The RAIL Principles for Neurosymbolic AI: Reasoning, Assurances, Interfacing and Learning

The RAIL Principles for Neurosymbolic AI: Reasoning, Assurances, Interfacing and Learning

AGNESE CHIATTI, Politecnico di Milano, Italy
MICHAEL COCHEZ, ELLIS Institute Finland & Åbo Akademi University, Finland
CRISTINA CORNELIO, Samsung AI, UK
SEBASTIJAN DUMANCIC, Delft University of Technology, The Netherlands
ARTUR D'AVILA GARCEZ, City St. George's, University of London, UK
LUIS C. LAMB, Stony Brook University, NY, USA
LIA MORRA, Politecnico di Torino, Italy
MATHIAS NIEPERT, University of Stuttgart, Germany
ROBERT PEHARZ, Graz University of Technology, Austria
ALBERTO SPERANZON, Lockheed Martin, Advanced Technology Labs, MN, USA
MAARTEN STOL, BrainCreators & Vrije Universiteit Amsterdam, The Netherlands
ANNETTE TEN TEIJE, Vrije Universiteit Amsterdam, The Netherlands
THIVIYAN THANAPALASINGAM, University of Amsterdam, The Netherlands
FRANK VAN HARMELEN, Vrije Universiteit Amsterdam, The Netherlands
EMILE VAN KRIEKEN, Vrije Universiteit Amsterdam, The Netherlands
ANTONIO VERGARI, University of Edinburgh, UK
BENJIE WANG, UCLA, CA, USA

Neurosymbolic AI systems that integrate machine learning and symbolic reasoning are rapidly gaining attention. They complement the data-intensive statistical approaches of neural networks and language models with symbolic reasoning algorithms to function in high-stakes domains or in low-data regimes that characterize many real-world applications. We argue that the neurosymbolic combination of machine learning and formal reasoning is not a niche approach within AI, but rather includes many already successful techniques that are of crucial importance to the development of reliable, efficient and, ultimately, trustworthy systems. This perspective prompts a re-examination of the design of current AI systems. We show that many leading AI

Authors' Contact Information: Agnese Chiatti, Politecnico di Milano, Italy, agnese.chiatti@polimi.it; Michael Cochez, ELLIS Institute Finland & Åbo Akademi University, Finland, michael.cochez@abo.fi; Cristina Cornelio, Samsung AI, UK, c.cornelio@samsung.com; Sebastijan Dumancic, Delft University of Technology, The Netherlands, S.Dumancic@tudelft.nl; Artur d'Avila Garcez, City St. George's, University of London, UK, a.garcez@city.ac.uk; Luis C. Lamb, Stony Brook University, NY, USA, luislamb@acm.org; Lia Morra, Politecnico di Torino, Italy, lia.morra@polito.it; Mathias Niepert, University of Stuttgart, Germany, mathias.niepert@ki.uni-stuttgart.de; Robert Peharz, Graz University of Technology, Austria, robert.peharz@tugraz.at; Alberto Speranzon, Lockheed Martin, Advanced Technology Labs, MN, USA, alberto.speranzon@lmco.com; Maarten Stol, BrainCreators & Vrije Universiteit Amsterdam, The Netherlands, maarten.stol@braincreators.com; Annette ten Teije, Vrije Universiteit Amsterdam, The Netherlands, annette.ten.teije@vu.nl; Thiviyan Thanapalasingam, University of Amsterdam, The Netherlands, t.singam@uva.nl; Frank van Harmelen, Vrije Universiteit Amsterdam, The Netherlands, Frank.van.Harmelen@vu.nl; Emile van Krieken, Vrije Universiteit Amsterdam, The Netherlands, e.van.krieken@vu.nl; Antonio Vergari, University of Edinburgh, UK, avergari@exseed.ed.ac.uk; Benjie Wang, UCLA, CA, USA, benjiewang@cs.ucla.edu.

ACM 1557-735X/2026/0-ART0
https://doi.org/10.1145/3830022

systems, including some that are not traditionally considered as neurosymbolic, can be analysed from the perspective of four principles of neurosymbolic AI design: *Reasoning*, *Assurances*, *Interfacing* and *Learning* (RAIL). Applying the RAIL framework offers a unified view of seemingly disparate AI systems, ranging from physics-aware machine learning to neuro-guided search (such as Google DeepMind's Alpha-* suite), causal learning and tool-augmented Large Language Models. Importantly, the RAIL principles will enable engineers to make better-informed and more principled decisions about the design and deployment of production-level AI systems. In this article, we introduce the RAIL principles, examine how they can be applied across major areas of AI, and illustrate how they may guide practitioners to integrate neurosymbolic methods into next-generation AI technologies.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**.

ACM Reference Format:
Agnese Chiatti, Michael Cochez, Cristina Cornelio, Sebastijan Dumancic, Artur d'Avila Garcez, Luis C. Lamb, Lia Morra, Mathias Niepert, Robert Peharz, Alberto Speranzon, Maarten Stol, Annette ten Teije, Thiviyan Thanapalasingam, Frank van Harmelen, Emile van Krieken, Antonio Vergari, and Benjie Wang. 2026. The RAIL Principles for Neurosymbolic AI: Reasoning, Assurances, Interfacing and Learning. *J. ACM* 00, 0, Article 0 (2026), 12 pages. https://doi.org/10.1145/3830022

1 Neurosymbolic AI

Neurosymbolic AI brings together the statistical nature of machine learning and the rigorous logical nature of reasoning [16]. The field has grown in relevance as it became clear that the combination of learning and reasoning within a formal semantic framework is crucial for AI to reach its full promise in science and engineering [7, 14].

Industrial AI systems that ingest sensor, text, and video streams are typically assembled from ad-hoc, task-specific modules that are designed and optimized independently. This divide-and-conquer approach accelerates development but results in non-compositional pipelines with no formal guarantees on overall behavior, leading to error propagation, discontinuities at module interfaces, and difficulties in interpreting the reasoning of generative models. The persistent problem of LLM hallucinations, four years after the release of ChatGPT, underscores the need for a more principled solution. Neurosymbolic AI addresses this gap by embedding sound symbolic reasoning into machine learning components, enabling verifiable constraints on model outputs and allowing the system to reason about what has been learned. We therefore frame neurosymbolic AI around four core principles, Reasoning, Assurances, Interfacing, and Learning (RAIL), which together provide a unified blueprint for building reliable, explainable, and compositional industrial AI pipelines. We will show that the RAIL principles also apply to systems that may not traditionally be considered neurosymbolic, but which do inject knowledge into their learning architecture, such as physics informed neural networks and industrial modular pipelines.

2 The RAIL Principles of Neurosymbolic AI

2.1 Reasoning

Reasoning, i.e., chaining a series of inference steps to derive a conclusion, has been essential in the study of machine intelligence. It has been deployed successfully in AI systems for question-answering, decision support, planning, and story understanding, among others. Reasoning can be deductive (*Which consequences follow from a premise?*), abductive (*Which premise best explains an observation?*), inductive (*Which conclusion is justified by given data?*) or analogical (*What are the correspondences between sequences of inference steps?*). Such reasoning capabilities have typically been realized through explicit symbol manipulation in a wide variety of formal frameworks [32].

The benefits of sound reasoning are a key advantage of neurosymbolic AI [14], based on results in classical and non-classical logics. Through representing modal, temporal, and probabilistic

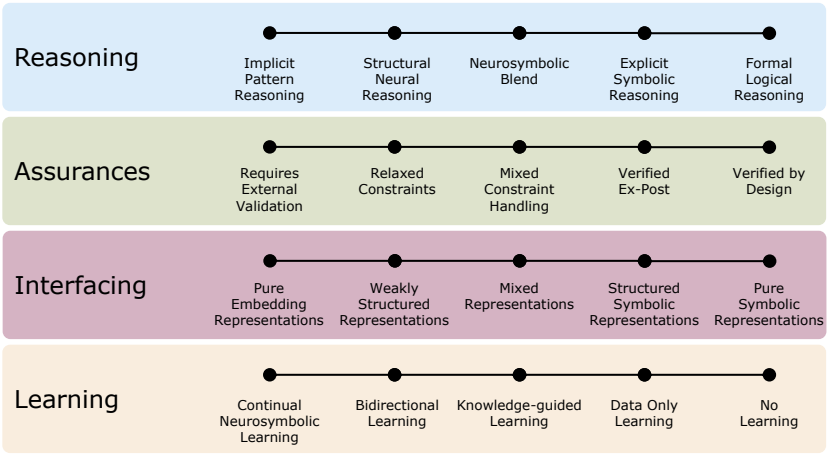


Fig. 1. The four RAIL principles of Neurosymbolic AI. Each principle exists on a qualitative spectrum, with intermediate positions showing different combinations of neural and symbolic approaches.

reasoning, neurosymbolic approaches support the representation of knowledge, actions, and beliefs of interacting agents [7, 14]. Beyond logic, the symbolic component can, in fact, be any computer program, e.g. used as a tool by an LLM for the purpose of reasoning. Recent advances in LLMs show how GenAI displays an informal, emergent, and latent form of reasoning based on a prompting process known as *chain of thought*. Our design of the Reasoning dimension of the RAIL principles (see Fig. 1) acknowledges this possibility, with such entirely implicit reasoning on one end—e.g. emergent reasoning by similarity—and *formal reasoning* on the other. In between, neurosymbolic systems such as DeepProbLog [20] and Logic Tensor Networks [4] explore a wide scale of implicit and explicit reasoning. This includes neurosymbolic systems with *explicit symbolic reasoning*, where neural models call symbolic computation (e.g. generating code in Python and calling an interpreter), and *structural neural reasoning* [14] where reasoning is encoded in the structure and weights of a neural network, with a variety of neurosymbolic blends in between [7]. Our Reasoning dimension also includes systems beyond those typically considered as neurosymbolic, specifically we consider systems that encode background knowledge, either explicitly in symbols or implicitly in their architecture. Despite rapid advances in LLM reasoning, recent results [5] show that the geometric structure that enables semantic generalisation also makes interference, forgetting, and false recall inevitable. These inherent limitations of systems on the left side of the Reasoning spectrum imply that future systems will have to inhabit the entire spectrum for fully reliable inference.

2.2 Assurances

This dimension of RAIL represents the *assurances* that a neurosymbolic system can provide with respect to given *requirements*. These can take the form of a formal specification that designers and domain experts elicit for safety-critical systems, formalized using one of the many suitable logics (propositional, temporal, or spatio-temporal logic), or can be unit-tests in the case of programs. The specifications may formalise diverse non-functional requirements too: safety ("the system must never enter an unsafe state"), performance ("the system must reach a desired state in a given time"), or reliability ("given a set of initial conditions, the system must always reach a target set of states").

For systems that learn, particularly those using deep neural networks, such requirements are notoriously difficult to check. While the field of neural network verification has devised several

techniques to ensure that certain required properties like robustness and bounded behaviour are met, verification guarantees in these systems often rely on simplified assumptions or computationally expensive procedures. On one side of the Assurances spectrum we find neural components trained without any requirements, thus requiring external validation. On the other side of the spectrum, we find neural components that are *verified by design*, i.e. the requirements are baked in the network architecture. In the middle are the approaches that treat requirements as soft or hard symbolic constraints during learning. This is typically done by relaxing the logical representation of a requirement into differentiable form, e.g. using fuzzy logic [4, 33], making it suitable for training, runtime monitoring and inference-time reasoning. It also includes neural networks with *shield layers* that assign zero probability to the output configurations that may violate a given constraint, e.g. algorithms to steer LLMs to guarantee that they produce outputs satisfying a given grammar or syntactical structure [38]. Finally, neural networks' properties or outcomes may be verified ex-post, after the system has been trained. Inherent limitations on assurances by neural systems shown in [5] imply that symbolic approaches to assurances will remain essential. Overall, balancing these assurances against their inherent limitations and scalability barriers remains a core open challenge for neurosymbolic architectures especially in complex real-world deployments.

2.3 Interfacing

Neural networks have been criticized for not being *interpretable* or *explainable*, understood as the capacity for internal processes or network outcomes to be inspected by or explained to humans. These requirements are particularly important when decisions made by AI are high-risk or when outcomes may induce discrimination. Interpretability and explainability presuppose a *communication channel* from the system to the user. Conversely, robust and generalizable reasoning often depends on the ability to communicate *domain knowledge* to the system in a well-defined form, requiring a channel also from the user to the system. These bidirectional requirements motivate the notion of *interfacing* as a RAIL principle: establishing shared representations that support meaningful information exchange between agents, whether human or artificial.

Like the other RAIL principles, interfacing is characterised on a spectrum (Fig. 1). At one end are the latent representations in a neural network, which excel at capturing high-dimensional statistical structure but offer limited direct inspection or user intervention. Moving along the spectrum, weakly structured representations introduce partial organization, e.g., attention patterns or neuron groupings, without committing to a defined semantics. Mixed representations, including graphs and causal models, support intervention while remaining compatible with subsymbolic systems. Similarly, natural languages have syntax and semantics, but are also used to convey other information (e.g., intentions, emotions) that often reside on a subsymbolic level. Further along, structured symbolic representations such as logic programs, annotated code, or structured data formats prioritize explicit compositional structure. Finally, pure symbolic representations, grounded in formal knowledge representation with unambiguous semantics, can be used to make interfacing requirements explicit, albeit at the cost of often requiring representations to be manually-specified. Each position on this spectrum affords distinct trade-offs in abstraction, expressivity, and handling of uncertainty in the interfaces from user to system (guidance) and from system to user (explanation).

2.4 Learning

The *learning* dimension of RAIL captures the degree to which a system applies machine learning to shape its behaviour. At one end of the spectrum, symbolic systems use pre-defined knowledge *without learning*, and depend on knowledge engineering rather than statistical generalisation from examples. Deep learning models acquire their behaviour through *data-only learning* by extracting statistical regularities, which often requires extensive data and offers limited control over what

is learned. Neurosymbolic systems go beyond learning from data alone and combine the use of structured knowledge to guide learning and decision-making [34]. Many systems employ some form of *knowledge-guided learning* to shape learned representations, improve generalisation, or reduce data requirements [34]. Knowledge can be added implicitly via architectural inductive bias—Convolutional Neural Networks make implicit use of knowledge about invariance—or by curating and augmenting data with informed transformations. Neurosymbolic systems enable a more explicit use of knowledge, that may include encyclopedic facts, taxonomies, abstract rules, logical constraints, program sketches or expert scientific structures like differential equations in physics-informed machine learning [25]. Incorporating knowledge can bring substantial improvements in data efficiency compared to purely neural baselines [4, 10, 20]. Further towards the end of the spectrum, some systems learn to reason, that is to learn the very mechanisms that operate over logical formalisms rather than treating logic as an external interpreter. Such systems do not treat knowledge as fixed background knowledge, but learn to evolve it as part of what we call *bidirectional learning* [13]. Finally, models may be fine-tuned after training through feedback loops such as reinforcement learning or self-training. In *continual neurosymbolic learning*, learning takes place over time with evolving knowledge to inform the learning process.

3 Putting Existing AI Systems on RAILs

We illustrate RAIL's usefulness by positioning prominent AI systems within the RAIL spectrum across five key AI domains, discussing the opportunities afforded by moving along the RAIL scales, and summarizing industrial neurosymbolic applications from the perspective of the RAIL principles.

3.1 Knowledge Discovery

Knowledge graphs (KGs) are a standard data model for representing entities and their relationships, governed by an ontology of types and constraints [19]. Originally designed for data integration and query answering, KGs are increasingly exploited to predict new knowledge, i.e., novel edges between entities, through neurosymbolic techniques. Two main families of methods address this link prediction task. Embedding-based approaches [9] map graph nodes to high-dimensional vectors, training a loss function that aligns symbolic patterns with geometric ones so that vector similarities can suggest new relations (e.g., discovering chemical bonds). Rule-based systems such as AnyBURL [21] learn explicit rules from the graph and use symbolic reasoning to infer new rules (e.g., drug-gene-symptom patterns for drug repurposing).

Reasoning. An embedding-based technique reasons implicitly on the patterns in vector space (e.g. vector similarity or distances). A drastic shift along the Reasoning dimension is taken by rule-learning systems. Instead of reasoning over geometric patterns in vector spaces, they perform explicit symbolic reasoning applying the rules that were discovered.

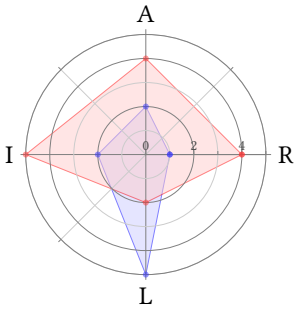
Assurances. Only soft assurances with *relaxed constraints* are given on the correctness of newly predicted relations. Minimization of the loss function increases the likelihood of a predicted relation but it does not provide guarantees. Assurances can move from relaxed constraints to verified constraints via ex-post verification of ontological demands, resulting in the removal of predicted relations that violate constraints such as types, cardinality, anti-symmetry.

Interfacing. The combination of a symbolic graph with numerical vector spaces offers a mixed level of interfacing. In the rule-learning methods, all representations (the graph, the rules and the newly discovered relations) are symbolically represented and expected to be humanly interpretable. By contrast, embeddings are weakly structured with difficult interpretability.

Learning. When a KG is used simply as a source of data, learning will be data-only learning because no further knowledge is injected. The Learning dimension shifts to knowledge-guided learning when using more informed (domain specific) sampling strategies to learn embeddings,

e.g. type constraints [24] or other ontological properties of the graph [1]. Rule-based learning is typically bi-directional. They first learn the rules from the graph data and then apply the rules to the graph data to discover new relations.

Lessons for RAIL. The two families of knowledge discovery methods are located very differently on the RAIL dimensions. For both families, assurances would move from relaxed constraints to verified constraints if verification techniques were applied to ontological knowledge. The radar plot shows that embedding methods (blue) place more emphasis on learning, while the symbolic rule-learning methods (red) provide more assurances and better interfacing.



3.2 Neuro-Guided Search

Neuro-guided search is a neurosymbolic approach where neural networks modify a structured symbolic representation, often using reasoning to ensure correctness. As a case-in-point, the Google DeepMind Alpha family of systems (AlphaGo, AlphaFold, AlphaProof, etc.) combine neural learning with Monte Carlo Tree Search [29]. In these systems, the neural component actively navigates and expands the symbolic problem structure (game trees, proof trees) rather than invoking external tools. This tight integration of neural and symbolic parts is the core inventive step behind their state-of-the-art performance across domains.

Reasoning. Neuro-guided search systems employ explicit structured representations of the problems they solve: AlphaGo explicitly represents the game state, neuro-guided program synthesis methods represent programs as symbolic structures, AlphaProof and AlphaGeometry explicitly represent the proofs in the LEAN programming language. Consequently, they all support explicit formal reasoning about their outputs. The outcome of the reasoning step can be explicitly used during the learning process. At the same time, the learned problem-solving process itself, e.g. the action selection in AlphaGo or the lemma proposition in AlphaProof, is implicit and opaque.

Assurances. By using structured representations, these systems offer strong assurances by design. For instance, AlphaProof guarantees proof correctness through LEAN. The other Alpha family systems are prevented from exploring invalid programs or strategies that violate the game rules.

Interfacing. Neuro-guided search systems have high interfacing from user to system: the user can provide new lemmas to AlphaProof and AlphaGeometry or new game rules to AlphaGo. However, the interfacing from system to user is limited because the decisions of the system take place within the black box neural network. AlphaProof cannot explain why it proposes a specific lemma to be considered, and AlphaGo cannot explain why a particular move is promising.

Learning. Learning in most neuro-guided search systems is knowledge-guided: the neural components learn how to make decisions constrained by the structured representations. For instance, in AlphaGo, the neural network learns to propose actions, and the rules of the game constrain which actions are possible; neuro-guided program synthesisers propose completions for abstract-syntax trees constrained by the syntax of a programming language. The structured problem representation remains unchanged while all the learning happens in the neural network. In some systems, however, learning is bidirectional. For instance, AlphaProof changes the representation of the problem when proposing new lemmas. That new representation can also be used in the solution of other problems. AlphaTensor similarly explicitly changes the algorithm structures through decompositions [15].

Lessons for RAIL. The Alpha systems highlight key trade-offs across the RAIL dimensions that inform the design of neurosymbolic systems. (1) *Reasoning–Assurance trade-off:* The progression from AlphaGo to AlphaProof illustrates how stronger formal assurances necessitate more explicit reasoning components. AlphaGo's game rules provide implicit constraints, whereas AlphaProof's

integration with LEAN enables machine-verified proofs, showing that domains requiring high-stake guarantees benefit from tighter coupling with formal verification systems. (2) *Interfacing asymmetry:* A consistent pattern emerges wherein interfacing *into* these systems (modifying rules, providing lemmas) is high, while interfacing *out of* the systems (explaining decisions) remains limited. The outputs are transparent and verifiable, but decision-making is opaque. (3) *Learning at the Knowledge-Guided level:* in most Alpha systems, learning occurs within the neural component and the symbolic structure remains fixed. In Alpha Proof and AlphaTensor's bidirectional learning, however, the symbolic representation evolves with new lemmas and decompositions, indicating that richer neurosymbolic integration is achievable when the symbolic element is flexible.

3.3 Tool Use by Language Models

A rapidly evolving area where neurosymbolic principles have re-emerged is the coupling of LLMs with computer program, either through *program synthesis*, where LLMs generate executable code from partial specifications in natural language or examples, or *via tool usage*, where LLMs invoke existing APIs or database queries. Program synthesis originated in symbolic AI, emphasizing explicit representations and guaranteed search [11]. Although purely neural LLM methods [3] have emerged, current systems are increasingly neurosymbolic, combining LLMs with programs, tests, grammars, interpreters, and verification engines. Tool-augmented frameworks like ReAct [37] exemplify this shift, operating where the model reasons about the problem, selects a concrete action, executes it within an external tool, and refines its behavior based on the result. This coupling is valuable because part of the LLM tasks are delegated to components with explicit semantics. However, reliability depends not only on tool correctness, but also on which tool the model selects and on how it integrates results from different calls, both of which are often brittle.

Reasoning. LLMs that use tools go beyond implicit pattern completion. While the base language model still performs latent statistical inference, the explicit generation and execution of programs or structured actions introduces symbolic semantics into the reasoning process. In the case of ReAct, reasoning is made more explicit by alternating between deliberation steps and symbolic actions (tool calls and environment interactions). Interleaving reasoning and actions can reduce hallucination and error propagation, but in longer sequences early tool-selection errors or misinterpreted observations may still cascade into later steps.

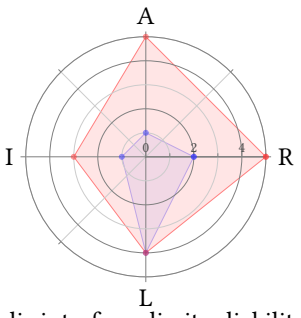
Assurances. Tool use enables guarantees currently unavailable to LLMs. At one end, execution, compilation, and unit testing provide the possibility of empirical checks to filter invalid outputs. More advanced approaches integrate formal verification [8], differential testing, or oracle-guided refinement, yielding assurances about functional correctness. In these approaches, selection and integration of external tools is crucial but remains difficult across the latent/explicit barrier.

Interfacing. LLMs with tools offer more interfacing possibilities than standalone LLMs. Programs, grammars and API calls provide compositional, inspectable and executable interfaces between neural and symbolic agents, as well as human users. Visual programming systems like ViperGPT [30] extend to multimodal settings where symbolic programs orchestrate learned perception modules from a query. However, interfaces that integrate LLMs with tools can introduce a degree of brittleness, as APIs, schemas, and grammars often require exact syntax.

Learning. Most current tool-augmented systems combine pre-trained foundation models with prompting, in-context learning, or reinforcement learning from execution feedback. Toolformer-style approaches[26] explicitly learn when and how to invoke tools. Other AI agents will tackle a task by generating a function from a few examples, testing it against a differentiable oracle that provides feedback for further refinement, e.g. agents tackling the *Abstraction and Reasoning Corpus* (ARC) challenge [35]. These bidirectional learning cycles closely mirror neurosymbolic learning

paradigms where the outcomes of symbolic execution guide neural network updates, and the neural networks' heuristics guide symbolic search.

Lessons for RAIL. Tool-augmented LLMs offer design directions that can inform broader advances in Neurosymbolic AI. The integration of structured programs and representations with LLMs achieves a trade-off between bottom-up learning from data and top-down enforcement of constraints. Tools can also serve as a method to check for correctness of LLM-based systems. Combining natural language with other data modalities impacts the interfacing dimension of RAIL, increasing interoperability among tools that require heterogeneous input and output representations. The radar plot shows the differences between LLMs with tools (red) and without (blue).Tool-augmented LLMs expose scalability barriers for neurosymbolic AI. Tool-selection instability, cascading reasoning errors, and brittle symbolic interfaces limit reliability, especially as tools grow in number and diversity. These limitations suggest that scaling and reliability depend on selecting the right balance between symbolic components for explicit reasoning and controllability, and subsymbolic components for flexibility and generalisation [39].



3.4 Causal Learning and Reasoning

Causal modeling [22] formalizes the representation and learning of causal knowledge, most often using causal graphs in which nodes denote variables and edges denote causal relationships. Within this framework, two central tasks are: *causal discovery*, which seeks to recover the graph structure from data and *causal inference*, which estimates the effects of interventions by combining data with partial knowledge of the graph. Because causal methods integrate statistical learning with explicit structural constraints and domain knowledge, they naturally embody neurosymbolic principles. *Identifiability* specifies when causal effects can be uniquely determined from observed data and graph assumptions, so progress in this area has depended on fusing symbolic representations and formal reasoning with statistical estimation rather than relying solely on black-box models.

Reasoning. Causal reasoning is formalized through a three-level causal hierarchy: association, intervention, and counterfactuals [23]. *Association* concerns reasoning about statistical dependencies and predictive tasks in standard machine learning. Various limitations of machine learning, such as poor extrapolation, and shortcuts, have been studied from a causal perspective. A causal perspective at the level of *Intervention* opens the possibility of systems with improved generalization and extrapolation capabilities, in particular under shifts in data distribution. One of the most widespread applications of intervention analysis is in econometrics and e-commerce [28]. Understanding and designing biological pathways is another prominent example. The third level of Pearl's causal hierarchy is counterfactual reasoning, seeking to answer the question *If X had not occurred, would Y have been the case?*, a question closely connected with explainability in neural networks [36].

Assurances. Probabilistic inference under observational, interventional and counterfactual distributions guarantees that, as long as the Structural Causal Model (SCM) specification and probabilistic inference routines are correct, strong assurances can be provided. Computational complexity issues aside, probabilistic inference also provides an opportunity to tackle epistemic challenges with assurances. The *do-calculus* [22] is a complete and consistent algorithm to decide whether interventional queries can be answered from the observational distribution and causal structure alone.

Interfacing. The intuitive graphical structure of causal models makes them easy to interface with on a qualitative level. Furthermore, causal models are naturally modular, in that it is not only possible to, but natural to modify the causal structure of subgraphs.

Learning. Causal learning often involves discovering the causal structure (graph), or causal mechanisms and probability distributions from data. A key constraint in causal learning comes from fundamental *identifiability* barriers, which restrict what can be learned from available data. This motivates the use of knowledge guided learning approaches. Another important challenge is to connect causal reasoning, which is typically defined over a set of interpretable variables, with high-dimensional unstructured data such as images. This is the goal of *causal representation learning*, which aims to mine causal representations from unstructured data [27].

Lessons for RAIL. Causality exemplifies a neurosymbolic design pattern in which a symbolic structure, typically a causal graph, mediates between data-driven learning and formal reasoning. On Reasoning, causality explicitly distinguishes associative prediction from intervention and counterfactual analysis. For Assurances, it provides guarantees when causal queries are answerable from data and assumptions, and when they are fundamentally underdetermined. On Interfacing, causal models offer modular representations that support human intervention. Finally, on Learning, causal discovery and representation learning illustrate knowledge-guided and bidirectional learning where statistical estimation is constrained by symbolic structure and vice versa.

3.5 Physics-Aware Machine Learning

Machine learning for science and engineering has developed largely in parallel with neurosymbolic AI, yet the overlap is substantial when viewed through the lens of the RAIL principles. Neurosymbolic AI traditionally injects *explicit structure* via logical rules, programs, or constraints. Physics-aware ML injects structure via *physical laws* [10], *symmetries* [2], *conservation principles* [25], and *mathematical operators* [6]. These ingredients are not always described as symbolic, but play the same conceptual role. They encode human knowledge in declarative form, constrain what solutions are admissible, and support more reliable generalization beyond the training distribution.

Reasoning. Many *AI for Science* methods employ forms of reasoning that closely mirror neurosymbolic inference. Forward simulation is akin to *deduction*: given assumptions (initial conditions, parameters), derive consequences (system trajectories). Inverse problems and parameter identification resemble *abduction*: infer latent causes or mechanisms that explain observations. Changing boundary conditions or forcing terms produce *counterfactuals*, analogous to interventions in causal reasoning. Multi-physics models combine multiple modules (e.g. fluid and structure), similar to compositional reasoning, where complex behaviour is assembled from reusable parts.

Assurances. Assurances may come from physical structure and invariances. Some assurances can be enforced *by design* through architecture [17] rather than through a loss function. For instance, equivariant neural networks [6] guarantee the encoding of symmetries (e.g. rotation, translation). Hamiltonian architectures [18] enforce conservation laws by construction. As in neurosymbolic systems, these assurances rely on the correctness of the assumed structure (e.g. the conserved quantity) and can degrade under misspecification, discretization artifacts, or missing physics.

Interfacing. Neurosymbolic AI emphasizes interfaces that let experts state *what* they know without prescribing *how* to compute it. Science has similarly rich declarative languages: differential equations, symmetries, conservation laws, and energy functions. These are natural symbolic interfaces between humans and models. Thus, work in neurosymbolic AI on intermediate representations (constraints, knowledge graphs) suggests a direction for scientific ML: enable incremental, partially specified, and revisable scientific knowledge to guide learning and reasoning.

Learning. In neurosymbolic learning, knowledge is incorporated either in the network architecture [17] or by augmenting the training objective with penalties for violating symbolic constraints. Physics-aware ML follows this pattern, with physical models as symbolic knowledge. In physics-informed approaches [12], learning includes penalties for inconsistency with physics equations, boundary conditions, or conservation laws. Such physics-based terms encourage solutions that

respect known physical structure, even when observations are sparse or noisy. In [12], knowledge-based penalties are implemented as soft constraints using weighted regularization terms.

Lessons for RAIL. Physics-aware ML is in many ways neurosymbolic, as it integrates learned models with explicit, human-authored structures. Viewing it through the RAIL lens helps clarify design trade-offs: where to enforce knowledge (loss function or network architecture), how to represent such constraints (rules or operators), and how to learn such constraints (PDEs or logic).

3.6 Industrial Neurosymbolic Architectures

The neurosymbolic paradigm in industrial AI has an additional driver: architectural necessity. Mature software products typically do not deploy a single neural model but consist of a composite software stack in which pragmatic demands on control and flexibility, together with the economy of product maintenance, drive the designs toward modular architectures. These architectural patterns are implicitly neurosymbolic: deep learning components that rely on learning from data combine with surrounding procedural glue code addressing logical requirements and encapsulating domain knowledge. Industry has matured this design pattern over the last decade largely independent of academic neurosymbolic AI. These software stacks function effectively as they allow ML engineers to enforce strict symbolic controls on stochastic neural components. Consequently, the lack of formal neurosymbolic methods is not a bottleneck that prevents industry from deploying products, using pragmatic, albeit disjointed, integration. The value proposition of formal neurosymbolic research for industry lies in providing more principled integration. The joint academic-industrial roadmap therefore concerns the transformation of these ad-hoc architectures into formal design patterns [31]. The goal is to replace brittle glue code and symbolic knowledge interfacing in the stack with the expressive, differentiable methods highlighted in previous sections (e.g., replacing a pre-processing script with a symbolic regularizer, or a post-processing heuristic with a differentiable reasoner). This upgrade path would move the system from an assembly of black boxes towards differentiable interfacing, ideally without sacrificing the engineering benefits of modularity.

Lessons for RAIL. RAIL principles are clearly present in industrial AI architectures. Learning modules are data only, or knowledge-guided by domain informed regularization. Interfacing ranges from using embeddings to structured symbolic representations, and reasoning in the form of symbolic glue code is typically informal. Assurances remain the most challenging RAIL dimension for such architectures. But this is also why assurances based on increasingly formal yet differentiable methods are how the neurosymbolic paradigm will have the biggest impact on industrial AI.

4 Conclusion: The Neurosymbolic Path Forward

We have shown that many high-impact AI systems, such as those from the Alpha* family, can be re-interpreted as neurosymbolic, unwittingly following the RAIL principles. The RAIL principles map the trade-offs and synergies that shape neurosymbolic architectures. They define a four-dimensional design space in which the inter-dependent axes constrain viable neurosymbolic architectures. Mapping existing systems onto this space revealed such trade-offs and synergies. By visualising these relationships, RAIL offers a framework for exploring new design opportunities, guiding future AI research and industrial deployments. We have provided only a qualitative treatment of the RAIL dimensions. Turning them into a practical tool will require operational criteria, formal definitions, and concrete metrics. We trust the community will rise to this challenge.

Acknowledgments

This paper is the result of Dagstuhl Seminar 25452: A Roadmap Towards Practical Applications of Neurosymbolic Learning and Reasoning. We thank the other participants for valuable discussions: Byron Cook (Amazon Web Services & University College London), Egor Kostylev (U. of Oslo), Robin

Manhaeve (KU Leuven), Guy Van den Broeck (UCLA), Maria-Esther Vidal (Leibniz U. Hannover), and Ute Schmid (U. of Bamberg). We thank the anonymous reviewers for valuable comments.

References

[1] Ralph Abboud, Ilkan Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. BoxE: a box embedding model for knowledge base completion. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 809, 13 pages.

[2] Tara Akhound-Sadegh, Laurence Perreault-Levasseur, Johannes Brandstetter, Max Welling, and Siamak Ravanbakhsh. 2023. Lie point symmetry and physics-informed networks. *Advances in Neural Information Processing Systems* 36 (2023), 42468–42481.

[3] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732* (2021).

[4] Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. 2022. Logic tensor networks. *Artificial Intelligence* 303 (2022), 103649.

[5] Sambartha Ray Barman, Andrey Starenky, Sofia Bodnar, Nikhil Narasimhan, and Ashwin Gopinath. 2026. The price of meaning: Why every semantic memory system forgets. arXiv:2603.27116 https://arxiv.org/abs/2603.27116

[6] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess Smidt, and Boris Kozinsky. 2022. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications* 13, 1 (2022), 1–11. https://www.nature.com/articles/s41467-022-29939-5.pdf

[7] Tarek Besold, Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luis C. Lamb, Priscila Vieira Lima, Leo de Penning, Gadi Pinkas, et al. 2022. Neural-symbolic learning and reasoning: A survey and interpretation. In *Neuro-symbolic artificial intelligence: The state of the art*. IOS press, 1–51.

[8] Yufan Cai, Zhe Hou, David Sanan, Xiaokun Luan, Yun Lin, Jun Sun, and Jin Song Dong. 2025. Automated Program Refinement: Guide and Verify Code Large Language Model with Refinement Calculus. *Proc. ACM Program. Lang.* 9, POPL, Article 69 (Jan. 2025), 33 pages. doi:10.1145/3704905

[9] Zhe Chen, Yuehan Wang, Bin Zhao, Jing Cheng, Xin Zhao, and Zongtao Duan. 2020. Knowledge graph completion: A review. *IEEE Access* 8 (2020), 192435–192456. doi:10.1109/ACCESS.2020.3030076

[10] Ryan Cory-Wright, Cristina Cornelio, Sanjeeb Dash, Bachir El Khadir, and Lior Horesh. 2024. Evolving scientific discovery by unifying data and background knowledge with AI Hilbert. *Nature Communications* 15, 1 (2024), 5922.

[11] Andrew Cropper and Sebastijan Dumančić. 2022. Inductive logic programming at 30: A new introduction. *Journal of Artificial Intelligence Research* 74 (2022), 765–850.

[12] Salvatore Cuomo, Vittorio Di Cola, Francesco Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. 2022. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal of Scientific Computing* 92, 3 (2022), 1–56. https://link.springer.com/content/pdf/10.1007/s10915-022-01939-z.pdf

[13] Artur d'Avila Garcez and Luis C. Lamb. 2003. Reasoning about Time and Knowledge in Neural-Symbolic Learning Systems. In *Advances in Neural Information Processing Systems (NIPS)*, S. Thrun, L. Saul, and B. Schölkopf (Eds.), Vol. 16. MIT Press. https://proceedings.neurips.cc/paper_files/paper/2003/file/347665597cbfaef834886adbb848011f-Paper.pdf

[14] Artur d'Avila Garcez, Luis C. Lamb, and Dov M. Gabbay. 2009. *Neural-Symbolic Cognitive Reasoning*. Springer.

[15] Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, et al. 2022. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature* 610, 7930 (2022), 47–53.

[16] Artur d'Avila Garcez and Luís C. Lamb. 2023. Neurosymbolic AI: the 3rd wave. *Artif. Intell. Rev.* 56, 11 (2023), 12387–12406. doi:10.1007/S10462-023-10448-W

[17] Eleonora Giunchiglia, Alex Tatomir, Mihaela Cătălina Stoian, and Thomas Lukasiewicz. 2024. CCN+: A Neuro-symbolic Framework for Deep Learning with Requirements. *International Journal of Approximate Reasoning* 171 (2024), 109124. doi:10.1016/j.ijar.2024.109124

[18] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. 2019. Hamiltonian neural networks. *Advances in Neural Information Processing Systems* 32 (2019). https://proceedings.neurips.cc/paper_files/paper/2019/file/26cd8ecadce0d4efd6cc8a8725cbd1f8-Paper.pdf

[19] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge Graphs. *ACM Comput. Surv.* 54, 4, Article 71 (July 2021), 37 pages. doi:10.1145/3447772

[20] Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. 2018. DeepProbLog: Neural Probabilistic Logic Programming. In *Advances in Neural Information Processing Systems*, Vol. 31. Curran

Associates, Inc. arXiv:1805.10872 https://arxiv.org/abs/1805.10872

[21] Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. 2019. An Introduction to AnyBURL. In *KI 2019: Advances in Artificial Intelligence - 42nd German Conference on AI, Kassel, Germany, September 23-26, 2019, Proceedings (Lecture Notes in Computer Science, Vol. 11793)*, Christoph Benzmüller and Heiner Stuckenschmidt (Eds.). Springer, 244–248. doi:10.1007/978-3-030-30179-8_20

[22] Judea Pearl. 2000. *Causality.* Cambridge University Press. 2nd edition, 2009.

[23] J. Pearl and D. Mackenzie. 2018. *The book of why: the new science of cause and effect.* Basic Books.

[24] D. Purohit, Y. Chudasama, M. Torrente, and M.-E. Vidal. 2024. VISE: Validated and invalidated symbolic explanations for knowledge graph integrity. In *Proceedings of the First Workshop on Explainable Artificial Intelligence for the Medical Domain (EXPLIMED).*

[25] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378 (2019), 686–707. https://www.sciencedirect.com/science/article/pii/S0021999118307125

[26] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems* 36 (2023), 68539–68551.

[27] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward causal representation learning. *Proc. IEEE* 109, 5 (2021), 612–634.

[28] Amit Sharma, Jake Hofman, and Duncan Watts. 2015. Estimating the causal impact of recommendation systems from observational data. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. 453–470.

[29] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484–489.

[30] Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. ViperGPT: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*. 11888–11898.

[31] Michael van Bekkum, Maaike de Boer, Frank van Harmelen, André Meyer-Vitali, and Annette ten Teije. 2021. Modular design patterns for hybrid learning and reasoning systems: a taxonomy, patterns and use cases. *Applied Intelligence* 51, 9 (Sept. 2021), 6528–6546. doi:10.1007/s10489-021-02394-3

[32] Frank Van Harmelen, Vladimir Lifschitz, and Bruce Porter. 2008. *Handbook of knowledge representation*. Vol. 1. Elsevier.

[33] Emile van Krieken, Erman Acar, and Frank van Harmelen. 2022. Analyzing differentiable fuzzy logic operators. *Artificial Intelligence* 302 (2022), 103602.

[34] Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, Michal Walczak, Jochen Garcke, Christian Bauckhage, and Jannis Schuecker. 2023. Informed Machine Learning – A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems. *IEEE Transactions on Knowledge and Data Engineering* 35, 1 (2023), 614–633. doi:10.1109/TKDE.2021.3079836

[35] Anjiang Wei, Tarun Suresh, Jiannan Cao, Naveen Kannan, Yuheng Wu, Kai Yan, Thiago Teixeira, Ke Wang, and Alex Aiken. 2025. CodeARC: Benchmarking reasoning capabilities of llm agents for inductive program synthesis. In *Proceedings of the 2025 Conference on Language Modeling*.

[36] Adam White, Kwun Ho Ngan, James Phelan, Kevin Ryan, Saman Sadeghi Afgeh, Constantino Carlos Reyes-Aldasoro, and Artur d'Avila Garcez. 2023. Contrastive counterfactual visual explanations with overdetermination. *Machine Learning* 112 (2023), 3497–3525. doi:10.1007/s10994-023-06333-w

[37] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. ReAct: Synergizing reasoning and acting in language models. In *Eleventh International Conference on Learning Representations*.

[38] Honghua Zhang, Po-Nien Kung, Masahiro Yoshida, Guy Van den Broeck, and Nanyun Peng. 2024. Adaptable logical control for large language models. *Advances in Neural Information Processing Systems* 37 (2024), 115563–115587.

[39] Yedi Zhang, Yufan Cai, Xinyue Zuo, Xiaokun Luan, Kailong Wang, Zhe Hou, Yifan Zhang, Zhiyuan Wei, Meng Sun, Jun Sun, Jing Sun, and Jin Song Dong. 2025. Position: Trustworthy AI Agents Require the Integration of Large Language Models and Formal Methods. In *42nd Int. Conf. on Machine Learning (ICML) Position Paper Track*. https://openreview.net/forum?id=wkisIZbntD