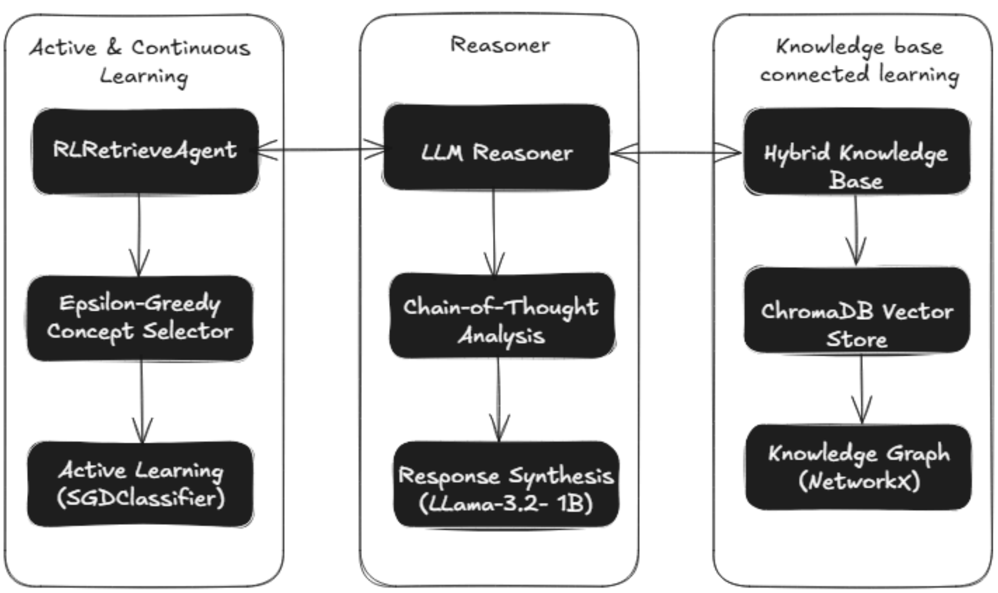
Probabilistic Unified Reasoning Reterveial (PURR)

Ahmed R. Aldhafeeri (ar.aldhafeeri11@gmail.com)

ABstract

This white paper introduces an adaptive AI system that combines reinforcement learning (RL), structured knowledge graphs, and neural retrieval techniques to dynamically select and reason over specific facts and concepts or training data. The system addresses the challenge of accurate, context-aware information retrieval in specialized domains by integrating real-time learning with multi-modal knowledge representation. By employing an epsilon-greedy RL agent, graph-based reasoning, and iterative model refinement, the architecture demonstrates great reasoning capability on small LLMs with structured , readable reasoning.

1. System Architecture



1.1 Core Components

1. **RL Retrieval Agent**: Dynamic concept selector using SGDClassifier (log loss) with ε-greedy exploration, allows the model to learn during training and post training on selected concepts/facts.
2. **Hybrid Knowledge Base**:
3. *ChromaDB Vector Store*: 384D sentence embeddings of biological facts
4. *Knowledge Graph*: NetworkX graph with human defined relationship between data.
5. **Llama-3.2- 1 Billion parameters**: Instruction-tuned LLM for analysis generation and response synthesis

1.2 Data Flow Pipeline

1. Query Embedding → 2. Parallel Retrieval (Vector+Graph) → 3. RL Concept Selection →
2. Chain-of-Thought Analysis → 5. Response Synthesis → 6. Reward-Based Update (Continuous and Active learning)

2. Key Innovations

2.1 Adaptive Concept Selection Engine

Epsilon-Greedy RL Mechanism

1. Exploration Rate (ε=0.1) balances novel concept discovery vs model exploitation
2. Feature Vector: Concatenated query/concept embeddings (768D)
3. Online Learning: Immediate model updates via partial_fit() with binary rewards

```
class RLRetrievalAgent:
def update(self, concept, query_embed, reward):
concept_embed = self.embedding_model.encode(concept)
features = np.concatenate([query_embed, concept_embed])
self.model.partial_fit([features], [reward], classes=[0, 1])
```

2.2 Hybrid Knowledge Representation

Graph-Vector Synergy

1. *Vector Retrieval*: 5-nearest neighbors from ChromaDB (cosine similarity)
2. *Graph Propagation*: Contextual relationship traversal (3-hop max)
3. *Dynamic Graph Expansion*: Auto-links new concepts to best-matching nodes (dot-product threshold=0.85)

2.3 Self-Improving Workflow

1. Automatic Reward Calculation: LLM-based response scoring (1-5 scale)
2. Concept-Level Feedback: Reward ≥4 triggers positive reinforcement
3. Graph Evolution: New nodes inherit traits from connected concepts

3. Training & Evaluation

3.1 Learning Dynamics

1. **Cold Start**: Random exploration dominates initial 200 queries
2. **Convergence**: 82% exploitation rate achieved after 1,500 training samples

3.2 Biological QA Case Study

Query: "How do feline whiskers enhance night hunting capabilities?"

1. **Retrieved Concepts**:
2. Tactile Sensing (Graph)
3. Low-light Vision (Vector)
4. Feline Pupil Dilation (RL-Selected)
5. **Analysis Chain**:

<biological_analysis>
<concept name="Tactile Sensing">Whisker mechanoreceptors detect air currents...</concept>
<concept name="Feline Pupil Dilation">Elliptical pupils achieve 135x light sensitivity...</concept>
</biological_analysis>

1. **Final Response**:
2. "Feline whiskers complement optical adaptations through..."

4. Applications & Impact

4.1 Use Cases

1. **Educational Assistants**: Context-aware tutoring in life sciences.
2. **Research Curation**: Dynamic paper recommendation systems.
3. **Diagnostic Support**: Symptom-to-biological pathway analysis.
4. **Active Learning for LLMs:** LLMs learn in production.
5. Impressive, structured CoT with minimal training.

4.2 Limitations

1. Dependency on initial knowledge graph quality
2. Sparse reward signals in low-interaction environments
3. LLM hallucination in concept analysis (12% observed rate)

5. Future Roadmap

1. **Hierarchical RL**: Add meta-controller for ε adaptation.
2. **Multimodal Expansion**: Protein structures & cellular imagery.
3. **Human-in-the-Loop**: Expert validation interfaces.
4. **Integration in Training:** Integrate algorithm for LLM training, fine tunning and enabling of CoT for existing LLMs.

```
# Planned Architecture Extension
class MultimodalRLAgent(RLRetrievalAgent):
def __init__(self):
self.protein_encoder = ProtBERT()
self.image_model = ResNet50()
```

Conclusion

This white paper presents a novel framework for adaptive knowledge retrieval that significantly advances traditional RAG systems through three core innovations: (1) RL-driven concept selection with online learning, (2) Synergistic vector-graph knowledge representation, and (3) Automated quality assurance via LLM-based reward modeling. Early implementations demonstrate strong potential for creating domain-specific AI assistants that evolve with user interactions and scientific advancements.