set with high probability, formally $\mathbb{P}(y_{n+1} \in C(x_{n+1})) \geq 1 - \alpha$. However, standard CP controls only miscoverage (0-1 loss). Database workloads require guarantees over aggregate, set-valued metrics (e.g., recall, False Negative Rate).

2.3.2 Conformal Risk Control. To accommodate these system-level metrics, Conformal Risk Control (CRC) [6] generalizes the standard CP framework to bound arbitrary loss functions $L(C_\lambda(x), y) \in [0, B]$. CRC introduces a tunable parameter λ that dictates the "strictness" or size of the prediction set $C_\lambda(x)$. Crucially, the chosen loss function must be non-increasing with respect to λ. For example, as a selection operator becomes more permissive (larger λ), the loss stays the same or decreases. The goal of CRC is to identify a threshold $\hat{\lambda}$ that guarantees the expected risk (e.g., FNR) remains strictly below a user-defined risk budget α. CRC achieves this by finding the tightest parameter that satisfies a corrected empirical risk bound on the calibration set:

$$\hat{\lambda} = \inf\{\lambda : \frac{n}{n+1}\widehat{R}(\lambda) + \frac{B}{n+1} \leq \alpha\} \tag{3}$$

where $\widehat{R}(\lambda) = \frac{1}{n}\sum_{i=1}^{n} L(C_\lambda(x_i), y_i)$ is the empirical risk over $\mathcal{D}_{\text{cal}}$ and B is an upper bound of the loss. The correction terms account

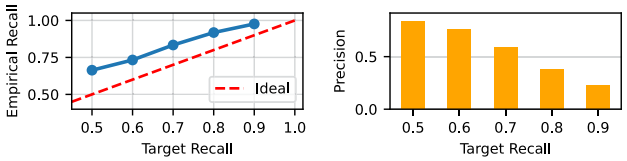

Figure 2: The cost of independent calibration on a 3-hop query (FB15k-237). (Left) The naive union bound approach consistently overshoots target recall, underutilizing the global risk budget. (Right) Consequently, excessively permissive thresholds admit cascading false positives, plunging end-to-end precision below 40% for all target recalls.

for the finite size of the calibration data, guaranteeing that the expected risk on unseen queries remains bounded:

$$\mathbb{E}[L(C_{\hat{\lambda}}(x_{n+1}), y_{n+1})] \leq \alpha \tag{4}$$

CRC calibrates a single operator. Extending it to composed query plans without massive precision degradation is the core challenge we address in Sec. 4.1.

3 PROBLEM FORMULATION

We formalize the execution of EPFO queries over a Neural Graph Database as a constrained optimization problem, replacing ad-hoc threshold tuning with a system-managed calibration process. Let $q = \omega_1 \circ \cdots \circ \omega_k$ be a query represented as a DAG of k operators. The execution of the query is governed by a threshold vector $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_k] \in [0,1]^k$, where each λ_i is assigned to a distinct inference operator $\tilde{f}^{(i)}$ in the query DAG (under any fixed logical ordering). To ensure a uniform threshold interface, we assume each inference operator $\tilde{f}^{(i)}$ produces a normalized score in the range $[0,1]$ (e.g., via a softmax or sigmoid activation). We assume queries are drawn from a distribution $\mathcal{D}$. We define the query-level ground truth $Y^{(k)} \subseteq \mathcal{V}$ as the set of entities reachable over the complete graph $\mathcal{G}$, and denote by $\hat{Y}^{(k)}_{\boldsymbol{\lambda}}$ the set of entities retrieved by the final operator in the pipeline under threshold vector $\boldsymbol{\lambda}$.

Risk Constraint. We define the risk function $\mathcal{R}(\boldsymbol{\lambda})$ as the expected end-to-end False Negative Rate (FNR) over $\mathcal{D}$. A threshold vector $\boldsymbol{\lambda}$ is valid if and only if the expected FNR remains below a user-specified risk budget $\alpha \in (0,1)$:

$$\mathcal{R}(\boldsymbol{\lambda}) = \mathbb{E}\left[1 - \frac{|\hat{Y}^{(k)}_{\boldsymbol{\lambda}} \cap Y^{(k)}|}{|Y^{(k)}|}\right] \leq \alpha \tag{5}$$

In a query topology of monotonic operations (selections, projections, joins), errors are asymmetric. A false positive adds downstream overhead but can be filtered by subsequent operators. A false negative is irrecoverable: once a valid tuple is pruned, no downstream join can restore it. By treating FNR as a hard constraint, ConRAD preserves the database expectation of completeness while delegating precision maximization to the cost objective. Equivalently, bounding FNR $\leq \alpha$ guarantees recall $\geq 1-\alpha$. We use both formulations interchangeably, referring to α as the *risk budget* and to $1-\alpha$ as the *recall target.*

Optimization Objective. Among all valid threshold vectors $\boldsymbol{\lambda}$, the optimizer seeks the plan that maximizes result quality. We minimize expected cardinality of the final answer set:

$$C(\boldsymbol{\lambda}) = \mathbb{E}\left[|\hat{Y}^{(k)}_{\boldsymbol{\lambda}}|\right] \tag{6}$$

Cardinality minimization serves as a tractable proxy for precision maximization rather than a strict equivalence: two feasible threshold vectors can, in principle, differ in both true-positive and false-positive counts. We adopt this objective following the CP literature, which commonly characterizes the *informativeness* of a calibrated predictor by the size of its prediction set [4, 55]: smaller sets are more informative under a fixed coverage guarantee. $C(\boldsymbol{\lambda})$ lifts this notion from a single predictor to a multi-operator query plan, minimizing the size of the end-to-end answer set under the recall-coverage constraint. The surrogate is tight in the regime ConRAD operates in: the recall constraint guarantees that the bulk of the ground-truth answers $Y^{(k)}$ are returned, so the TP count is largely fixed by the constraint, and what varies across feasible threshold vectors is the number of non-answers admitted alongside. Minimizing $|\hat{Y}^{(k)}_{\boldsymbol{\lambda}}|$ therefore reduces FPs, aligning closely with end-to-end precision. We confirm this empirically in Section 6.2.

Risk-Constrained Optimization. The objective of our optimization framework is to identify the threshold vector $\boldsymbol{\lambda}$ that minimizes C without violating the risk constraint. Formally:

$$\begin{aligned} \min_{\boldsymbol{\lambda} \in [0,1]^k} \quad & C(\boldsymbol{\lambda}) \\ \text{subject to} \quad & \mathcal{R}(\boldsymbol{\lambda}) \leq \alpha, \quad \alpha \in (0,1) \end{aligned} \tag{7}$$

This formulation differs from classical cost-based query optimization, where all physical plans produce correct and complete results and the optimizer aims to minimize the execution time. In our setting, the answer set itself is a function of $\boldsymbol{\lambda}$: the optimizer must navigate the continuous trade-off between selectivity and coverage, finding the tightest threshold vector that still satisfies the recall target. We treat the risk constraint as hard and non-negotiable in expectation, backed by finite-sample statistical validity over the query distribution; the cost objective is best-effort. Throughout, we assume that test-time queries are exchangeable with the calibration workload. The exact solution of this k-dimensional constrained problem is intractable for any practical optimizer and, more fundamentally, incompatible with the nestedness property required by CRC (Sec. 4.1). Sec. 4 introduces a tractable relaxation that searches a one-dimensional monotone family of threshold vectors; the resulting $\hat{\boldsymbol{\lambda}}$ is optimal along this trajectory and feasible for the original problem, but is not guaranteed to be its global optimum. We discuss this trade-off in detail in Sec. 5.2. We discuss the implications and limitations of this assumption in Sec. 7.

4 THE CONRAD FRAMEWORK

We first show that independent per-operator calibration is provably wasteful, then introduce a scalarization strategy that reduces the k-dimensional threshold search to a single parameter, and finally design the conformal gate operator that dynamically routes between retrieval and inference.

The Pessimism of Independent Calibration. A naive baseline for calibrating a k-operator EPFO query allocates the global risk budget α equally, assigning α/k to each operator and bounding the total risk via the union bound ($\mathcal{R}(\lambda) \leq \sum_{i=1}^{k} \alpha/k$). While statistically valid, the union bound assumes worst-case dependence between

operators, ignoring the correlations between operators. The effect is systematic over-allocation, as each operator receives a more generous threshold than necessary, and the surplus false positives compound across hops. Figure 2 illustrates this for a 3p query topology on FB15k-237 [11]. At a 50% recall target, independent calibration overshoots to 0.66 empirical recall, demonstrating an inability to constrain cardinality growth. At a 90% target, the thresholds provide no meaningful pruning: empirical recall reaches 0.976 while precision collapses to 0.23. This motivates the joint, pipeline-aware calibration strategy we propose next.

4.1 Conformal Calibration for Composed Plans

Calibrating k thresholds jointly is both expensive and theoretically incompatible with CP. A grid search over $[0,1]^k$ is intractable for any practical optimizer. Furthermore, CRC requires that lowering a threshold can only grow the result set (nestedness), but moving in multiple dimensions at once can shrink one operator's output while expanding another's, breaking this property. Our solution is to collapse the search to one dimension: we derive the full threshold vector $\boldsymbol{\lambda}$ from a single scalar parameter $\eta \in [0,1]$. This is a deliberate relaxation. The resulting calibration finds the tightest feasible $\boldsymbol{\lambda}$ along the chosen monotone trajectory rather than the global optimum of Eq. 7; the relaxation is what makes CRC's finite-sample guarantees possible.

4.1.1 Scalarization Strategies. We define a scalarization as a mapping $\mathcal{M} : [0,1] \to [0,1]^k$ that derives the threshold vector $\boldsymbol{\lambda}$ from a single global parameter η. Any component-wise monotonic mapping preserves the nestedness required by CRC, making it statistically valid. A key challenge is that different operators produce confidence scores on incomparable scales. For instance, a score of 0.7 may represent high confidence for one predictor and near-random guessing at another. To resolve this, our scalarization operates in quantile space. Figure 4 illustrates this on a `ip` topology. A single global parameter $\eta = 0.8$ is routed through each operator's empirical quantile function $\hat{Q}_j$, derived from the calibration set, producing per-operator thresholds ($\lambda_1 = 0.82$, $\lambda_2 = 0.77$, $\lambda_3 = 0.91$) that normalize across incomparable score distributions. Each threshold can further be scaled by a per-operator exponent $\gamma_j > 0$:

$$\lambda_j(\eta) = \hat{Q}_j(\eta^{\gamma_j}), \quad j = 1, \ldots, k \tag{8}$$

The quantile function $\hat{Q}_j$ normalizes across operators with different score distributions, ensuring consistent selectivity regardless of raw score range. The exponent γ_j controls how aggressively the j-th operator tightens relative to others: a *loose-early, tight-late* $\boldsymbol{\gamma}$ can preserve recall at initial hops to maximize final precision, while a *tight-early, loose-late* $\boldsymbol{\gamma}$ can aggressively prune intermediate cardinalities, analogous to predicate pushdown. Since any strictly positive $\boldsymbol{\gamma}$ yields a component-wise monotonic mapping, all such choices preserve nestedness and are valid under the conformal prediction framework, as we formally prove below.

4.1.2 Theoretical Validity. We prove that the nested prediction sets required by CRC hold globally across composed query plans for any monotonic scalarization.

Assumption 1 (Pointwise Scoring). *The inference operator $\tilde{f}^{(i)}_{\lambda_i}$ evaluates each candidate triple (h,r,t) independently. The inclusion of entity t depends solely on whether $\phi(h,r,t) \geq \lambda_i$.*

This assumption holds for most standard KG scoring functions [12, 19, 51, 53], which compute $\phi(h,r,t)$ as a function of the triple alone. Each inference operator thus evaluates a per-tuple selection predicate. A candidate is admitted if and only if its score exceeds λ_i, independently of the scores assigned to other candidates. This holds by construction for all threshold-based operators but not for rank-based selections (e.g., top-k).

Theorem 4.1 (Global Monotonicity). *Let q be an EPFO query satisfying Assumption 1. If the configuration $\boldsymbol{\lambda}(\eta)$ is determined by any component-wise non-decreasing scalarization $\mathcal{M}$, then for any $\eta_a < \eta_b$:*

$$\hat{Y}^{(k)}_{\boldsymbol{\lambda}(\eta_b)} \subseteq \hat{Y}^{(k)}_{\boldsymbol{\lambda}(\eta_a)}$$

Consequently, $\mathcal{R}(\boldsymbol{\lambda})$ is non-decreasing in η.

Proof. By induction over the DAG in topological order.
Base case: For leaf operators, the input is a fixed anchor set independent of η. Since $\mathcal{M}$ is component-wise non-decreasing, $\eta_a < \eta_b$ implies $\lambda_j(\eta_a) \leq \lambda_j(\eta_b)$ for all j. By Assumption 1, raising a threshold can only shrink a neural operator's output, so $\tilde{f}^{(j)}_{\lambda_j(\eta_b)}(S,r) \subseteq \tilde{f}^{(j)}_{\lambda_j(\eta_a)}(S,r)$ for any fixed input set S.
Inductive step: Assume the subset property holds for all predecessors of ω_i in topological order. Every EPFO operator is input-monotone: if $A \subseteq B$ then $\omega_i(A) \subseteq \omega_i(B)$, which holds for projection (fewer sources yield fewer targets), intersection ($A_j \subseteq B_j$ implies $\bigcap A_j \subseteq \bigcap B_j$), and union ($A_j \subseteq B_j$ implies $\bigcup A_j \subseteq \bigcup B_j$). For projection operators, the stricter threshold further shrinks the output. Combined, $\hat{Y}^{(i)}_{\boldsymbol{\lambda}(\eta_b)} \subseteq \hat{Y}^{(i)}_{\boldsymbol{\lambda}(\eta_a)}$ holds at stage i, and by induction to the final output $\hat{Y}^{(k)}$.
Risk monotonicity follows: since the ground truth $Y^{(k)}$ is fixed, a smaller prediction set can only capture fewer true positives, so $\mathcal{R}(\boldsymbol{\lambda})$ is non-decreasing in η. □

This nestedness enables tractable calibration. CRC performs a monotone search over η to find the tightest threshold vector satisfying the marginal risk bound $\mathcal{R}(\boldsymbol{\lambda}) \leq \alpha$, with the finite-sample guarantee of Eq. 3 applying directly. All scalarizations in the family of Sec. 4.1.1 are valid by construction. Importantly, the search finds the optimum along a given trajectory but not necessarily the global minimum of $C(\boldsymbol{\lambda})$: an unconstrained optimizer could independently tighten one operator while loosening another, potentially yielding smaller result sets at the same recall level. This is a necessary trade-off for tractability and statistical validity, analogous to heuristic plan-space pruning in classical query optimizers; we discuss it further in Sec. 5.2.

4.2 The Conformal Gate

Invoking a neural model at every hop is expensive and injects unnecessary false positives. We therefore map each logical operator ω_i to a *conformal gate*, a physical operator that dynamically routes between deterministic retrieval $f^{(i)}$ and neural inference $\tilde{f}^{(i)}_{\lambda_i}$ based on the calibrated threshold λ_i.

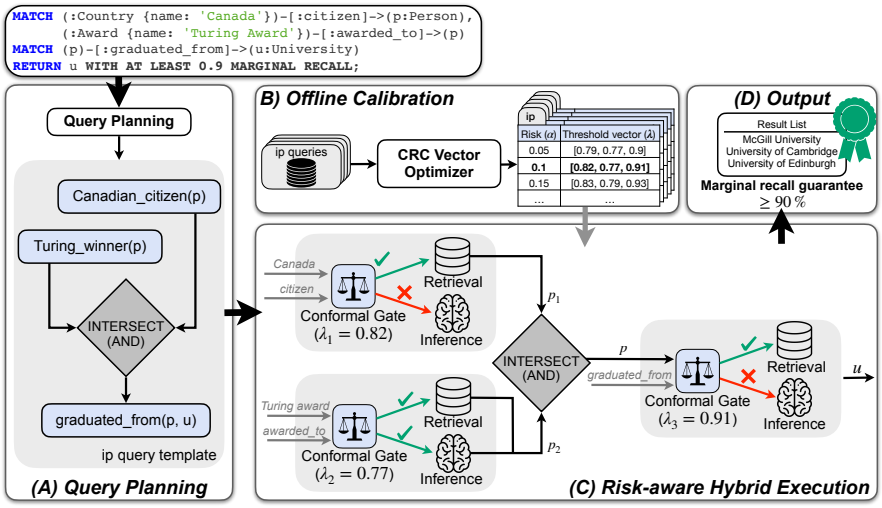

Figure 3: ConRAD overview.

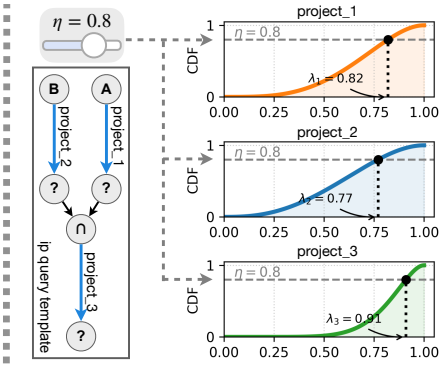

Figure 4: `ip` scalarization example.

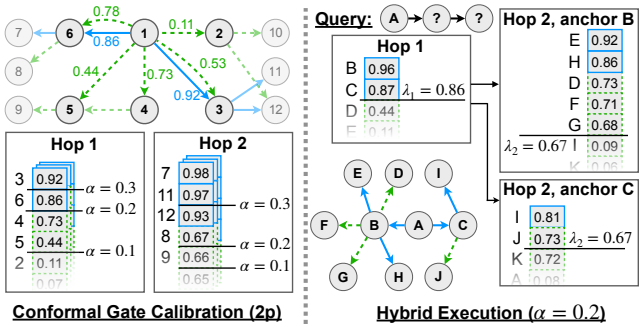

Figure 5: The conformal gate on a two-hop query topology. Left: Calibration maps risk budgets to per-operator thresholds; tighter budgets force inclusion of uncertain neural predictions (green). Right: Online execution at $\alpha = 0.2$ demonstrates adaptive routing. Node A's neighborhood is dense enough for retrieval-only execution (blue), while nodes B and C require hybrid mode to meet the recall target.

For the calibrated threshold λ_i to act as a universal gate, retrieval results (correct by construction) and neural predictions (uncertain) must be scored on a single comparable scale. A naive approach assigning confidence 1.0 to all retrieval facts creates a mass of tied scores, preventing the calibrator from selecting fine-grained thresholds, as it must either accept or reject all retrieval facts at once. We resolve this by partitioning the score space $[0, 1]$ at a routing threshold $\delta \in (0, 1)$:

$$S(\tau) = \begin{cases} \delta + (1-\delta) \cdot \mathcal{H}(\tau) & \text{if } \tau \in f^{(i)}(\hat{\mathcal{G}}) \\ \phi(\tau) \cdot (\delta - \epsilon) & \text{otherwise} \end{cases} \tag{9}$$

where $\tau = (h, r, t)$ is a candidate triple, $\epsilon > 0$ is a negligible numerical margin, and $\mathcal{H}$ is a consistent hash function mapping triples to random scalar values in $[0, 1]$. Retrieval facts receive scores in $[\delta, 1]$, while neural predictions are strictly bounded to $[0, \delta - \epsilon)$. This guarantees that the calibrator prioritizes explicit facts over uncertain predictions. $\mathcal{H}$ spreads retrieval scores uniformly across the upper interval, breaking ties and allowing the calibrator to select thresholds at arbitrary granularity, similar to tie-breaking mechanisms used in standard CP methods [4, 55]. The tie-breaking enables subsampling within the retrieval-only regime to prevent downstream cardinality blowup. Because CRC calibrates over the joint scores, the choice of δ does not affect accept/reject behavior.

At runtime, the relationship between the calibrated threshold λ_i and the routing threshold δ determines the gate's execution mode: *Retrieval-only ($\lambda_i \geq \delta$):* The recall target is satisfied by $\hat{\mathcal{G}}$. The gate executes only $f^{(i)}$ and subsamples retrieval facts where $S(\tau) \geq \lambda_i$, pruning intermediate cardinalities with zero inference cost. *Hybrid ($\lambda_i < \delta$):* Local graph density is insufficient. The gate returns all retrieval facts and invokes $\tilde{f}^{(i)}_{\lambda_i}$ to recover missing answers.

Figure 5 illustrates both modes on a two-hop query topology. At $\alpha = 0.2$, hop 1 calibrates $\lambda_1 = 0.86$: nodes B and C are admitted via retrieval alone. At hop 2, the gate routes adaptively per anchor. Node A's neighborhood is dense enough for retrieval-only execution, while nodes B and C require hybrid mode to meet the recall target. This example illustrates the full ConRAD pipeline: the offline calibration maps α to $\hat{\boldsymbol{\lambda}}$ via the scalarization of Sec. 4.1.1; at runtime, each conformal gate compares λ_i against δ to select its execution mode, producing the final result set with a marginal recall guarantee.

5 IMPLEMENTATION

We now describe the end-to-end implementation of ConRAD, bridging the theoretical guarantees of Sec. 4 with a practical query execution engine. ConRAD is implemented as a middleware layer coordinating a deterministic graph store (exposing a standard traversal API) with an external neural scoring service ($\phi(h, r, t) \to [0, 1]$); either component can be swapped independently. Figure 3 illustrates the workflow. A user submits a query alongside a declarative recall target. The system (A) parses the query into a logical DAG and

Algorithm 1 Calibration Phase (Offline)

Require: Risk budget α, calibration queries $\mathcal{D}_{\text{cal}}$, query topology $\mathcal{T}$ with k inference operators, routing threshold δ, candidate strategies Γ, cardinality objective C
Ensure: Calibrated threshold vector $\hat{\boldsymbol{\lambda}}$

Split $\mathcal{D}_{\text{cal}}$ into disjoint subsets $\mathcal{D}_{\text{opt}}$ and $\mathcal{D}_{\text{valid}}$
$n \leftarrow |\mathcal{D}_{\text{opt}}|$
for each operator $j \leftarrow 1$ to k **do**
  Compute scores on $\mathcal{D}_{\text{opt}}$ ▷ *Eq. 9*
  Compute empirical quantile function $\hat{Q}_j$
best_c $\leftarrow \infty$
for each strategy $\boldsymbol{\gamma} \in \Gamma$ **do** ▷ *Evaluate scalarization strategies*
  for each η in a discretized grid over $[0, 1]$ **do**
    $\boldsymbol{\lambda} \leftarrow [\hat{Q}_j(\eta^{\gamma_j})]_{j=1}^{k}$ ▷ *Scalarization (Eq. 8)*
    for each $(q_i, Y_i) \in \mathcal{D}_{\text{opt}}$ **do**
      Execute q_i under $\boldsymbol{\lambda}$ to obtain $\hat{Y}_{i,\boldsymbol{\lambda}}^{(k)}$
      $L_i \leftarrow 1 - |\hat{Y}_{i,\boldsymbol{\lambda}}^{(k)} \cap Y_i| \,/\, |Y_i|$ ▷ *Per-query FNR*
    $\hat{\mathcal{R}}_\eta \leftarrow \frac{1}{n} \sum L_i$
  $\hat{\eta} \leftarrow \sup\left\{\eta : \frac{n}{n+1}\hat{\mathcal{R}}_\eta + \frac{B}{n+1} \le \alpha\right\}$ ▷ *CRC on $\mathcal{D}_{opt}$*
  $\boldsymbol{\lambda}^* \leftarrow [\hat{Q}_j(\hat{\eta}^{\gamma_j})]_{j=1}^{k}$
  $c \leftarrow$ Evaluate $C(\boldsymbol{\lambda}^*)$ over $\mathcal{D}_{\text{valid}}$
  if $c <$ best_c **then**
    $\hat{\boldsymbol{\lambda}} \leftarrow \boldsymbol{\lambda}^*$; best_c $\leftarrow c$
return $\hat{\boldsymbol{\lambda}}$

maps it to a known query topology, (B) retrieves the pre-computed calibration parameters for that topology and risk budget α, (C) compiles the logical plan into a physical DAG of conformal gates, and (D) returns the result set with a marginal recall guarantee. We detail each stage below, beginning with the offline calibration process that provides the statistical foundation for runtime execution.

5.1 Offline Calibration and Runtime Execution

For each query topology $\mathcal{T}$ and risk budget α, the system determines the threshold vector $\hat{\boldsymbol{\lambda}}$ that satisfies the recall target while minimizing result-set cardinality C. This phase runs entirely offline, decoupled from the query critical path. Algorithm 1 details the procedure, which proceeds in three stages over a calibration dataset $\mathcal{D}_{\text{cal}}$ partitioned into disjoint optimization ($\mathcal{D}_{\text{opt}}$) and evaluation ($\mathcal{D}_{\text{valid}}$) sets. First, for each inference operator j in the topology, we execute the queries in $\mathcal{D}_{\text{opt}}$ over the incomplete graph $\hat{\mathcal{G}}$ and collect both retrieval and inference scores, unified via $S(\tau)$ (Eq. 9). The routing threshold δ is fixed across calibration and online execution, ensuring that $\boldsymbol{\lambda}$ induces identical gate behavior in both phases. From these scores, we compute the empirical quantile function $\hat{Q}_j$ for each operator. Second, we evaluate each candidate scalarization strategy $\boldsymbol{\gamma} \in \Gamma$ (Sec. 4.1.1), a small predefined set of per-operator exponent vectors, over a discretized grid of 100 values $\eta \in [0, 1]$. At each grid point, the scalarization (Eq. 8) maps η to a threshold vector $\boldsymbol{\lambda}$, and the full pipeline is executed over $\mathcal{D}_{\text{opt}}$ to compute the empirical FNR $\hat{\mathcal{R}}_\eta$. The CRC correction (Eq. 3) then identifies the largest η whose corrected risk remains within budget, yielding a valid threshold vector $\boldsymbol{\lambda}^*$ for that trajectory. Third, each valid $\boldsymbol{\lambda}^*$ is evaluated on the held-out set $\mathcal{D}_{\text{valid}}$, and the system retains the threshold vector achieving the lowest cardinality $C(\boldsymbol{\lambda}^*)$. The resulting $\hat{\boldsymbol{\lambda}}$ is stored in a lookup table mapping $(\mathcal{T}, \alpha) \rightarrow \hat{\boldsymbol{\lambda}}$, as shown in Figure 3 (B). This table is computed once per dataset and query topology. Like updating statistics in a traditional DBMS, recalibration is needed only when the graph structure or model weights shift enough to alter score distributions. We quantify this cost in Sec. 6.2 (Table 4). The calibration set requires queries with known ground-truth answers, which can be bootstrapped from historical query logs, expert validation, or synthetic generation over trusted subgraphs.

At query time, the planner retrieves the calibrated threshold vector $\hat{\boldsymbol{\lambda}}$ for the given query topology $\mathcal{T}$ and risk budget α. The logical DAG is compiled into a physical plan of conformal gates $\tilde{f}_{\lambda_1}^{(1)}, \ldots, \tilde{f}_{\lambda_k}^{(k)}$, evaluated in topological order. Each gate compares its operator threshold λ_i against the routing threshold δ to select its execution mode: retrieval-only when local graph evidence suffices, or hybrid when inference is needed to meet the recall target. Intersection and union nodes apply deterministic set operations and introduce no additional uncertainty. The final output $\hat{Y}_{\hat{\boldsymbol{\lambda}}}^{(k)}$ satisfies a marginal guarantee that the expected end-to-end recall remains above $1 - \alpha$.

5.2 Practical considerations

Optimization Tractability. Our formulation seeks the tightest threshold vector that satisfies the recall target. To achieve this, the quantile-space scalarization reduces the k-dimensional threshold space to a one-dimensional search over η, enabling tractable calibration with finite-sample guarantees. However, this is not guaranteed to be the global optimum over the full k-dimensional space, as the scalarization constrains the search to a single monotonic path. An unconstrained search could theoretically find threshold vectors yielding smaller prediction sets at the same recall level by tightening one hop more aggressively while loosening another. While advanced search techniques, such as Bayesian optimization or gradient-based searches, could theoretically explore this high-dimensional space, they face two critical hurdles. First, evaluating arbitrary threshold vectors incurs prohibitive computational costs. Second, and more fundamentally, unconstrained multi-dimensional tuning violates the strict nestedness property required by CRC. We view this as an inherent tradeoff between statistical validity and optimization power; the scalarization sacrifices precision optimality in exchange for rigorous finite-sample guarantees that hold without distributional assumptions. Richer monotone families (e.g., parameterized higher-dimensional surfaces in threshold space that preserve component-wise monotonicity) are a viable middle ground and represent a promising direction for future work.

Compositionality vs. End-to-End Inference. A common alternative in neural graph reasoning is end-to-end inference, where a model predicts a multi-hop destination in a single latent step. While such models can offer superior precision by capturing global dependencies, they function as black boxes that lack the plan transparency and intermediate filtering required by modern DBMSs. ConRAD explicitly favors a compositional approach by decomposing queries into primitive link-prediction operators. This modularity ensures that each step remains interpretable and allows for frontier pruning in dense regions. Importantly, the statistical foundations of

Table 2: Statistics of the KG datasets used in our evaluation.

Dataset	# Entities	# Relations	# Triples
FB15k-237	14541	237	310116
NELL-995	75492	200	154213
YAGO3-10	123182	37	1089040

ConRAD are operator-agnostic. Future neural query optimizers could treat fused multi-hop predictors as individual physical operators within our conformal framework, trading plan granularity for predictive latency while maintaining the same formal guarantees.

6 EXPERIMENTS

We evaluate ConRAD on multi-hop EPFO queries over incomplete knowledge graphs, addressing four research questions:
(RQ1) Validity: Does ConRAD satisfy user-specified recall targets across diverse query topologies and graph incompleteness levels?
(RQ2) Precision: Does ConRAD's risk-constrained optimization achieve precision competitive with best-case static baselines?
(RQ3) Efficiency and Overheads: Does the conformal gate reduce neural invocations by exploiting local graph evidence, and are the offline and online query costs tractable?
(RQ4) Robustness: Are the guarantees preserved when the underlying neural model is swapped?

6.1 Setup

Datasets. We evaluate on three established KG benchmarks spanning distinct structural characteristics (Table 2): FB15k-237 [11] (dense, high degree), NELL-995 [56] (noisy, irregular), and YAGO3-10 [38] (large-scale). We simulate incompleteness by removing 5%, 20%, and 40% of edges uniformly at random. A stratified deletion strategy that concentrates removal in low-degree neighborhoods confirms that ConRAD's validity is preserved under non-uniform incompleteness (details in our repository).
The Neural Predictor. We employ the UltraQuery foundation model [19] as a black-box scoring function ϕ. UltraQuery provides inductive, zero-shot predictions without task-specific fine-tuning, allowing us to isolate ConRAD's calibration efficacy from the underlying model quality. While UltraQuery is capable of end-to-end multi-hop predictions in a single inference step, we explicitly deploy it as a per-hop link predictor. This allows us to evaluate ConRAD's ability to maintain recall guarantees across composed query plans. To test robustness to predictor quality, we also evaluate UltraQuery-T, a variant with different weights and lower predictive accuracy. On the NELL-995 tail-prediction task (i.e., one-hop prediction), UltraQuery-T achieves a significantly lower MRR (48.2% vs. 54.2%) and Hits@10 (67.0% vs. 70.9%) compared to the base model. This lower-fidelity variant tests ConRAD's ability to adaptively adjust thresholds when paired with a weaker predictor, demonstrating that the recall guarantee is preserved through recalibration alone.
Baselines. We compare against three execution strategies. `neo4j` executes queries deterministically over the observed graph $\hat{\mathcal{G}}$, guaranteeing zero false positives but with recall bounded by graph incompleteness. `neural` executes queries hop-by-hop using UltraQuery with fixed, global thresholds ($\theta \in \{0.7, 0.8, 0.9, 0.99\}$), representing the standard workflow of manual threshold tuning without formal guarantees. `hybrid` implements the conformal gate with fixed global thresholds ($\theta \in \{0.4, 0.5, 0.6, 0.7\}$) and routing threshold $\delta = 0.5$. This baseline represents the best a practitioner can achieve by combining retrieval and inference with manual threshold tuning, without offline calibration or per-operator adaptation.
Metrics. We report four indicators, averaged over the evaluation query set. First, the *Empirical Recall* (1 − FNR) is defined as the fraction of ground-truth entities retrieved. Next, *Precision* is defined as the fraction of returned entities that are ground-truth answers. We also report *Average Query Latency* (wall-clock time per query) to characterize the end-to-end cost of the recall guarantee. Lastly, the *Neural Invocations* are defined as the total number of inference operator calls per query, our primary proxy for computational overhead. Queries where the system abstains are assigned zero precision and zero recall.
Calibration Procedure. We employ a split-conformal strategy, executing 5000 queries per topology over the incomplete graph $\hat{\mathcal{G}}$. The workload is partitioned into disjoint sets for CRC calibration ($|\mathcal{D}_{\text{opt}}| = 2000$), cardinality-driven strategy selection ($|\mathcal{D}_{\text{valid}}| = 2000$), and final evaluation (1000). For the most materialization-heavy topologies (`3p`, `ip`, `2u`, and `up` on YAGO3-10), the evaluation set is reduced to 500 queries to bound overhead; the corresponding panels therefore show wider 95% confidence bands. We discretize the scalarization (Section 4.1.1) into a grid of 100 candidate values of η. The candidate strategy set Γ contains six scalarization strategies per query topology, selected on $\mathcal{D}_{\text{valid}}$ by lowest cardinality (cost). Two are neutral: a pooled strategy applying a single shared quantile across all components, and a per-component strategy ($\gamma_j = 1$). For chain topologies, two asymmetric strategies prioritize early vs. late pruning (e.g. $[1.5, 1.0, 0.5]$ and $[0.5, 1.0, 1.5]$ for 3p); for intersection topologies, these reduce to symmetric permissive/tight variants. Two balanced strategies set $\gamma_j = 0.7$ and $\gamma_j = 1.5$ for all j. The calibration is performed independently per query topology.
Frontier Management. During calibration, loose thresholds can require materializing up to $|\mathcal{V}|^3$ candidate trajectories for `3p` queries. To maintain tractability, we restrict intermediate propagation to the union of ground-truth entities and the top-1000 highest-scoring neural candidates per hop. First, because conformal thresholds are determined primarily by ground-truth scores, this pruning preserves the calibration guarantees [4, 55]. Further, valid end-to-end paths can traverse intermediate entities absent from any sub-query's ground truth, necessitating the inclusion of top-scoring neural candidates. We retain 1000 per hop as a practical trade-off between path coverage and materialization cost; an ablation over top-K with $K \in \{10, 100, 1000\}$ confirms that all settings produce valid recall guarantees across all three datasets; precision varies by up to 12 percentage points depending on the dataset and top-K (detailed analysis in our repository). At runtime, calibrated thresholds are applied to all candidates without restriction.
Experimental Scalability. We restrict the calibration and evaluation to queries with intermediate ground-truth frontiers $|Y^{(i)}| \leq 50$, due to the memory overhead of materializing $|Y^{(i)}| \times |\mathcal{V}|$ score matrices on the GPU. An ablation with caps of 100 and 200 on FB15k-237 `3p` confirms that validity and precision are preserved, but the calibration times grows super-linearly (from 38 minutes for

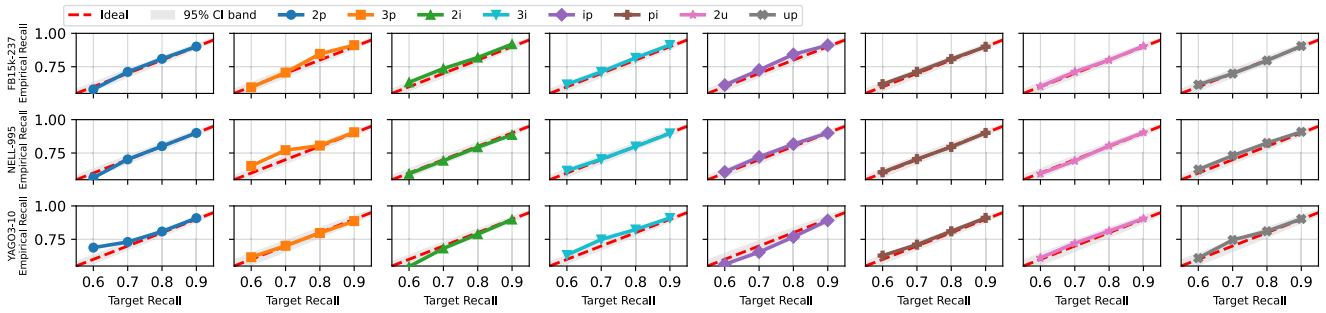

Figure 6: Validity results across query topologies, and datasets under 20% data sparsity. ConRAD provides effective recall control across all tested configurations.

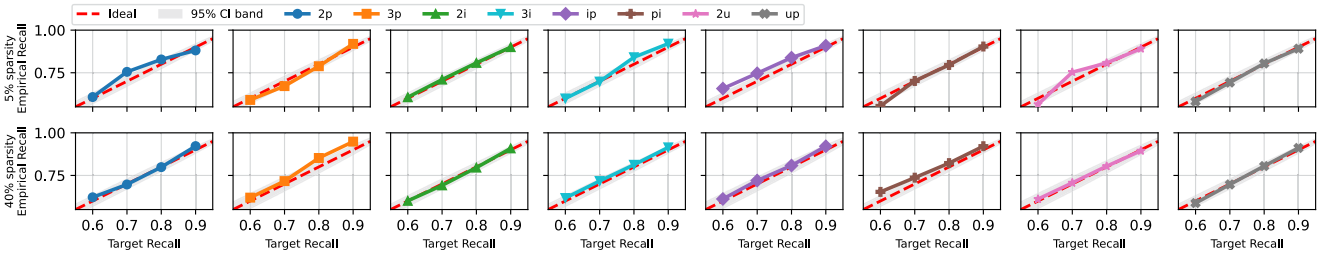

Figure 7: Validity results across query topologies for the FB15k-237 dataset. ConRAD offers effective recall control across all tested data incompleteness levels.

a cap of 50 to 258 minutes for the 200 cap). The results are included and discussed in our repository.

Implementation Details. The graph store is Neo4j v5.20.0, handling all retrieval traversals and set operations (intersection, union). Inference uses a PyTorch deployment of UltraQuery on a single NVIDIA H100 NVL (96GB VRAM) with batch size 64. The routing threshold δ is set to 0.5 in all experiments. The hash function $\mathcal{H}$ in Eq. 9 is a deterministic MD5 hash normalized to $[0, 1]$.

6.2 Results

Our experiments demonstrate four key findings. Additional ablations are available in the linked repository.

(1) ConRAD satisfies declarative marginal recall targets across all the tested settings. Even under severe 40% graph incompleteness and difficult query topologies, the empirical recall tightly tracks the target with a maximum downward deviation of only 0.0547.
(2) ConRAD achieves precision competitive with (and often exceeding) best-case static baselines, reaching up to 96% on FB15k-237 3p queries.
(3) The conformal gate dynamically bypasses neural inference when local graph evidence suffices, reducing invocations to 0% in near-complete graphs at moderate recall targets. ConRAD's latency is comparable to the hybrid baseline at matched recall levels, and increases only as the gate routes more hops through inference to meet stricter targets under higher sparsity.
(4) ConRAD's statistical guarantees are robust to predictor quality. Substituting UltraQuery with UltraQuery-T preserves the recall target across all query topologies (maximum downward deviation: 0.017). Precision degrades proportionally to model quality, confirming that ConRAD provides formal recall guarantees without imposing accuracy requirements on the underlying predictor.

6.2.1 Validity. To answer RQ1, we evaluate whether ConRAD's conformal calibration translates into reliable recall control across datasets, query topologies, and incompleteness levels. Figure 6 reports empirical recall as a function of the recall target (0.6-0.9) at 20% incompleteness across all three datasets and query topologies. ConRAD consistently satisfies the declared risk budget α. Empirical recall tightly tracks or marginally exceeds the target, generally falling within or above the shaded 95% confidence bands (derived from the standard error over the evaluation query set). The maximum upward deviation across all settings was 0.0877 (YAGO3-10, 2p, target 0.6); the maximum downward deviation was 0.0547 (YAGO3-10, 2i, target 0.6). Both are consistent with finite-sample calibration and threshold grid quantization [4, 55]. Across topologies, 2u exhibits the tightest tracking: the parallel union structure provides inherent recall redundancy, allowing the calibrator to select tighter, higher-precision thresholds. The 3p topology is the most difficult, as errors compound sequentially across three hops. Yet, the guarantee holds robustly across all datasets.

Figure 7 isolates the effect of incompleteness by evaluating ConRAD on FB15k-237 under 5% (near-complete) and 40% (high incompleteness) conditions. Even under these extremes, empirical recall

Table 3: Maximum precision subject to empirical recall ≥ 0.60 at 20% incompleteness. For each baseline, we report the best-performing threshold. For ConRAD, we report the risk budget α yielding the highest precision at the same recall floor. *Fails (X)* means the method did not reach 60% recall, where X is the maximum achieved.

Dataset	Method	2p	3p	2i	3i	ip	pi	2u	up
FB15k-237	neo4j	0.86	Fails (0.59)	0.73	Fails (0.52)	Fails (0.57)	Fails (0.54)	0.99	0.80
	neural (best θ)	0.88 ($\theta = 0.7$)	0.82 ($\theta = 0.7$)	0.80 ($\theta = 0.7$)	Fails (0.54)	0.70 ($\theta = 0.7$)	0.72 ($\theta = 0.7$)	0.99 ($\theta = 0.8$)	0.87 ($\theta = 0.7$)
	hybrid (best θ)	0.86 ($\theta = 0.4$)	0.95 ($\theta = 0.4$)	0.73 ($\theta = 0.4$)	Fails (0.52)	0.92 ($\theta = 0.4$)	Fails (0.54)	1.00 ($\theta = 0.4$)	0.80 ($\theta = 0.4$)
	conrad (best α)	**0.96** ($\alpha = 0.9$)	**0.96** ($\alpha = 0.9$)	**0.95** ($\alpha = 0.9$)	**0.90** ($\alpha = 0.9$)	**0.93** ($\alpha = 0.9$)	**0.92** ($\alpha = 0.9$)	**1.00** ($\alpha = 0.9$)	**0.95** ($\alpha = 0.9$)
NELL-995	neo4j	0.73	Fails (0.36)	0.71	Fails (0.51)	Fails (0.57)	Fails (0.58)	0.97	0.75
	neural (best θ)	0.72 ($\theta = 0.7$)	0.70 ($\theta = 0.7$)	0.74 ($\theta = 0.7$)	Fails (0.55)	0.65 ($\theta = 0.7$)	0.65 ($\theta = 0.7$)	0.96 ($\theta = 0.9$)	0.73 ($\theta = 0.8$)
	hybrid (best θ)	0.73 ($\theta = 0.4$)	**0.94** ($\theta = 0.4$)	0.71 ($\theta = 0.4$)	Fails (0.51)	0.90 ($\theta = 0.4$)	Fails (0.58)	0.97 ($\theta = 0.4$)	0.75 ($\theta = 0.4$)
	conrad (best α)	**0.93** ($\alpha = 0.9$)	0.93 ($\alpha = 0.9$)	**0.91** ($\alpha = 0.9$)	**0.88** ($\alpha = 0.9$)	**0.90** ($\alpha = 0.9$)	**0.91** ($\alpha = 0.9$)	**0.99** ($\alpha = 0.9$)	**0.93** ($\alpha = 0.9$)
YAGO3-10	neo4j	0.86	Fails (0.56)	0.71	Fails (0.53)	Fails (0.53)	Fails (0.55)	0.99	0.84
	neural (best θ)	0.72 ($\theta = 0.7$)	Fails (0.56)	0.69 ($\theta = 0.7$)	Fails (0.48)	Fails (0.50)	Fails (0.43)	0.93 ($\theta = 0.7$)	Fails (0.54)
	hybrid (best θ)	0.86 ($\theta = 0.4$)	0.85 ($\theta = 0.4$)	0.71 ($\theta = 0.4$)	Fails (0.53)	**0.85** ($\theta = 0.4$)	Fails (0.55)	0.99 ($\theta = 0.5$)	0.84 ($\theta = 0.4$)
	conrad (best α)	**0.92** ($\alpha = 0.8$)	**0.87** ($\alpha = 0.8$)	**0.91** ($\alpha = 0.9$)	**0.89** ($\alpha = 0.9$)	0.83 ($\alpha = 0.9$)	**0.82** ($\alpha = 0.8$)	**0.99** ($\alpha = 0.8$)	**0.92** ($\alpha = 0.8$)

remains tightly bound to the target. The maximum upward deviation was 0.0573 (ip, 5%, target 0.60), while the maximum downward deviation was 0.045 (pi, 5%, target 0.6). These results confirm that ConRAD's guarantees hold across incompleteness levels. The calibration procedure is agnostic to the degree of incompleteness and requires no manual intervention or prior knowledge of the graph's structural state. By design, the conformal gate adapts selectivity per query based on local graph density rather than uniformly inflating prediction sets. Per-query cardinality analyses confirming this behavior are available in our repository.

6.2.2 Precision. To answer RQ2, we compare ConRAD's precision against static baselines that require manual threshold tuning and provide no formal recall guarantees. Table 3 reports the maximum precision achieved by each execution strategy subject to an empirical recall constraint of ≥ 0.6 at 20% data incompleteness level. For static baselines, we report the best-performing threshold from a coarse grid search. For ConRAD, we report the risk budget α that achieves the same recall constraint with the highest precision.

At 20% incompleteness, deterministic retrieval (neo4j) fails to reach 60% recall on all multi-hop topologies involving intersection (3p, 3i, ip, pi) across all three datasets, because a missing edge at any hop prunes all downstream paths. It succeeds on simpler or redundant topologies (2p, 2i, 2u, up), where shorter paths or parallel structure provide resilience. All methods achieve near-perfect precision on 2u, confirming that union queries do not stress the predictive pipeline.

Conversely, the neural baseline demonstrates why uncalibrated inferences are insufficient for rigorous database querying. While pure inference manages to satisfy the 60% recall constraint on FB15k-237 and NELL-995, it suffers a severe drop in precision. The hybrid baseline reaches significantly higher precision ranges, confirming the value of combining retrieval with inference. However, ConRAD's system-managed recall bounding matches or exceeds this hybrid's exhaustive static thresholding in terms of precision in the large majority of settings. On FB15k-237, ConRAD reaches 0.96 precision on 3p queries compared to 0.95, and 0.95 on 2i queries compared to 0.73. Even on the challenging YAGO3-10 dataset, ConRAD outperforms hybrid on 3p queries (0.87 versus 0.85). Where hybrid leads marginally (e.g., NELL-995 3p, 0.94 vs. 0.93; YAGO3-10 ip, 0.85 vs. 0.83), the gap is narrow. Crucially, the hybrid results

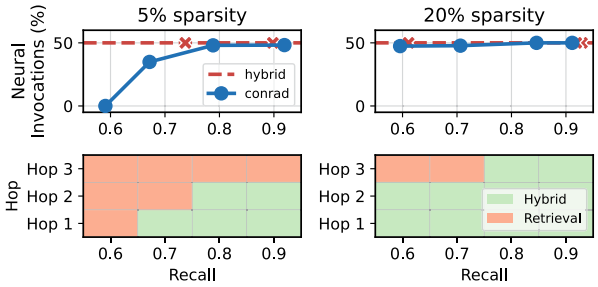

Figure 8: Conformal gate efficiency on FB15k-237 3p. Top: aggregate inference invocations as a fraction of total hops. Bottom: per-hop routing decisions. The gate bypasses inference at early hops where retrieval suffices and switches to hybrid mode at later hops as local evidence degrades.

Table 4: Offline calibration wall-clock time (sec). Each cell reports the Total Time (λ *Optimization only*), where Total = Scores Computation + λ Optimization. The scores computation dominates the overhead.

Dataset	3p	2u	ip
FB15k-237	1302.2 (344)	233.1 (4.5)	750.2 (22.1)
NELL-995	2095.0 (381.6)	603.3 (7.4)	6142.5 (81.4)
YAGO3-10	3711.3 (927.2)	1327.9 (7.7)	4476.7 (91.2)

reflect the best threshold found by a coarse grid search with no guarantee of transfer to unseen queries. ConRAD derives its thresholds automatically from the declared risk budget α and provides a formal recall guarantee over the query distribution.

6.2.3 Efficiency. To answer RQ3, we evaluate whether the conformal gate reduces neural invocations by exploiting local graph evidence. Figure 8 reports the percentage of inference operator calls across the FB15k-237 3p pipeline. The hybrid baseline triggers inference at every hop regardless of local graph density or recall target. ConRAD's conformal gate bypasses inference when retrieval alone satisfies the calibrated threshold. The savings depend on both the risk budget and the incompleteness level. At a recall target of 0.6 under 5% incompleteness, ConRAD bypasses inference entirely, reducing invocations to zero. As the target increases to 0.7 and 0.9, invocation rates rise to 34.9% and 48.2% respectively.

The per-hop routing breakdown reveals where the savings originate. Initial hops often traverse densely connected neighborhoods where retrieval alone satisfies the threshold. At moderate recall targets, the conformal gate operates in retrieval-only mode for the first hop, reducing both the immediate inference cost and the intermediate result set size, which limits false-positive propagation to downstream operators. As execution progresses to later hops, or as incompleteness increases to 20%, local graph evidence degrades and the conformal gate switches to hybrid mode to meet the recall target. Despite bypassing inference at early hops, the aggregate invocation rate is dominated by later hops, which process a larger candidate set due to frontier expansion. At 20% incompleteness, ConRAD's invocation rate plateaus at 50% for recall targets of both 0.8 and 0.9, reflecting this effect. The conformal gate thus concentrates inference where retrieval evidence is insufficient, reducing total model calls while satisfying the recall guarantee.

6.2.4 Overheads. Table 4 reports wall-clock calibration time across datasets and query topologies at 20% incompleteness. The total cost remains tractable, reaching at most approximately 1.5 hours for the largest setting (YAGO3-10 3p). A breakdown reveals that score computation dominates the overhead. On YAGO3-10 2u, it accounts for over 99% of the total time, while the threshold optimization requires only 7.7 seconds. This phase consists entirely of evaluating the neural model over graph triples, a process that can be decoupled from the optimization itself. In a production environment, the system can gather non-conformity scores asynchronously by piggybacking on online query execution. Once a sufficient sample is collected, the optimizer only needs to trigger the lightweight threshold search to produce updated calibration parameters.

At query time, the gate's routing logic requires only computing the unified score via a lightweight, deterministic hash function for retrieving facts and a scalar projection for inference scores, followed by a single floating-point comparison against the calibrated threshold per candidate tuple. This ensures that the formal recall guarantees add only constant-time overhead per candidate triple.

End-to-end query latency depends on how many neural invocations the gate issues. Figure 9 plots target recall against average per-query latency on FB15k-237, 3p. At every sparsity level, static baselines hit hard recall ceilings (e.g., `neural` saturates around 0.4 and `hybrid` around 0.7 under 40% sparsity). ConRAD is the only method that reaches the high-recall regime, trading latency for the formal recall guarantee: at $\alpha = 0.4$ under 5% sparsity, latency is ~55 ms (comparable to `hybrid`), while a target of 0.9 under 40% sparsity costs ~4.5 s as the gate routes most hops through inference.

6.2.5 Robustness. To answer RQ4, we evaluate whether ConRAD's statistical guarantees transfer when the underlying neural model is replaced. We substitute UltraQuery with UltraQuery-T, a less accurate variant of the same architecture, and recalibrate on NELL-995 at 20% sparsity. No other system component is modified. Figure 10 shows that the recall guarantee is preserved across all three query templates, with a maximum upward deviation of 0.0259 (target 0.80, 2u) and a maximum downward deviation of 0.0169 (target 0.70, `ip`). Both are well within the finite-sample variance, as noted also in Sec. 6.2.1. As expected, precision degrades relative to UltraQuery, reflecting the weaker model's noisier score distribution. Maximum precision across templates reaches 0.70 (3p), 0.96 (2u), and 0.66 (`ip`),

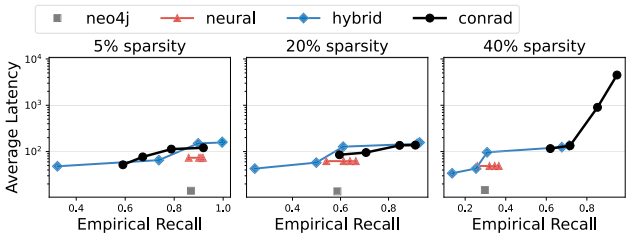

Figure 9: Average query latency on FB15k-237 3p across three sparsity levels. For ConRAD, we show four target recalls: 0.6, 0.7, 0.8, 0.9. The baselines sweep θ over the ranges defined in Section 6.1. ConRAD trades higher latency for recall guarantees that no static baseline can reach.

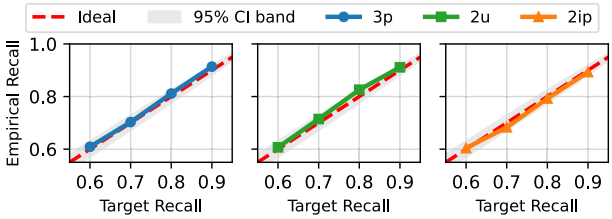

Figure 10: Robustness to predictor quality: ConRAD with UltraQuery-T on NELL-995 (20% sparsity). Despite a weaker underlying model, the recall guarantee is preserved across all three query templates, requiring only offline recalibration. Maximum precision reaches 0.70 (3p), 0.66 (`ip`), and 0.96 (2u), compared to 0.93, 0.9, and 0.99 for UltraQuery under identical conditions (Table 3).

compared to 0.93, 0.9, and 0.99 for UltraQuery under identical conditions (Table 3). This confirms that ConRAD cannot compensate for a fundamentally less accurate predictor, but the safety contract remains intact. Practitioners can therefore upgrade or replace the scoring model and restore guarantees through recalibration alone, without modifying the framework or query plans.

7 DISCUSSION

Marginal Guarantees and Exchangeability. ConRAD provides marginal guarantees, ensuring the recall target is satisfied on average across the query distribution rather than per individual query instance. Achieving strict conditional coverage is provably impossible without restrictive distributional assumptions or infinite calibration data [17]. While this trade-off enables distribution-free guarantees, it fundamentally relies on the exchangeability of calibration and test queries. The premise of exchangeability over non-i.i.d. graph structures has been formally established for Graph Neural Networks [27] and specifically for link prediction when queries are sampled uniformly [57]. However, as we empirically demonstrated in prior work [26], conformal guarantees can degrade under severe distribution shifts (e.g., sudden query skew toward unseen, high-variance relations), a limitation that applies equally to ConRAD's predictive pipelines. Adaptive conformal methods designed for distribution shift, such as Weighted Conformal Prediction [9] and Adaptive Conformal Inference [20], are a promising mitigation strategy.

Zero-Shot Transfer Constraints. ConRAD requires a representative calibration workload for each deployment setting and does not

currently support zero-shot transfer to novel query topologies or out-of-distribution workloads without recalibration. When deploying on a new query template without calibration data, the system must invoke a conservative fallback, such as the union bound alternative described in Sec. 4. However, this limitation does not prevent the modular composition of pre-calibrated branches. If disjoint query branches have already been independently calibrated, they can be safely composed into novel query topologies via deterministic set operators. In such cases, the system could apply structural risk composition rules to primitive sub-plans rather than at the individual operator level, potentially avoiding the severe precision loss of a pipeline-wide fallback. We leave formal analysis of this compositional strategy to future work.

8 RELATED WORK

AI-Integrated Database Systems. Modern data management systems increasingly embed learned components to replace or augment traditional operations, from learned indexes [32] and query optimizers [39] to systems that execute queries over incomplete data. Prior works optimize queries involving expensive ML-based predicates through cascade routing and cost-quality trade-offs (e.g., NoScope [29], BlazeIt [28], VIVA [48], EVAPORATE [8]). Raven [30, 41] presents a unified optimization framework for prediction queries that compose data processing operators with ML inference in a single plan. Neural Graph Databases [10, 24, 25, 45] compose exact graph traversals with neural link prediction, CAESURA [54] integrates ML inference into relational SQL plans, InferDB [49] accelerates in-database inference by approximating ML pipelines through index lookups, and compound AI frameworks such as DSPy [31], LOTUS [43], and Palimpzest [35] orchestrate multi-model workflows over structured and unstructured data. While these frameworks improve expressiveness, orchestration, or latency, they fundamentally lack formal, distribution-free guarantees for end-to-end correctness.

Approximate Query Processing and Uncertainty in Databases. Managing uncertainty natively within the query engine is a foundational database problem. Traditional Approximate Query Processing systems such as BlinkDB [2], VerdictDB [42], and online aggregation [22] provide statistical error bounds, but these quantify sampling variance over completely observed data rather than uncertainty from learned models. A cornerstone of database research, Probabilistic Databases (e.g., MayBMS [7], Trio [3], and Dalvi and Suciu [15]) rigorously propagate tuple-level uncertainty through query evaluation. While foundational, these systems typically model a closed world of explicitly annotated probabilities. In contrast, ConRAD addresses open-world incompleteness, where unobserved facts must be dynamically recovered and bounded using uncalibrated neural scoring. To establish formal guarantees over such black-box models, the database community has recently begun adopting Conformal Prediction. For example, dbET [34] uses CP to bound execution time distributions for cost-based plan selection, Liu et al. [36] apply it to verify learned query optimizers with latency bounds, and ConANN [26] provides recall guarantees for approximate kNN search. While these methods successfully manage model-driven uncertainty in databases, they operate exclusively at the single-operator level or target system performance metrics. ConRAD elevates conformal calibration from isolated operators to the multi-hop pipeline level.

Uncertainty in Knowledge Graph Reasoning. A rich landscape of neural models has been developed for KG reasoning. Link prediction models such as TransE [12], ComplEx [53], and RotatE [51] learn expressive scoring functions for individual triples. Concurrently, neural query embedding frameworks (e.g., Query2Box [46], BetaE [47], NQE [37], GNN-QE [59]) and foundation models (ULTRA [18], UltraQuery [19]) extend this to complex multi-hop logical queries. However, these models optimize for pointwise accuracy and produce uncalibrated confidence scores. Standard post-hoc calibration techniques (e.g., Platt scaling [44], temperature scaling [21]) improve point-wise score reliability but provide no finite-sample guarantees for set-valued outputs. To address this, recent work has applied Conformal Prediction to Graph Neural Networks (CF-GNN [27]) and individual KG link predictors [57, 58], while Bayesian approaches [14] explicitly model predictive uncertainty in evolving graphs. However, these methods calibrate individual operators in isolation. Most closely related, UAG [40] applies Learn Then Test [5], a conformal procedure based on multiple hypothesis testing, to multi-step KG Query-Answering over LLM-generated reasoning paths. ConRAD extends rigorous uncertainty quantification beyond isolated predictors and beyond fixed KGQA pipelines, delivering dynamically routed, end-to-end recall guarantees for composed EPFO query plans.

9 CONCLUSIONS AND FUTURE WORK

ConRAD addresses a fundamental gap in modern data systems by transitioning reliability from a manually tuned hyperparameter to a declarative system constraint. By extending Conformal Risk Control to multi-operator query topologies, we provide the first framework capable of delivering finite-sample recall guarantees for complex queries over incomplete knowledge graphs. Our evaluation demonstrates that ConRAD strictly preserves user-specified recall targets across varying query topologies and incompleteness levels, achieving precision competitive with best-case static baselines. As the industry moves toward composing stochastic AI primitives into broader data processing stacks, ConRAD provides a principled methodology for maintaining the database correctness contract without sacrificing the predictive power of learned models.

Several directions remain open. The cost objective (Sec. 3) can target computational effort rather than cardinality, creating a stochastic analogue to predicate pushdown. The offline calibration can be made continuous by piggybacking score collection onto online query execution and integrating adaptive conformal methods for distribution shift [9, 20]. Finally, empirical validation beyond KGs, including RAG pipelines, learned join operators, and multi-stage retrieval systems, is an important next step.

ACKNOWLEDGMENTS

This work was supported by the Knut and Alice Wallenberg Foundation (WASP NEST Data-bound computing). Vasiliki Kalavri was supported by the National Science Foundation under Grant No. 2237193. Ioanis Kontoyiannis was supported in part by the EPSRC-funded INFORMED-AI project EP/Y028732/1.

REFERENCES

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. 1995. *Foundations of databases*. Vol. 8. Addison-Wesley Reading.

[2] Sameer Agarwal, Barzan Mozafari, Aurojit Panda, Henry Milner, Samuel Madden, and Ion Stoica. 2013. BlinkDB: queries with bounded errors and bounded response times on very large data. In *Proceedings of the 8th ACM European conference on computer systems*. 29–42.

[3] Charu C Aggarwal. 2009. Trio a system for data uncertainty and lineage. In *Managing and Mining Uncertain Data*. Springer, 1–35.

[4] Anastasios N. Angelopoulos and Stephen Bates. 2023. Conformal Prediction: A Gentle Introduction. *Found. Trends Mach. Learn.* 16, 4 (2023), 494–591.

[5] Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. 2021. Learn then Test: Calibrating Predictive Algorithms to Achieve Risk Control. *CoRR* abs/2110.01052 (2021). arXiv:2110.01052 https://arxiv.org/abs/2110.01052

[6] Anastasios Nikolas Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. 2024. Conformal Risk Control. In *ICLR*. OpenReview.net.

[7] Lyublena Antova, Christoph Koch, and Dan Olteanu. 2007. MayBMS: Managing Incomplete Information with Probabilistic World-Set Decompositions. In *ICDE*. IEEE Computer Society, 1479–1480.

[8] Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. 2023. Language Models Enable Simple Systems for Generating Structured Views of Heterogeneous Data Lakes. *Proc. VLDB Endow.* 17, 2 (2023), 92–105. https://doi.org/10.14778/3626292.3626294

[9] Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. 2023. Conformal prediction beyond exchangeability. *The Annals of Statistics* 51, 2 (April 2023). https://doi.org/10.1214/23-AOS2276

[10] Maciej Besta, Patrick Iff, Florian Scheidl, Kazuki Osawa, Nikoli Dryden, Michal Podstawski, Tiancheng Chen, and Torsten Hoefler. 2022. Neural Graph Databases. In *LoG (Proceedings of Machine Learning Research)*, Vol. 198. PMLR, 31.

[11] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*. 2787–2795.

[12] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[13] Payal Chandak, Kexin Huang, and Marinka Zitnik. 2023. Building a knowledge graph to enable precision medicine. *Scientific data* 10, 1 (2023), 67.

[14] Xuelu Chen, Muhao Chen, Weijia Shi, Yizhou Sun, and Carlo Zaniolo. 2019. Embedding Uncertain Knowledge Graphs. In *AAAI*. AAAI Press, 3363–3370.

[15] Nilesh N. Dalvi and Dan Suciu. 2004. Efficient Query Evaluation on Probabilistic Databases. In *VLDB*. Morgan Kaufmann, 864–875.

[16] Vayena Effy, Blasimme Alessandro, and I. Glenn Cohen. 2018. Machine learning in medicine: Addressing ethical challenges. *PLOS Medicine* 15, 11 (11 2018), e1002689. https://doi.org/10.1371/journal.pmed.1002689

[17] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. 2021. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA* 10, 2 (2021), 455–482.

[18] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. 2024. Towards Foundation Models for Knowledge Graph Reasoning. In *ICLR*. OpenReview.net.

[19] Michael Galkin, Jincheng Zhou, Bruno Ribeiro, Jian Tang, and Zhaocheng Zhu. 2024. A Foundation Model for Zero-shot Logical Query Reasoning. In *NeurIPS*.

[20] Isaac Gibbs and Emmanuel Candes. 2021. Adaptive Conformal Inference Under Distribution Shift. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 1660–1672. https://proceedings.neurips.cc/paper/2021/hash/0d441de75945e5acbc865406fc9a2559-Abstract.html

[21] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR, 1321–1330.

[22] Joseph M Hellerstein, Peter J Haas, and Helen J Wang. 1997. Online aggregation. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*. 171–182.

[23] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (Csur)* 54, 4 (2021), 1–37.

[24] Sonia Horchidan. 2023. Query Optimization for Inference-Based Graph Databases. In *PhD@VLDB (CEUR Workshop Proceedings)*, Vol. 3452. CEUR-WS.org, 33–36.

[25] Sonia Horchidan and Paris Carbone. 2023. ORB: Empowering Graph Queries through Inference. In *ESWC Workshops (CEUR Workshop Proceedings)*, Vol. 3443. CEUR-WS.org.

[26] Sonia Horchidan, Fabian Zeiher, Henrik Boström, and Paris Carbone. 2025. ConANN: Conformal Approximate Nearest Neighbor Search. *Proc. VLDB Endow.* 19, 1 (2025), 29–42.

[27] Kexin Huang, Ying Jin, Emmanuel Candes, and Jure Leskovec. 2023. Uncertainty quantification over graph with conformalized graph neural networks. *Advances in Neural Information Processing Systems* 36 (2023), 26699–26721.

[28] Daniel Kang, Peter Bailis, and Matei Zaharia. 2019. BlazeIt: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. *Proc. VLDB Endow.* 13, 4 (2019), 533–546.

[29] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. 2017. NoScope: Optimizing Neural Network Queries over Video at Scale. *Proceedings of the VLDB Endowment* 10, 11 (2017).

[30] Konstantinos Karanasos, Matteo Interlandi, Fotis Psallidas, Rathijit Sen, Kwanghyun Park, Ivan Popivanov, Doris Xin, Supun Nakandala, Subru Krishnan, Markus Weimer, Yuan Yu, Raghu Ramakrishnan, and Carlo Curino. 2020. Extending Relational Query Processing with ML Inference. In *10th Conference on Innovative Data Systems Research, CIDR 2020, Amsterdam, The Netherlands, January 12-15, 2020, Online Proceedings*. www.cidrdb.org. https://vldb.org/cidrdb/2020/extending-relational-query-processing-with-ml-inference.html

[31] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714* (2023).

[32] Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. 2018. The Case for Learned Index Structures. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, Gautam Das, Christopher M. Jermaine, and Philip A. Bernstein (Eds.). ACM, 489–504. https://doi.org/10.1145/3183713.3196909

[33] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 33 (2020), 9459–9474.

[34] Yifan Li, Xiaohui Yu, Nick Koudas, Shu Lin, Calvin Sun, and Chong Chen. 2023. dbET: Execution Time Distribution-based Plan Selection. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–26.

[35] Chunwei Liu, Matthew Russo, Michael J. Cafarella, Lei Cao, Peter Baile Chen, Zui Chen, Michael J. Franklin, Tim Kraska, Samuel Madden, Rana Shahout, and Gerardo Vitagliano. 2025. Palimpzest: Optimizing AI-Powered Analytics with Declarative Query Processing. In *CIDR*. www.cidrdb.org.

[36] Hanwen Liu, Shashank Giridhara, and Ibrahim Sabek. 2025. Conformal Prediction for Verifiable Learned Query Optimization. *Proc. VLDB Endow.* 18, 8 (2025), 2653–2666.

[37] Haoran Luo, Haihong E, Yuhao Yang, Gengxian Zhou, Yikai Guo, Tianyu Yao, Zichen Tang, Xueyuan Lin, and Kaiyang Wan. 2023. NQE: N-ary Query Embedding for Complex Query Answering over Hyper-Relational Knowledge Graphs. In *AAAI*. AAAI Press, 4543–4551.

[38] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *CIDR*. www.cidrdb.org.

[39] Ryan Marcus, Parimarjan Negi, Hongzi Mao, Nesime Tatbul, Mohammad Alizadeh, and Tim Kraska. 2022. Bao: Making Learned Query Optimization Practical. *SIGMOD Rec.* 51, 1 (2022), 6–13. https://doi.org/10.1145/3542700.3542703

[40] Bo Ni, Yu Wang, Lu Cheng, Erik Blasch, and Tyler Derr. 2025. Towards Trustworthy Knowledge Graph Reasoning: An Uncertainty Aware Perspective. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, Toby Walsh, Julie Shah, and Zico Kolter (Eds.). AAAI Press, 12417–12425. https://doi.org/10.1609/AAAI.V39I12.33353

[41] Kwanghyun Park, Karla Saur, Dalitso Banda, Rathijit Sen, Matteo Interlandi, and Konstantinos Karanasos. 2022. End-to-end Optimization of Machine Learning Prediction Queries. In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, Zachary G. Ives, Angela Bonifati, and Amr El Abbadi (Eds.). ACM, 587–601. https://doi.org/10.1145/3514221.3526141

[42] Yongjoo Park, Barzan Mozafari, Joseph Sorenson, and Junhao Wang. 2018. Verdictdb: Universalizing approximate query processing. In *Proceedings of the 2018 International Conference on Management of Data*. 1461–1476.

[43] Liana Patel, Siddharth Jha, Melissa Pan, Harshit Gupta, Parth Asawa, Carlos Guestrin, and Matei Zaharia. 2025. Semantic Operators and Their Optimization: Enabling LLM-Based Data Processing with Accuracy Guarantees in LOTUS. *Proc. VLDB Endow.* 18, 11 (July 2025), 4171–4184. https://doi.org/10.14778/3749646.3749685

[44] John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers* 10, 3 (1999), 61–74.

[45] Hongyu Ren, Mikhail Galkin, Zhaocheng Zhu, Jure Leskovec, and Michael Cochez. 2024. Neural Graph Reasoning: A Survey on Complex Logical Query Answering. *Trans. Mach. Learn. Res.* 2024 (2024).

[46] H Ren, W Hu, and J Leskovec. 2020. Query2box: Reasoning Over Knowledge Graphs In Vector Space Using Box Embeddings. In *International Conference on Learning Representations (ICLR)*.

[47] Hongyu Ren and Jure Leskovec. 2020. Beta embeddings for multi-hop logical reasoning in knowledge graphs. *Advances in Neural Information Processing Systems* 33 (2020), 19716–19726.

[48] Francisco Romero, Johann Hauswald, Aditi Partap, Daniel Kang, Matei Zaharia, and Christos Kozyrakis. 2022. Optimizing Video Analytics with Declarative Model Relationships. *Proc. VLDB Endow.* 16, 3 (2022), 447–460. https://doi.org/10.14778/3570690.3570695

[49] Ricardo Salazar-Díaz, Boris Glavic, and Tilmann Rabl. 2024. InferDB: In-Database Machine Learning Inference Using Indexes. *Proc. VLDB Endow.* 17, 8 (2024), 1830–1842. https://doi.org/10.14778/3659437.3659441

[50] Glenn Shafer and Vladimir Vovk. 2008. A Tutorial on Conformal Prediction. *J. Mach. Learn. Res.* 9 (2008), 371–421.

[51] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197* (2019).

[52] Pedro Tabacof and Luca Costabello. 2020. Probability Calibration for Knowledge Graph Embedding Models. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=S1g8K1BFwS

[53] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*. PMLR, 2071–2080.

[54] Matthias Urban and Carsten Binnig. 2024. CAESURA: Language Models as Multi-Modal Query Planners. In *14th Conference on Innovative Data Systems Research, CIDR 2024, Chaminade, HI, USA, January 14-17, 2024*. www.cidrdb.org. https://vldb.org/cidrdb/2024/caesura-language-models-as-multi-modal-query-planners.html

[55] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. 2005. *Algorithmic learning in a random world*. Springer.

[56] Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In *EMNLP*. Association for Computational Linguistics, 564–573.

[57] Tianyi Zhao, Jian Kang, and Lu Cheng. 2024. Conformalized Link Prediction on Graph Neural Networks. In *KDD*. ACM, 4490–4499.

[58] Yuqicheng Zhu, Nico Potyka, Jiarong Pan, Bo Xiong, Yunjie He, Evgeny Kharlamov, and Steffen Staab. 2025. Conformalized answer set prediction for knowledge graph embedding. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 731–750.

[59] Zhaocheng Zhu, Mikhail Galkin, Zuobai Zhang, and Jian Tang. 2022. Neural-symbolic models for logical queries on knowledge graphs. In *International conference on machine learning*. PMLR, 27454–27478.