Crystalline Intelligence: A Novel Paradigm for Continuous Knowledge Acquisition Without Neural Networks

Nicola Spieser[1]
nicola@spieser.ch

[1]Independent Researcher, Winterthur, Switzerland

February 2026

Abstract

We introduce **Crystalline Intelligence (CI)**, a fundamentally new paradigm for machine intelligence that achieves continuous, autonomous knowledge acquisition without neural networks, gradient descent, or GPU computation. Our reference implementation, AXON, constructs a self-growing knowledge graph from unstructured web content through incremental entity extraction, relation mining, and confidence-weighted temporal learning. Unlike Large Language Models (LLMs), which require billions of parameters and substantial compute for training, CI systems operate on commodity hardware with sub-100 MB memory footprints while continuously expanding their knowledge base. We further present PROMETHEUS, an automated scientific discovery engine built atop the CI framework, capable of identifying structural gaps in knowledge graphs, generating testable hypotheses, and validating them against existing evidence. Experiments on 12 Wikipedia domains demonstrate that axon extracts 9,250+ entities with 190 relations from raw HTML, while Prometheus discovers recurring structural patterns across knowledge domains. We argue that CI represents a complementary—and in resource-constrained settings, superior—approach to knowledge representation compared to parametric neural models.

Keywords: knowledge graphs, continuous learning, automated discovery, non-neural AI, graph intelligence

1 Introduction

The dominant paradigm in artificial intelligence relies on parametric models—neural networks trained on massive datasets via stochastic gradient descent [LeCun et al., 2015]. While remarkably effective, this approach has fundamental limitations:

1. **Catastrophic forgetting**: Neural networks struggle to incorporate new knowledge without degrading previously learned representations [Kirkpatrick et al., 2017].
2. **Computational cost**: Training and inference require specialized hardware (GPUs/TPUs) with significant energy consumption [Strubell et al., 2019].
3. **Opacity**: Knowledge is encoded in billions of opaque floating-point parameters, making it difficult to inspect, verify, or explain [Lipton, 2018].
4. **Static knowledge**: Once trained, models cannot acquire new knowledge without retraining or fine-tuning.

We propose **Crystalline Intelligence (CI)**, a paradigm inspired by the distinction between *fluid* and *crystallized* intelligence in cognitive psychology [Cattell, 1963]. Just as crystallized intelligence in humans represents accumulated knowledge that grows throughout life, CI systems build structured knowledge representations that:

- Grow *continuously* without retraining

- Require *minimal compute* (single CPU core, <100 MB RAM)
- Store knowledge *explicitly* in inspectable graph structures
- Support *temporal dynamics*: confidence reinforcement and forgetting
- Enable *automated discovery* through structural analysis

1.1 Contributions

1. We formalize the **Crystalline Intelligence** paradigm and contrast it with neural approaches (Section 2).
2. We present **axon**, a reference implementation in Rust comprising 8,641 lines of code with 190 tests (Section 3).
3. We introduce **Prometheus**, an automated hypothesis generation and validation engine operating over CI knowledge graphs (Section 4).
4. We provide experimental results demonstrating the system's knowledge acquisition capabilities across 12 scientific domains (Section 5).

2 The Crystalline Intelligence Paradigm

2.1 Formal Definition

A Crystalline Intelligence system $\mathcal{C}$ is defined as a tuple:

$$\mathcal{C} = (G, \mathcal{E}, \mathcal{L}, \mathcal{D}, \mathcal{T}) \tag{1}$$

where:

- $G = (V, E, \phi)$ is a *temporal knowledge graph* with vertices V (entities), edges E (relations), and a confidence function $\phi : V \cup E \to [0, 1] \times \mathbb{R}^+$ mapping each element to a confidence–timestamp pair.
- $\mathcal{E}$: Text $\to \{(v, e)\}$ is an *extraction function* that maps unstructured text to sets of entities and relations.
- $\mathcal{L} : G \times \{(v, e)\} \to G'$ is a *learning operator* that integrates new extractions into the graph with confidence updates.
- $\mathcal{D} : G \to G'$ is a *decay operator* modeling temporal forgetting.
- $\mathcal{T} : G \to \mathcal{H}$ is a *theory operator* that generates hypotheses $\mathcal{H}$ from structural graph analysis.

2.2 Confidence Dynamics

The confidence of an entity or relation x evolves according to:

$$\phi(x, t+1) = \begin{cases} \min(1, \phi(x, t) + \alpha(1 - \phi(x, t))) & \text{if } x \text{ observed at } t \\ \phi(x, t) \cdot e^{-\lambda \Delta t} & \text{otherwise} \end{cases} \tag{2}$$

where $\alpha \in (0, 1)$ is the *reinforcement rate* and $\lambda > 0$ is the *decay constant*. This models two key cognitive phenomena:

- **Reinforcement**: Repeated observation increases confidence with diminishing returns, analogous to spaced repetition in human memory [Ebbinghaus, 1885].
- **Temporal decay**: Unobserved knowledge gradually loses confidence, modeling the forgetting curve.

2.3 Comparison with Neural Paradigms

Table 1: Crystalline Intelligence vs. Neural Networks

Property	**CI**	**Neural**
Knowledge storage	Explicit graph	Implicit weights
Learning	Incremental	Batch training
Forgetting	Controlled decay	Catastrophic
Hardware	CPU only	GPU/TPU
Memory	<100 MB	1–100+ GB
Explainability	Full provenance	Opaque
New knowledge	Instant	Requires retraining
Inference	Graph traversal	Forward pass

3 System Architecture: axon

AXON implements the CI paradigm as a single Rust binary with 13 modules (Table 2).

3.1 Knowledge Graph Storage

The knowledge graph is stored in SQLite with three primary tables:

Table 2: axon module architecture (8,641 LOC total)

Module	Function	LOC
`db.rs`	Knowledge graph (SQLite)	802
`nlp.rs`	Multi-language NLP	1,352
`prometheus.rs`	Discovery engine	1,607
`plugin.rs`	Domain extractors	1,089
`embeddings.rs`	HNSW vector index	892
`tui.rs`	Terminal UI	673
`graph.rs`	Graph algorithms	513
`main.rs`	CLI interface	448
`server.rs`	REST API	342
`export.rs`	Multi-format export	307
`crawler.rs`	Web crawler	234
`query.rs`	Query engine	227
`config.rs`	Configuration	155

```sql
CREATE TABLE entities (
  id INTEGER PRIMARY KEY,
  name TEXT UNIQUE,
  entity_type TEXT,
  confidence REAL DEFAULT 0.5,
  first_seen TIMESTAMP,
  last_seen TIMESTAMP,
  access_count INTEGER DEFAULT 1
);

CREATE TABLE relations (
  id INTEGER PRIMARY KEY,
  subject_id INTEGER REFERENCES
      entities,
  predicate TEXT,
  object_id INTEGER REFERENCES
      entities,
  confidence REAL DEFAULT 0.5,
  source_url TEXT,
  learned_at TIMESTAMP
);

CREATE TABLE facts (
  id INTEGER PRIMARY KEY,
  entity_id INTEGER REFERENCES
      entities,
  key TEXT, value TEXT,
  confidence REAL,
  source_url TEXT
);
```

SQLite was chosen for its zero-configuration deployment, ACID compliance, and proven reliability for datasets up to terabyte scale [SQLite, 2020].

3.2 Multi-Language NLP Pipeline

The extraction function $\mathcal{E}$ operates as a five-stage pipeline:

1. **Language Detection**: Stopword frequency analysis across English, German, French, Italian, and Spanish corpora.
2. **Sentence Segmentation**: Abbreviation-aware boundary detection (handles Dr., Prof., etc.).
3. **Entity Extraction**: Pattern-based recognition of:
   - Proper nouns (capitalization heuristics)
   - Dates (ISO, US, EU, German, French formats)
   - Numbers with units ("750 GB", "3.5 GHz")
   - Currencies ("$1.2M", "€500", "CHF 1'200")
   - URLs and email addresses
4. **Relation Extraction**: Subject–verb–object patterns including passive voice ("was developed by") and appositions ("Berlin, the capital of Germany").
5. **Deduplication**: Levenshtein distance-based entity merging ($d < 3$).

3.3 HNSW Vector Index

For semantic similarity search, axon implements the Hierarchical Navigable Small World (HNSW) algorithm [Malkov and Yashunin, 2018] from scratch in 892 lines of Rust. Entity representations are computed as TF-IDF vectors over their associated text, enabling k-nearest-neighbor queries without external embedding models.

The cosine similarity between entities a and b is:

$$\mathrm{sim}(a, b) = \frac{\mathbf{v}_a \cdot \mathbf{v}_b}{\|\mathbf{v}_a\| \cdot \|\mathbf{v}_b\|} \tag{3}$$

where $\mathbf{v}_x$ is the sparse TF-IDF vector of entity x.

4 Automated Discovery: Prometheus

Prometheus implements the theory operator $\mathcal{T}$ through four stages:

4.1 Pattern Discovery

Algorithm 1 Structural Pattern Mining

Require: Knowledge graph $G = (V, E)$
Ensure: Set of patterns P
$P \leftarrow \emptyset$
for all pairs $(e_1, e_2) \in E \times E$ where $\text{subject}(e_1) = \text{subject}(e_2)$ **do**
  $m \leftarrow (\text{pred}(e_1), \text{pred}(e_2))$ {predicate motif}
  $P[m].\text{freq} \leftarrow P[m].\text{freq} + 1$
end for
for all $v_i, v_j \in V$ where $|\text{neighbors}(v_i) \cap \text{neighbors}(v_j)| \geq 2$ **do**
  if $(v_i, v_j) \notin E$ and $(v_j, v_i) \notin E$ **then**
    $P \leftarrow P \cup \{\textsc{StructuralHole}(v_i, v_j)\}$
  end if
end for
return P

4.2 Hypothesis Generation

From detected gaps, Prometheus generates hypotheses of the form:

$$h = (s, p, o, c, \mathbf{r}) \tag{4}$$

where s is the subject, p the predicted predicate, o the object, $c \in [0, 1]$ the confidence, and $\mathbf{r}$ is the reasoning chain—a sequence of evidence steps leading to the hypothesis.

4.2.1 Type-Based Gap Hypotheses

If entities of type τ commonly exhibit predicate p:

$$\frac{|\{v \in V_\tau : \exists e \in E, \text{pred}(e) = p, \text{subj}(e) = v\}|}{|V_\tau|} > \theta \tag{5}$$

then for any $v^* \in V_\tau$ lacking predicate p, we generate hypothesis $h = (v^*, p, ?, c)$ where c is proportional to the ratio above.

4.2.2 Analogy-Based Hypotheses

Using the HNSW index, if entity a is similar to entity b ($\text{sim}(a, b) > \sigma$), and b has relation (b, p, o) that a lacks, we hypothesize (a, p, o') where o' is the most similar entity to o in a's neighborhood.

4.3 Meta-Learning

Prometheus tracks which pattern types lead to confirmed discoveries:

$$w_p(t+1) = w_p(t) + \eta \cdot (\text{confirmed}_p - \text{rejected}_p) \tag{6}$$

This allows the system to focus on the most productive discovery strategies over time, implementing a form of reinforcement learning without neural networks.

5 Experiments

5.1 Setup

We evaluated axon on 12 Wikipedia articles spanning diverse scientific domains (Table 3). All experiments were conducted on a single Apple M-series CPU core with 8 GB RAM.

Table 3: Knowledge acquisition across 12 domains

Domain	**Entities**	**Relations**	**Facts**
Artificial Intelligence	2,033	22	10
Climate Change	1,961	57	10
Deep Learning	1,424	41	10
Neural Networks	1,492	32	10
Fusion Power	1,463	53	10
CRISPR Gene Editing	1,235	26	10
Machine Learning	1,230	32	10
RISC-V	1,070	6	10
Linux	1,026	22	10
Quantum Computing	984	12	10
Rust (Programming)	602	11	10
Knowledge Graphs	317	3	10
Total (deduplicated)	**9,250**	**190**	**120**

5.2 Performance

Table 4: Resource usage

Metric	**Value**
Total extraction time (12 pages)	48 s
Database size	1.7 MB
Peak memory usage	<50 MB
Binary size (release)	12 MB
Entities per second	~190

5.3 Prometheus Discovery Results

After ingesting all 12 domains, Prometheus identified 4 structural patterns:

- **Frequent subgraph**: The `(is, is)` predicate motif appeared 128 times, indicating dense taxonomic structure across domains.
- **Temporal sequence**: `is` → `is` chains (26 occurrences), suggesting hierarchical classification patterns.
- **Co-occurrence**: `using` predicates cluster (4 motifs), indicating tool/technology relationships.

While no hypotheses were generated at this scale (the graph requires denser cross-domain connections), the pattern detection demonstrates the viability of automated structural analysis.

6 Discussion

6.1 Strengths

Radical efficiency. axon processes 12 scientific domains in 48 seconds on a single CPU core, producing a 1.7 MB knowledge base. An equivalent LLM would require gigabytes of parameters and GPU hours for comparable knowledge coverage.

Perfect transparency. Every fact in the knowledge graph has a source URL, extraction timestamp, and confidence score. There are no hidden representations.

True continuous learning. New knowledge is integrated instantly without any retraining, finetuning, or parameter updates. The daemon mode enables fully autonomous 24/7 knowledge acquisition.

6.2 Limitations

Extraction quality. Pattern-based NLP cannot match transformer-based NER systems in accuracy. Future work will explore lightweight neural extractors as optional components.

Reasoning depth. CI systems excel at relational queries but lack the generative fluency of language models. We view CI and LLMs as complementary, not competing, paradigms.

Scale validation. Our experiments cover 12 domains; larger-scale evaluation across thousands of sources is needed to assess Prometheus's discovery capabilities.

6.3 Future Work

1. **Distributed CI**: Gossip protocol for knowledge sharing between axon instances.
2. **Active learning**: Attention mechanisms to prioritize high-value crawling targets.
3. **Hybrid CI-LLM**: Use CI as a structured knowledge backend for LLM reasoning.
4. **Domain-specific deployment**: Scientific literature mining (PubMed, arXiv).

7 Related Work

Knowledge Graphs. Google's Knowledge Graph [Singhal, 2012], Wikidata [Vrandečić and Krötzsch, 2014], and DBpedia [Auer et al., 2007] are large-scale knowledge bases, but require centralized curation. CI automates the acquisition process.

Open Information Extraction. NELL [Carlson et al., 2010] and OpenIE [Banko et al., 2007] extract knowledge from text but lack the temporal dynamics and discovery components of CI.

Automated Scientific Discovery. BACON [Langley et al., 1987] and more recently AI Scientist [Lu et al., 2024] pursue automated discovery, but rely on predefined hypothesis spaces or LLM generation. Prometheus discovers hypotheses purely from graph structure.

Continual Learning. EWC [Kirkpatrick et al., 2017] and progressive networks [Rusu et al., 2016] address catastrophic forgetting in neural networks. CI avoids the problem entirely through explicit knowledge storage.

8 Conclusion

We have presented Crystalline Intelligence, a novel paradigm for machine intelligence that achieves continuous knowledge acquisition through explicit graph construction rather than parametric learning. Our implementation, axon (8,641 LOC Rust, 190 tests), demonstrates that meaningful knowledge extraction and pattern discovery are possible without neural networks, GPUs, or massive datasets. The Prometheus discovery engine represents a first step toward automated scientific reasoning grounded in structural graph analysis.

We believe CI fills a critical gap in the AI landscape: not every intelligent system needs a billion parameters. Sometimes, a crystal is enough.

Code availability. AXON is open source: `https://github.com/redbasecap-buiss/axon`

References

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a web of open data. In *ISWC*, pages 722–735, 2007.

M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, pages 2670–2676, 2007.

A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, volume 24, pages 1306–1313, 2010.

R. B. Cattell. Theory of fluid and crystallized intelligence: A critical experiment. *Journal of Educational Psychology*, 54(1):1–22, 1963.

H. Ebbinghaus. *Über das Gedächtnis*. Duncker & Humblot, Leipzig, 1885.

J. Kirkpatrick, R. Pascanu, N. Rabinowitz, et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13):3521–3526, 2017.

P. Langley, H. A. Simon, G. L. Bradshaw, and J. M. Zytkow. *Scientific Discovery: Computational Explorations of the Creative Processes*. MIT Press, 1987.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Z. C. Lipton. The mythos of model interpretability. *Queue*, 16(3):31–57, 2018.

C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha. The AI Scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE TPAMI*, 42(4):824–836, 2018.

A. A. Rusu, N. C. Rabinowitz, G. Desjardins, et al. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

A. Singhal. Introducing the Knowledge Graph: things, not strings. *Google Blog*, May 2012.

D. R. Hipp. SQLite: Most widely deployed database engine. `https://sqlite.org`, 2020.

E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in NLP. In *ACL*, pages 3645–3650, 2019.

D. Vrandečić and M. Krötzsch. Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.