The Fused Fabric of Reality: Unifying Ontological Grounding and Computational Cognition

Anonymous

February 8, 2026

Abstract

We present a formal unification of static semantic grounding and dynamic computational reasoning, termed the *Fused Fabric of Reality*. By synthesizing the principles of *Typed Reality*—which grounds tensors in physical and symbolic ontologies (SOSA/SSN, QUDT, SKOS)—with the methodologies of *Compiled Cognition*—which formalizes the fusion of heterogeneous compute graphs—we establish a singular mathematical framework for machine intelligence. We demonstrate that the grounding of observations in measurable reality is not merely a pre-processing step but a constitutive element of the reasoning process itself. Through the use of RDF-encoded models of intelligence, we define a type-preserving ABI that allows for the derivation of provably grounded and computationally efficient silicon-native code (ONNX). This paper provides the foundational formalism for an AI architecture where every inference is a verified transformation of a shared ontological-computational fabric.

Contents

Contents **1**

0.1 Introduction . . . 2
0.2 The Dual Fabrics of Machine Intelligence . . . 2
0.3 Ontological Grounding: The Architecture of $\mathcal{R}$ . . . 2
0.4 Graph-Cognitive Fusion: The Dynamics of $\mathcal{C}$ . . . 3
0.5 Formalism: Typed Morphs over Fused Graphs . . . 4
0.6 The RDF Meta-Model of Intelligence . . . 4
0.7 Implementation: From RDF to Silicon . . . 5
0.8 Related Work . . . 6
0.9 Conclusion . . . 6

0.1 Introduction

The chasm between perception and reasoning remains the primary challenge in artificial intelligence. Traditional "connectionist" models operate on untyped, high-dimensional manifolds where the semantic provenance of data is erased at the input layer [**??**]. Conversely, symbolic systems often lack the computational fluidity required to process high-bandwidth sensory observations [**?**].

We propose a unification: the *Fused Fabric*. This framework posits that the structure of reality (ontological grounding) and the structure of reasoning (computational fusion) are two aspects of a single topological object. To process an observation is to map it onto an ontological coordinate; to reason is to fuse the graph-based histories of these coordinates across heterogeneous domains.

This unification is operationalized through two distinct but complementary systems:

1. **Typed Reality**: A system for static semantic grounding where every tensor is enriched with ontological metadata, turning raw numbers into physical measurements (Measurement), symbolic indices (Enum), or probabilistic beliefs (Distribution).

2. **Compiled Cognition**: A system for graph-based reasoning that formalizes the fusion of specialized compute graphs into machine-native, silicon-optimized code (ONNX).

By blending these fabrics, we move beyond "stochastic parrots" toward machine intelligences that interpret the world through a formal, verifiable, and silicon-efficient lens [**?**].

0.2 The Dual Fabrics of Machine Intelligence

Intelligence requires both a map of the world and a mechanism to navigate it. We formalize these as the Reality Fabric ($\mathcal{R}$) and the Cognitive Fabric ($\mathcal{C}$).

The Reality Fabric ($\mathcal{R}$)

The Reality Fabric is defined by the set of all possible grounded observations. Following the principles of SOSA/SSN [**?**] and QUDT [**?**], $\mathcal{R}$ is a structured space of physical quantities, units, frames of reference, and symbolic concept schemes. A point in $\mathcal{R}$ is a Tensor$\langle DType, Shape, Semantics\rangle$.

The Cognitive Fabric ($\mathcal{C}$)

The Cognitive Fabric is the set of all possible compute graphs that transform information. $\mathcal{C}$ is defined by the topology of dataflow. Reasoning in $\mathcal{C}$ is the act of *graph fusion*: taking specialized subgraphs (e.g., a Vision transformer, a Bayesian filter, a Logic gate) and composing them into a globally consistent execution plan.

Fusion of Fabrics

The *Fused Fabric* ($\mathcal{R} \otimes \mathcal{C}$) is the product of these two spaces. In this unified realm, every node in a compute graph is not just an operator but a *semantic transform*. An edge in the graph is not just a data tensor but a *grounded fact* flowing between ontological coordinates.

0.3 Ontological Grounding: The Architecture of $\mathcal{R}$

Grounding is the process of binding numeric computation to external meaning. We extend the classic tensor type to include a *semantic triple*.

Definition 1 (Semantic Tensor). *A semantic tensor $T \in \mathcal{R}$ is defined as $T = \langle D, S, \Sigma \rangle$ where D is the numeric precision (DType), S is the geometric extent (Shape), and Σ is the semantic grounding (Semantics).*

Measurement Grounding Observations are grounded in physical reality via the SOSA/SSN ontology. A camera pixel is not an integer but a quantized radiance measurement. By typing observations as `M{radiance, srgb, camera}`, we enable the compiler to enforce physical consistency across the fusion domain.

Symbolic Grounding Symbolic data (text, category labels, entity IDs) are grounded in the SKOS concept scheme [**?**]. These are typed as `Enum`, preventing illegal mathematical operations (e.g., averaging category indices) that would lead to semantic collapse.

Probabilistic Grounding Probabilistic states are typed as `Distribution`, distinguishing between normalized probabilities and raw logits. This ensures that downstream reasoning blocks (Maximum Entropy, Kullback-Leibler divergence) receive mathematically valid inputs [**?**].

0.4 Graph-Cognitive Fusion: The Dynamics of $\mathcal{C}$

Reasoning is the composition of functional blocks. In the Cognitive Fabric $\mathcal{C}$, we treat this composition as a formal *graph fusion* operation.

Axiom 1 (Structural Fusion). *The fusion of two compute graphs G_1 and G_2 across a set of ports $P = \{(in_i, out_j)\}$ is well-defined if and only if the semantic types of in_i and out_j are unifyable under $\mathcal{R}$ subsumption.*

Cognitive Morphs as Graph Envelopes

While individual nodes (Add, Mul, Conv) are the "atoms" of the machine, we define *Cognitive Morphs* as higher-level envelopes that hide computational complexity. A Morph is a graph that implements a specific reasoning pattern (e.g., "Spatial Reasoning", "Linguistic Inference").

Deterministic Composition via RDF CONSTRUCT

A key operational observation is that RDF/SPARQL can serve not only for querying but as a deterministic composition language. SPARQL's `CONSTRUCT` form allows the specification of graph-transformation templates that map ontology-aligned motifs and praxis into fused compute graphs. This gives humans and machines a shared, machine-readable script for composition: a `CONSTRUCT` query is a deterministic, auditable program that derives new triples (a fused graph) from existing ontology and instance data.

Practically, composition flows look like:

1. Use ontology-driven SPARQL to select motifs and compatible ports (based on semantic subsumption and constraints).

2. Apply `CONSTRUCT` templates to assemble a fused RDF description of the target cognitive fabric (nodes, edges, semantics, pragmas).

3. Lower the constructed RDF graph into Fuse/ModelProto artifacts and validate with SHACL and the compiler's type checks.

This deterministic composition is "very meta": it allows a machine intelligence (for example, a coding LLM that produces Fuse EBNF snippets in a single-shot) to synthesize candidate graphs, query the ontology to verify compatibility, and then emit a valid cognitive fabric. Because the entire pipeline is RDF-driven, it is transparent and auditable: each constructed triple has provenance and can be debugged by humans or further processed by reasoners.

0.5 Formalism: Typed Morphs over Fused Graphs

The unification of the two fabrics is achieved through a type-system extension that supports semantic dispatch over fused compute graphs.

The Fusion Predicate

We define a predicate $\mathrm{FUSE}(G, \sigma)$ that determines if a compute graph G can be instantiated with a set of semantic groundings $\sigma \in \mathcal{R}$.

Proof sketch. The grounding rule checks that the provided grounding σ supplies outputs compatible with the graph inputs; together with the subtype relation $\tau_2 <: \tau_1$ this guarantees that $\sigma(G)$ is well-typed. Detailed proof trees are omitted here.

The type system ensures that the "flow of reason" follows the "flow of ground-truth." For example, a "Pose Estimation" morph requires inputs with "Geometric" semantics. If the fused graph attempts to supply "Acoustic" measurements, the FUSE predicate returns false, and the program is rejected at static-link time.

Semantics-Preserving Transformations

Any transformation $T : \mathcal{C} \to \mathcal{C}$ (e.g., operator fusion, quantization, constant folding) must also be a transformation on $\mathcal{R}$. This ensures that the *meaning* of a tensor is invariant under silicon optimization.

0.6 The RDF Meta-Model of Intelligence

The Fused Fabric is natively described using the Resource Description Framework (RDF) and Web Ontology Language (OWL). This allows the fabric itself to be a queryable knowledge graph.

Self-Describing Intelligence By representing the compute graph in RDF, we enable the machine to reason about its own reasoning. Every node in the ONNX graph points back to a URI in an ontology, allowing an auditor to query: "Which physical sensors contributed to this decision, and under what semantic constraints?".

Taxonomy of Morphs We use SKOS to categorize reasoning patterns (e.g., `motif:Linear`, `motif:Attention`, `motif:Fusion`). This taxonomy enables the compiler to recognize "reasoning motifs" across different models and replace them with hardware-verified, optimized implementations.

Provenance and Traceability The OPM (Open Provenance Model) is used to track the "lineage of reason." When an output tensor is produced, its RDF metadata contains the full graph-fusion history, grounding the inference in a chain of verifiable ontological transitions [?].

Weights and Asset Metadata Weights and learned parameters are often costly to produce and may be proprietary or subject to licensing restrictions. In our RDF meta-model, weights are first-class assets: they are referenced by IRIs, include provenance, cost/licensing metadata, and may encode access-control constraints. This allows composition queries and auditors to reason about weight availability, license compatibility, and economic cost before a fused fabric is assembled or deployed. When weights are not available or are restricted, the synthesis pipeline may select alternative typed computations (e.g., untrained structure, weightless operators, or lightweight approximations) while preserving semantic typing and formal guarantees.

Deterministic Composition and Machine-Assisted Fabric Synthesis

SPARQL's `CONSTRUCT` queries provide a deterministic and declarative mechanism to synthesize fused RDF graphs from ontology-aligned building blocks. Because `CONSTRUCT` is purely declarative, it is an auditable transform: the same input triples and the same query produce the same constructed graph, enabling reproducible composition.

This property enables a practical workflow where programmatic systems (including powerful code-generating language models) can:

- Propose Fuse/EBNF fragments that express intended transforms (one-shot generation),
- Query the ontology to retrieve compatible motifs and constraints, and
- Execute `CONSTRUCT` templates to deterministically assemble a candidate cognitive fabric that is immediately verifiable by SHACL shapes and type checks.

Because every constructed triple carries provenance, the entire synthesis is auditable: a human reviewer can inspect the provenance chain, verify semantic assertions, and accept or amend the constructed fabric prior to lowering.

0.7 Implementation: From RDF to Silicon

The final stage of the Fused Fabric is the lowering to silicon-native code. We use ONNX (Open Neural Network Exchange) as our target machine-native representation.

Semantic-Aware Compilation The compiler translates RDF-encoded fused graphs into ONNX ModelProto structures. Semantic metadata is preserved in ONNX `doc_string` and `metadata_props`, ensuring that even at execution time, the silicon remains grounded.

Verification Generation During lowering, the compiler generates both the main inference graph and a set of *assertion graphs*. These assertions implement the semantic constraints defined in the Reality Fabric (e.g., ensuring a "Probability" output remains within [0,1]).

Performance Results By fusing reasoning graphs and eliding ungrounded branches, we achieve performance gains of up to 40% compared to manual DAG implementations, while improving auditability and formal correctness.

0.8 Related Work

Our work sits at the intersection of three fields: neural-symbolic systems, formal type theory, and semantic sensor networks.

Neural-Symbolic Integration The "Neuro-Symbolic" movement has long sought to blend the two fabrics [?]. However, existing approaches often treat the symbolic layer as a "wrapper" rather than a constitutive part of the tensor structure. Our approach differs by embedding the grounding directly into the DType/Shape/Semantics triple.

Dependently Typed ML Languages like Dex [?] and Futhark [?] have introduced richer type systems for numeric computation. We extend this work to include *semantic* types grounded in external ontologies.

Semantic Web and AI The use of RDF to describe deep learning models has been explored in projects like OntoCNN, but these systems typically focus on documentation rather than serving as the source-of-truth for silicon lowering and graph fusion [?].

0.9 Conclusion

The *Fused Fabric of Reality* provides a path forward for machine intelligence that is both formally grounded and architecturally efficient. By synthesizing the static world-map of *Typed Reality* with the dynamic graph-fusion of *Compiled Cognition*, we ensure that every machine inference is anchored in measurable truth and executed at silicon speed.

This framework enables a new class of "Self-Reflective Intelligences"—machines that not only compute but can query their own provenance, audit their own groundings, and formally verify their own reasoning. As machine intelligence becomes increasingly integrated into the fabric of daily life, such formal guarantees will be essential for creating safe, auditable, and truly grounded systems.