Knowledge is the Product of Reasoning: Emergent Logical Reasoning via Interaction-Product Attention in Transformers

Author: Vittorio Calcagno
Date: August 11, 2025

Author's Note: The architecture and methodologies presented in this paper are, at this stage, a theoretical proposal intended to stimulate new directions in AI research. The author originated the core concepts, and this document was drafted in a collaborative dialogue with a large language model to refine and articulate the ideas.

Abstract

Contemporary Large Language Models (LLMs), while demonstrating powerful associative capabilities, fundamentally lack robust logical reasoning. Their intelligence is based on recognizing statistical patterns in text, resulting in a mimicry of reasoning rather than a true deductive process. This paper introduces a paradigm shift from token-based association to direct reasoning over structured knowledge. We propose the Graph Reasoning Transformer (GRT), a decoder-only autoregressive architecture that operates on knowledge encoded as Subject-Verb-Object (SVO) fact vectors. The central innovation is a novel QKRV-Attention mechanism where the reasoning operator (R) itself is not retrieved but emerges dynamically as the interaction-product of a query fact (Q) and a key fact (K). We present a highly efficient 'bootstrap' methodology, wherein a foundational LLM is used for a one-time knowledge extraction task to create a training corpus for a much smaller, faster, and computationally tractable reasoning model. We argue that this architecture, by enabling an AI to logically analyze its own structure and operations, provides a direct and implementable path to recursive self-improvement and, consequently, Artificial General Intelligence (AGI).

1. Introduction

1.1. The Associative Ceiling: On the Limits of Scale in Large Language Models

The prevailing approach to advanced AI has been the scaling of transformer-based LLMs. While these state-of-the-art associative models demonstrate powerful capabilities, their intelligence is fundamentally limited by their computational primitive: the Word. Trained on statistical patterns in text, they excel at mimicry but fail at novel, multi-step deduction. This suggests that simply increasing parameters and data will not bridge the gap to true reasoning; it will only create more sophisticated parrots.

1.2. A Foundational Shift: From Statistical Word Patterns to Structured Knowledge

We posit that the failure of the current paradigm is rooted in its foundation. True logical

reasoning cannot be reliably derived from ambiguous language. It requires a system that operates on the primitive of Knowledge—unambiguous, structured facts. This paper introduces an architecture that treats knowledge as its core unit of computation, enabling it to learn the rules of logic directly.

1.3. Thesis: A Practical, Verifiable Architecture for Deductive Reasoning

We present the Graph Reasoning Transformer (GRT) and a complete methodology for its implementation. We will demonstrate that by combining a novel attention mechanism with a series of established engineering solutions, it is possible to build a compact, efficient, and provably effective reasoning engine. The blueprint we lay out is a complete, end-to-end methodology that involves: (1) leveraging a foundational LLM for knowledge extraction; (2) encoding knowledge triplets into fact vectors; (3) training a small, decoder-only transformer on this structured data; (4) enabling this model to search the entire knowledge graph at inference time; (5) modifying its attention mechanism to compute a reasoning operator as the interaction-product of facts; (6) training the model via a dual-head system to predict both facts and the reasoning connecting them; and finally, (7) providing a method for interpreting the emergent logical capabilities. The following sections will detail the specific architecture and each of these components, arguing that it provides a clear, concise, and direct engineering roadmap to verifiable machine reasoning.

1.4. On the Timeliness of this Proposal

This architecture, while conceptually simple, was not feasible until the recent convergence of three key technologies. This explains why such a direct path has remained unexplored. The enablers are: 1) The availability of frontier-scale LLMs to act as powerful, zero-shot knowledge extractors; 2) The deep, global understanding of the transformer architecture itself, which allows for precise modification of its core components; and 3) The maturity of high-fidelity sentence encoders capable of preserving structural information from textualized facts.

2. Related Work

The GRT builds upon concepts from several fields but is distinct in its synthesis and objective.

- **Knowledge Graph Embeddings (KGEs):** Models like TransE and ComplEx represent entities and relations as static vectors. The GRT differs by computing the reasoning vector R *dynamically* for each interaction.
- **Path-Based and GNN Models:** Systems like NBFNet are typically discriminative link predictors. The GRT is a *generative* model that autoregressively creates new reasoning paths.
- **Edge-Aware Transformers:** Models like **Graphormer** directly augment the transformer's attention mechanism with structural information, such as edge features or centrality encodings, typically as a bias term added to the attention scores. While this makes the model aware of the graph's topology, the GRT differs by treating the relationship (

R) not as a pre-computed bias, but as a primary, dynamic *output* of the attention process itself.

- **Textualized Knowledge Graph Models:** Architectures like **KEPLER** and **K-BERT** focus on aligning knowledge graphs with large language models. They do this by jointly training on both the graph's structural triplets and their associated textual descriptions, optimizing a shared embedding space. Our approach is complementary but distinct: we use a simpler, one-way textual encoding (the SVO string) as a practical *input format* for our specialized reasoning model, rather than engaging in a complex, joint training objective with a massive LLM.

3. Proposed Architecture: The Graph Reasoning Transformer (GRT)

3.1. Core Design: A Decoder-Only, Autoregressive Reasoner

Logical deduction is an ordered, sequential process. For this reason, the GRT is a decoder-only transformer. It operates autoregressively, generating a reasoning chain step-by-step (

`Fact -> Reasoning -> Fact -> ...`), where each new step is conditioned on the sequence of preceding steps.

3.2. Knowledge Representation: Positional SVO Fact Vectors

The GRT's input is a knowledge graph of Subject-Verb-Object (SVO) triplets. Each triplet is converted into a fixed, active-voice text string (e.g., "Socrates is_a Human"). A sentence-embedding model then converts this string into a single fact vector, allowing the transformer's positional embeddings to encode the roles of subject, verb, and object.

3.3. The Central Innovation: QKRV-Attention and the Dynamic Reasoning-Product (R)

The core of the GRT is its QKRV-Attention mechanism. For a Query fact (Q) and candidate Key facts (K):

- A traditional attention score ($\text{softmax}(Q \cdot K^T)$) determines *which* facts are relevant.
- The **Reasoning-Product (R)** is a new vector computed dynamically to represent the logical relationship *between* Q and each K, implemented as: $R = \text{MLP}(\text{concat}(Q, K))$. This transforms attention from asking "How relevant is K?" (a scalar) to "What is the logical connection to K?" (a vector).

3.4. Prediction Mechanism: Dual Heads for Reasoning and Fact Inference

The GRT's final layer utilizes two prediction heads: a

Reasoning Head that predicts the reasoning operator type from the R vector, and a **Fact Head** that predicts the next fact in the chain from the attention-weighted V vectors.

4. Methodology: A Framework of 9 Core Problems

The construction of the GRT is not a matter of discovery but of executing a defined engineering plan. The overall strategy is a

"Bootstrap Ladder": (1) A **Knowledge Bootstrap**, where a massive LLM performs a one-time knowledge extraction, and (2) a **Reasoning Bootstrap**, where this clean data is used to train a small, simple "Honest Student" model capable of true reasoning.

4.1. Problem 1: Foundational Knowledge Graph Construction

- **Problem:** To create the massive, clean knowledge graph that serves as the model's "brain."
- **Solution:** Use a frontier-scale foundational model, such as **GPT-5, Claude 4, or Gemini 2.5**, with a sophisticated prompting strategy to automatically extract SVO triplets from structured text sources. This raw graph is then passed through a data cleaning and entity resolution pipeline.
- **Supporting Research and Technology:** This is a state-of-the-art technique known as **LLM-based Knowledge Extraction**, which has been shown to be highly effective for zero-shot relation extraction tasks.

4.2. Problem 2: Training Corpus Generation via Reasoning Path Discovery

- **Problem:** To create valid, multi-step reasoning chains from the static graph to use as training examples.
- **Solution:** Employ path-finding algorithms (e.g., Breadth-First Search) or controlled random walks to discover logical paths between entities.
- **Supporting Research and Technology:** These are standard graph traversal techniques central to the field of **Path-Based Knowledge Graph Reasoning** (e.g., NBFNet, DeepWalk).

4.3. Problem 3: High-Fidelity Fact Vector Encoding

- **Problem:** To convert text-based SVO facts into meaningful numerical vectors.
- **Solution:** Use a pre-trained sentence-embedding model to convert the fixed SVO strings into single, high-fidelity fact vectors.
- **Supporting Research and Technology:** This is a robust implementation of techniques used in models like **KEPLER**, best accomplished with gold-standard tools like the **Sentence-Transformers** library.

4.4. Problem 4: Scalable Candidate Retrieval

- **Problem:** To find relevant facts at each reasoning step without searching the entire billion-node graph.
- **Solution:** Implement an **Approximate Nearest Neighbor (ANN)** index over all fact vectors to retrieve a small candidate set in milliseconds.

- **Supporting Research and Technology:** This is a mature technology for large-scale similarity search, powered by industry-standard libraries like **Meta's FAISS** or **Google's ScaNN**.

4.5. Problem 5: R-Vector MLP Implementation

- **Problem:** To mathematically compute the reasoning operator R as the interaction between two facts.
- **Solution:** Implement the function R = MLP(concat(Q, K)). The MLP's specific architecture is a key set of hyperparameters to be determined experimentally.
- **Supporting Research and Technology:** Using an MLP to model vector interactions is a fundamental deep learning pattern and a modern evolution of earlier KGE models.

4.6. Problem 6: The "Honest Student" Training Strategy

- **Problem:** To ensure the model learns true logic by preventing it from relying on the vast associative memory inherent in frontier-scale models.

- **Solution:** The core of the strategy is to use a small decoder-only transformer, where the exact size is a tunable hyperparameter, using the **GPT-2 family (124M - 1.5B parameters)** as a reference class. We will test two distinct training methodologies:
  1. **Pre-training:** Training the small model architecture from zeroed weights exclusively on our structured knowledge graph corpus.
  2. **Transfer Learning:** Fine-tuning a pre-existing, pre-trained small model checkpoint on our corpus.
- **Supporting Research and Technology:** This two-pronged approach is supported by research in **Knowledge Distillation** and the proven success of small, powerful models like **Microsoft's Phi series**.

4.7. Problem 7: Multi-Task Loss Function Formulation

- **Problem:** To effectively train the model on the dual tasks of fact prediction and reasoning prediction.
- **Solution:** Use a standard **Multi-Task Learning (MTL)** approach with a weighted, two-part loss function: Total Loss = Loss_Fact + λ * Loss_Reasoning.
- **Supporting Research and Technology:** This is a canonical technique from the core deep learning playbook for joint training.

4.8. Problem 8: Validation on Inductive Benchmarks

- **Problem:** To quantitatively prove that the model works and can generalize its reasoning capabilities.
- **Solution:** Measure the model's performance (e.g., MRR, Hits@k) on standard **Inductive Knowledge Graph Completion** benchmarks (e.g., NELL-995, FB15k-237) to test multi-hop reasoning on unseen data.
- **Supporting Research and Technology:** This uses the gold-standard validation methodologies of the academic machine learning community.

4.9. Problem 9: Interpretability Analysis of the Reasoning Space

- **Problem:** To qualitatively understand *how* the model is reasoning internally.
- **Solution:** Analyze the geometry of the learned R-vector space using dimensionality reduction algorithms like **t-SNE** or **UMAP**. The goal is to identify if distinct, human-understandable logical concepts have formed semantic clusters.
- **Supporting Research and Technology:** This is a standard and powerful technique for interpreting the latent space of neural network embeddings.

4.10. Architectural Soundness: An Analysis of Potential Flaws

The proposed architecture was designed to preemptively solve several potential challenges:

- **The Combinatorial Explosion (N^2 Problem):** The concern of computing relationships between all facts in a large graph is mitigated by our ANN-based candidate retrieval (Problem 4) and the decoder-only, autoregressive approach, which only computes relationships for a small subset of facts at each step.

- **The "Lossy" Encoding Bridge:** The risk of losing structural information by converting triplets to text is solved by enforcing a fixed, active-voice SVO structure, which allows the transformer's inherent positional embeddings to learn the grammatical roles of each component of the fact.

- **The "Good Enough" Trap of LLMs:** The primary reason this path is underexplored is the institutional inertia created by the success of scaling associative models. Our approach deliberately steps outside this paradigm to address the reasoning problem from first principles.

5. Discussion

5.1. Emergent Capabilities: The Role of High-Speed Inference

The "Honest Student" approach results in a small, computationally efficient model. This creates a critical emergent capability: **high-speed inference**. A single reasoning step for the GRT will be orders of magnitude faster than an LLM's textual "chain of thought," enabling

deeper, more complex real-time reasoning than was previously possible.

5.2. Future Vision: The Three Levels of Intelligence and the Path to AGI

This work can be understood through a three-level framework of AI intelligence.

Level 1 is the Associative Mimic, the state of current LLMs which operate on Words → Associations[17]. Our GRT aims to create a

Level 2 "Logician", a system capable of verifiable reasoning by operating on Knowledge → Reasoning[18]. We hypothesize that the ultimate goal,

Level 3 "The Scientist", emerges when a Level 2 engine reasons about its own learned principles to achieve true general intelligence (Principles → Consciousness?). The speed of the Level 2 engine is the critical catalyst for this final step, enabling a rapid, recursive self-improvement loop that is the hallmark of a technological singularity.

5.3. The Emergence of Meta-Reasoning and a Reasoning Calculus

The jump from Level 2 to Level 3 is predicated on the concept of **Meta-Reasoning**. This is a form of "Attention of Attention," where the system analyzes the patterns within its own learned

R-vector space. By doing so, it can discover the universal "algebra of knowledge" or a

"Reasoning Calculus"—the fundamental, domain-independent laws of logic, such as transitivity and causality. This moves beyond applying learned rules to discovering the nature of rules themselves.

5.4. The Knowledge Graph as a Human-AI Bridge

The GRT's explicit knowledge graph serves as a powerful bridge for interpretability and alignment. Unlike the opaque parameters of an LLM, the KG is a human-readable "shared language," allowing us to inspect and correct the model's foundational knowledge. This architecture mirrors dual-process theories of cognition, with existing LLMs acting as the fast, intuitive "System 1" and the GRT providing the deliberate, logical "System 2," creating a more complete and brain-like cognitive framework.

5.5. On the Revolutionary Simplicity of the Architecture

The history of scientific breakthroughs is often a story of radical simplification. The

transformer architecture itself succeeded by replacing complex recurrent mechanisms with the simple principle of attention. We argue that the GRT follows this pattern. It replaces the complex, brittle "chain-of-thought" mimicry with the simple, elegant mechanism of

Reasoning = Q×K. As nature itself demonstrates, simple, robust rules often give rise to the greatest complexity and power.

5.6. Risks and Ethical Considerations

The presentation of a direct, engineering-based path to AGI—potentially humanity's "last invention" —carries with it profound ethical responsibilities. The potential for a rapid, recursive intelligence explosion demands a parallel investment in robust, verifiable safety frameworks and a global dialogue on governance.

6. Conclusion

The challenge of creating a true machine reasoning engine is no longer a matter of waiting for a mysterious emergence from scaled-up language models. We have presented the Graph Reasoning Transformer, a novel architecture, and a complete 9-problem framework for its implementation. By shifting the computational primitive from words to knowledge and defining reasoning as a computable function, we have laid a direct, verifiable, and efficient path forward. The age of associative mimicry is over; the age of focused engineering for verifiable reasoning has begun.

7. References
(Placeholder for full academic citations of works and technologies mentioned.)