ChatGPT

Hybrid Symbolic–Geometric Reasoning in AI

Introduction

Hybrid symbolic–geometric reasoning refers to AI approaches that combine **symbolic logic** (discrete, rule-based reasoning) with **geometric deep learning** (continuous vector representations and learning). The motivation is to build systems that can perform **logical, consistent reasoning** while also making **creative associative leaps** from data. A key challenge is enabling an AI to **hold contradictory ideas yet remain coherent**, as in paraconsistent or "LOL" logic. *Paraconsistent logic* allows reasoning with inconsistent information without exploding into triviality – unlike classical logic, a contradiction doesn't entail *everything* is true [1]. This means an AI could entertain conflicting hypotheses or nuanced perspectives and still draw sensible conclusions. Recent neurosymbolic research seeks to realize this vision by merging *both/and* reasoning: blending **vector-based knowledge (for flexibility and intuition)** with **symbolic rules (for global consistency)**.

Paraconsistent Reasoning ("LOL Logic")

Traditional logic systems explode when faced with a contradiction: if one fact is both true and false, anything can follow. **Paraconsistent logics** avoid this by weakening the explosive inference rules [1]. In practice, this might let an AI **preserve multiple conflicting beliefs** and reason about them without collapsing into incoherence [2]. Implementations of paraconsistent reasoning in AI often involve multi-valued logics or truth-value *bounds*. For example, IBM's **Logical Neural Networks (LNN)** use *weighted real-valued logic* neurons and specifically include a loss term for logical contradictions, giving the model **resilience to inconsistent knowledge** [3]. LNNs can maintain *bounds on truth values* (supporting an open-world assumption), meaning a statement can be "possibly true" without being firmly true or false [4]. This enables an AI to handle uncertainty or temporary contradictions until more evidence arrives. In summary, paraconsistent or contradiction-tolerant methods are a cornerstone for any AI that must **"not fall to pieces"** when faced with paradoxes or dilemmas – it can represent "both A and ¬A" as a manageable state rather than a fatal error.

Neurosymbolic AI Approaches

Neurosymbolic AI is the broad field integrating neural networks with symbolic reasoning. Over the past 2–3 years, there have been significant advances in this area, aiming to get the best of both worlds: robust learning from data and rigorous logical reasoning [5] [6]. Below we outline key approaches and systems:

Differentiable Logic Networks

One line of research makes logic *differentiable* so it can train with gradient-based methods. For instance, **Logic Tensor Networks (LTN)** define a *fully differentiable first-order logic* called *Real Logic*, grounding logical symbols in real-valued vectors and using fuzzy semantics [7]. In an LTN, truth values are in [0,1] and logical formulas become continuous functions; learning adjusts embeddings to maximize formula satisfiability.

Similarly, IBM's **Logical Neural Networks (LNN)** represent each neuron as part of a logical formula (with weighted truth values) and perform inference akin to theorem proving, but in a network that trains end-to-end [3]. These networks use *soft versions* of AND, OR, NOT so that logical consistency becomes a differentiable loss term. Crucially, such frameworks maintain interpretability (each neuron or embedding has a semantic meaning) and can integrate prior knowledge as constraints. They have shown success in tasks like multi-label classification, knowledge base completion, and reasoning under uncertainty [7].

Another example is **Neural Theorem Provers (NTPs)**, which map logic proofs into neural computations. NTPs replace discrete unification with *soft unification*: symbols (predicates and entities) are embedded as vectors, and unification is judged by vector similarity (e.g. cosine distance) [8]. Layers of neural modules imitate logic operators (AND as a product or min, OR as sum or max, etc.), making the *entire proof search differentiable* [9] [10]. Rocktäschel et al.'s *end-to-end differentiable prover* demonstrated that one can learn first-order logic rules from data by gradient descent, achieving interpretable rules and multi-hop reasoning. These systems allow **learning and reasoning simultaneously** – e.g. inducing the rule *locatedIn(X,Y) :- locatedIn(X,Z) ∧ locatedIn(Z,Y)* from raw facts [11]. A challenge, however, is scalability (differentiable proof graphs can grow large), but recent work combines neural guidance and vector indexing to handle larger knowledge bases [12]. The upshot is that *differentiable logic* approaches can blend symbolic rules with neural networks, enabling an AI to refine its knowledge and logical inferences via gradient-based learning.

Geometric Embeddings of Knowledge and Logic

Another key theme is encoding symbolic knowledge (like relations, facts, or rules) as **geometric constraints in a vector space**. Knowledge graph embedding (KGE) techniques pioneered this idea: they embed entities and relations as vectors such that known facts correspond to low-distance or high-score configurations in the space. For example, **TransE** and its variants treat a relation as a translation vector *r* where *head + r ≈ tail* for a true triple (head, r, tail). More recent embeddings represent relations as linear transformations, cones, or **regions** in space to capture hierarchy and intersection. A powerful result is that even complex logical queries (conjunctive queries with AND, existential variables, etc.) can be answered by geometric operations. Ren *et al.* (2018) showed that one can map a logical query to a sequence of *vector projections (for relational steps) and intersections (for conjunctions)* such that the embedding of the query retrieves the correct answers [13] [14]. In other words, there is a formal equivalence between certain first-order queries and navigational geometry in the embedding space – the logical structure is encoded in the algebra of vectors.

To achieve this, models like **Query2Box** (2019) represent sets of possible answers as geometric regions (e.g. hyper-rectangles, or "boxes") in embedding space. Each query constraint (relation or filter) reduces the region (intersection of boxes), and the final region contains the embeddings of all answers that satisfy all constraints. This way, logical **AND corresponds to geometric intersection**, logical OR to union of regions, and existential quantifiers to projections. Newer approaches like **BetaE** (2020) use probabilistic embeddings (Beta distributions) to represent uncertainty in query answers. These **geometric neurosymbolic systems** can even handle *incomplete* knowledge graphs by learning to infer missing facts via interpolation in the vector space. Moreover, they are fully differentiable, so the ability to answer logical queries improves as the embeddings are trained [15]. The takeaway is that *"logical relations as geometric constraints"* is not just a metaphor – it's implemented in systems where distance, angle, and region overlap correspond to truth values of logical expressions. This enables fast reasoning (nearest-neighbor search in a vector database can retrieve answers) and continuous generalization (similar entities might satisfy similar queries).

Vector Symbolic Architectures (High-Dimensional Symbol Binding)

A complementary approach uses **Vector Symbolic Architectures (VSA)**, also known as *Hyperdimensional Computing*, to marry symbolic structure with neural representations. VSAs represent symbols (and composite symbolic structures) as very high-dimensional vectors, where operations like binding (associating two symbols) and superposition (set union or summation) are defined via algebraic manipulations on vectors. The remarkable property is that these vector operations mimic logical or symbolic operations while staying within a continuous vector space. Recent work by IBM and others has revived VSA as a way to build more *brain-like cognitive architectures*. IBM's **NeuroVSA** project, for example, combines a small set of algebraic operations on high-dimensional vectors to achieve reasoning and learning [16]. In NeuroVSA, information is stored in superposed vector form (distributed representation) rather than localized bits, which makes it robust to noise and capable of one-shot learning [17]. Operations like convolution or permutation in the vector space can implement analogues of pointer binding or tuple construction.

The advantage of VSA-based reasoning is that it inherently blends symbolic discreteness with vector continuity. *Variables* and values can be bound into a single vector, meaning a neural network can manipulate that vector and thereby perform logical inference implicitly. For instance, a query like *f(A,B)?* could be encoded into a vector and compared with memory vectors that encode facts like *f(X,Y)*; a high dot-product might indicate a match with *X=A, Y=B*. VSA's have been used in solving Raven's Progressive Matrices (visual puzzles) by encoding perceptual patterns and logical rules together [18]. They are also being explored for **analogy making** and **abductive reasoning** (inferring the best explanation) by learning vector representations of rules [19]. One appeal of VSAs is their potential compatibility with **neuromorphic hardware** – since operations are simple (XOR, rotate, add) and can be done with in-memory computing, they promise efficient, brain-inspired reasoning [20]. Overall, VSAs provide a *vector calculus of symbols*, allowing an AI to manipulate structured knowledge (like graphs, sequences, sets) in a neural network-friendly way.

Category-Theoretic and Topological Knowledge Integration

To ensure **global consistency while combining local reasoning contexts**, researchers are turning to category theory – in particular, *topos theory*. A **topos** is a mathematical structure that generalizes set theory and comes equipped with an *internal logic* (often intuitionistic logic) and sheaf-like structures to glue local data into global wholes [21] [22]. In AI, category-theoretic approaches provide a principled way to compose different knowledge modules and modalities. For example, *sheaf theory* (from algebraic topology) has been applied to knowledge graphs: each relation can be seen as a constraint that local entity embeddings must satisfy, and a global section of the sheaf represents a **globally consistent embedding assignment** [23]. In 2023, Gebhart *et al.* framed knowledge graph embedding as *sheaf learning*, which brought tools to measure **local vs. global consistency** of the learned embeddings [23] [24]. If local pieces (triples or subgraphs) are inconsistent when glued together, the sheaf formalism can detect that (via a *sheaf Laplacian* measuring inconsistency) and guide the model to correct it. This is analogous to ensuring different "views" or contexts in the knowledge graph agree on overlaps – much like ensuring different maps of a city align on the boundaries when forming a single global map.

More broadly, **topos-based learning** has been proposed as a next-generation AI framework that can handle **hierarchical, multi-context reasoning**. A recent survey (Jia *et al.*, 2023) argues that standard deep learning lacks "structural constraints to ensure semantic consistency, logical validity, and global coherence" in complex architectures [25]. Their solution is to introduce *sheaf and stack structures* (from topos theory) on top of neural networks to enforce that local decisions are compatible globally [25]. In such a design, each subsystem (e.g. a vision module, a language module) might have its own internal logic (a local topos), and a higher-order structure ensures these logics align when combined. Topos theory provides constructs like **pullbacks and pushouts** (for aligning ontologies or contexts) and **subobject classifiers** (to represent predicates and truth values categorically) [26] [27]. The benefit is an AI that can integrate heterogeneous knowledge (say physics laws and commonsense heuristics) without forcing them into one flat inconsistent theory – instead, they remain in their appropriate contexts but are *glued together* consistently on overlapping concepts. This approach is still theoretical, but early work has shown it can improve interpretability and guarantee certain invariances or logical constraints in learned models [28]. In essence, category-theoretic AI ensures the **"logic of glue"**: it focuses on how to connect parts so that the whole system maintains coherence, much like topos logic glues local truths into a global truth without losing nuance or introducing contradictions.

Notable Systems and Frameworks (2022–2025)

Research on hybrid reasoning has yielded several frameworks and prototypes. Here we highlight a few, along with their key ideas:

- **DeepProbLog (2018–2020)** – Extends the probabilistic logic programming language Prolog by embedding neural predicates. Perception tasks (like classifying an image) can be done by a neural network, then symbolic ProbLog handles the logical reasoning with those results. It's a neurosymbolic system that retains exact logical inference but delegates uncertain perception to deep nets, enabling multimodal reasoning (vision + logic).

- **Logic Tensor Networks (LTN, 2022)** – A fully differentiable first-order logic system using fuzzy logic truth values [7]. Implemented in frameworks like *LTNtorch*, it has been used for knowledge graph completion and logical query answering with neural networks. LTN is notable for providing a *uniform language* to perform both learning and reasoning [7], making it a general template for neurosymbolic tasks.

- **Logical Neural Networks (LNN, 2021–2022)** – IBM's neurosymbolic engine where each neuron acts as a logic gate (with continuous truth values). LNNs support **bidirectional inference** (you can query in any direction, not just forward-chaining) and incorporate logical consistency in the loss function [3]. They have demonstrated applications in natural language understanding, where background knowledge (facts or ontologies) is encoded into the network to improve comprehension and consistency of answers.

- **Neural Theorem Prover and NTP$^\lambda$ (2017–2021)** – University of Oxford and later work by UCL/DeepMind introduced NTPs, which learn relational reasoning end-to-end. NTP$^\lambda$ (2018) could learn first-order rules by gradient descent while achieving competitive results on knowledge base completion. These systems inspired many follow-ups combining embedding-based models (like ComplEx, ConvE) with rule learning – for example, **RulE (2020)** which learns embeddings for logical rules themselves to inject into KGE models.

- **Vector Symbolic Architecture Systems (2023)** – IBM's NeuroVSA team has shown VSA's strength in few-shot learning and abstract reasoning [29] [17]. A 2023 Nature Machine Intelligence paper used a *NeuroVSA model to solve Raven's Progressive Matrices* (visual IQ test) at a level on par with deep CNNs, but with far greater data efficiency and interpretability. By binding visual features with candidate answer patterns in high-D vectors, the system could logically infer the correct answer. This exemplifies how *symbolic structure (the puzzle rules)* and *geometric representation (the image features)* worked together.

- **Sheaf-Based Knowledge Graph Embedding (2023)** – By treating each relation as a *fiber* and entities as base spaces, Gebhart *et al.* defined an embedding model that can natively handle *typed relations and complex queries* with a guarantee of consistency [23] [24]. They showed improvements on answering multi-hop queries by using *harmonic extension* (a concept from sheaf theory) to fill in missing facts in a logically sound way. This framework can be seen as a step toward *topology-informed machine learning*, where the shape of data and knowledge is considered in the learning process.

- **Topos Architectures for LLMs (2025)** – Very recent theoretical work (Mahadevan, 2025) proposes viewing *the space of all language model behaviors as a topos*. This implies one can use category operations like pullbacks to combine different language models or knowledge sources in a principled way [30] [31]. While highly abstract, this direction could lead to new architectures for large models that are more modular and *globally coherent* by design (for example, ensuring that different subsystems of a multi-modal AI all obey a shared internal logic).

Implementing a Hybrid "DonutOS" Planner

Bringing these ideas together, we consider how one might implement an AI planner for a system (like the hypothetical *DonutOS*) that **mixes creative associative leaps with rule-based consistency checks**. The goal is a planner that can brainstorm freely (thanks to geometric neural networks) yet remain grounded and non-contradictory (thanks to symbolic logic). Here are some suggestions:

- **Neural Proposal, Symbolic Verification:** The planner could use a large neural model (such as a language model or a graph neural network over a knowledge graph) to generate candidate plans or associations. These might be creative and non-obvious – e.g. suggesting an unconventional tool for a task by analogy. Each candidate, represented in an embedding form, would then be checked by a symbolic module: for example, an LNN or logic rule base ensures the plan doesn't violate hard constraints or known facts. This is akin to **"generate-and-test"**: generation in vector space, testing with logic. Because the vector proposals come with similarity scores, the system can explore *analogical ideas* (close matches) and even contradictory hypotheses (divergent ideas) simultaneously. The logic checker, being paraconsistent, can handle this without blowing up – it would mark some proposals as partially inconsistent, guiding the refinement rather than rejecting everything.

- **Embedding Knowledge Graph + Rules:** DonutOS could maintain a knowledge graph of the environment and user goals. Using a knowledge graph embedding (perhaps enhanced with sheaf-based consistency), the planner can quickly predict likely relations or missing steps by geometric reasoning. For instance, if the goal requires an intermediate state that's not explicitly in the graph, the embedding might suggest a bridging concept. These suggestions can then be fed into a **symbolic reasoner (like a PDDL planner or logical theorem prover)** which checks if the sequence of actions is actually valid. The symbolic planner can enforce global constraints (e.g. resource limits, temporal logic of events), while the neural embeddings provide the intuition to fill gaps. This hybrid loop would iterate until a plan is found that satisfies all symbolic constraints – effectively using neural creativity under a symbolic safety net.

- **Multi-Modal Integration via Category Theory:** In a system dealing with text, images, and other modalities, category theory offers a blueprint for integration. One could design each modality-specific subsystem as a *category* with its own representations and morphisms (transformations). A higher-order *functor* or *sheaf* could align these – for example, linking the concept of "red circle" as described in text to the detection of a red circle in an image. The planner could employ a **topos approach to context**: maintain separate logical contexts for, say, "what the user says" vs "what the camera sees," which are combined by functors (mappings) that identify common entities. This ensures the planner's global state is **consistent across modalities** (no contradictions between vision and language understanding) yet allows local processing to be specialized. If inconsistencies arise (e.g. the user says an object is present but the camera doesn't see it), the system can represent that ambiguity without immediate collapse, possibly asking a clarifying question or using a paraconsistent logic to assign a "both true and false" value temporarily.

- **Vector Symbolic Memory for Plans:** The planner might use a VSA-based memory to store and manipulate plan states. Each potential plan (a sequence of steps) can be encoded as a high-dimensional vector via binding step representations together. By design, similar plans will produce similar vectors, allowing the neural part to do a fuzzy search in the space of plans (creative variations), while a symbolic decoder can extract an exact sequence from a vector if needed. This yields a kind of **analogical planning**: if a novel situation is "close" in vector space to a known scenario, the system can retrieve that scenario's plan and adapt it, all while ensuring via symbolic checks that the adapted plan still meets logical requirements. The neuro-vector approach also makes the planner inherently explainable: each element in the vector has a meaning (due to how it's constructed from symbols), so one can trace back *why* a certain action was chosen by decoding the vector (as IBM's NeuroVSA notes, the computing is transparent and traceable to causes [17]).

Conclusion

Hybrid symbolic–geometric reasoning is an emerging paradigm aimed at AI that is **both intuitive and trustworthy**. Recent advances show that it's feasible to encode logic in neural terms – from differentiable logic programs that learn from data [5] , to knowledge graph embeddings that treat relations as geometric transformations [23] , to vector architectures that carry out algebraic reasoning in high dimensions [16] . Meanwhile, category theory and topology provide high-level formalisms to ensure that as we combine these pieces, the AI's **global reasoning remains coherent** [28] . For a system like "DonutOS" which must orchestrate multiple modalities and cognitive styles, these approaches offer a roadmap: use neural networks to expand what the AI can see, imagine, and suggest, but use symbolic logic (perhaps enriched with paraconsistent tolerance and category-theoretic glue) to ensure it stays consistent, explains itself, and respects constraints. The key papers and systems highlighted – from **LTN** and **LNN** to **NTPs**, **NeuroVSA**, and **sheaf-based learners** – form a toolkit for building such an AI. Going forward, research will likely focus on improving the **scalability and integration** of these components. If successful, the result will be AI planners and reasoners that truly "mix and match" creative *both/and* thinking with reliable rule-following – achieving alignment with human complexities without collapsing nuance in the process.

Sources: Recent literature and systems integrating neural learning with symbolic reasoning were referenced, including neuro-symbolic AI surveys [5] , differentiable logic frameworks [7] [10] , knowledge graph embedding techniques [14], IBM's neuro-symbolic projects [16] [3], and category-theoretic AI research [25] [28], among others. These illustrate the state-of-the-art methods for hybrid reasoning and their relevance to planning in multimodal AI systems.

---

[1] [2] Paraconsistent Logic | Internet Encyclopedia of Philosophy
https://iep.utm.edu/para-log/

[3] [4] Extensions and NLU Applications of Logical Neural Networks - IBM Research
https://research.ibm.com/projects/extensions-and-nlu-applications-of-logical-neural-networks

[5] A review of neuro-symbolic AI integrating reasoning and learning for advanced cognitive systems - ScienceDirect
https://www.sciencedirect.com/science/article/pii/S2667305325000675

[6] [7] Logic Tensor Networks – Sony AI
https://ai.sony/publications/Logic-Tensor-Networks/

[8] [9] [10] [11] [12] Neural Theorem Proving: Integrating AI and Logic
https://www.emergentmind.com/topics/neural-theorem-proving

[13] [14] [15] cs.stanford.edu
https://cs.stanford.edu/people/jure/pubs/netquery-neurips18.pdf

[16] [17] [18] [19] [20] [29] Neuro-Vector-Symbolic Architecture - IBM Research
https://research.ibm.com/projects/neuro-vector-symbolic-architecture

[21] [22] [25] [28] Category-Theoretical and Topos-Theoretical Frameworks in Machine Learning: A Survey
https://www.mdpi.com/2075-1680/14/3/204

[23] [24] proceedings.mlr.press
https://proceedings.mlr.press/v206/gebhart23a/gebhart23a.pdf

[26] [27] [30] [31] [2508.08293] Topos Theory for Generative AI and LLMs
https://arxiv.org/abs/2508.08293