CORRECTION AND SCOPE

Omnidirectional Tensor Reasoning: A Unified Framework for Grounded, Multi-Field LLMs and Neuro-Symbolic Inference

AI and software theory. The tensor framework is a proposal for multi-field language-model reasoning and neuro-symbolic inference.

Manuscript · p. 1, introduction

Evidence · Live publication library

ChatGPT

Omnidirectional Tensor Reasoning: A Unified Framework for Grounded, Multi-Field LLMs and Neuro-Symbolic Inference

Introduction

Large language models (LLMs) and vision-language models are increasingly being extended beyond text-only inputs. Recent layout-aware document understanding models like **LayoutLM** demonstrate that incorporating spatial and style information along with text yields markedly better performance on tasks like form or receipt understanding [1]. Likewise, the OCR-free **Donut** transformer directly ingests raw document images (text, graphics and layout) and achieves state-of-the-art results on various visual document tasks [2]. These successes illustrate that *synchronized multi-field inputs* – e.g. text content plus layout, image and other metadata – can greatly reduce predictive distortion by providing complementary views of the data.

At the same time, neurosymbolic frameworks have shown how to combine rich, structured knowledge with neural learning. For example, Logic Tensor Networks (LTNs) embed fuzzy first-order logic into differentiable computations, allowing deep nets to answer queries about abstract facts [3]. Similarly, TensorLog compiles logical clauses into differentiable message-passing operations over factor graphs, yielding end-to-end learnable inference networks [4]. Probabilistic Soft Logic (PSL) ties first-order rules to scalable convex inference via soft truth values [5], and other recent work (e.g. **Tensor Logic** by Domingos) shows that **Einstein-summation** ("einsum") over tensors can realize logical inference: logical rules and tensor contractions are formally analogous [6]. These developments hint at a unified "tensor-native" paradigm for reasoning that spans sub-symbolic perception and high-level logic.

In this paper we propose **Omnidirectional Tensor Reasoning**: a unified framework that (1) treats all contextual information (layout, time, gaze, depth, provenance, etc.) as N-dimensional synchronized inputs, (2) performs neuro-symbolic reasoning in a differentiable tensor architecture, and (3) maintains a *redundant logical graph* of knowledge. Our contributions are as follows:

- **Information-Theoretic Guarantee:** Using the Information Bottleneck (IB) and Rate–Distortion (R-D) principles, we show that *noiseless prediction of a target from data requires including all relevant information fields*. In other words, excluding any synchronized input dimension (layout, modality, temporal cues, etc.) forces higher distortion given a fixed representation budget [7] [8].
- **Deduce–Assimilate–Create Pipeline:** We introduce a neuro-symbolic inference pipeline comprising three phases: **Deduce** (perform logic-based inference to propose new facts), **Assimilate** (integrate these facts into the knowledge graph), and **Create** (generate outputs or novel deductions using the enriched graph). This pipeline builds a *logical fabric* of interconnected truths ("knots") and inference chains ("threads"), incrementally updating the model's internal knowledge.
- **Holographic Intelligence:** We define *holographic intelligence* as the property that any local portion of the knowledge graph suffices to reconstruct the whole, due to distributed redundancy – akin to a

hologram where each fragment contains the entire image [9]. We propose a metric **Holo_r** that quantifies the degree of local reconstructability in the graph (e.g. mutual information between subgraphs and global state).

- **Differentiable Logic Layers:** We outline tensor-based implementations of logical inference: (i) **Einstein-summation proofs**, realizing rule application as tensor contractions (per "Tensor Logic" theory [6]); (ii) **Soft (fuzzy) rules**, embedding first-order predicates as real-valued tensor operations (as in LTN/PSL [3] [5]); (iii) **Sparse-matrix logic**, representing knowledge bases as sparse tensors and using differentiable sparse matrix multiplication for multi-hop inference [10]; and (iv) **Field-gated relations**, which gate inference on auxiliary dimensions (e.g. attention masks or learned gates for fields like salience or time) to focus reasoning.
- **System Architecture:** We propose an open-source modular architecture comprising (a) an **N-D Input Encoder** (jointly encoding multiple synchronized fields, e.g. text tokens with layout and time features, images with gaze maps, etc.), (b) a **Semantic Tensor Memory** (a logic-augmented knowledge store, e.g. a tensor of learned embeddings or a differentiable KB) , (c) a **Gated Inference Engine** (which selects relevant subgraphs and applies the differentiable logic layers), and (d) a **Constrained Decoder** (a generator conditioned on retrieved facts, ensuring outputs adhere to logical constraints).
- **Evaluation Plan:** We outline benchmark tasks and ablations to validate the framework. Key metrics include task-specific utility, factuality/hallucination rates, answer *consistency*, and information-theoretic *rate–distortion curves*. We hypothesize that our model will outperform text-only baselines by leveraging additional fields, yielding fewer hallucinations (due to grounded knowledge) and higher consistency. We also propose measuring R–D efficiency by estimating mutual information between inputs and outputs under different compressions.
- **Philosophical Implications:** Finally, we discuss how "tensor-world" models relate to classical problems of symbol grounding and explainability. Our holographic-graph view echoes brain-inspired holonomy ideas [9] and provides an explicit semantics for symbols (each symbolic assertion is grounded in redundant patterns over data fields). By construction, every inference is traceable through tensor operations, offering transparency not present in black-box LLMs. This perspective advances debates on grounded cognition and trustworthiness in AI.

In the following sections we elaborate these ideas with formalism, examples, and references to prior work [1] [3] [7] [6].

Information-Theoretic Justification for Multi-Field Inputs

To understand why *omnidirectional* inputs are necessary, consider predicting a target variable Y from a high-dimensional observation X that actually comprises multiple synchronized subfields $(X_1, X_2, \dots, X_N)$ (e.g. text, layout, gaze, timestamp, depth). By the **Information Bottleneck** principle [8] [7], any representation T used for prediction must trade off between compression $I(X;T)$ and relevance $I(T;Y)$. Concretely, IB optimizes $$\min_{p(t|x)}\; I(X;T) - \beta\, I(T;Y)\;, $$ which is equivalent to a rate–distortion problem where the distortion measures prediction error [8] [7]. In this view, each input field X_i contributes some mutual information about Y. If we drop any field (i.e. compress X to a lower-dimensional $\tilde X$ omitting X_i), the maximum achievable $I(\tilde X;Y)$ can only decrease, forcing higher distortion at fixed $I(\tilde X;Y)$. Thus, to *minimize* prediction distortion given a fixed representation rate, one must include all relevant fields of X [7] [8].

This multi-view IB argument is supported by recent work on multimodal embeddings. For instance, a **Multimodal Information Bottleneck (MIB)** analysis shows that naively fusing modalities can introduce redundancy or ignore useful unimodal cues, degrading prediction [11]. MIB methods maximize $I(T;Y)$ while constraining $I(T;X)$ for *both* the combined and individual modalities, highlighting that each field can carry unique, non-redundant information. In our context, this implies that dimensions like **layout, gaze or provenance** often encode different aspects of semantics that text alone cannot capture. Formally, if X is partitioned into $X = (X_{\mathrm{text}}, X_{\mathrm{layout}}, X_{\mathrm{time}}, \dots)$, then the IB-optimal distortion $D^*(R)$ *(as a function of representation cost $R=I(X;T)$) is a* decreasing *function of N and of the inclusion of each X_i* [7] [8]*. In other words, synchronized multi-field encoding expands the feasible region in the R–D tradeoff, enabling lower distortion for the same compression. Empirically, this principle explains why models like LayoutLM (using text+layout) outperform text-only models* [1]*, and motivates adding* all* available structured inputs to minimize hallucinated or ambiguous predictions.

Deduce → Assimilate → Create: A Neuro-Symbolic Reasoning Pipeline

We propose a three-stage reasoning pipeline that incrementally builds and uses a *logical fabric* of knowledge. At each step, the model performs:

1. **Deduce:** Apply logical rules and inference to current knowledge to derive new candidate facts. This uses differentiable reasoning layers (e.g. tensor logic rules) to "spin" new threads between existing knots. For example, given known predicates, the model infers implied relations (akin to forward-chaining) by performing tensor contractions (Einstein sums) that instantiate rule heads. TensorLog and related methods show how unrolling logic queries yields differentiable functions [4] [6].

2. **Assimilate:** Integrate the newly deduced facts back into the knowledge base, updating the graph of truths. Concretely, this may involve adding new nodes (knot facts) or edges (inference threads) to a symbolic graph that is embedded in tensors. The **Semantic Tensor Memory** serves here like a differentiable knowledge graph: each node's embedding and each relation is stored in tensor form. Learning occurs as we adjust these embeddings so that the inferred facts are consistent (e.g. using soft constraint satisfaction). In effect, Deduce generates hypotheses and Assimilate instantiates them into the memory structure.

3. **Create:** Use the enriched knowledge to answer queries or generate text. In this phase, the model attends to the logical graph and the input fields to produce outputs. Generation is *constrained* by the current knowledge: only statements that can be reconciled with the memory are output (akin to constrained decoding). For example, to answer "What is the author's age?", the model fetches relevant facts and uses differentiable reasoning to compute the answer, then verbalizes it. This creation step often resembles neural generation but with conditioning on the logic fabric.

This Deduce–Assimilate–Create cycle thus continuously weaves a fabric of knowledge: *knots* are the grounded facts in the memory and *threads* are the inference chains between them. Each cycle increases the graph's redundancy and connectivity. By contrast to end-to-end neural "black box" generation, this pipeline maintains explicit structure: all outputs trace back through Deduce/Assimilate steps in the graph. Our design is inspired by cognitive architectures and by recent neural-proof generators, yet it enforces a clear separation of reasoning, memory updating, and generation [4] [12].

Holographic Intelligence and the Holo_r Metric

We define a **holographic intelligence** system as one whose knowledge graph is *locally reconstructable*: any sufficiently large local subgraph contains enough information to recover the global state. In analogy to optical holograms (where every portion of the image encodes the whole) [9], our logical fabric is holographic if information is redundantly encoded across multiple paths. This redundancy ensures robustness (damage to some facts can be compensated by others) and rapid inference (local neighborhoods suffice to answer queries about the broader world).

To quantify this, we propose the **Holo_r** metric: informally, for a given node or subgraph S, let $D(S)$ be how well the remaining graph predicts or reconstructs the truths in S. One definition is $$ \mathrm{Holo}*r(G) \;=\; \frac{1}{|G|}\sum* I(v; N_r(v))\,, $$
where $I(v; N_r(v))$ is the mutual information between a node v and its r-hop neighborhood in graph G. High Holo$_r$ means each local patch has a lot of the information about the whole graph. An equivalent view is that the tensor embeddings of graph neighborhoods have low reconstruction error for missing nodes. In a truly holographic memory, even if many individual facts are removed, the remaining graph (at radius r) suffices to predict them. We will explore precise definitions (choosing r and how to measure I in practice) in future work.

This concept is related to associative/distributed memory models in neuroscience [9]. In our framework, holographic intelligence arises naturally: as Deduce and Assimilate run, facts get entangled in multiple inferences. For example, if “A$\to$B” and “B$\to$C” and “A$\to$C” are all represented, then any one link can be inferred from the others. Holography thus ensures that knowledge is not localized but spread across threads, aligning with the idea that meaning should not reside in a single symbol but in the network of relationships.

Differentiable Logic Layers (Tensor Implementations)

Our system implements logical reasoning as tensor operations, enabling end-to-end learning. Key building blocks include:

- **Einstein-summation proofs:** Following Tensor Logic [6], we represent predicates as multi-way tensors and rules as summation constraints. For example, a rule $\forall x,y: R(x,y)\,\wedge\,S(y,z)\to T(x,z)$ becomes a tensor contraction $T_{i,k} = \sum_j R_{i,j}\,S_{j,k}$ (an einsum). In general, each rule instantiates a small tensor network whose forward pass computes inferred relations. Logical unification and substitution correspond to aligning tensor indices. This realizes symbolic inference purely via batched tensor arithmetic, making proofs differentiable.

- **Soft (fuzzy) rules and continuous truth values:** To allow gradient learning, we use soft logic as in LTN/PSL [12] [5]. Predicates evaluate to [0,1] and logical connectives (AND/OR/Implication) are replaced by differentiable t-norms (e.g. $a\wedge b \approx \min(a,b)$ or a learned smooth approximation). Rule satisfaction becomes a differentiable loss (e.g. minimizing violations of $a \wedge b \le c$). For instance, one can add a penalty $\max(0, R(x,y)+S(y,z)-T(x,z)-1)$ for each grounding. This soft layer ensures that rules influence embeddings without hard boolean thresholds.

- **Sparse-matrix knowledge inference:** We store factual triples (e,r,f) in a sparse tensor M_r per relation r (an adjacency matrix for each predicate). Inference then reduces to sparse matrix multiplies: e.g. multi-hop reasoning is repeated sparse multiplication $v M_{r_1} M_{r_2} \cdots$. Scalable implementations like the Sparse KB method [10] distribute these matrices and allow efficient gradient-based retrieval. A benefit is faithfulness: reasoning steps directly mirror logical graph traversal.

- **Field-gated relations:** Each inference can be gated or weighted by an auxiliary "field" vector. Concretely, we augment tensor operations with learned gates dependent on input features (e.g. time or salience). For example, an inference tensor $R_{i,j}$ might be multiplied elementwise by a gate $g_{i,j}(z)$ parameterized by field z (such as focus of attention). This allows the model to emphasize or suppress certain logical paths based on context (similar in spirit to attention and FiLM layers). While we are not aware of prior work explicitly using field gating in logic inference, it naturally extends ideas of context-aware retrieval [13] to the symbolic domain.

By combining these layers, the model can perform **differentiable logical inference** end-to-end. All operations are tensor (PyTorch/TensorFlow) primitives: einsums, elementwise functions, sparse matmuls, etc. This means traditional neural training (backprop with respect to a loss) can adjust both the symbolic parameters (rule weights, truth values) and sub-symbolic embeddings (entity vectors) simultaneously. In effect, symbolic deduction becomes a learnable component of the deep network, realized in common tensor algebra [6] [12].

System Architecture (Proposed Open-Source Stack)

We envision a modular system, implemented as open libraries, with the following core components:

- **N-D Input Encoder:** A front-end that tokenizes and embeds all input modalities and fields. For example, this could be a Transformer that takes as input a sequence of "tokens" each augmented with vectors encoding word, bounding-box layout, timestamp, salience score, etc. Image-based inputs (layout images or eye-gaze heatmaps) can be processed by vision CNNs whose output tensors align with token positions. Ultimately the encoder produces a unified tensor embedding for every "concept" in the input, with separate channels for each modality. This idea extends multimodal models like LayoutLM [1] and recent multi-sensory Transformers by treating fields (layout, time, depth, etc.) as extra input dimensions.

- **Logic-based Memory (Semantic Tensor Memory):** A differentiable knowledge base storing facts and learned background knowledge. This could be implemented as a set of parameter matrices or high-dimensional tensors. For example, each unary predicate has an embedding vector per entity, and each binary predicate a matrix. Memory may initially be seeded from a knowledge graph or extracted facts; it is continuously updated by the Assimilate step. We propose "tensor memory" in the spirit of TensorLog/PSL: the memory is itself composed of tensors that represent symbolic objects [4] [10]. This module is accessible via tensor queries (subtensor lookup or sparse tensor multiplies).

- **Gated Inference Engine:** A neural-neuro-symbolic core that performs Deduce and Assimilate. It iteratively applies the differentiable logic layers (einsum, soft rules, sparse inference) to the encoded input and memory. Gating modules select relevant predicates or subgraphs: for instance, a small

neural network may score which relations are likely useful given current query context, thus controlling which tensor operations to execute. This is analogous to the differentiable retrieval of DrKIT [13], but extended to symbolic graphs. During training, the engine computes losses for violated rules and missing facts, driving gradient updates.

- **Constrained Decoder:** A generation module (e.g. an LLM or seq2seq model) that produces text or answers conditioned on both the encoder embeddings and the current memory. Crucially, we enforce *constraint decoding*: if the decoder attempts to state a fact not supported by memory, it incurs a penalty or is disallowed. Implementation can use masking or beam search constraints (as in constrained generation literature). The decoder therefore "halls" only that which the memory and logic allow. We can re-use existing libraries (e.g. HuggingFace Transformers) and modify their attention and softmax steps to include memory-based constraints.

These modules communicate via tensor data structures: e.g. the encoder outputs an N×D tensor of features, the memory is a K×K×R tensor of relations, etc. Because everything is differentiable, the entire pipeline can be trained end-to-end or in stages. We envision open-source reference implementations (in PyTorch/JAX) with clear APIs: one module for input encoding, one for memory graph management, one for logic inference, and one for generation/decoding.

Experimental Benchmarks and Ablation Plans

To empirically validate Omnidirectional Tensor Reasoning, we propose the following evaluation strategy:

- **Utility Gains:** Test on tasks requiring multimodal grounding and reasoning (e.g. document QA, visual question answering with textual overlays, temporal document understanding). For each task, compare our model against a baseline LLM that uses only text (or text+OCR) and against ablations that remove one or more input fields. We expect the full N-D model to have significantly higher accuracy or F1 due to richer context [1] [2].

- **Hallucination Reduction:** Use factuality benchmarks (e.g. TruthfulQA, or synthetic QA with known answers) to measure hallucination rate. We will track the percentage of outputs that assert unsupported facts. Our hypothesis is that grounding answers in the semantic memory will reduce unsupported statements, aligning with findings that augmenting LLMs with knowledge (via retrieval or rules) improves factuality [13]. Specifically, ablate by disabling the Constrained Decoder to see how hallucinations increase.

- **Consistency Improvement:** Evaluate *internal consistency* by testing logically equivalent questions or adding known facts as premises. For example, if the system knows "All A are B" and is asked "Is X an A?" versus "Is X a B?", consistent reasoning should agree. We will measure consistency scores as in recent benchmarks (see e.g. consistency metrics in QA literature). We predict the logical fabric will enforce consistency where the raw LLM often does not.

- **Rate–Distortion Superiority:** Quantify information-theoretic performance by estimating the mutual information $I((X_1,\dots,X_N);Y)$ captured by different models. One can approximate this by computing the IB tradeoff curve: vary a regularization parameter to get different compression levels and measure prediction error. We expect that including more fields shifts the curve downward (lower

distortion for a given rate) consistent with theory [7]. Practically, we can do this by training with noise or bottleneck constraints on the latent representation and plotting accuracy vs. information.

- **Ablation Studies:** Systematically remove or "black out" each input dimension (e.g. remove layout features, remove provenance info) to see its marginal contribution. Also test variants of the Deduce–Assimilate pipeline (e.g. skip the Deduce step) to demonstrate its effect on performance. These ablations will highlight which components drive gains and verify that the model indeed leverages all dimensions as intended.

Throughout, all methods and dataset splits will be openly shared. We will report statistically robust results (means ± variance) to support claims of improved utility and reduced hallucinations. In particular, we aim to show not only raw performance gains but *qualitative* improvement in the structure of answers (e.g. logical coherence), illustrating the advantage of our neuro-symbolic integration.

Philosophical Implications: Symbolic Grounding and Explainability

By construction, Omnidirectional Tensor Reasoning offers a new perspective on age-old philosophical questions in AI. **Symbol Grounding:** Our approach grounds symbols in a *tensorial world model* that includes all modalities. In other words, each "symbol" (like a word or concept) is linked via the memory tensor to sensory and contextual data. This directly addresses Harnad's symbol-grounding problem [14]: meaning is not just other symbols, but is anchored in the redundant logical fabric of real observations (layout, images, contexts).

Holographic World-Model: The notion of holographic intelligence bridges computational and cognitive theories. Just as Pribram proposed that brain memory might operate holographically [9], our model distributes every fact across many dimensions and relations. This suggests a path toward AI whose "memories" are inherently robust and distributed, potentially mirroring aspects of human episodic knowledge. The metric Holo_r gives a formal handle on this holography, unlike standard black-box models.

Explainability and Transparency: In standard LLMs, reasoning is implicit and often inscrutable. Here, every inference is a (differentiable) logic step: we can trace an output back to specific deduced facts and tensor operations. The separation into Deduce/Assimilate/Create means the model has explicit intermediate representations (the memory graph) which can be inspected. For instance, if the model answers a question, we can examine which "knots" and "threads" it used. This satisfies a strong notion of explainability: the network's use of each symbol is encoded in the architecture itself.

AGI Context: In the quest for more general intelligence, our framework suggests combining "symbolic" and "subsymbolic" seamlessly. It points to an agent that continually builds a logical model of its observations in rich context, arguably an essential step toward human-like understanding. Of course, many questions remain (e.g. how to learn the rules themselves, or how to scale to world-sized knowledge). But by explicitly integrating rate-distortion-optimal multi-view perception with symbolic inference, we lay conceptual groundwork for AGI that is robust, grounded, and intelligible.

Overall, this work highlights that *tensors* – the bread and butter of deep learning – can directly implement logical reasoning when augmented with information-theoretic and architectural principles. By making inputs and memories truly N-dimensional, we align machine perception with the complexity of the real

world, while retaining the rigor of formal reasoning. The result is a promising step towards AI systems that are both powerful learners and clear thinkers [6] [12].

Acknowledgments: We thank the research community for prior work (cited) on multimodal learning, neuro-symbolic reasoning, and information theory that inspired this unified vision.

---

[1] [1912.13318] LayoutLM: Pre-training of Text and Layout for Document Image Understanding
https://arxiv.org/abs/1912.13318

[2] [2111.15664] OCR-free Document Understanding Transformer
https://arxiv.org/abs/2111.15664

[3] [12] Logic Tensor Networks
https://arxiv.org/pdf/2012.13635

[4] [1605.06523] TensorLog: A Differentiable Deductive Database
https://arxiv.org/abs/1605.06523

[5] Probabilistic soft logic - Wikipedia
https://en.wikipedia.org/wiki/Probabilistic_soft_logic

[6] [2510.12269] Tensor Logic: The Language of AI
https://arxiv.org/abs/2510.12269

[7] arXiv:physics/0004057 v1 24 Apr 2000
https://www.princeton.edu/~wbialek/our_papers/tishby+al_99.pdf

[8] Information bottleneck method - Wikipedia
https://en.wikipedia.org/wiki/Information_bottleneck_method

[9] karlpribram.com
https://www.karlpribram.com/wp-content/uploads/2025/08/T-167.pdf

[10] openreview.net
https://openreview.net/pdf?id=BJlguT4YPr

[11] [2210.17444] Multimodal Information Bottleneck: Learning Minimal Sufficient Unimodal and Multimodal Representations
https://arxiv.org/abs/2210.17444

[13] [2002.10640] Differentiable Reasoning over a Virtual Knowledge Base
https://arxiv.org/abs/2002.10640

[14] Symbol grounding problem - Wikipedia
https://en.wikipedia.org/wiki/Symbol_grounding_problem