Symbolic Reasoning Methods in Neural Architectures and Large Language Models

Adrita Barua[,a,∗], Avishek Das[a,∗], Moumita Sen Sarma[a,∗] and Pascal Hitzler[a,∗]
∗ Kansas State University, Department of Computer Science, Manhattan, Kansas, USA
[a] Corresponding authors: adrita@ksu.edu, avishek@ksu.edu, moumita@ksu.edu, hitzler@ksu.edu
[a] ORCID IDs: 0000-0002-3287-7443, 0000-0002-1589-8322, 0009-0003-6644-6531, 0000-0001-6192-3472

Abstract

This chapter presents a comprehensive overview of contemporary approaches that integrate neural networks and large language models (LLMs) with classical symbolic reasoning. We review the evolution of methods designed to embed logical inference within neural architectures and explore recent advances in prompting strategies, hybrid reasoning frameworks, retrieval-augmented learning, and reinforcement-based reasoning optimization methods. We discuss the symbolic foundations of logical reasoning and then analyze how neural and LLM-based methods have progressively evolved to emulate, extend, and optimize symbolic reasoning across diverse tasks. Finally, we explore emerging neurosymbolic paradigms that unify neural and symbolic reasoning to achieve interpretable, scalable, and generalizable intelligence. Our analysis underscores the growing importance of neurosymbolic AI as a foundational direction for developing reliable and explainable reasoning systems.

Keywords: Symbolic Reasoning, LLMs, Neurosymbolic Reasoning

1. Introduction

Symbolic reasoning is the foundation of classical artificial intelligence (AI), concerned with the representation and manipulation of explicit symbols and rules to derive logical conclusions. In symbolic systems, knowledge is represented as well-structured symbols, such as predicates, logical operators, or relations; and reasoning is carried out through rule-based inference mechanisms, including deduction, induction, and abduction, to produce new conclusions from given premises [73, 89]. This paradigm relies on the explicit representation of facts (e.g., *Human(Socrates)*) and rules (e.g., $\forall x(Human(x) \rightarrow Mortal(x))$) that can be systematically combined to derive new knowledge (e.g., *Mortal(Socrates)*). Symbolic reasoning thus emphasizes interpretability, consistency, and guaranteed soundness, qualities that stem from its reliance on formal logic and deterministic computation rather than statistical generalization.

Traditional symbolic reasoners, such as Prolog-based logic engines [17, 54], first-order theorem provers [53], or ontology-based reasoners like Pellet [100] and HermiT [98], operate deterministically, guaranteeing soundness (no false conclusions) and completeness (all true conclusions can be derived) under well-defined logical frameworks [3]. While these systems excel at formal correctness and interpretability, they rely on explicitly defined knowledge bases and rules, making them brittle in open-ended environments where knowledge is incomplete, noisy, or linguistically expressed. Their dependence on symbolic representations also makes them sensitive to the “knowledge engineering bottleneck”: the difficulty of manually encoding human knowledge in logic-based formalisms [4].

Between the dominance of classical logic-based systems and the recent emergence of large language models, an important intermediate phase transformed how symbolic reasoning was implemented in AI. During this period, neural architectures began to internalize symbolic structure through differentiable and embedding-based formulations. Approaches such as TransE and DistMult [10, 128] demonstrated how entities and relations could be encoded as vectors in continuous space, while differentiable-logic frameworks like Logic Tensor Networks and Neural Theorem Provers [21, 88] reformulated logical inference as gradient-based computation. Collectively, these methods bridged the gap between symbolic reasoning and distributed representations, providing a conceptual foundation for the reasoning behaviors that later emerged in transformer-based language models.

In recent years, neural networks and large language models (LLMs) have gained traction for performing symbolic reasoning tasks due to their impressive ability to model associations, compositionality, and analogical patterns from large-scale data [120, 52, 44]. Unlike traditional reasoners that operate over hand-crafted symbols, neural models learn latent representations of knowledge that encode implicit relationships between entities and concepts. Through techniques such as chain-of-thought prompting [120], program-aided reasoning [27], and tool-augmented inference [93], LLMs can generate step-by-step logical explanations, produce executable symbolic expressions, or interface with external solvers to emulate reasoning behavior. Similarly, retrieval-augmented or self-reflective architectures [36, 50] combine generative reasoning with grounding mechanisms to improve factual accuracy and interpretability. These methods enable LLMs to perform structured reasoning in mathematical, logical, and commonsense domains, and in some cases even outperform specialized symbolic systems on certain benchmarks.

However, the reasoning abilities of LLMs remain contentious and limited [118]. Although they can simulate symbolic reasoning through the generation of linguistic patterns, empirical studies show that their reasoning traces are often weak, inconsistent, or semantically shallow, more reflective of statistical association than formal inference [9, 110]. They lack the soundness and verifiability of symbolic systems, and are prone to hallucination, producing logically coherent but factually false or unverifiable statements [6]. Moreover, the reliance of LLMs on textual reasoning makes them sensitive to prompt formulation and incapable of guaranteeing logical correctness without external verification [90, 63]. Consequently, while neural models demonstrate emergent reasoning behavior, they do not yet exhibit genuine symbolic reasoning in the classical sense.

To overcome these limitations, neurosymbolic approaches have emerged that integrate the expressivity and generalization of neural models with the rigor and interpretability of symbolic reasoning [8, 29, 28]. In such hybrid architectures, neural networks can learn latent features or embeddings from unstructured data, while symbolic modules, such as logic-based reasoners, knowledge graphs, or differentiable logic layers, provide structure, constraints, and explainability. Recent works have demonstrated that combining symbolic inductive bias with neural representations improves both data efficiency and logical consistency in reasoning tasks [59]. This synthesis of learning and reasoning represents a promising direction toward AI systems that can not only generate plausible solutions but also justify and verify them according to formal principles. The remainder of this chapter explores the evolution of such models, focusing on how neural networks and large language models have been leveraged for symbolic reasoning tasks, the spectrum of neurosymbolic integration methods, and the open challenges that define the future of this rapidly evolving field.

2. Historical Foundations

Before exploring how neural and large language models emulate or integrate symbolic reasoning, it is essential to understand the classical methods that shaped this field. Symbolic reasoning emerged from the early days of artificial intelligence as a way to represent and manipulate explicit knowledge through logic and rules [30, 5, 48, 115]. These systems reasoned over well-defined symbols—such as predicates, relations, and operators—to derive new conclusions in a sound and interpretable manner [48, 115, 30].

In this section, we review the foundational paradigms that underpin modern reasoning research. We begin with rule-based and logic programming systems, which operationalize logical inference through rule chaining and unification [83, 91]. We then discuss automated theorem proving, which formalizes deduction through proof search in first-order logic [65, 40], followed by ontology-based reasoning methods that leverage structured knowledge representations for semantic inference [83, 64]. Together, these approaches form the conceptual and algorithmic basis from which neurosymbolic and LLM-based reasoning frameworks have evolved.

2.1. Rule-Based and Logic Programming Systems

Early symbolic reasoning in artificial intelligence was primarily expressed through production systems and rule-based expert systems, where reasoning was formalized as a collection of conditional rules of the form "`IF condition THEN action`." While these systems demonstrated the feasibility of representing expert knowledge in a symbolic form, they operated procedurally, lacking the formal semantics of logic and the flexibility required for compositional inference. The emergence of logic programming in the early 1970s represented a conceptual breakthrough—recasting symbolic reasoning as a declarative computational paradigm grounded in first-order logic. A central theoretical milestone in this transition was the formulation of predicate logic as a programming language, first articulated in [54]. The author proposed that implications of the form:

$$B \leftarrow A_1, A_2, ..., A_n$$

could be interpreted procedurally as "To prove B, prove each of $A_1, A_2, \ldots, A_n$." This interpretation established a unifying perspective between logical deduction and computation, a relationship often encapsulated in the principle that computational processes can be viewed as controlled logical inference. Here, it was demonstrated that proof procedures such as resolution could serve as the operational counterpart to logical implication, with proofs corresponding to computations and goals representing procedure calls. Also, the author further identified the Horn Clause, subset of predicate logic as the computationally tractable core of this paradigm, enabling deterministic yet declarative reasoning.

These theoretical ideas culminated in the development of Prolog (Programmation en Logique), the first practical logic programming language. Prolog operationalized logical inference through linear resolution with unification and backtracking control, providing an efficient and expressive mechanism for declarative computation [19]. It demonstrated that first-order logic could serve simultaneously as a representation formalism and an executable specification language—capable of performing syntactic, semantic, and inferential tasks in a unified framework. Prolog's design embodied the dual interpretation of logic as both knowledge and procedure: programs consist of logical clauses defining relations among entities, while computation corresponds to automated reasoning over these relations through goal satisfaction. This paradigm provided a foundation for declarative AI systems that could explain their inferences in symbolic terms, while also influencing subsequent developments in automated theorem proving, knowledge representation, and neurosymbolic reasoning.

2.2. Automated Theorem Proving

Automated theorem proving (ATP) represents one of the earliest and most influential branches of symbolic reasoning, concerned with the mechanical derivation of logical consequences from a set of formal premises. In contrast to logic programming, which operationalizes reasoning as goal-directed computation, theorem proving seeks general algorithms capable of verifying or discovering proofs within formal logical systems. These approaches provided the first demonstration that deductive reasoning could be fully automated under well-defined logical calculi [125].

The theoretical foundation of modern theorem proving lies in the Resolution principle, introduced as a complete inference rule for first-order logic [87]. The resolution method reduces the problem of entailment to one of unsatisfiability, systematically resolving pairs of clauses until a contradiction is derived. This mechanism elegantly unifies inference and proof search under a single rule, eliminating the need for multiple inference schemata. Its reliance on unification—the computation of the most general substitution that makes two logical atoms identical—became a cornerstone of later reasoning systems and directly influenced the design of logic programming languages such as Prolog [25].

Building upon this foundation, numerous systems extended and optimized the resolution calculus for large-scale automated deduction. Tools such as Vampire [85], E Prover [96], and Prover9 [71] introduced sophisticated indexing, clause weighting, and heuristic search mechanisms to improve proof efficiency. These systems enabled reasoning over complex first-order theories and found extensive application in formal verification, software correctness, [94, 20] and ontology consistency checking [97], where logical soundness and completeness are essential. Through these developments, theorem provers established a rigorous computational framework for reasoning within both first-order and higher-order logics [79, 102].

2.3. Ontology-Based and Description Logic Reasoning

Ontology-based reasoning extends symbolic inference to structured knowledge representations, where concepts, entities, and relations are explicitly modeled within formal ontological frameworks. Unlike logic programming or theorem proving, which operate on general-purpose logical expressions, ontology reasoning emphasizes terminological knowledge—the formal specification of classes, properties, and constraints that describe a domain of discourse. This paradigm underlies the field of Description Logics (DLs), a family of knowledge representation formalisms designed to balance logical expressivity with computational tractability [3].

In Description Logic, knowledge is divided into two components: the TBox, which encodes conceptual hierarchies and axioms about classes (e.g., "every mammal is an animal"), and the ABox, which contains assertions about individual instances (e.g., "Dolphin is a mammal"). Reasoning in this framework involves tasks such as subsumption checking (determining whether one class is a subclass of another), consistency testing (verifying the logical coherence of an ontology), and instance classification (inferring the most specific concept that an individual belongs to). These reasoning services form the basis of ontology management in systems such as the Semantic Web and are essential for ensuring logical soundness and interoperability across heterogeneous knowledge sources.

To support automated reasoning over ontologies, specialized engines known as Description Logic (DL) reasoners were developed. Among the most prominent are Pellet [100], HermiT [98], and FaCT++ [108], each implementing distinct algorithms for DL inference. Pellet was one of the first practical OWL-DL reasoners, supporting full reasoning over Web Ontology Language (OWL) constructs and providing capabilities such as consistency checking and query answering through SPARQL-DL interfaces. HermiT introduced a highly efficient hypertableau algorithm, enabling scalable reasoning over large and expressive ontologies. These systems demonstrated that sound and complete reasoning could be achieved over ontologies encoded in OWL, a W3C standard grounded in Description Logic semantics.

3. Symbolic Reasoning in Modern Neural Architectures

As neural networks matured, researchers began exploring methods to enable these models to perform symbolic reasoning without explicit logic engines or rule-based systems. This phase, preceding the era of large language models, focused on demonstrating that neural architectures could internally learn and approximate symbolic structures—capturing relations such as hierarchy, transitivity, and entailment through learned representations.

This section reviews key developments from this pre-LLM stage, covering three main directions:

1. **Embedding-based methods**, which encode symbolic entities and relations as geometric patterns in vector space;
2. **Differentiable-logic approaches**, which express logical inference as continuous, trainable operations within neural systems.

3.1. Embedding-Based Symbolic Approximation

A major advance in symbolic reasoning within neural architectures is the development of neural embeddings—vector-space representations of symbols, relations, and logical axioms. These embeddings allow neural networks to approximate symbolic semantics while operating on continuous data.

3.1.1. Knowledge Graph and Relational Embeddings

Knowledge-graph embeddings such as TransE [10], DistMult [128], ComplEx [107], and RotatE [103] treat each triple ⟨head h, relation r, tail t⟩ as a constraint in latent space.

- *TransE* models relations as translations: h + r ≈ t. Training minimizes the distance between translated heads and true tails while maximizing it for corrupted triples, encouraging geometric regularities that correspond to relational patterns such as symmetry or hierarchy. Although simple and scalable, TransE struggles with one-to-many or many-to-one relations because a single translation vector cannot represent multiple valid tails.
- *DistMult* tried to address this issue by introducing bilinear scoring functions f(h,r,t) = $h^\top diag(r)$, where diag(r) is a diagonal matrix of relation parameters. This formulation allows interactions between each dimension of h and t weighted by r, but it is inherently symmetric (f(h,r,t)=f(t,r,h)), making it capturing only symmetric relations and not the asymmetric relations such as parent-of.
- *ComplEx* overcomes the limitation of models like DistMult that can only represent symmetric relations, by extending the bilinear formulation into the complex domain. In this approach, entity and relation embeddings are complex-valued vectors, $h, r, t \in C^k$, rather than real-valued ones. The scoring function for a triple ⟨h, r, t⟩ is defined as $f(h, r, t) = Re(\langle \mathbf{h} \odot \mathbf{r} \odot \mathbf{t} \rangle)$ where $\odot$ is the Hadamard (element-wise) product, and Re() extracts the real part.
- *RotatE* represents relations as rotations in complex space, improving modeling of inversion and composition.

These embeddings implicitly encode symbolic regularities—transitivity, reflexivity, and type hierarchies—through geometry, enabling reasoning tasks like link prediction and rule discovery. RDF2Vec [86] adapts this paradigm to RDF/OWL graphs by performing random walks over triples and training a skip-gram (Word2Vec-style) model on the resulting sequences. The learned embeddings capture semantic proximity based on graph co-occurrence, thereby translating structured ontologies into distributional representations that neural networks can process.

3.1.2. Ontology- and Logic-Aware Embeddings

While graph-based embeddings capture relational proximity, they do not explicitly preserve the formal semantics of symbolic logic. Ontology- and logic-aware embeddings address this limitation by incorporating axioms, constraints, and reasoning rules from formal knowledge representation languages—such as OWL 2 or Description Logics. Each approach seeks to translate the rules of logic into geometry, allowing neural architectures to approximate symbolic reasoning through differentiable operations.

- Order Embeddings [138] provide one of the simplest ways to represent hierarchical or taxonomic relations—for example, that cat *is a* kind of animal. The key idea is to encode these "is-a" relations directly in vector space using coordinate-wise inequalities. Each concept is represented as a vector $\mathbf{x} \in \mathbb{R}^n$, and the model enforces that if concept x is more specific (a subclass) than concept y, then every coordinate of its vector is smaller or equal: $x \sqsubseteq y$ is modeled by $x \sqsubseteq y \iff x_i \leq y_i, \forall i$. In simple terms, a subclass vector lies below its superclass in every dimension of the space. If any coordinate of x exceeds the corresponding coordinate of y, it violates the hierarchy. The model penalizes such violations using the loss

$$L(x, y) = ||\max(0, \mathbf{x} - \mathbf{y})||^2$$

which becomes zero when all coordinates respect the order.

- Box Embeddings [111] extend the idea of hierarchy by representing each concept not as a single point, but as a region — specifically, a box — in a multi-dimensional space. Instead of assigning one vector to a class, the model defines it by two vectors: a lower corner l_c and an upper corner u^c . These define an n-dimensional box:

$$B_C = \{\mathbf{x} \mid \mathbf{l}_C \leq \mathbf{x} \leq \mathbf{u}_C\}$$

In this setup, logical inclusion (e.g., *dog* $\sqsubseteq$ *animal*) simply means that one box is completely contained inside another:

$$A \sqsubseteq B \iff B_A \subseteq B_B$$

If two concepts overlap (for instance, pet and mammal), their boxes intersect; if they are unrelated, their boxes are far apart. During training, the model adjusts the corners of these boxes so that containment holds for true relations and fails for false ones. Thus, Box Embeddings naturally combine logical hierarchy with uncertainty, giving neural models an intuitive way to reason about inclusion, overlap, and disjointness in a differentiable space.

- EL Box Embeddings [55] extend the basic box-embedding idea to represent formal logical axioms from the Description Logic EL — the logical foundation of OWL 2 EL ontologies. The goal is to make geometric regions in vector space behave like logical concepts and relations, so that reasoning follows the same patterns as in symbolic logic. Each concept (or class) is modeled as a box, just like in standard Box Embeddings. In addition, relations (or roles) are represented as translation vectors that can move one box to another, allowing the model to express existential restrictions and other logical constructs. Logical axioms in EL are then reinterpreted as geometric containment rules:

$$C \sqsubseteq D \Rightarrow B_C \subseteq B_D \quad \text{(subsumption)}$$

$$C \sqcap D \sqsubseteq E \Rightarrow B_C \cap B_D \subseteq B_E \quad \text{(conjunction)}$$

$$\exists R.C \sqsubseteq D \Rightarrow B_C + r_R \subseteq B_D \quad \text{(existential restriction)}$$

Here B_x denotes the box for concept x, and r_R is the translation vector for relation R. During training, the model minimizes small violations of these geometric constraints, effectively learning a continuous space where the ontology's logical rules hold approximately true. In this way, EL Box Embeddings provide a bridge between symbolic description logic and neural representation learning—allowing logical reasoning to be expressed entirely through smooth, differentiable operations.

- OWL2Vec [38] and its enhanced version OWL2Vec* [13] integrate lexical, structural, and logical contexts of ontologies into corpus-based embeddings, providing a bridge between symbolic graphs and neural text models. The process begins by linearizing the ontology: axioms, classes, relations, and labels are converted into sequences of tokens, much like sentences. Then, random walks are performed over the ontology graph—tracing paths such as

$$\text{Animal} \xrightarrow{\text{hasPart}} \text{Leg} \xrightarrow{\text{isPartOf}} \text{Body}.$$

  Each of these walks becomes a "sentence," and a skip-gram model (like Word2Vec) is trained to predict nearby tokens within a context window. The objective is to maximize the likelihood of observing context words c given a target word w:

$$\max_{\theta} \sum_{(w,c)\in D} \log P(c|w;\theta)$$

  where $P(c|w)$ is computed using vector similarity between the embeddings of w and c. Essentially, the model learns that entities or concepts frequently appearing in similar logical or structural neighborhoods should have similar embeddings. OWL2Vec* further enriches this process by adding lexical information—such as class labels, synonyms, and textual descriptions—so that linguistic and logical cues jointly shape the embeddings. The result is a representation space where proximity reflects semantic relatedness rather than strict logical entailment.

Together, these models illustrate a continuum: from distributional embeddings that capture semantic co-occurrence, to geometric and logic-preserving embeddings that encode formal relationships. The former excel at scalability and generalization, while the latter achieve higher logical fidelity at the cost of computational complexity.

3.2. Differentiable Logic and Neural Reasoners

Building upon embedding-based representations, a complementary line of research explores how logical inference itself can be formulated as a differentiable process within neural architectures. Although some of these differentiable-logic frameworks were proposed slightly earlier than ontology-embedding models, we present them here to maintain a conceptual progression—from neural representations of symbolic knowledge to neural mechanisms that perform reasoning itself. Instead of merely encoding symbolic relationships, these approaches define logical operators—such as conjunction, implication, and unification—as continuous functions that can be optimized through gradient descent. This allows neural networks to perform approximate reasoning while remaining fully differentiable and end-to-end trainable.

Representative models—including Logic Tensor Networks [21], Neural Theorem Provers [88], Neural Logic Programming [129], and DeepProbLog [70] illustrate different strategies for embedding formal reasoning principles into neural computation.

- *Logic Tensor Networks(LTN)* introduced one of the first differentiable formulations of first-order logic. Each logical predicate is represented by a neural function that outputs a soft truth value in the range [0,1]. Formulas are then composed using smooth approximations of logical connectives, for example:

$$AND : a \wedge b \approx a \cdot b$$

$$OR : a \vee b \approx a + b - a \cdot b$$

$$NOT : \neg a \approx 1 - a$$

  Training maximizes the overall satisfaction of known axioms, effectively turning logical consistency into a learning objective. LTNs thus allow background knowledge and logical constraints to guide representation learning, unifying logic and neural optimization. Their limitation lies mainly in scalability, since evaluating large sets of grounded formulas can be computationally expensive.
- *Neural Theorem Provers(NTP)* reinterpret the process of symbolic theorem proving in continuous space. In classical logic, unification checks whether two symbols can be made identical by substitution; NTP replaces this discrete operation with a differentiable similarity function:

$$\text{unify}(x, y) = \exp\left(-\|x - y\|^2\right)$$

  where x and y are the embeddings of symbols x and y. Proofs are built recursively as soft inference trees, and each proof path receives a confidence score based on the product of its unification strengths. Training encourages embeddings that maximize the scores of valid proofs, effectively learning to make true statements geometrically consistent. NTP thereby converts logical deduction into a differentiable pattern-matching process—retaining the structure of theorem proving while enabling gradient-based learning. However, the recursive expansion of proof trees can become computationally intensive, motivating more scalable successors such as Neural LP.
- *Neural Logic Programming (Neural LP)* extends differentiable reasoning to large knowledge bases by expressing multi-hop inference as matrix operations. Each relation R is represented by an adjacency matrix M_R; multi-step reasoning chains

are realized through matrix multiplication: $M_{R_2} M_{R_1} v_A$, where v_A is a one-hot vector representing entity A. By learning attention weights over relation sequences, Neural LP can infer rules such as

$$\text{ancestor}(x, z) \leftarrow \text{parent}(x, y) \wedge \text{parent}(y, z)$$

The model learns both the structure and the confidence of such rules in a fully differentiable way, combining the interpretability of logic programs with the flexibility of neural learning. Compared with NTP, it scales more efficiently while still producing explicit, human-readable rules.

- DeepProbLog extends probabilistic logic programming by allowing neural networks to provide probabilities for certain facts or predicates. It builds on ProbLog, a system where each logical fact has an associated probability and inference is performed over all possible worlds. DeepProbLog adds neural modules so that some of these probabilities are learned from data rather than fixed manually. For example, a neural classifier might estimate $p = f_\theta(image)$ that an image contains a cat, while symbolic rules reason about ownership:

$$\text{ancestor}(x, z) \leftarrow \text{parent}(x, y) \wedge \text{parent}(y, z)$$

The neural output p serves as the probabilistic weight of the predicate isCat(y) within the logic program. During training, both the neural parameters θ and the probabilistic rule weights are optimized together to increase the likelihood of correct conclusions. This creates a single differentiable pipeline in which neural perception feeds into symbolic reasoning, and reasoning feedback can refine the neural models. This design unites low-level perception and high-level reasoning in a single differentiable framework, enabling probabilistic inference over neural predictions. DeepProbLog exemplifies how neural and symbolic components can cooperate seamlessly to perform reasoning grounded in real-world sensory data.

Differentiable-logic models mark a pivotal stage in the evolution of neural reasoning. While LTNs express logic as continuous truth values, NTP and Neural LP translate deductive inference into differentiable search or matrix computation, and DeepProbLog extends the paradigm to probabilistic and perceptual domains. Together, these systems show that the algebra of logic can be embedded directly within the calculus of gradients—providing the conceptual and methodological foundations upon which large language models later build their emergent reasoning capabilities.

4. Large Language Models and Symbolic Reasoning Capabilities

Large Language Models (LLMs) have shown remarkable reasoning capabilities with the emergence of advanced reasoning models like OpenAI o3 [1] and DeepSeek-R1 [32] in performing tasks that traditionally required symbolic reasoning. While symbolic reasoning conventionally depends on explicit representations of facts and rules, LLMs rely on vast linguistic corpora to learn latent representations that implicitly encode relations, regularities, and compositional patterns. Consequently, the study of symbolic reasoning in LLMs has evolved from early prompting-based simulations of reasoning to more sophisticated grounding and fine tuning of these models, retrieval and agent-based reasoning frameworks, and the emerging reinforcement learning architectures. This section reviews these methods, emphasizing their methodology, applications, advantages, and limitations.

4.1. Prompt-Based and Chain-of-Thought Reasoning

Early efforts to elicit symbolic reasoning behavior from LLMs centered on the use of *prompt engineering*—the design of carefully structured natural language inputs that guide models to produce stepwise, logically coherent responses. The introduction of the *Chain-of-Thought (CoT)* prompting framework by Wei et al. [121] demonstrated that encouraging models to "think aloud" by generating intermediate reasoning steps could significantly improve accuracy in arithmetic and logic-based reasoning tasks. This approach was later refined through methods such as *Self-Consistency* [116] and *Least-to-Most Prompting* [139], which enhance reliability by sampling multiple reasoning paths or decomposing complex problems into smaller sub-problems. To further improve LLM reasoning capabilities *Tree-of-Thoughts (ToT)* is introduced. ToT allows LLMs to make deliberate decisions by considering multiple different reasoning paths and self-evaluating choices to decide the next course of action, as well as looking ahead or backtracking when necessary to make global choices [132].

Prompt-based techniques are effective for tasks such as symbolic arithmetic, logical deduction, and commonsense reasoning benchmarks (e.g., GSM8K [18], and LogiQA [66]). Their major advantage lies in their ability to leverage pretrained LLMs without finetuning, which makes it scalable and generalizable to many use cases. However, these methods remain limited by their dependence on natural language formulations, making reasoning fragile to prompt variations, and often fail to maintain logical consistency or soundness, since generated reasoning traces are not verified against formal symbolic rules [109, 11].

[1] `https://openai.com/index/o3-o4-mini-system-card/`

4.2. Program-Aided and Tool-Augmented Reasoning

To overcome the constraints of purely linguistic reasoning, subsequent research introduced methods that combine LLMs with external tools capable of performing formal or arithmetic computations. The *Program-Aided Language Models (PAL)* framework by [27] exemplifies this trend: the model generates Python code representing logical or mathematical reasoning steps, which is then executed by an interpreter to obtain the final answer. Similarly, the *Toolformer* model [93] trained LLMs to autonomously decide when and how to call external APIs such as calculators or knowledge bases, effectively bridging neural text generation with symbolic computation.

These models have been successfully applied in symbolic reasoning domains such as theorem proving, equation solving, and knowledge graph querying [27, 93, 136]. Their integration of external tools enhances factual correctness and numerical precision. However, the reliance on external execution environments and the absence of internal symbolic representations mean that reasoning remains procedural rather than conceptual—LLMs act as controllers rather than genuine symbolic reasoners.

4.3. Retrieval-Augmented and Knowledge-Guided Reasoning

A more recent direction integrates LLMs with external knowledge sources or structured memory to support symbolically grounded reasoning. Retrieval-Augmented Generation (RAG) frameworks [56] enhance reasoning accuracy by fetching relevant symbolic or factual information from a database or knowledge graph, which is then incorporated into the model’s generative reasoning chain. In the symbolic reasoning context, such systems have been used to assist rule induction, ontology-based reasoning, and logical inference [58]. Variants of RAG, such as Self-RAG [2], Fusion-in-Decoder (FiD) [45], and GraphRAG [34], further extend retrieval-augmented reasoning by improving evidence integration, multi-hop reasoning, and symbolic grounding through knowledge-graph–based retrieval.

These architectures achieve improved factual grounding, interpretability, and generalization to multi-hop reasoning tasks. Nevertheless, they still rely on the probabilistic text generation capabilities of LLMs to perform inference, which can lead to semantic inconsistencies, hallucinated premises, or incorrect logical deductions if retrieval results are misinterpreted [12].

4.4. Reflection, Planning, and Meta-Reasoning Frameworks

The most advanced approaches toward symbolic reasoning with LLMs introduce elements of meta-cognition and self-evaluation. Frameworks such as *ReAct* [133] and *Reflexion* [99] allow LLMs to interleave reasoning and action: generating hypotheses, testing them using external tools, and revising their conclusions based on feedback. Self-improvement frameworks such as Self-Taught Reasoner [134] and Self-

Refine [69] extend this concept by enabling LLMs to iteratively assess and refine their own reasoning chains.

These techniques represent a convergence between neural reasoning and symbolic control, where models not only produce logical sequences but also evaluate, back-track, and optimize reasoning paths. Their advantages include robustness, adaptability to complex reasoning domains, the ability to reflect on their mistakes, and provide guidance to improve problem solving [84].

However, even advanced self-reflective architectures struggle to guarantee logical validity and verifiable soundness, and often fail to scale to reasoning tasks that require quantification, recursion, or higher-order logic - because their internal reflection mechanisms can suffer 'performance collapse', losing consistency when reasoning paths grow long or complex [101].

4.5. Instruction Fine-Tuning and Alignment for Symbolic Reasoning

Instruction Fine-Tuning (IFT) has played a pivotal role in enhancing the reasoning and symbolic understanding capabilities of LLMs. Building upon the success of supervised instruction models such as *FLAN* [119] and *T0* [92], IFT exposes models to large-scale collections of task-oriented examples framed as natural language instructions. By learning to follow instructions phrased in diverse linguistic forms, the models generalize to unseen symbolic reasoning tasks without task-specific training. This training paradigm was later expanded through *Super-NaturalInstructions* [117] and *Alpaca* [105], which curate large instruction datasets that implicitly encode logical, arithmetic, and compositional reasoning.

IFT-based models have demonstrated strong performance on structured reasoning benchmarks such as GSM8K and MATH [18, 37], as well as logical reasoning benchmarks [112]. Their primary advantage lies in scalability and versatility. IFT enables LLMs to generalize reasoning behavior across symbolic domains without explicit rule encoding. Moreover, IFT can be combined with reasoning traces (e.g., Chain-of-Thought or stepwise rationales) to produce interpretable symbolic reasoning chains [134].

Despite these strengths, IFT-based reasoning remains inherently data-driven and limited by the coverage and quality of the instruction data. Since IFT optimizes for linguistic compliance rather than logical soundness, models may produce fluent but logically inconsistent explanations or fail to generalize beyond seen symbolic structures [109]. Furthermore, reliance on human-curated instruction datasets introduces annotation biases and restricts symbolic expressivity to natural language formulations. Consequently, while IFT enhances reasoning fluency and versatility, it alone cannot guarantee formally valid symbolic reasoning.

4.6. Agent-Based Models for Autonomous Reasoning

Recent research has explored *agentic frameworks* that conceptualize LLMs as autonomous reasoning agents capable of interacting with tools, environments, and other agents to accomplish symbolic reasoning tasks. In these systems, the LLM is decomposed into modular roles—such as *planner*, *reasoner*, and *executor*—that coordinate iteratively through explicit communication protocols or memory updates [126]. This paradigm is exemplified by frameworks such as *AutoGPT* [2], *MetaGPT*, and *Voyager*, which allow LLMs to perform long-horizon reasoning and goal-directed planning using autonomous, multi-agent architectures [39, 113]. Symbolic reasoning arises naturally in these systems as agents must construct, manipulate, and evaluate structured representations of goals, plans, and world states. To further enhance existing multi-agent reasoning approaches, Interactive Learning for LLM Reasoning (ILR) introduces an interaction-based paradigm where agents engage in structured discussions to collaboratively refine their reasoning before producing individual final answers [61]. This framework aims to improve reasoning quality through inter-agent dialogue, enabling models to independently resolve similar questions in future tasks.

Agent-based reasoning architectures have found applications in multi-step decision-making, automated theorem proving, robotic task planning, and scientific discovery [67, 114]. Their advantages include modularity, interpretability, and the ability to perform recursive reasoning across temporal contexts. Through multi-agent collaboration or reflection, such systems can collectively refine symbolic hypotheses or correct logical inconsistencies [22]. Additionally, their alignment with classical symbolic AI paradigms (e.g., planning and rule-based inference) makes them a natural interface between LLM-based reasoning and traditional symbolic frameworks.

Nevertheless, current agent-based systems face significant challenges in scalability, coordination, and reliability. Reasoning quality often degrades over long reasoning horizons due to accumulated errors or inconsistent message passing between agents [57]. Moreover, most agentic frameworks lack grounding in formal logic, relying instead on emergent coordination behaviors that remain fragile and non-deterministic [35]. Despite these limitations, agent-based reasoning represents a promising direction for integrating symbolic structure, procedural control, and interactive reasoning within large language models, moving closer to the vision of autonomous neurosymbolic intelligence.

4.7. Reinforcement Learning–Based Methods for Reasoning

Reinforcement learning (RL) has opened new paradigms in optimizing large language models (LLMs) to advance symbolic reasoning by training models to prefer logi-

[2] `https://github.com/Significant-Gravitas/AutoGPT`

cally valid reasoning trajectories through feedback and reward signals. Reinforcement Learning from Human Feedback (RLHF) [16, 75] laid the groundwork for aligning large language models with human-preferred reasoning quality through large-scale optimization. Reinforcement learning enables models to learn through trial and error, guided by reward signals rather than explicit labeled data. The model takes actions based on its current state and receives feedback in the form of rewards or penalties. Over time, this feedback allows the model to adjust its parameters and discover strategies that maximize cumulative rewards. Building on this foundation, recent advancements such as OpenAI’s o1 model [46] have demonstrated the power of reinforcement learning to enhance structured and coherent reasoning. These developments have inspired a new generation of reinforcement learning–based approaches aimed at improving Chain-of-Thought (CoT) reasoning, and mitigating issues like formulaic outputs and limited depth in multi-step inference. A notable leap forward is represented by DeepSeek-R1 [32], which introduces a novel reinforcement learning framework for logical reasoning. Its variant, DeepSeek-R1-Zero, is trained entirely through reinforcement learning without Instruction Fine-Tuning (IFT), achieves strong reasoning performance but faces challenges in readability and linguistic coherence [63]. To address this, DeepSeek-R1 incorporates a minimal amount of long-form CoT IFT data as a “cold start,” striking a balance between reasoning quality and linguistic fluency. Through iterative self-improvement and reasoning-driven data synthesis, DeepSeek-R1 effectively overcomes the limitations of human annotation, reducing mechanistic and repetitive reasoning patterns. This paradigm marks a significant shift toward RL–optimized reasoning, showcasing how RL can refine and extend the symbolic reasoning capabilities of LLMs beyond traditional supervised fine-tuning. The recent advancement in RL-based reasoning enhances multi-step inference, generalization across domains, and long-context reasoning capabilities.

A recent survey on Reinforced Reasoning with Large Language Models [127] provides a comprehensive overview of this emerging direction, outlining the three-stage RL process (data generation → reward modeling → optimization) for reasoning optimization:

(1) generating reasoning trajectories or problem–solution pairs through self-play or supervised demonstrations;

(2) designing reward models that evaluate reasoning quality based on correctness, process validity, and interpretability; and

(3) optimizing the policy via reinforcement objectives such as the PPO [95] method from RLHF or Direct Preference Optimization (DPO) [82] to reinforce high-quality reasoning patterns.

It also discusses the advancement of multi-step reasoning with the Outcome Reward Model (ORM) and Process Reward Model (PRM). ORM is primarily applied to complex reasoning tasks—such as mathematical problem solving—where LLMs must perform multi-step reasoning, like Chain-of-Thought, to ultimately reach an accurate solution. In these tasks, the reward feedback is typically only available after all reasoning steps are completed and the final solution is obtained. In contrast, PRM emphasizes the evaluation of intermediate steps rather than focusing solely on end-state outcomes. The reward in PRM is distributed across each reasoning step, rather than being concentrated at the final outcome. By providing nuanced feedback throughout the reasoning trajectory, PRM enables models to optimize their behavior with greater alignment to human preferences and complex task requirements. This approach is crucial for tasks that involve sequential decision-making, where intermediate steps or decisions significantly contribute to achieving the final goal.

More specialized methods apply symbolic reasoning rewards: for instance, [60] propose RL systems that are rewarded for consistency, logical validity, or execution correctness.

Reinforcement learning has thus emerged as a powerful alternative framework for training models to master reasoning processes, as it can mitigate the limitations associated with the high reliance on expensive, high-quality labeled datasets and the high computational costs of supervised fine-tuning. However, while RL-based reasoning improves coherence and alignment with correct symbolic outputs, it remains compute-intensive and heavily dependent on carefully engineered reward functions. Suboptimal rewards can reinforce shallow heuristics rather than genuine logical reasoning.

In summary, while the progression from prompt-based reasoning to tool-augmented, retrieval-based, and reflective reasoning architectures demonstrates substantial advancement in symbolic reasoning capabilities, current LLMs still operate primarily as probabilistic pattern generators that mimic reasoning behavior without genuine logical understanding or formal consistency. These limitations have motivated hybrid neurosymbolic approaches that explicitly integrate symbolic knowledge representations or logical constraints into neural architectures, which we discuss in the following section.

5. Neurosymbolic Reasoning with Large Language Models

Neurosymbolic approaches seek to combine the statistical generalization and linguistic fluency of modern large language models (LLMs) with the explicit, verifiable, and compositional reasoning afforded by symbolic methods. The motivating observation is straightforward: while LLMs can generate plausible multi-step explanations, their outputs are often fragile with respect to logical validity, verifiability, and long-chain compositionality. Conversely, symbolic systems (logic solvers, planners, constraint engines, knowledge graphs) provide algorithmic guarantees and exact inference, but do not naturally handle noisy, ambiguous, or language-rich inputs. Neurosymbolic approaches therefore pursue a spectrum of integration strategies, ranging from using symbolic systems to generate training data for LLMs, to pipelines where LLMs translate natural language into symbolic forms that are then solved, to tightly coupled end-to-end hybrids that embed symbolic modules into neural architectures or training objectives. Below, we review these directions, describe representative methods, discuss how they mitigate LLM reasoning failure modes, and summarize their application contexts.

5.1. $\text{LLM}_{\text{Symbolic}}$: Symbolic-Trace guided LLM reasoning

A growing class of neurosymbolic methods uses symbolic systems to generate reasoning data that can be distilled into LLMs through fine-tuning. In this paradigm, symbolic methods such as formal solvers or theorem provers first produce high-quality symbolic reasoning traces, which are then converted into natural language or programmatic reasoning trajectories to train the LLM. Such approaches have been adopted in domains like geometry theorem proving and algorithmic reasoning, where formal deduction engines or logical planners generate logically consistent solution paths in systems such as AlphaGeometry [106] and Planformer [76]. The resulting traces act as high-quality supervision signals, allowing LLMs to internalize structured reasoning. Recent methods such as LoGiPT [24] uses a solver-generated reasoning processes by learning to strictly adhere to solver syntax and grammar. Stream of Search (SoS) [26] follow a similar approach by using search-derived reasoning trajectories with search algorithms, such as DFS, BFS, MCTS, etc, and fine-tune LLMs to enable the model to learn to search and backtrack in the reasoning procedure. These methods aim to internalize the abilities of symbolic solvers in LLMs by constructing data sets to fine-tune the model, thus enhancing the LLM's reasoning abilities. It mitigates the scaling problem of formal logical solvers, while substantially improving the reasoning robustness of LLMs.

5.2. $\text{Symbolic}_{\text{LLM}}$: LLM aided Symbolic Reasoning

Another line of work operates in the direction, where the LLM acts as a front-end that translates natural language problems into symbolic representations, which are then subsequently solved or verified by external reasoning systems. In this framework, the model generates intermediate symbolic outputs, such as logical formulas, programs, or queries that can be executed or checked for correctness.

Program-aided methods like PAL [27], and Program of Thoughts (PoT) [14] prompt the LLM to express the reasoning process as executable code (e.g., Python or SQL) and the computation is relegated to an external program executor, thereby separating reasoning from computation. Related methods have been successfully applied to mathematical reasoning, code generation, robotics, etc.

Other methods such as LogicLM [77] and LINC [74] emphasize translation of natural language problems into formal logics or expressions and then leverage logic solvers to generate results. By delegating exact reasoning steps to formal solvers, these systems effectively reduce hallucinations and logical errors. However, their performance depends heavily on accurate autoformalization, the ability of the LLM to produce semantically correct symbolic expressions.

Both the symbolic solver or program-aided methods rely on external execution environments and the absence of internal symbolic representations mean that reasoning remains procedural rather than conceptual—LLMs act as controllers rather than genuine symbolic reasoners. Unlike these models, which translate natural language problems into formalized representations and directly invoke symbolic solvers or program executors to derive solutions, tool-augmented reasoning approaches leverage a broader range of external functionalities through structured interactions. These methods are inherently more complex, as they typically involve multiple stages of reasoning, including task planning, tool selection, tool invocation, and response generation [130]. For instance, Toolformer [93] demonstrated how LLMs can be fine-tuned to autonomously decide when and how to call external APIs such as calculators, translation systems, etc., thereby augmenting their reasoning capacity without explicit supervision. Similarly, VisProg [33] employs LLMs to generate Python-like modular programs that coordinate a variety of tools, including image processing subroutines (e.g., OpenCV), and pretrained vision models to perform complex visual reasoning tasks based on natural language instructions. These ideas have inspired a growing family of tool-using LLM frameworks, such as ViperGPT [104], Chameleon [68], and VisualSketchpad [42], all of which extend the compositional reasoning capability of LLMs across modalities. In addition, ToolLLM [81] and Gorilla [78] enhance model robustness and adaptability in real-world tool use by enabling open-ended API grounding and dynamic retrieval of tool documentation. Binder [15] extends tool-augmented reasoning by binding LLMs within symbolic programming environments. The model generates executable programs in languages such as Python or SQL to interface with structured APIs and symbolic tools. Each API call is then prompted to the LLM to generate its corresponding output, and finally, the entire program is executed to produce the overall answer. For mathematical and multimodal reasoning, systems such as Tora [31] and MM-ReAct [131] demonstrate that combining LLMs with external computational tools (e.g., symbolic solvers, code executors, calculators, and vision experts) not only improves accuracy but also yields more interpretable reasoning chains.

5.3. LLM ↔ Symbolic : End-to-End & Hybrid Integration

Moving beyond loosely coupled neurosymbolic pipelines, recent research has explored tighter integration between symbolic reasoning and large language models (LLMs). In current reasoning architectures, reasoning trajectories are typically expressed either in natural language or as latent embeddings within neural networks. However, these representations are inherently approximate and may accumulate semantic drift as the reasoning chain grows longer. Such deviations often result in logical inconsistencies or reasoning failures, particularly in tasks requiring multi-step or deductive reasoning. To mitigate these issues, several approaches have introduced formally symbolic representations for intermediate reasoning states. Symbolic reasoning structures enable more explicit and semantically precise formulations of intermediate steps, avoiding the ambiguity of natural language and the opaqueness of latent representations. This sequence of intermediate symbolic representations guide the model toward the final answer. For example, Chain-of-Symbol Prompting [41] constrains LLMs to operate over compact symbolic alphabets rather than free-form text, reducing linguistic noise and enhancing consistency in spatial reasoning tasks. Similarly, NaturalProgram [62] proposes a natural language–based deductive reasoning format that enables reasoning chains to be both verifiable and interpretable in natural language. Further advancing this line, LogicGuide [80] provides an explicit logical reasoning layer that interacts with LLMs through guided formal inference. By embedding symbolic logic within the reasoning loop, it ensures that each reasoning step remains logically sound and traceable, thereby improving factual consistency and reliability. Other works, such as [122], use programmatic representations (e.g., Python code) as symbolic scaffolds for intermediate reasoning—serving not as executable code, but as structured prompts that constrain the model's reasoning trajectory toward the final solution. By introducing symbolic representations into the reasoning chains of LLMs, these methods reduce the representation errors that often arise in natural language representations during the reasoning processes.

Beyond static symbolic formulations, emerging methods explore differentiable symbolic reasoning, which enables end-to-end optimization by relaxing discrete symbolic operations into continuous differentiable approximations. This paradigm bridges the gap between neural computation and logical inference, allowing symbolic modules to be optimized jointly with the language model. For instance, DiLA [137] integrates an LLM with a differentiable logic solver that parses natural language into formal logical expressions. The LLM proposes initial solutions that are iteratively refined through constraint checking within a differentiable logic layer, ensuring both logical satisfiability and semantic coherence. Similarly, OREO-LM [43] consists of a Knowledge Interaction Layer that can be flexibly plugged into existing Transformer-based LMs to interact with a differentiable Knowledge Graph Reasoning module. It enables symbolic reasoning to influence the internal representations of the LLM during both training and inference. In the visual reasoning domain, NS-VQA [1] pioneered a framework that separates visual representation learning from the inference mechanism and introduced a differentiable first-order logic formalism to enable compositional reasoning for visual question answering. More recent works, such as [124], integrate

Compiled Neural Networks (CoNNs) into standard transformer architectures, embedding symbolic rules directly within attention mechanisms to enhance efficiency, and interpretability for symbolic operations.

Recent studies also highlight optimized probabilistic reasoning enhancements in end-to-end neurosymbolic systems. For instance, TRACE [123] introduces a tractable probabilistic reasoning approach for controllable text generation, which distills a Hidden Markov Model (HMM) from a base LM and combines it with simple, efficiently trained classifiers to tractably compute the Expected Attribute Probability (EAP), which guides generation towards desired attributes. Ctrl-G [135] offers an adaptable framework that combines any production-ready LLM with a Hidden Markov Model, enabling LLM outputs to adhere to logical constraints represented as deterministic finite automata.

Collectively, these efforts represent a paradigm shift toward end-to-end neurosymbolic architectures, where formal symbolic reasoning and neural representations coevolve within a unified system. By encoding intermediate reasoning states symbolically, integrating differentiable logical modules, and optimizing probabilistic inference, these hybrids achieve a more principled balance between the interpretability of symbolic reasoning and the expressive power of neural computation, though this comes at the cost of higher architectural complexity and optimization challenges.

5.3.1. Symbolic Feedback based Methods

Another line of end-to-end neurosymbolic integration involves symbolic feedback-based reasoning systems, which provide a powerful framework for integrating logical validation as regularization terms within large language model (LLM) reasoning. Building upon this foundation, a growing body of research explores the intersection of neurosymbolic reasoning with reinforcement learning (RL) and agentic architectures. In agentic systems, LLMs are trained to decompose complex problems into subproblems, invoke symbolic modules (such as verifiers or solvers), and iteratively refine their reasoning based on structured feedback signals. Reinforcement learning facilitates this iterative refinement by rewarding reasoning trajectories that align with symbolic correctness. Recent models such as DeepSeek-R1 [32] and OpenAI's o1 model [46] demonstrate how reinforcement-driven reasoning optimization can yield consistent improvements in factuality and logical accuracy. The integration of symbolic feedback directly into the reward model [127] represents a particularly promising direction, allowing symbolic verifiers to influence the training objective without requiring differentiable symbolic components. A number of recent neurosymbolic systems further exemplify this trend. [23] employs a small frozen language model equipped with an adapter that translates natural language problems into formal symbolic expressions. The adapted model is then trained using reinforcement learning, guided by feedback from a non-differentiable symbolic solver, ensuring correctness at the formal reasoning level. [72] introduces Rule Based Rewards (RBR) that penalize reasoning violations and enhance model safety, while RLSF [49] integrates feedback from external symbolic or domain-specific tools, such as theorem provers or knowledge bases, to guide reasoning optimization. LLM-Modulo [51] applies symbolic consistency checks to planning problems, and CoTran [47] introduces symbolic feedback loops for code generation tasks. These systems share the principle of using symbolic verification or structured evaluation as a source of external reward, offering a form of neurosymbolic reinforcement that aligns model behavior with logical correctness and interpretable reasoning. Another hybrid neurosymbolic approach leverages symbolic feedback through iterative loops [7], where a symbolic reasoner verifies the LLM-generated outputs and provides feedback to the LLM via prompts to refine its responses. While these methods do not alter the internal reasoning weights of the model, they help the LLM produce more accurate answers and maintain logical consistency.

The main advantage of symbolic feedback-based approaches lies in their ability to provide reliable and interpretable guidance without requiring differentiable symbolic components. By incorporating symbolic verifiers or solvers into the feedback loop, these methods bridge the gap between continuous neural optimization and discrete logical reasoning. However, their effectiveness often comes at the cost of computational complexity, as symbolic validation and feedback evaluation can be resource-intensive. Despite this, symbolic feedback-driven reinforcement remains one of the most promising directions for improving the reasoning fidelity and interpretability of LLMs, particularly in domains requiring structured, logically sound decision-making.

Neurosymbolic approaches represent a promising direction for enhancing the reasoning abilities of LLMs. By integrating symbolic reasoning with neural network-based learning, these methods address the limitations of LLMs in handling complex reasoning tasks. While challenges remain in integrating these components effectively, ongoing research continues to advance the development of neurosymbolic systems, paving the way for more capable and interpretable AI systems.

6. Open Challenges and Future Directions

Despite the significant progress in neurosymbolic reasoning with large language models (LLMs), several open challenges remain before such systems can achieve robust, generalizable, and interpretable reasoning. This section outlines key limitations and future research directions for advancing the integration of symbolic reasoning and neural language models.

6.1. Advanced Hybrid Architectures and Optimization

While LLMs have demonstrated remarkable inductive reasoning abilities, they remain fundamentally statistical learners rather than formal logical reasoners. The design of advanced hybrid architectures that tightly couple symbolic inference with neural representations remains an open frontier [41, 137, 124]. A promising direction involves constructing architectures where symbolic reasoning modules and neural layers can be jointly optimized—either through differentiable approximations of logical constraints or via reinforcement learning with symbolic feedback [32, 46]. However, training such systems at scale poses significant computational and optimization challenges, especially in ensuring stability and convergence across symbolic and neural components. Future work should focus on scalable learning frameworks, efficient symbolic-verification loops, and adaptive reasoning strategies that enable dynamic switching between neural intuition and symbolic precision.

6.2. Theoretical Foundations and Interpretability

A deeper theoretical understanding of how symbolic reasoning improves the generalization and reliability of LLMs is still lacking. Current progress has largely been empirical, with limited formal analysis of how symbolic constraints shape neural representations or influence reasoning trajectories. Open questions remain regarding the nature of symbolic regularization in large-scale optimization, the emergence of structured representations during training, and the interplay between symbolic supervision and LLM scaling laws. Establishing mathematical and cognitive models for neurosymbolic reasoning could provide crucial insights into the design of interpretable and controllable reasoning systems. Furthermore, understanding reasoning shortcuts, where models arrive at correct answers without genuine inference, remains essential for evaluating true reasoning capabilities rather than surface-level performance.

6.3. Multi-Modal and Embodied Reasoning

Most existing neurosymbolic approaches focus primarily on textual reasoning tasks, leaving multi-modal reasoning relatively underexplored. However, real-world reasoning often involves integrating information from multiple modalities, such as vision, language, spatial context, and sensory feedback, to perform grounded and embodied reasoning [131, 42]. Developing systems that combine symbolic logic with multi-modal LLMs is a key challenge, as these models must align symbolic abstractions (e.g., spatial relations or visual predicates) with perceptual data. Future research could explore symbolic pipelines that emulate human-like reasoning, where abstract reasoning interacts dynamically with perception and action.

6.4. Data, Benchmarking, and Evaluation Metrics

Progress in neurosymbolic reasoning is also constrained by the lack of standardized datasets and evaluation benchmarks. Current reasoning benchmarks often focus on linguistic correctness or numerical accuracy, but overlook structural or logical validity. Creating large-scale, diverse datasets annotated with symbolic reasoning traces, verifiable logical steps, or formal proofs could significantly enhance supervised and reinforcement learning of reasoning processes [28]. The categorization of neurosymbolic approaches remains an active topic of discussion, while their respective theoretical foundations are not yet well defined. In addition, developing evaluation metrics that capture interpretability, logical soundness, and reasoning faithfulness would allow a more rigorous assessment of the symbolic reasoning capabilities of LLMs.

6.5. Toward Generalizable and Adaptive Symbolic Reasoning

Ultimately, the long-term goal is to develop neurosymbolic systems capable of adaptive, generalizable reasoning that transfers across domains. This requires learning symbolic abstractions dynamically, rather than relying on fixed ontologies or handcrafted logic templates, and grounding them in data-driven contexts. Future directions include exploring meta-reasoning frameworks that allow LLMs to autonomously induce, verify, and revise symbolic rules, as well as incorporating continual learning strategies to maintain reasoning consistency across evolving knowledge domains. Achieving this synthesis would mark a major step toward interpretable, scalable, and human-aligned artificial intelligence.

7. Conclusion

This chapter provided a concise overview of symbolic reasoning and its modern evolution through neurosymbolic approaches that integrate symbolic knowledge with neural and large language models. This integration represents a pivotal step toward developing AI systems that are both interpretable and capable of complex reasoning. This chapter has reviewed key advances in this direction, including methods that embed symbolic rules within neural architectures, LLMs trained on structured reasoning traces, and hybrid frameworks that combine symbolic inference with data-driven learning. These approaches collectively illustrate the growing synergy between neural and symbolic paradigms, where neural components provide flexibility and scalability, while symbolic systems contribute precision, compositionality, and explainability.

Despite remarkable progress, several challenges persist. Theoretical understanding of how symbolic reasoning enhances neural generalization remains limited, and systematic frameworks for evaluating reasoning quality are still evolving. Moreover, the scalability of hybrid architectures and their applicability across modalities and domains require further exploration. Addressing these challenges will not only advance the field of neurosymbolic AI but also bring us closer to realizing general, trustworthy, and human-aligned reasoning systems.

The convergence of symbolic and neural methods suggests a promising research frontier, one that moves beyond the dichotomy between logic and learning toward a unified paradigm of intelligence. As the field progresses, developing principled, adaptable, and transparent reasoning mechanisms will be central to shaping the next generation of AI systems capable of both understanding and reasoning about the world.

Abbreviations

Declaration on Generative AI

During the preparation of this work, the author(s) used X-GPT-4 in order to: Grammar, spelling check and sentence correction. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

Acknowledgments

The authors acknowledge partial funding under the National Science Foundation grant 2333782 "Proto-OKN Theme 1: Safe Agricultural Products and Water Graph (SAW-Graph): An OKN to Monitor and Trace PFAS and Other Contaminants in the Nation's Food and Water Systems", and by Kansas State University through the Game-changing Research Initiation Program (GRIP) and the GRIPex: AI in the Disciplines program.

REFERENCE

1. Amizadeh, S., Palangi, H., Polozov, A., Huang, Y., Koishida, K., 2020. Neuro-symbolic visual reasoning: Disentangling "visual" from "reasoning", in: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, PMLR. pp. 279–290. URL: http://proceedings.mlr.press/v119/amizadeh20a.html.
2. Asai, A., Wu, Z., Wang, Y., Sil, A., Hajishirzi, H., 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection, in: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024, OpenReview.net. URL: https://openreview.net/forum?id=hSyW5go0v8.
3. Baader, F., Horrocks, I., Lutz, C., Sattler, U., 2017. An Introduction to Description Logic. Cambridge University Press. URL: http://www.cambridge.org/de/academic/subjects/computer-science/knowledge-management-databases-and-data-mining/introduction-description-logic?format=PB\#17zVGeWD2TZUeu6s.97.
4. Bader, S., Hitzler, P., 2005. Dimensions of neural-symbolic integration - A structured survey, in: Artëmov, S.N., Barringer, H., d'Avila Garcez, A.S., Lamb, L.C., Woods, J. (Eds.), We Will Show Them! Essays in Honour of Dov Gabbay, Volume One, College Publications. pp. 167–194.
5. Badreddine, S., d'Avila Garcez, A., Serafini, L., Spranger, M., 2022. Logic tensor networks. Artif. Intell. 303. URL: https://doi.org/10.1016/j.artint.2021.103649, doi:10.1016/j.artint.2021.103649.
6. Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., Do, Q.V., Xu, Y., Fung, P., 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity, in: Park, J.C., Arase, Y., Hu, B., Lu, W., Wijaya, D., Purwarianti, A., Krisnadhi, A.A. (Eds.), Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP 2023 -Volume 1: Long Papers, Nusa Dua, Bali, November 1 - 4, 2023, Association for Computational Linguistics. pp. 675–718. URL: https://doi.org/10.18653/v1/2023.ijcnlp-main.45.
7. Barua, A., Hitzler, P., 2025. Description logic concept learning using large language models, in: 19th International Conference on Neurosymbolic Learning and Reasoning. URL: https://openreview.net/forum?id=ebVC7S5VMF.
8. Besold, T.R., d'Avila Garcez, A.S., Bader, S., Bowman, H., Domingos, P.M., Hitzler, P., Kühnberger, K., Lamb, L.C., Lima, P.M.V., de Penning, L., Pinkas, G., Poon, H., Zaverucha, G., 2021. Neural-symbolic learning and reasoning: A survey and interpretation, in: Hitzler, P., Sarker, M.K. (Eds.), Neuro-Symbolic Artificial Intelligence: The State of the Art. IOS Press. volume 342 of *Frontiers in Artificial Intelligence and Applications*, pp. 1–51. URL: https://doi.org/10.3233/FAIA210348.
9. Binz, M., Schulz, E., 2022. Using cognitive psychology to understand GPT-3. CoRR abs/2206.14576. URL: https://doi.org/10.48550/arXiv.2206.14576, arXiv:2206.14576.
10. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O., 2013. Translating embeddings for modeling multi-relational data. Advances in neural information processing systems 26.
11. Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y.T., Li, Y., Lundberg, S.M., Nori, H., Palangi, H., Ribeiro, M.T., Zhang, Y., 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. CoRR abs/2303.12712. URL: https://doi.org/10.48550/arXiv.2303.12712, arXiv:2303.12712.
12. Chan, B.J., Chen, C., Cheng, J., Huang, H., 2025. Don't do RAG: when cache-augmented generation is all you need for knowledge tasks, in: Long, G., Blumestein, M., Chang, Y., Lewin-Eytan, L., Huang, Z.H., Yom-Tov, E. (Eds.), Companion Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025 - 2 May 2025, ACM. pp. 893–897. URL: https://doi.org/10.1145/3701716.3715490, doi:10.1145/3701716.3715490.
13. Chen, J., Hu, P., Jimenez-Ruiz, E., Holter, O.M., Antonyrajah, D., Horrocks, I., 2021. Owl2vec*: embedding of owl ontologies. Machine Learning 110, 1813–1845.
14. Chen, W., Ma, X., Wang, X., Cohen, W.W., 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. Trans. Mach. Learn. Res. 2023. URL:

https://openreview.net/forum?id=YfZ4ZPt8zd.
15. Cheng, Z., Xie, T., Shi, P., Li, C., Nadkarni, R., Hu, Y., Xiong, C., Radev, D., Ostendorf, M., Zettlemoyer, L., Smith, N.A., Yu, T., 2023. Binding language models in symbolic languages, in: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023, OpenReview.net. URL: https://openreview.net/forum?id=lH1PV42cbF.
16. Christiano, P.F., Leike, J., Brown, T.B., Martic, M., Legg, S., Amodei, D., 2017. Deep reinforcement learning from human preferences, in: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp. 4299–4307. URL: https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.
17. Clocksin, W.F., Mellish, C.S., 1994. Programming in Prolog (4. ed.). Springer.
18. Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., Schulman, J., 2021. Training verifiers to solve math word problems. CoRR abs/2110.14168. URL: https://arxiv.org/abs/2110.14168, arXiv:2110.14168.
19. Colmerauer, A., Roussel, P., 1996. The birth of Prolog. Association for Computing Machinery, New York, NY, USA. p. 331–367. URL: https://doi.org/10.1145/234286.1057820.
20. Denney, E., Fischer, B., Schumann, J., 2006. An empirical evaluation of automated theorem provers in software certification. International Journal on Artificial Intelligence Tools 15, 81–107. doi:10.1142/S0218213006002576.
21. Donadello, I., Serafini, L., d'Avila Garcez, A., 2017. Logic tensor networks for semantic image interpretation, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pp. 1596–1602. URL: https://doi.org/10.24963/ijcai.2017/221, doi:10.24963/ijcai.2017/221.
22. Du, Y., Li, S., Torralba, A., Tenenbaum, J.B., Mordatch, I., 2024. Improving factuality and reasoning in language models through multiagent debate, in: Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024, OpenReview.net. URL: https://openreview.net/forum?id=zj7YuTE4t8.
23. Dutta, S., Pandey, I., Singh, J., Manchanda, S., Chakrabarti, S., Chakraborty, T., 2024. Frugal lms trained to invoke symbolic solvers achieve parameter-efficient arithmetic reasoning, in: Wooldridge, M.J., Dy, J.G., Natarajan, S. (Eds.), Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada, AAAI Press. pp. 17951–17959. URL: https://doi.org/10.1609/aaai.v38i16.29750.
24. Feng, J., Xu, R., Hao, J., Sharma, H., Shen, Y., Zhao, D., Chen, W., 2024. Language models can be deductive solvers, in: Duh, K., Gómez-Adorno, H., Bethard, S. (Eds.), Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024, Association for Computational Linguistics. pp. 4026–4042. URL: https://doi.org/10.18653/v1/2024.findings-naacl.254.
25. Fitting, M., 2012. First-order logic and automated theorem proving. Springer Science & Business Media. doi:10.1007/978-1-4612-2360-3.
26. Gandhi, K., Lee, D., Grand, G., Liu, M., Cheng, W., Sharma, A., Goodman, N.D., 2024. Stream of search (sos): Learning to search in language. CoRR abs/2404.03683. URL: https://doi.org/10.48550/arXiv.2404.03683, arXiv:2404.03683.
27. Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., Callan, J., Neubig, G., 2023. PAL: program-aided language models, in: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (Eds.), International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, PMLR. pp. 10764–10799. URL: https://proceedings.mlr.press/v202/gao23f.html.
28. d'Avila Garcez, A., Lamb, L.C., 2023. Neurosymbolic AI: the 3rd wave. Artif. Intell. Rev. 56, 12387–12406. URL: https://doi.org/10.1007/s10462-023-10448-w, doi:10.1007/

s10462-023-10448-w.
29. d'Avila Garcez, A.S., Gori, M., Lamb, L.C., Serafini, L., Spranger, M., Tran, S.N., 2019. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. FLAP 6, 611–632. URL: https://collegepublications.co.uk/ifcolog/?00033.
30. Garnelo, M., Shanahan, M., 2019. Reconciling deep learning with symbolic artificial intelligence: representing objects and relations. Current Opinion in Behavioral Sciences 29, 17–23. URL: https://www.sciencedirect.com/science/article/pii/S2352154618301943, doi:10.1016/j.cobeha.2018.12.010.
31. Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Huang, M., Duan, N., Chen, W., 2024. Tora: A tool-integrated reasoning agent for mathematical problem solving, in: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024, OpenReview.net. URL: https://openreview.net/forum?id=Ep0TtjVoap.
32. Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al., 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. CoRR abs/2501.12948. URL: https://doi.org/10.48550/arXiv.2501.12948, arXiv:2501.12948.
33. Gupta, T., Kembhavi, A., 2023. Visual programming: Compositional visual reasoning without training, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023, IEEE. pp. 14953–14962. URL: https://doi.org/10.1109/CVPR52729.2023.01436.
34. Han, H., Wang, Y., Shomer, H., Guo, K., Ding, J., Lei, Y., Halappanavar, M., Rossi, R.A., Mukherjee, S., Tang, X., He, Q., Hua, Z., Long, B., Zhao, T., Shah, N., Javari, A., Xia, Y., Tang, J., 2025. Retrieval-augmented generation with graphs (graphrag). CoRR abs/2501.00309. URL: https://doi.org/10.48550/arXiv.2501.00309, arXiv:2501.00309.
35. Han, S., Zhang, Q., Yao, Y., Jin, W., Xu, Z., He, C., 2024. LLM multi-agent systems: Challenges and open problems. CoRR abs/2402.03578. URL: https://doi.org/10.48550/arXiv.2402.03578, arXiv:2402.03578.
36. He, H., Zhang, H., Roth, D., 2023. Rethinking with retrieval: Faithful large language model inference. CoRR abs/2301.00303. URL: https://doi.org/10.48550/arXiv.2301.00303, arXiv:2301.00303.
37. Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., Steinhardt, J., 2021. Measuring mathematical problem solving with the MATH dataset, in: Vanschoren, J., Yeung, S. (Eds.), Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual. URL: https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.
38. Holter, O.M., Myklebust, E.B., Chen, J., Jiménez-Ruiz, E., 2019. Embedding owl ontologies with owl2vec, in: CEUR Workshop Proceedings, pp. 33–36.
39. Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S.K.S., Lin, Z., Zhou, L., Ran, C., Xiao, L., Wu, C., Schmidhuber, J., 2024. Metagpt: Meta programming for A multi-agent collaborative framework, in: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024, OpenReview.net. URL: https://openreview.net/forum?id=VtmBAGCN7o.
40. Hsiang, J., 1983. Topics in automated theorem proving and program generation. University of Illinois at Urbana-Champaign, USA.
41. Hu, H., Lu, H., Zhang, H., Song, Y.Z., Lam, W., Zhang, Y., 2024a. Chain-of-symbol prompting for spatial reasoning in large language models, in: First Conference on Language Modeling.
42. Hu, Y., Shi, W., Fu, X., Roth, D., Ostendorf, M., Zettlemoyer, L., Smith, N.A., Krishna, R., 2024b. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models, in: Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J.M., Zhang, C. (Eds.), Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, De-

cember 10 - 15, 2024. URL: http://papers.nips.cc/paper_files/paper/2024/hash/fb82011040977c7712409fbdb5456647-Abstract-Conference.html.
43. Hu, Z., Xu, Y., Yu, W., Wang, S., Yang, Z., Zhu, C., Chang, K., Sun, Y., 2022. Empowering language models with knowledge graph reasoning for open-domain question answering, in: Goldberg, Y., Kozareva, Z., Zhang, Y. (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, Association for Computational Linguistics. pp. 9562–9581. URL: https://doi.org/10.18653/v1/2022.emnlp-main.650.
44. Huang, J., Chang, K.C., 2023. Towards reasoning in large language models: A survey, in: Rogers, A., Boyd-Graber, J.L., Okazaki, N. (Eds.), Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023, Association for Computational Linguistics. pp. 1049–1065. URL: https://doi.org/10.18653/v1/2023.findings-acl.67.
45. Izacard, G., Grave, E., 2021. Leveraging passage retrieval with generative models for open domain question answering, in: Merlo, P., Tiedemann, J., Tsarfaty, R. (Eds.), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021, Association for Computational Linguistics. pp. 874–880. URL: https://doi.org/10.18653/v1/2021.eacl-main.74.
46. Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., et al., 2024. Openai o1 system card. CoRR abs/2412.16720. URL: https://doi.org/10.48550/arXiv.2412.16720, arXiv:2412.16720.
47. Jana, P., Jha, P., Ju, H., Kishore, G., Mahajan, A., Ganesh, V., 2024. Cotran: An llm-based code translator using reinforcement learning with feedback from compiler and symbolic execution, in: Endriss, U., Melo, F.S., Bach, K., Diz, A.J.B., Alonso-Moral, J.M., Barro, S., Heintz, F. (Eds.), ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024), IOS Press. pp. 4011–4018. URL: https://doi.org/10.3233/FAIA240968.
48. Janowicz, K., Hitzler, P., Bianchi, F., Ebrahimi, M., Sarker, M.K., 2020. Neural-symbolic integration and the semantic web. Semant. Web 11, 3–11. URL: https://doi.org/10.3233/SW-190368, doi:10.3233/SW-190368.
49. Jha, P., Jana, P., Arora, A., Ganesh, V., 2024. RLSF: reinforcement learning via symbolic feedback. CoRR abs/2405.16661. URL: https://doi.org/10.48550/arXiv.2405.16661, arXiv:2405.16661.
50. Ji, Z., Yu, T., Xu, Y., Lee, N., Ishii, E., Fung, P., 2023. Towards mitigating hallucination in large language models via self-reflection. CoRR abs/2310.06271. URL: https://doi.org/10.48550/arXiv.2310.06271, arXiv:2310.06271.
51. Kambhampati, S., Valmeekam, K., Guan, L., Verma, M., Stechly, K., Bhambri, S., Saldyt, L., Murthy, A., 2024. Position: Llms can’t plan, but can help planning in llm-modulo frameworks, in: Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024, OpenReview.net. URL: https://openreview.net/forum?id=Th8JPEmH4z.
52. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y., 2022. Large language models are zero-shot reasoners, in: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (Eds.), Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022. URL: http://papers.nips.cc/paper_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html.
53. Kovács, L., Voronkov, A., 2013. First-order theorem proving and vampire, in: Sharygina, N., Veith, H. (Eds.), Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings, Springer. pp. 1–35. URL: https://doi.org/10.1007/978-3-642-39799-8_1.
54. Kowalski, R.A., 1974. Predicate logic as programming language, in: Rosenfeld, J.L. (Ed.), Information Processing, Proceedings of the 6th IFIP Congress 1974, Stockholm, Sweden, August 5-10, 1974, North-Holland. pp. 569–574.
55. Kulmanov, M., Liu-Wei, W., Yan, Y., Hoehndorf, R., 2019. El embeddings: Geometric construction

of models for the description logic el++, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization. pp. 6103–6109. URL: https://doi.org/10.24963/ijcai.2019/845, doi:10.24963/ijcai.2019/845.

56. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D., 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks, in: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.
57. Li, H., Chong, Y.Q., Stepputtis, S., Campbell, J., Hughes, D., Lewis, C., Sycara, K.P., 2023a. Theory of mind for multi-agent collaboration via large language models, in: Bouamor, H., Pino, J., Bali, K. (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, Association for Computational Linguistics. pp. 180–192. URL: https://doi.org/10.18653/v1/2023.emnlp-main.13.
58. Li, Y., Zhang, W., Yang, Y., Huang, W., Wu, Y., Luo, J., Bei, Y., Zou, H.P., Luo, X., Zhao, Y., Chan, C., Chen, Y., Deng, Z., Li, Y., Zheng, H., Li, D., Jiang, R., Zhang, M., Song, Y., Yu, P.S., 2025. Towards agentic RAG with deep reasoning: A survey of rag-reasoning systems in llms. CoRR abs/2507.09477. URL: https://doi.org/10.48550/arXiv.2507.09477, arXiv:2507.09477.
59. Li, Z., Huang, Y., Li, Z., Yao, Y., Xu, J., Chen, T., Ma, X., Lu, J., 2023b. Neuro-symbolic learning yielding logical constraints, in: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (Eds.), Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/4459c3c143db74ee52afebdf56836375-Abstract-Conference.html.
60. Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., Cobbe, K., 2024. Let's verify step by step, in: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024, OpenReview.net. URL: https://openreview.net/forum?id=v8L0pN6EOi.
61. Lin, H., Cao, S., Li, M., Wang, S., Wu, H., Yang, L., Zheng, J., Qin, C., 2025. Interactive learning for llm reasoning. arXiv preprint arXiv:2509.26306 .
62. Ling, Z., Fang, Y., Li, X., Huang, Z., Lee, M., Memisevic, R., Su, H., 2023. Deductive verification of chain-of-thought reasoning, in: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (Eds.), Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/72393bd47a35f5b3bee4c609e7bba733-Abstract-Conference.html.
63. Liu, H., Fu, Z., Ding, M., Ning, R., Zhang, C., Liu, X., Zhang, Y., 2025a. Logical reasoning in large language models: A survey. CoRR abs/2502.09100. URL: https://doi.org/10.48550/arXiv.2502.09100, arXiv:2502.09100.
64. Liu, J., Zhang, X., Li, Y., Wang, J., jin Kim, H., 2019. Deep learning-based reasoning with multi-ontology for iot applications. IEEE Access 7, 124688–124701. doi:10.1109/ACCESS.2019.2937353.
65. Liu, P., Chen, S., Liu, J., Xu, Y., Cao, F., Wu, G., 2023. An efficient contradiction separation based automated deduction algorithm for enhancing reasoning capability. Knowledge-Based Systems 261, 110217. doi:10.1016/j.knosys.2022.110217.
66. Liu, T., Xu, W., Huang, W., Zeng, Y., Wang, J., Wang, X., Yang, H., Li, J., 2025b. Logic-of-thought: Injecting logic into contexts for full reasoning in large language models, in: Chiruzzo, L., Ritter, A., Wang, L. (Eds.), Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025, Association

for Computational Linguistics. pp. 10168–10185. URL: https://doi.org/10.18653/v1/2025.naacl-long.510.

67. Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K., et al., 2024. Agentbench: Evaluating llms as agents, in: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024, OpenReview.net. URL: https://openreview.net/forum?id=zAdUB0aCTQ.
68. Lu, P., Peng, B., Cheng, H., Galley, M., Chang, K., Wu, Y.N., Zhu, S., Gao, J., 2023. Chameleon: Plug-and-play compositional reasoning with large language models, in: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (Eds.), Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/871ed095b734818cfba48db6aeb25a62-Abstract-Conference.html.
69. Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B.P., Hermann, K., Welleck, S., Yazdanbakhsh, A., Clark, P., 2023. Self-refine: Iterative refinement with self-feedback, in: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (Eds.), Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.
70. Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., De Raedt, L., 2018. Deepproblog: Neural probabilistic logic programming, in: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/dc5d637ed5e62c36ecb73b654b05ba2a-Paper.pdf.
71. McCune, W., 2005. Prover9 and Mace4. Technical Report. Argonne National Laboratory. URL: https://www.cs.unm.edu/~mccune/prover9/. version 2005–2010.
72. Mu, T., Helyar, A., Heidecke, J., Achiam, J., Vallone, A., Kivlichan, I., Lin, M., Beutel, A., Schulman, J., Weng, L., 2024. Rule based rewards for language model safety, in: Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J.M., Zhang, C. (Eds.), Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024. URL: http://papers.nips.cc/paper_files/paper/2024/hash/c4e380fb74dec9da9c7212e834657aa9-Abstract-Conference.html.
73. Newell, A., Simon, H.A., 1976. Computer science as empirical inquiry: Symbols and search. Commun. ACM 19, 113–126. URL: https://doi.org/10.1145/360018.360022.
74. Olausson, T., Gu, A., Lipkin, B., Zhang, C.E., Solar-Lezama, A., Tenenbaum, J.B., Levy, R., 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers, in: Bouamor, H., Pino, J., Bali, K. (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, Association for Computational Linguistics. pp. 5153–5176. URL: https://doi.org/10.18653/v1/2023.emnlp-main.313.
75. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P.F., Leike, J., Lowe, R., 2022. Training language models to follow instructions with human feedback, in: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (Eds.), Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022. URL: http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.
76. Pallagani, V., Muppasani, B., Murugesan, K., Rossi, F., Horesh, L., Srivastava, B., Fabiano, F., Loreggia, A., 2022. Plansformer: Generating symbolic plans using transformers. CoRR abs/2212.08681. URL: https://doi.org/10.48550/arXiv.2212.08681, arXiv:2212.08681.

77. Pan, L., Albalak, A., Wang, X., Wang, W.Y., 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning, in: Bouamor, H., Pino, J., Bali, K. (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023, Association for Computational Linguistics. pp. 3806–3824. URL: https://doi.org/10.18653/v1/2023.findings-emnlp.248.
78. Patil, S.G., Zhang, T., Wang, X., Gonzalez, J.E., 2024. Gorilla: Large language model connected with massive apis, in: Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J.M., Zhang, C. (Eds.), Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024. URL: http://papers.nips.cc/paper_files/paper/2024/hash/e4c61f578ff07830f5c37378dd3ecb0d-Abstract-Conference.html.
79. Pavlov, V., Schukin, A., Cherkasova, T., 2013. Exploring automated reasoning in first-order logic: Tools, techniques and application areas, in: Klinov, P., Mouromtsev, D. (Eds.), Knowledge Engineering and the Semantic Web, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 102–116. doi:10.1007/978-3-642-41360-5_9.
80. Poesia, G., Gandhi, K., Zelikman, E., Goodman, N.D., 2024. Certified deductive reasoning with language models. Trans. Mach. Learn. Res. 2024. URL: https://openreview.net/forum?id=yXnwrs2Tl6.
81. Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., Lin, Y., Cong, X., Tang, X., Qian, B., Zhao, S., Hong, L., Tian, R., Xie, R., Zhou, J., Gerstein, M., Li, D., Liu, Z., Sun, M., 2024. Toolllm: Facilitating large language models to master 16000+ real-world apis, in: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024, OpenReview.net. URL: https://openreview.net/forum?id=dHng2O0Jjr.
82. Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C., 2023. Direct preference optimization: Your language model is secretly a reward model, in: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (Eds.), Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.
83. Rakib, A., Faruqui, R.U., 2021. Model checking ontology-driven reasoning agents using strategy and abstraction. Concurrency and Computation: Practice and Experience 33, e5205.
84. Renze, M., Guven, E., 2024. Self-reflection in LLM agents: Effects on problem-solving performance. CoRR abs/2405.06682. URL: https://doi.org/10.48550/arXiv.2405.06682, arXiv:2405.06682.
85. Riazanov, A., Voronkov, A., 2002. The design and implementation of vampire. AI communications 15, 91–110.
86. Ristoski, P., Paulheim, H., 2016. Rdf2vec: Rdf graph embeddings for data mining, in: Groth, P., Simperl, E., Gray, A., Sabou, M., Krötzsch, M., Lecue, F., Flöck, F., Gil, Y. (Eds.), The Semantic Web – ISWC 2016, Springer International Publishing, Cham. pp. 498–514.
87. Robinson, J.A., 1965. A machine-oriented logic based on the resolution principle. J. ACM 12, 23–41. URL: https://doi.org/10.1145/321250.321253, doi:10.1145/321250.321253.
88. Rocktäschel, T., Riedel, S., 2017. End-to-end differentiable proving, in: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/b2ab001909a8a6f04b51920306046ce5-Paper.pdf.
89. Russell, S., Norvig, P., 2020. Artificial Intelligence: A Modern Approach (4th Edition). Pearson. URL: http://aima.cs.berkeley.edu/.
90. Sahoo, P., Singh, A.K., Saha, S., Jain, V., Mondal, S., Chadha, A., 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. CoRR abs/2402.07927. URL: https://doi.org/10.48550/arXiv.2402.07927, arXiv:2402.07927.
91. Samuel, K., Obrst, L., Stoutenburg, S., Fox, K., Franklin, P., Johnson, A., Laskey, K., Nichols, D., Lopez, S., Peterson, J., 2007. Translating owl and semantic web rules into prolog: Moving toward description logic programs. Theory and Practice of Logic Programming 8, 301 – 322. doi:10.

1017/S1471068407003249.

92. Sanh, V., Webson, A., Raffel, C., Bach, S.H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T.L., Raja, A., et al., 2022. Multitask prompted training enables zero-shot task generalization, in: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022, OpenReview.net. URL: https://openreview.net/forum?id=9Vrb9D0WI4.
93. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., Scialom, T., 2023. Toolformer: Language models can teach themselves to use tools, in: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (Eds.), Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.
94. Schlichtkrull, A., Blanchette, J.C., Traytel, D., 2019. A verified prover based on ordered resolution, in: Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs, Association for Computing Machinery, New York, NY, USA. p. 152–165. URL: https://doi.org/10.1145/3293880.3294100, doi:10.1145/3293880.3294100.
95. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. CoRR abs/1707.06347. URL: http://arxiv.org/abs/1707.06347, arXiv:1707.06347.
96. Schulz, S., 2002. E - a brainiac theorem prover. AI Commun. 15, 111–126.
97. Schulz, S., 2013. System description: E 1.8, in: McMillan, K., Middeldorp, A., Voronkov, A. (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 735–743. doi:10.1007/978-3-642-45221-5_49.
98. Shearer, R.D.C., Motik, B., Horrocks, I., 2008. Hermit: A highly-efficient OWL reasoner, in: Dolbear, C., Ruttenberg, A., Sattler, U. (Eds.), Proceedings of the Fifth OWLED Workshop on OWL: Experiences and Directions, collocated with the 7th International Semantic Web Conference (ISWC-2008), Karlsruhe, Germany, October 26-27, 2008, CEUR-WS.org. URL: https://ceur-ws.org/Vol-432/owled2008eu_submission_12.pdf.
99. Shinn, N., Labash, B., Gopinath, A., 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. CoRR abs/2303.11366. URL: https://doi.org/10.48550/arXiv.2303.11366, arXiv:2303.11366.
100. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y., 2007. Pellet: A practical OWL-DL reasoner. J. Web Semant. 5, 51–53. URL: https://doi.org/10.1016/j.websem.2007.03.004.
101. Stechly, K., Valmeekam, K., Kambhampati, S., 2025. On the self-verification limitations of large language models on reasoning and planning tasks, in: The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025, OpenReview.net. URL: https://openreview.net/forum?id=4O0v4s3IzY.
102. Steen, A., Benzmüller, C., 2019. The higher-order prover leo-iii (extended abstract), in: KI 2019: Advances in Artificial Intelligence: 42nd German Conference on AI, Kassel, Germany, September 23–26, 2019, Proceedings, Springer-Verlag, Berlin, Heidelberg. p. 333–337. URL: https://doi.org/10.1007/978-3-030-30179-8_30, doi:10.1007/978-3-030-30179-8_30.
103. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J., 2019. Rotate: Knowledge graph embedding by relational rotation in complex space, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=HkgEQnRqYQ.
104. Surís, D., Menon, S., Vondrick, C., 2023. Vipergpt: Visual inference via python execution for reasoning, in: IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023, IEEE. pp. 11854–11864. URL: https://doi.org/10.1109/ICCV51070.2023.01092.
105. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B., 2023. Alpaca: A strong, replicable instruction-following model. Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html 3, 7. URL: https://crfm.stanford.edu/2023/03/13/alpaca.html.

106. Trinh, T.H., Wu, Y., Le, Q.V., He, H., Luong, T., 2024. Solving olympiad geometry without human demonstrations. Nat. 625, 476–482. URL: https://doi.org/10.1038/s41586-023-06747-5.
107. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G., 2016. Complex embeddings for simple link prediction, in: International conference on machine learning, PMLR. pp. 2071–2080.
108. Tsarkov, D., Horrocks, I., 2006. Fact++ description logic reasoner: System description, in: International joint conference on automated reasoning, Springer. pp. 292–297. doi:10.1007/11814771_26.
109. Valmeekam, K., Hernandez, A.O., Sreedharan, S., Kambhampati, S., 2022. Large language models still can't plan (A benchmark for llms on planning and reasoning about change). CoRR abs/2206.10498. URL: https://doi.org/10.48550/arXiv.2206.10498, arXiv:2206.10498.
110. Valmeekam, K., Marquez, M., Sreedharan, S., Kambhampati, S., 2023. On the planning abilities of large language models - A critical investigation, in: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (Eds.), Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/efb2072a358cefb75886a315a6fcf880-Abstract-Conference.html.
111. Vilnis, L., Li, X., Murty, S., McCallum, A., 2018. Probabilistic embedding of knowledge graphs with box lattice measures, in: Gurevych, I., Miyao, Y. (Eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia. pp. 263–272. URL: https://aclanthology.org/P18-1025/, doi:10.18653/v1/P18-1025.
112. Wang, C., Li, X., Liu, H., Wu, X., He, W., 2024a. Efficient logical reasoning in large language models through program-guided learning. Authorea Preprints .
113. Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., Anandkumar, A., 2024b. Voyager: An open-ended embodied agent with large language models. Trans. Mach. Learn. Res. 2024. URL: https://openreview.net/forum?id=ehfRiF0R3a.
114. Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W.X., Wei, Z., Wen, J., 2024c. A survey on large language model based autonomous agents. Frontiers Comput. Sci. 18, 186345. URL: https://doi.org/10.1007/s11704-024-40231-1.
115. Wang, W., Yang, Y., Wu, F., 2025a. Towards data-and knowledge-driven ai: A survey on neuro-symbolic computing. IEEE Transactions on Pattern Analysis and Machine Intelligence 47, 878–899. doi:10.1109/TPAMI.2024.3483273.
116. Wang, X., Wei, J., Schuurmans, D., Le, Q.V., Chi, E.H., Narang, S., Chowdhery, A., Zhou, D., 2023. Self-consistency improves chain of thought reasoning in language models, in: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023, OpenReview.net. URL: https://openreview.net/forum?id=1PL1NIMMrw.
117. Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Arunkumar, A., Ashok, A., Dhanasekaran, A.S., Naik, A., Stap, D., et al., 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks, in: Goldberg, Y., Kozareva, Z., Zhang, Y. (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, Association for Computational Linguistics. pp. 5085–5109. URL: https://doi.org/10.18653/v1/2022.emnlp-main.340.
118. Wang, Y., Yu, Y., Liang, J., He, R., 2025b. A comprehensive survey on trustworthiness in reasoning with large language models. arXiv preprint arXiv:2509.03871 .
119. Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V., 2022a. Finetuned language models are zero-shot learners, in: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022, OpenReview.net. URL: https://openreview.net/forum?id=gEZrGCozdqR.
120. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V., Zhou, D., 2022b. Chain-of-thought prompting elicits reasoning in large language models, in: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (Eds.), Advances in Neural Information Processing Systems 35: Annual Conference on Neural Infor-

mation Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022. URL: http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

121. Wei, J., Wang, X., Schuurmans, D., et al., 2022c. Chain of thought prompting elicits reasoning in large language models. arXiv preprint arXiv:2201.11903 URL: https://arxiv.org/abs/2201.11903.
122. Weir, N., Khalifa, M., Qiu, L., Weller, O., Clark, P., 2024. Learning to reason via program generation, emulation, and search, in: Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J.M., Zhang, C. (Eds.), Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024. URL: http://papers.nips.cc/paper_files/paper/2024/hash/401ece9f5d1cfa8600c22049ef43930e-Abstract-Conference.html.
123. Weng, G.Y., Wang, B., den Broeck, G.V., 2025. TRACE back from the future: A probabilistic reasoning approach to controllable language generation. CoRR abs/2504.18535. URL: https://doi.org/10.48550/arXiv.2504.18535, arXiv:2504.18535.
124. Weng, Y., Zhu, M., Xia, F., Li, B., He, S., Liu, K., Zhao, J., 2024. Mastering symbolic operations: Augmenting language models with compiled neural networks, in: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024, OpenReview.net. URL: https://openreview.net/forum?id=9nsNyN0vox.
125. Wos, L., McCune, W., 1991. Automated theorem proving and logic programming: a natural symbiosis. The Journal of Logic Programming 11, 1–53. URL: https://www.sciencedirect.com/science/article/pii/074310669190008D, doi:https://doi.org/10.1016/0743-1066(91)90008-D.
126. Wu, J., Zhu, J., Liu, Y., 2025. Agentic reasoning: Reasoning llms with tools for the deep research. CoRR abs/2502.04644. URL: https://doi.org/10.48550/arXiv.2502.04644, doi:10.48550/ARXIV.2502.04644, arXiv:2502.04644.
127. Xu, F., Hao, Q., Zong, Z., Wang, J., Zhang, Y., Wang, J., Lan, X., Gong, J., Ouyang, T., Meng, F., Shao, C., Yan, Y., Yang, Q., Song, Y., Ren, S., Hu, X., Li, Y., Feng, J., Gao, C., Li, Y., 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. CoRR abs/2501.09686. URL: https://doi.org/10.48550/arXiv.2501.09686, arXiv:2501.09686.
128. Yang, B., Yih, S.W.t., He, X., Gao, J., Deng, L., 2015. Embedding entities and relations for learning and inference in knowledge bases, in: Proceedings of the International Conference on Learning Representations (ICLR) 2015. URL: https://www.microsoft.com/en-us/research/publication/embedding-entities-and-relations-for-learning-and-inference-in-knowledge-bases/.
129. Yang, F., Yang, Z., Cohen, W.W., 2017. Differentiable learning of logical rules for knowledge base reasoning, in: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/0e55666a4ad822e0e34299df3591d979-Paper.pdf.
130. Yang, X., Shao, J., Guo, L., Zhang, B., Zhou, Z., Jia, L., Dai, W., Li, Y., 2025. Neuro-symbolic artificial intelligence: Towards improving the reasoning abilities of large language models, in: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025, ijcai.org. pp. 10770–10778. URL: https://doi.org/10.24963/ijcai.2025/1195.
131. Yang, Z., Li, L., Wang, J., Lin, K., Azarnasab, E., Ahmed, F., Liu, Z., Liu, C., Zeng, M., Wang, L., 2023. MM-REACT: prompting chatgpt for multimodal reasoning and action. CoRR abs/2303.11381. URL: https://doi.org/10.48550/arXiv.2303.11381, arXiv:2303.11381.
132. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., Narasimhan, K., 2023a. Tree of thoughts: Deliberate problem solving with large language models, in: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (Eds.), Advances in Neural Information Processing Systems 36:

Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

133. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K.R., Cao, Y., 2023b. React: Synergizing reasoning and acting in language models, in: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023, OpenReview.net. URL: https://openreview.net/forum?id=WE_vluYUL-X.
134. Zelikman, E., Wu, Y., Mu, J., Goodman, N.D., 2022. Star: Bootstrapping reasoning with reasoning, in: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (Eds.), Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022. URL: http://papers.nips.cc/paper_files/paper/2022/hash/639a9a172c044fbb64175b5fad42e9a5-Abstract-Conference.html.
135. Zhang, H., Kung, P., Yoshida, M., den Broeck, G.V., Peng, N., 2024a. Adaptable logical control for large language models, in: Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J.M., Zhang, C. (Eds.), Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024. URL: http://papers.nips.cc/paper_files/paper/2024/hash/d15c16cf5619a2b1606da5fc88e3f1a9-Abstract-Conference.html.
136. Zhang, J., 2023. Graph-toolformer: To empower llms with graph reasoning ability via prompt augmented by chatgpt. CoRR abs/2304.11116. URL: https://doi.org/10.48550/arXiv.2304.11116, arXiv:2304.11116.
137. Zhang, Y., Zhen, H.L., Pei, Z., Lian, Y., Yin, L., Yuan, M., Yu, B., 2024b. Dila: Enhancing llm tool learning with differential logic layer. arXiv preprint arXiv:2402.11903 .
138. Zhapa-Camacho, F., Hoehndorf, R., 2024. Lattice-preserving alc ontology embeddings, in: International Conference on Neural-Symbolic Learning and Reasoning, Springer. pp. 355–369.
139. Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q.V., Chi, E.H., 2023. Least-to-most prompting enables complex reasoning in large language models, in: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023, OpenReview.net. URL: https://openreview.net/forum?id=WZH7099tgfM.