mathematics

Article

AI Reasoning in Deep Learning Era: From Symbolic AI to Neural–Symbolic AI

Baoyu Liang [1,2], Yuchen Wang [1] and Chao Tong [1,2,*]

[1] School of Computer Science and Engineering, Beihang University, 37 Xueyuan Road, Haidian District, Beijing 100191, China
[2] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, 37 Xueyuan Road, Haidian District, Beijing 100191, China
* Correspondence: tongchao@buaa.edu.cn

Abstract: The pursuit of Artificial General Intelligence (AGI) demands AI systems that not only perceive but also reason in a human-like manner. While symbolic systems pioneered early breakthroughs in logic-based reasoning, such as MYCIN and DENDRAL, they suffered from brittleness and poor scalability. Conversely, modern deep learning architectures have achieved remarkable success in perception tasks, yet continue to fall short in interpretable and structured reasoning. This dichotomy has motivated growing interest in Neural–Symbolic AI, a paradigm that integrates symbolic logic with neural computation to unify reasoning and learning. This survey provides a comprehensive and technically grounded overview of AI reasoning in the deep learning era, with a particular focus on Neural–Symbolic AI. Beyond a historical narrative, we introduce a formal definition of AI reasoning and propose a novel three-dimensional taxonomy that organizes reasoning paradigms by representation form, task structure, and application context. We then systematically review recent advances—including Differentiable Logic Programming, abductive learning, program induction, logic-aware Transformers, and LLM-based symbolic planning—highlighting their technical mechanisms, capabilities, and limitations. In contrast to prior surveys, this work bridges symbolic logic, neural computation, and emergent generative reasoning, offering a unified framework to understand and compare diverse approaches. We conclude by identifying key open challenges such as symbolic–continuous alignment, dynamic rule learning, and unified architectures, and we aim to provide a conceptual foundation for future developments in general-purpose reasoning systems.

Keywords: AI reasoning; symbolic AI; neural–symbolic reasoning

MSC: 68T01; 68T37

check for updates

Academic Editor: Jonathan Blackledge

Received: 24 April 2025
Revised: 15 May 2025
Accepted: 15 May 2025
Published: 23 May 2025

Citation: Liang, B.; Wang, Y.; Tong, C. AI Reasoning in Deep Learning Era: From Symbolic AI to Neural–Symbolic AI. *Mathematics* **2025**, *13*, 1707. https://doi.org/10.3390/math13111707

1. Introduction

Can machines think? Since Alan Turing raised this profound question [1], endowing machines with human-like reasoning abilities has remained one of the central challenges of artificial intelligence (AI) [2]. As AI research deepens and applications continue to expand across domains, a key question emerges: how can we enhance AI's generalization and adaptability to match or even surpass human cognitive capabilities? This question lies at the heart of the quest for Artificial General Intelligence (AGI) [3].

Unlike narrow AI systems—designed for specific tasks such as image classification or speech recognition [4]—AGI envisions AI systems capable of abstract reasoning, cross-domain generalization, autonomous learning, and the ability to adapt to novel environments [2]. While the path toward AGI remains unclear [5], it is widely agreed that achieving it will require breakthroughs in key areas such as commonsense reasoning [6], causal modeling [7], and learning from limited data [8]. Among these, AI reasoning—the ability to infer, explain, and make decisions based on knowledge—has re-emerged as a foundational component toward building more general, intelligent systems.

AI reasoning, broadly defined, refers to the computational ability to perform logical inference, knowledge-based deduction, and structured problem solving [9]. Early efforts in this domain were grounded in symbolic AI, particularly following the formulation of the Physical Symbol System Hypothesis by Newell and Simon [10] in the 1970s. Symbolic AI dominated from the 1950s to the 1980s, leveraging formal logic, handcrafted rules, and expert systems (e.g., MYCIN [11], DeepBlue [12]) to perform reasoning tasks. However, symbolic methods suffered from brittleness—high reliance on manually encoded knowledge made them ill suited for dynamic or open-ended environments with uncertainty and unstructured data. Since the 2010s, the deep learning revolution has brought tremendous success in perception-driven tasks [4]. End-to-end neural architectures like CNNs [13] and Transformers [14] have enabled machines to recognize images, generate text, and play games at superhuman levels. Yet, their capacity for robust, interpretable reasoning remains limited [5]. For example, while AlphaGo [15] defeated human champions, its success hinged on a hybrid system that combined deep neural networks with symbolic search techniques like Monte Carlo Tree Search. This highlighted a critical limitation: deep learning alone still struggles with structured reasoning, causal inference, and factual consistency [7]—capabilities essential for AGI.

This dichotomy—symbolic systems excel at reasoning but lack perception, while neural networks excel at perception but struggle with reasoning [16]—has led to growing interest in their integration. Leading researchers such as Yoshua Bengio [17] and Bo Zhang [18] have advocated for combining System 1 (intuitive, fast) and System 2 (deliberative, logical) reasoning within AI architectures. Real-world tasks—such as autonomous driving, where causal inference is needed in real time; medical diagnosis, where interpretability is crucial; and legal reasoning, where logical consistency is mandatory—further highlight the urgent need for AI systems that can reason as well as they perceive.

In response, the emerging paradigm of Neural–Symbolic AI has gained significant traction. This paradigm aims to unify symbolic logic and neural computation through techniques such as differentiable logic programming, knowledge-infused learning, and symbolic–neural interaction modules. Recent advances such as ∂ILP (Differentiable Inductive Logic Programming) [19,20], TransE-style knowledge graph embeddings [21], and NS-CL (Neural–Symbolic Concept Learner) [22] represent promising steps in this direction. Nonetheless, key challenges remain, including (1) the compatibility of discrete symbolic structures with continuous vector representations, (2) dynamic adaptation of logical rules, (3) balancing system complexity and efficiency, and (4) the absence of unified architectural frameworks.

This survey presents a comprehensive overview of AI reasoning in the deep learning era, with a focus on Neural–Symbolic AI, from historical, technical, and application perspectives. Beyond recounting historical developments, we contribute a unified and formalized view of AI reasoning by (1) introducing a formal definition and typology of reasoning functions across symbolic, statistical, and neural paradigms; (2) proposing a three-dimensional taxonomy that categorizes reasoning systems by representation type, inferential logic, and domain assumptions; and (3) offering a technically detailed, up-to-date synthesis of modern reasoning architectures, especially neuro-symbolic frameworks that integrate logic with deep learning. Our goal is not only to revisit reasoning history, but also to equip researchers with a structured conceptual map for navigating current challenges and future innovations in AI reasoning.

While several surveys have been conducted on reasoning in AI, some are temporally constrained or narrowly scoped—often limited to pre-deep learning symbolic systems or high-level descriptions of Neural–Symbolic AI [16,23,24]. Others focus solely on recent advancements in large language models (LLMs) [25] without fully examining their implications for structured reasoning. In contrast, this work seeks to offer a comprehensive, up-to-date, and technically grounded analysis of the field. We hope this survey provides a valuable resource for researchers and practitioners, facilitating deeper understanding and further innovation in AI reasoning.

Terminological Note. In this paper, the term *symbolic* refers to symbolic AI and logic-based reasoning, i.e., systems that perform inference via formal rules, symbolic structures, or discrete logic programs. This usage differs from that in *symbolic data analysis* (SDA) as studied in statistical learning, where "symbolic" refers to complex data types such as intervals, distributions, or multi-valued variables [26]. While both paradigms use symbolic representations, our focus is on symbolic *reasoning*, rather than symbolic *data summarization*.

2. Historical View: From Symbolic Systems to Hybrid Intelligence

2.1. A Brief History of AI Reasoning

Artificial intelligence reasoning has undergone a series of transformative paradigm shifts since the field's inception. Each era of development—symbolic, statistical, and neural–symbolic—has introduced new methods, capabilities, and limitations. In this section, we revisit the history of AI reasoning with a focus on identifying major methodological lineages within each era, their emergence timeline, core motivations, challenges, and representative systems. This lineage-based analysis serves as the conceptual foundation for constructing a reasoning paradigm evolution diagram in subsequent sections.

2.1.1. Symbolic Reasoning Era (1950s–1980s)

The symbolic era of AI, spanning roughly from the 1950s through the 1980s, was characterized by the dominance of logic-based and knowledge-driven methods for reasoning and decision-making. At its core was the belief that intelligent behavior could be achieved through explicit symbol manipulation, formalized primarily in logic systems and rule-based frameworks. Several distinct methodological lineages emerged during this era, each contributing uniquely to the field.

Logicism (Formal Logic-Based Reasoning). Pioneered in the 1950s, logicism posited that reasoning could be modeled using formal logic systems, primarily first-order logic. This tradition was rooted in cognitive science and mathematical logic and led to the development of early theorem provers and AI planning formalisms. Its key limitation was the inability to handle uncertainty, noise, and incomplete knowledge. This lineage also laid the foundation for logic programming paradigms such as Prolog [27], resolution-based theorem proving [28], and early symbolic rule learning systems such as Michalski's Variable-Valued Logic (VL1) framework [29]. This lineage also laid the foundation for logic programming paradigms, most notably Prolog, which became a cornerstone of symbolic AI in the 1970s and 1980s. VL1 extended classical logic by allowing predicates to take on flexible, multi-valued conditions, which enabled symbolic generalization from structured data and informed later developments in inductive learning.

Expert Systems. In the 1970s and 1980s, expert systems emerged as a practical application of symbolic reasoning. These systems used handcrafted rules encoded in knowledge bases to replicate human decision-making in specific domains. Despite early successes like MYCIN and DENDRAL [11,30], expert systems suffered from brittleness and the knowledge acquisition bottleneck. Notably, this lineage also encompassed production systems and rule-based engines such as CLIPS and OPS5 [31,32], as well as early constraint satisfaction solvers widely used in configuration and scheduling.

Non-monotonic and Default Logic. Traditional logic systems assumed monotonicity: once something is inferred, it remains true. However, real-world reasoning often requires retracting conclusions when new information arrives. This gave rise to non-monotonic reasoning formalisms such as default logic [33], circumscription [34], and later, answer set programming [35]. These methods better captured commonsense reasoning but introduced high computational complexity. Related developments such as truth maintenance systems (TMSs) [36] and belief revision frameworks were also introduced to support the dynamic consistency of knowledge bases under new information.

Planning Systems. Classical planning frameworks, such as STRIPS [37], emerged in the 1970s, framing reasoning as a sequence of state transitions governed by logical operators. These methods were effective in controlled environments but faced scalability issues in large or dynamic domains. A notable early planning robot was Shakey [38] which was the world's first mobile intelligent robot.

Argumentation-Based Reasoning. Emerging in the late 1980s, computational argumentation sought to model reasoning as a dialectical process of supporting and attacking claims. Dung's abstract argumentation framework [39] formalized argument acceptability, laying the foundation for structured approaches like ASPIC+ [40].

Semantic Networks and Description Logic. Efforts to formalize conceptual hierarchies led to semantic networks and later, description logic (DL). DL offered a decidable fragment of first-order logic with well-defined semantics, which eventually became the basis for ontologies and the Semantic Web [41–43].

Modal and Temporal Logic. Developed during the 1960s and 1970s, modal logic introduced operators to express necessity, possibility, belief, and knowledge—enabling reasoning about epistemic states and agent capabilities [44]. Temporal logic, including linear and computation tree logics, provided tools for modeling the evolution of system states over time [45,46]. These formalisms laid the foundation for AI planning under uncertainty, program verification, and multi-agent reasoning, though their adoption was often limited by expressiveness and tractability trade-offs.

These streams collectively shaped the symbolic foundations of AI reasoning (Table 1). Despite their strengths in interpretability and formal rigor, symbolic approaches struggled with scalability, brittleness, and perceptual grounding—limitations that motivated the shift toward data-driven methods in the following decades.

Table 1. Key methodological lineages in the symbolic AI reasoning era.

Methodological Lineage	Time of Emergence	Core Idea	Key Challenges	Representative Works/ Systems
Logicism (Formal Logic)	1950s–1960s	Deductive reasoning via formal logic	Noisy data, scalability	GPS [28], Resolution Theorem Proving, Prolog [27], VL1 [29]
Expert Systems	1970s–1980s	Rule-based knowledge representation	Brittleness, knowledge acquisition bottleneck	MYCIN [11], DENDRAL [30], CLIPS [31], OPS5 [32]
Non-monotonic Logic	1980s	Reasoning with defaults and retractable beliefs	Computational complexity	Default Logic [33], Circumscription [34], ASP [35], TMS [36]

Table 1. *Cont.*

Methodological Lineage	Time of Emergence	Core Idea	Key Challenges	Representative Works/ Systems
Planning Systems	1970s	Action sequences via logical state transitions	State-space explosion	STRIPS [37], Shakey [38]
Argumentation	Late 1980s–1990s	Reasoning via structured debates and conflicts	Semantics ambiguity, expressivity limits	Dung's AF [39], ASPIC+ [40]
Semantic Networks and DL	1970s–1990s	Hierarchical concept representation and ontology	Limited expressivity, integration with logic	KL-ONE [41], Description Logic [42], OWL [43]
Modal and Temporal Logic	1960s–1980s	Reasoning with modalities and temporal operators	Integration with learning, decidability	Epistemic Logic [44], LTL, and CTL [45,46]

2.1.2. Statistical and Data-Driven Era (1990s–2010s)

The second major era of AI reasoning emerged in response to the limitations of purely symbolic systems, particularly their brittleness, poor scalability, and inability to handle uncertainty. From the 1990s onward, the field increasingly embraced data-driven and statistical approaches, fueled by the rise in machine learning and the growing availability of real-world data. This era introduced reasoning frameworks capable of modeling uncertainty, learning from data, and making probabilistic inferences.

Probabilistic Graphical Models (PGMs). Probabilistic graphical models, including Bayesian Networks [47] and Markov Networks [48], provided a powerful formalism to represent and reason under uncertainty. They captured dependencies among random variables using graph structures, enabling efficient probabilistic inference and decision-making. PGMs were widely used in diagnosis, prediction, and robotics, but often required expert-designed structures and struggled with high-dimensionality.

Markov Logic Networks (MLNs). As a bridge between logic and probability, MLNs extended first-order logic by associating weights with formulas [49], thereby relaxing strict logical inference into probabilistic reasoning. This hybrid approach addressed the rigidity of symbolic logic while retaining structure. However, inference in MLNs remained computationally expensive, and scalability to large domains remained an open challenge.

Probabilistic Logic Programming. This lineage integrated probabilistic reasoning directly into logic programming paradigms. Languages such as ProbLog [50], PRISM [51], and LPAD [52] extended traditional logic programming with probabilistic facts and rules. These systems supported structured and interpretable inference under uncertainty, often leveraging symbolic reasoning engines as their computational backbone.

Statistical Relational Learning (SRL). SRL frameworks generalized machine learning methods to relational domains with complex structures. Approaches such as Relational Bayesian Networks [53] and Relational Markov Networks [54] enabled learning from relational data with shared statistical patterns. Historical precursors to SRL include Inductive Logic Programming (ILP), which aimed to generalize logic rules from structured examples [55]. However, classical ILP systems lacked robustness to noise and uncertainty, which limited their scalability and integration with probabilistic inference.

In parallel with the rise in Inductive Logic Programming and statistical relational learning, researchers also explored conceptual clustering as a form of symbolic structure discovery [56–58]. One notable example is COBWEB [58], an incremental conceptual clustering algorithm that constructs hierarchical taxonomies from symbolic attributes using category utility as a guiding metric. Unlike classical numerical clustering, COBWEB produces interpretable concept trees that reflect human-like abstraction processes. These systems demonstrated that symbolic representations and pattern discovery could be integrated in an unsupervised manner—an idea that remains relevant in modern efforts to learn structured latent spaces for reasoning.

Causal Inference and Structural Causal Models. Inspired by the philosophy of science and epidemiology, this stream focused on modeling causal relationships rather than mere correlations. Structural causal models (SCMs), popularized by Judea Pearl, introduced formal tools such as do-calculus for inferring interventions, counterfactuals, and causal effects [7,59]. These methods enriched the reasoning landscape with mechanisms for understanding and manipulating cause–effect relationships.

Kernel Methods and Shallow Statistical Learning. Before deep learning became dominant, statistical reasoning relied heavily on feature-based methods such as SVMs [60], decision trees [61], and ensemble models [62]. Though not symbolic, these methods provided interpretable decision rules and were effective in many structured reasoning tasks, particularly in classification, regression, and ranking.

Deep Neural Networks for Sub-symbolic Reasoning. With the rise in deep learning, neural networks began to exhibit emergent capabilities in representation-based reasoning. Architectures such as convolutional neural networks (CNNs) [63], recurrent neural networks (RNNs) [64], graph neural networks (GNNs) [65], and especially Transformers [14], enabled models to learn hierarchical abstractions, capture compositional patterns, and generalize to novel input combinations. Though these systems lack explicit symbolic representations, they demonstrated powerful inductive capabilities in vision, language, and decision-making tasks. For example, AlphaGo integrated deep neural policy networks with symbolic search (Monte Carlo Tree Search) [15], foreshadowing later neural–symbolic systems. However, purely neural models struggle with interpretability, logical consistency, and systematic generalization in abstract reasoning tasks.

A notable frontier within sub-symbolic reasoning is the emergent reasoning phenomenon observed in large foundation models. Although these models lack explicit symbolic structures or logical inference mechanisms, they exhibit surprising capabilities in multi-step problem solving, analogical inference, and even structured planning—solely by virtue of model scale and in-context learning. These emergent behaviors are best interpreted as an implicit extension of sub-symbolic reasoning, and are distinct from neural–symbolic systems that explicitly incorporate logical form.

Overall, the statistical era emphasized generalization from data, uncertainty modeling, and structural regularities, laying the foundation for modern machine learning (Table 2). However, many of these approaches sacrificed interpretability, relied on strong assumptions, and lacked the rich semantic expressivity of symbolic systems—limitations that would later motivate the integration of symbolic and neural paradigms.

Table 2. Historical lineage of statistical and sub-symbolic reasoning paradigms.

Methodological Lineage	Time of Emergence	Core Idea	Key Challenges	Representative Works / Systems
Probabilistic Graphical Models	1990s	Inference over uncertain variables using graphs	Structure design, scalability	Bayesian Networks [47], Markov Networks [48]
Markov Logic Networks	Mid-2000s	Softening logical rules with probabilistic weights	Inference complexity, large-scale training	MLNs [49], Alchemy [66]

Table 2. *Cont.*

Methodological Lineage	Time of Emergence	Core Idea	Key Challenges	Representative Works / Systems
Probabilistic Logic Programming	Early 2000s	Extending logic programming with probabilistic facts	Semantic complexity, grounding issues	ProbLog [50], PRISM [51], LPAD [52]
Statistical Relational Learning	2000s	Learning statistical patterns in relational domains	Overfitting, relational sparsity	Relational Bayes Nets [53], Relational Markov Networks [54], Tuffy [67], ILP [55]
Causal Inference and SCMs	1990s–2000s	Modeling interventions and counterfactuals	Assumptions, identifiability	SCMs [7], do-calculus [59]
Deep Neural Networks (DNNs)	2006	End-to-end learning for hierarchical abstraction	Lack of logic structure, poor interpretability	CNN [63], RNN [64], GNN [65], Transformer [14], AlphaGo [15]
Kernel and Ensemble Methods	1990s–2010s	Structured prediction using statistical learning	Feature engineering, limited abstraction	SVM [60], Decision Trees [61], Random Forests [62]

2.1.3. Neural–Symbolic Integration Era (2016–Present)

In recent years, the AI community has witnessed a renewed interest in combining the strengths of symbolic reasoning with the representational power of neural networks. This has given rise to the neural–symbolic integration paradigm, which aims to bridge the long-standing divide between logic-based inference and gradient-based learning. The rise in deep learning, the emergence of large-scale pretrained language and vision models, and the increasing demand for explainable and structured reasoning have all contributed to the rapid development of this hybrid field.

Differentiable Logic Programming. One of the foundational directions in neural–symbolic reasoning is the development of differentiable logic systems. Methods such as ∂ILP [19], Neural Logic Machines [68], and Logical Tensor Networks [69] introduce gradient-based mechanisms to induce and evaluate logical rules. These systems offer differentiability and symbolic interpretability, but often face scalability limitations and require careful design to avoid degeneracy in optimization.

Abductive Learning. Proposed by Zhou and colleagues [70], abductive learning is a neuro-symbolic reasoning paradigm that integrates neural perception with symbolic abductive inference. The central idea is to generate plausible symbolic explanations for observed data using background knowledge and abductive logic programming, and use these explanations to supervise the learning of neural perception modules. Unlike purely deductive or inductive approaches, abductive learning enables reasoning with incomplete observations and missing logical components. It has been applied to handwritten equation understanding, visual question answering, and semantic parsing, demonstrating strong generalization and interpretability. The original framework was proposed by Lin and Zhou [70]. However, it faces challenges in grounding symbols, scaling to large search spaces, and integrating with gradient-based models.

Neuro-symbolic Concept Learners and Program Induction. Another stream focuses on neural systems that learn structured programs, symbolic rules, or modular logic graphs from data. Approaches such as NS-CL [22], Neural Module Networks [71], and CLEVR-CoGenT [72] construct interpretable logic chains over visual or textual inputs.

These systems are especially effective in visual question answering, relational reasoning, and grounded language understanding.

LLM-guided Neural–Symbolic Reasoning. With the rise in LLMs such as GPT-4 [73] and Claude [74], researchers have explored their capacity to perform reasoning through chain-of-thought prompting, tree-structured generation, and API/tool augmentation. Frameworks like ReAct [75], Toolformer [76], and DSPy [77] use LLMs as symbolic planners, orchestrating external tools for structured decision-making. These approaches offer flexibility and generalization, though they often struggle with consistency, faithfulness, and verifiability. These models also demonstrate what some researchers describe as emergent reasoning—a form of symbolic-like behavior not grounded in explicit logical structure but arising from large-scale pattern learning and in-context processing.

Logic-aware Transformer Architectures. Recent work has extended standard Transformer architectures with symbolic inductive bias. Models such as Logical Transformers [78] and LogicBench [79] integrate logical constraints, graph structures, or discrete operators into the attention mechanism or decoder path. These systems aim to combine end-to-end learning with structural regularization and symbolic supervision.

Neural Theorem Provers and Knowledge Injection. Other approaches focus on integrating symbolic knowledge into neural architectures via knowledge graphs, ontology embeddings, or reasoning constraints. Systems like DeepProbLog [80], NeurASP [81], and K-BERT [20] use neural–symbolic hybrids to perform logical inference grounded in structured knowledge. They demonstrate strong performance in knowledge-based QA, commonsense inference, and scientific reasoning tasks.

Multimodal Neuro-symbolic Reasoning. Extending beyond text, recent work applies neuro-symbolic methods to multimodal domains including visual reasoning, video event understanding, and robotic planning. Models like NS-CL [22], CLEVR-CoGenT [72], VideoCoT [82], and ViperGPT [83] demonstrate that grounding symbolic structures in perception significantly improves generalization and reasoning robustness.

Overall, the neural–symbolic era represents a convergence of paradigms: the statistical generalization of deep learning, the formal rigor of symbolic logic, and the flexibility of large-scale pretrained models (Table 3). Despite ongoing challenges—including training stability, scalability, and semantic consistency—this paradigm is considered a promising path toward interpretable, robust, and generalizable AI reasoning.

Table 3. Historical lineage of neural–symbolic reasoning paradigms.

Methodological Lineage	Time of Emergence	Core Idea	Key Challenges	Representative Works/Systems
Differentiable Logic Programming	2016	Learnable logical rules via gradients	Optimization stability, scalability	∂ILP [19], Logical Tensor Networks [69], Neural Logic Machines [68]
Abductive Learning	2019–	Combining symbolic abduction with neural perception	Symbol grounding, search complexity	Abductive Learning Framework [70], ABL-KG [84]
Program Induction and NS-Concept Learners	2016–	Learning symbolic programs and structures	Compositionality, sample efficiency	NS-CL [22], Neural Module Networks [71], CLEVR-CoGenT [72]
LLM-based Reasoning and Tool Use	2022–	Prompt-driven symbolic planning via LLMs	Verifiability, hallucination, tool coverage	ReAct [75], Toolformer [76], DSPy [77]

Table 3. *Cont.*

Methodological Lineage	**Time of Emergence**	**Core Idea**	**Key Challenges**	**Representative Works/Systems**
Logic-aware Transformers	2021–	Structural priors in Transformer architectures	Complexity, generalization	LogicT5 [85], Logical Transformers [78]
Knowledge-augmented Reasoning	2019–	Injecting KG and symbolic structure into models	Alignment, representation mismatch	DeepProbLog [80], NeurASP [81], K-BERT [20]
Multimodal Neuro-symbolic Reasoning	2020–	Reasoning over visual and multimodal inputs	Visual grounding, temporal logic	NS-CL [22], CLEVR-CoGenT [72], VideoCoT [82], ViperGPT [83]

2.2. *Definition and Formalization of AI Reasoning*

AI reasoning refers to the capability of an artificial system to derive conclusions, explanations, or decisions based on a set of premises, background knowledge, or observed information—often through a structured, generalizable, and semantically meaningful process [7,9,86]. Reasoning is distinct from perception or raw classification in that it involves inferential relationships, abstract manipulation, and goal-oriented decision chains that may go beyond direct pattern recognition. Historically, AI reasoning has been grounded in formal logic, such as propositional and first-order logic [87], and it evolved through multiple computational paradigms. Depending on the paradigm, the underlying mechanism and representation of reasoning vary significantly. We review these core characterizations below.

2.2.1. Formal Characterizations Across Paradigms

We now formalize AI reasoning across the major paradigms discussed in Section 2.1. While their underlying assumptions differ, these paradigms share a common goal: to map structured knowledge and observations into logically or semantically grounded conclusions. Each paradigm instantiates the abstract reasoning function:

$$\mathcal{R} : (\text{ Knowledge, Observation }) \to \text{Inferred Conclusion.} \tag{1}$$

We analyze these three major paradigms and their representative instantiations below.

Symbolic Reasoning. Symbolic reasoning is grounded in classical logic, where reasoning involves deriving conclusions from explicit knowledge bases and input premises using syntactic rules, as follows:

$$\mathcal{R}_{\text{sym}} : (\mathcal{K}, \Gamma) \to \Delta \quad \text{where} \quad \Gamma \cup \mathcal{K} \models \Delta, \tag{2}$$

where

- $\mathcal{K}$ is background knowledge (e.g., axioms, ontologies);
- Γ are current inputs or observations;
- Δ are conclusions derived by logical entailment;
- $\models$ is syntactic or semantic entailment (e.g., modus ponens, resolution).

This paradigm underpins classical expert systems and theorem provers [31,88], with strong interpretability but limited generalization in open domains.

Statistical and Sub-symbolic Reasoning. Statistical reasoning models uncertainty and correlation between observed and unobserved variables using probabilistic structures. A general reasoning model follows:

$$\mathcal{R}_{\text{stat}} : P(Y \mid X, \theta) \rightarrow \arg\max_{Y} \mathbb{E}_{\theta}[Y \mid X], \tag{3}$$

where

- X are observed data (e.g., features, facts);
- Y is the target variable to be inferred;
- K is prior knowledge encoded in a probabilistic graphical model;
- θ are model parameters (e.g., conditional probabilities);
- $\hat{Y}$ is the most probable or expected outcome.

This paradigm includes Bayesian Networks [47], MLNs [49], probabilistic logic programming (e.g., ProbLog [50]), and probabilistic soft logic [89]. It supports reasoning under uncertainty but often requires handcrafted structures or supervision.

As a key modern subclass of statistical reasoning, sub-symbolic reasoning refers to inference performed via continuous representations learned from data, where inference is implicit within neural function approximators like the following:

$$\mathcal{R}_{\text{sub}} : f_{\theta}(X) \rightarrow Z \quad \text{where} \quad Z \in \mathbb{R}^{d}, \tag{4}$$

where

- X is the high-dimensional input (e.g., image pixels, word tokens, graph embeddings);
- f_{θ} is the neural model parameterized by θ (e.g., CNNs [63], GNNs [65], Transformers [14]);
- $\hat{Y}$ is the task-dependent output (e.g., answer, class label, entity).

Though not based on formal logic, such models demonstrate compositional generalization and relational abstraction [90] in tasks like visual question answering, multi-hop QA, and analogical reasoning.

Recent foundation models (e.g., GPT-4 [73], Claude [74], LLaMA [91]) demonstrate reasoning-like behaviors in tasks such as multi-step question answering, numerical inference, and analogical problem solving. These capabilities emerge without explicitly defined symbolic rules or logic modules, but instead arise from the scale and pattern capacity of Transformer architectures trained on large corpora [92,93]. We interpret such phenomena as an extension of sub-symbolic reasoning, where inference is not computed through logic chains, but statistically approximated through prompt-conditioned generation.

$$\mathcal{R}_{\text{emergent}} : T(X, P) \rightarrow \hat{Y}, \tag{5}$$

where

- X is the user input or natural context (e.g., question, image caption);
- P is the prompt scaffold or few-shot template (e.g., chain-of-thought);
- T is a large pretrained Transformer model;
- $\hat{Y}$ is the model-generated output (e.g., answer, plan, proof explanation).

While these behaviors are not grounded in verifiable logic, they exhibit surprising reasoning fluency across many domains. It is important to distinguish this prompt-only emergent reasoning from LLM-augmented neural-symbolic systems—discussed separately in Section 2.1.3—that explicitly integrate external symbolic modules or structured APIs into the reasoning loop.

Neural–Symbolic Reasoning. Neural–symbolic systems aim to combine the expressivity of symbolic structures with the flexibility and scalability of neural networks. These systems typically integrate logic-based priors or rules $\mathcal{K}$ into a neural model f_{θ}, resulting in hybrid architectures capable of both pattern recognition and structured reasoning. We formalize the reasoning process as follows:

$$\mathcal{R}_{\text{ns}} : f_\theta(X) + \mathcal{K} \to \Delta, \tag{6}$$

where

- X are input observations (e.g., a question, an image, a scene graph);
- f_θ is a neural encoder or predictor parameterized by θ;
- $\mathcal{K}$ is symbolic knowledge, such as rules, ontologies, or graphs;
- Δ are structured outputs inferred jointly from symbolic and neural components.

This reasoning paradigm supports end-to-end learning while allowing the incorporation of explicit reasoning structures. Depending on the implementation, the symbolic component may appear in different forms:

- Differentiable logic layers: logic rules are approximated using tensors or neural operators (e.g., ∂ILP [19], Logical Tensor Networks [69]);
- Neuro-symbolic concept learning: symbolic program execution is conditioned on visual or textual concept modules (e.g., NS-CL [22], Neural Module Networks [71]);
- Knowledge-guided Transformers: structured external knowledge is injected into pretrained models (e.g., K-BERT [20], NeurASP [81]).

Neural–symbolic reasoning offers a middle ground between interpretability and adaptability. It allows AI systems to learn from data while reasoning over known structures, making it particularly useful in domains such as scientific QA, medical diagnosis, and law, where both statistical inference and logical guarantees are essential.

2.2.2. Categorization of AI Reasoning Across Dimensions

Beyond historical paradigms and formal mechanisms, AI reasoning can be further categorized along multiple orthogonal axes. These dimensions help situate diverse reasoning systems in a unified framework and facilitate comparative analysis. We propose a three-dimensional taxonomy based on *representation type*, *task structure*, and *application context*.

By Representation Type. This axis characterizes the internal form of knowledge and reasoning operations ranging from explicit logical formulas to implicit statistical embeddings, including the following:

- **Symbolic Reasoning**: Relies on discrete, human-interpretable representations such as logic rules, graphs, and ontologies [94,95]. Inference is typically performed using deductive or rule-based systems, enabling traceability and formal verification [88]. Such systems dominate early expert systems and theorem provers.
- **Statistical Reasoning**: Model uncertainty using probability distributions over structured data. Reasoning tasks involve belief updating, probabilistic inference, and marginalization, often leveraging tools like Bayesian Networks [47], HMMs [96], or MLNs [49]. Logic-based probabilistic systems such as ProbLog [50] and PSL [89] also fall under this category.
- **Neural Reasoning**: Employs continuous, learned representations within neural networks. Inference emerges from multi-layer transformations and pattern abstraction, without explicit rule structures. Despite its black-box nature, neural reasoning has demonstrated success in perception-rich and language-heavy tasks [14,90].
- **Hybrid (Neural–Symbolic) Reasoning**: Attempts to unify the interpretability of symbolic models with the flexibility of neural networks. This includes architectures that inject symbolic priors into differentiable computation (e.g., ∂ILP [19], Log-

ical Tensor Networks [69]), or use neural controllers to invoke symbolic tools (e.g., NS-CL [22], NeurASP [81], K-BERT [20]).

By Task Structure. This axis focuses on the inferential logic underlying the reasoning process, reflecting different modes of human-like thinking, including the following:

- **Deductive Reasoning**: Draws logically valid conclusions from known premises or rules. It operates under certainty and preserves truth, forming the foundation of theorem provers [88], symbolic solvers [97], and classical logic programming [94].
- **Inductive Reasoning**: Generalizes from specific instances to broader rules or models. Typical in scientific discovery and machine learning, this paradigm underlies systems like Inductive Logic Programming (e.g., FOIL [98], Meta-ILP [99]), and concept generalization.
- **Abductive Reasoning**: Seeks the most plausible explanation for an observation. It is used extensively in diagnosis, plan recognition, and commonsense reasoning, where causes must be inferred from effects [70,84].
- **Analogical Reasoning**: Solves unfamiliar problems by mapping structures from previously encountered scenarios. This approach underlies analogical question answering, metaphor understanding, and visual analogy [100,101].

By Application Context. This axis describes the environment in which reasoning is applied, emphasizing the domain's structural assumptions and complexity, including the following:

- **Closed-domain Reasoning**: Operates in well-defined, highly structured environments where rules and ontologies are fixed and comprehensive. Common in robotic control [102], rule-based planning [37], and legal document validation [103], these systems prioritize correctness and determinism.
- **Open-domain Reasoning**: Engages with ambiguous, dynamic, and incomplete knowledge sources. It encompasses multi-hop question answering [104], visual reasoning [72,82], dialogue systems [105], and scientific exploration [106], where models must cope with noise, novelty, and partially observed states.

This multidimensional taxonomy enables a more fine-grained understanding of the diverse methodologies in AI reasoning and provides a scaffold for comparing their assumptions, strengths, and limitations. It also clarifies the trade-offs between generalization and precision, expressiveness and tractability, and learning and interpretability (Table 4).

Table 4. Three-dimensional categorization of AI reasoning across representation, task structure, and application context. Each cell lists representative methods under *Closed* (C) and *Open* (O) domains.

Task Structure	Symbolic Reasoning	Statistical Reasoning	Neural Reasoning	Neural–Symbolic Reasoning
Deductive Reasoning	C: Theorem Provers [88], Datalog [94] O: Ontology-based QA [107], Legal Inference [95]	C: Probabilistic Rules (MLN) [49] O: MLN for OIE QA [49]	C: Logic-aware Transformers [85] O: Chain-of-thought LLMs [92]	C: NeurASP [81], Neuro-symbolic Planners [108] O: Toolformer [76], DSP-Logic [77]
Inductive Reasoning	C: ILP [98], FOIL [98] O: Meta-ILP [99], Meta-Interpretive Learning [109]	C: Bayesian Nets [47], ProbLog [50] O: Probabilistic Program Synthesis [110]	C: NLM [68], TreeLSTM Learners [111] O: GPT Concept Learners [112]	C: NS-CL [22], Neural Logic Machines [68] O: SceneGraph Reasoning [113]

Table 4. *Cont.*

Task Structure	Symbolic Reasoning	Statistical Reasoning	Neural Reasoning	Neural–Symbolic Reasoning
Abductive Reasoning	C: Logic Diagnosis [114], ABox Completion [95] O: Plan Recognition [115], Commonsense Inference [70]	C: Probabilistic Causal Models [7] O: PSL for Temporal Explanations [89]	C: Causal LLMs, Plan Explanation [116] O: Emergent causal LLMs [93]	C: Abductive Learning [70], Rule Induction [80] O: DSPy [77], Script Induction + Symbol Decoders [117]
Analogical Reasoning	C: Structure Mapping Engines [100] O: Metaphor Resolution Systems [118]	C: Similarity Search over Graphs [119] O: Retrieval-aug. Analogical QA [120]	C: CLIP-based Visual Analogy [101] O: GPT-4 Analogical QA [121], Visual CoT [82]	C: Neural Module Networks (NMN) [71] O: Tool-enhanced Analogy QA [76]

3. Technical View: Architectures, Mechanisms, and Trends of AI Reasoning in Deep Learning Era

With the historical landscape and taxonomic structure of AI reasoning in place, we now turn our attention to recent developments of reasoning under the deep learning paradigm. While classical paradigms emphasized explicit logical representations and rule-based inference, the explosive progress of deep neural networks since 2015 has fundamentally reshaped how reasoning is conceptualized, modeled, and implemented in AI systems. A pivotal turning point emerged around 2019, when leading researchers such as Yoshua Bengio and Yann LeCun highlighted the need to transcend purely perceptual, reactive intelligence—what cognitive science terms “System 1”—and pursue “System 2” capabilities, which emphasize abstraction, composition, causality, and structured reasoning [122,123]. This transition reflects a shift from learning statistical correlations to discovering explainable mechanisms and inference chains. This vision inspired a new class of reasoning architectures that build on deep learning foundations while incorporating symbolic knowledge, logical supervision, causal priors, or in-context prompting. The resulting systems are not merely more accurate or scalable—they seek to endow AI with interpretability, generalization, and reasoning fidelity across diverse domains. In the following, we provide a systematic overview of current AI reasoning frameworks in the deep learning era, especially the neural-symbolic methods [124].

As artificial intelligence increasingly transitions from perception-dominated tasks to those requiring structured inference, the ability to reason has become a defining characteristic of advanced AI systems. In the deep learning era, reasoning is no longer confined to rigid symbolic rules or handcrafted pipelines but emerges from trainable, scalable, and data-driven architectures. This new generation of reasoning systems aims to combine the generalization strength of neural networks with the structure, consistency, and interpretability of classical reasoning. However, enabling reasoning in neural systems introduces three fundamental challenges:

- Knowledge-Representation Integration: How can background knowledge, logical rules, or ontologies be injected into gradient-based models while preserving learnability and robustness?
- Inference Control and Modularity: How can neural architectures learn and execute structured reasoning steps—such as multi-hop deduction or conditional branching—without explicit supervision or logic templates?

- Consistency, Verifiability, and Hallucination Mitigation: Neural models, especially LLMs, often generate plausible but factually incorrect inferences. How can we constrain these systems to produce faithful and logically sound conclusions?

To address these issues, a diverse array of architectures has emerged, blending symbolic logic, probabilistic structures, large-scale pretraining, and task-specific supervision. In the sections that follow, we survey these reasoning models along seven primary categories, highlighting their core mechanisms, strengths, limitations, and applications across different domains.

3.1. Differentiable Logic Programming

Differentiable Logic Programming (DLP) represents a foundational paradigm in Neural–Symbolic AI, aiming to reconcile the strengths of logical rule-based reasoning with the learning capacity and scalability of deep neural networks. The primary goal of this line of work is to allow logic-based inference to be embedded within end-to-end trainable models, enabling systems to both learn logical rules from data and reason over them using gradient descent. This paradigm is particularly relevant for tasks that require relational generalization, interpretable rule induction, or symbolic extrapolation from limited supervision.

DLP methods formulate logic programs using continuous, differentiable structures that approximate symbolic reasoning. The core mechanism involves mapping logical rules and atoms to differentiable tensors or attention mechanisms and using learnable parameters to control rule application. This design enables reasoning operations—such as unification, forward chaining, and conjunction—to be approximated within neural architectures. Formally, a DLP can be modeled as a tuple:

$$\mathcal{M} = (\mathcal{F}, \mathcal{H}_\theta, \mathcal{R}_\theta, \mathcal{L}), \tag{7}$$

where

- $\mathcal{F}$ is observable or given knowledge, such as symbolic facts extracted from structured data (e.g., knowledge graphs), outputs of perception modules (e.g., scene relations), or labeled logical atoms;
- $\mathcal{H}_\theta = \{(\theta_i : r_i) \mid r_i \in \text{ Rule Templates}, \theta_i \in [0,1]\}$ is a parameterized set of logic rule templates. In some settings, the structure of rules (e.g., Horn clauses) is predefined, and only rule weights are learned. In more general settings, both the structure and weights of rules can be induced from data—commonly referred to as Neural Inductive Logic Programming (Neural ILP). This variant enables the system to discover, select, or generate effective symbolic patterns during training;
- $\mathcal{R}_\theta$ is a differentiable reasoning engine that approximates logical inference, such as matrix computation, graph neural networks, tensor composition, and probabilistic logic semantics, which may differ among different methods;
- $\mathcal{L}$ is a loss function combining predictive error and rule regularization.

The goal is to learn parameters θ such that the model can approximate symbolic inference. The overall reasoning process is defined as follows:

$$\hat{y} = \mathcal{R}_\theta(\mathcal{F}, \mathcal{H}_\theta) \approx \text{Entail}(\mathcal{F}, \mathcal{H}). \tag{8}$$

Here, R_θ is a neural module that simulates the application of rules $\mathcal{H}_\theta$ over facts $\mathcal{F}$, $\text{Entail}(\mathcal{F}, \mathcal{H})$ denotes the set of conclusions logically derived from applying symbolic rules $\mathcal{H}$ to the facts $\mathcal{F}$ via forward chaining or resolution, and $\hat{y}$ represents the model's prediction (e.g., probabilities of inferred atoms or labels). The learning objective is formulated as

$$\min_{\theta} \mathcal{L}_{\text{task}}(\hat{y}, y) + \lambda \cdot \mathcal{L}_{\text{consistency}}(\mathcal{H}_\theta), \tag{9}$$

where

- $\mathcal{L}_{\text{task}}$ is standard task loss (e.g., cross-entropy);
- $\mathcal{L}_{\text{consistency}}$ is a rule-level regularization term that enforces sparsity, logical consistency, or syntactic constraints over the soft rule set;
- λ is a hyperparameter controlling the strength of the symbolic regularizer.

Example 1. *Consider a symbolic task where the goal is to learn and apply relational rules to infer family relationships. We define each component of the DLP model $\mathcal{M} = (\mathcal{F}, \mathcal{H}_\theta, \mathcal{R}_\theta, \mathcal{L})$ as follows:*

- ***Facts:*** $\mathcal{F} = \{parent(alice, bob),\ parent(bob, carol)\}$;
- ***Soft Rule Set:*** $\mathcal{H}_\theta = \{(r_1, \theta_1)\}$, *where the rule template r_1 is $grandparent(X, Z) \leftarrow parent(X, Y),\ parent(Y, Z)$ and $\theta_1 \in [0, 1]$ is a learnable rule confidence weight;*
- ***Reasoning Engine:*** $\mathcal{R}_\theta(\mathcal{F}, \mathcal{H}_\theta) = \sum_{(r_i, \theta_i) \in \mathcal{H}_\theta} \theta_i \cdot Apply(r_i, \mathcal{F})$ *aggregates soft predictions from each rule application. The operator $Apply(r_i, \mathcal{F})$ simulates differentiable forward chaining for rule r_i over facts $\mathcal{F}$;*
- ***Supervision Target:*** $y = \{grandparent(alice, carol)\}$;
- ***Training Objective:*** $\mathcal{L} = \mathcal{L}_{task}(\hat{y}, y) + \lambda \cdot \mathcal{L}_{consistency}(\mathcal{H}_\theta)$, *where $\mathcal{L}_{task}$ is cross-entropy loss between predicted and ground truth facts, and $\mathcal{L}_{consistency}$ regularizes rule sparsity or logical coherence.*

In this example, the rule r_1 is converted into a differentiable operator—typically a neural module that composes relational matrices. Suppose all binary predicates are encoded as adjacency matrices, e.g., $M_{\text{parent}} \in \{0, 1\}^{n \times n}$, where $[M_{\text{parent}}]_{i,j} = 1$ if $\text{parent}(i, j) \in \mathcal{F}$. Then the predicted matrix for the grandparent relation is computed as

$$\hat{M}_{\text{gp}} = \theta_1 \cdot M_{\text{parent}} \cdot M_{\text{parent}},$$

which performs soft relation composition, weighted by θ_1. The predicted score $\hat{y}_{alice,carol} = [\hat{M}_{\text{gp}}]_{0,2}$ reflects the model's belief in the fact $\text{grandparent}(alice, carol)$.

During training, supervision is provided as a binary label $y_{alice,carol} = 1$. The task loss is computed as

$$\mathcal{L}_{\text{task}} = -y \cdot \log \hat{y} - (1 - y) \cdot \log(1 - \hat{y}),$$

and gradients are backpropagated through the reasoning graph to update θ_1. Simultaneously, the consistency loss $\mathcal{L}_{\text{consistency}}$ may enforce inductive biases such as rule sparsity or prior knowledge.

Over the past decade, a rich and expanding body of work has emerged in DLP, addressing key challenges in rule induction, scalable inference, and neural integration. We categorize the representative methods in this space into four major classes based on their modeling emphasis and architectural design.

(1) Rule Learning and Structure Induction. This class of methods focuses on learning logical rules from data, typically in the form of differentiable Horn clauses or logic program templates. ∂ILP [19] introduces a soft unification mechanism and gradient-based optimization for learning weighted rule sets. Neural LP [125] uses differentiable attention over rule paths for relational reasoning. These foundational frameworks have been extended to accommodate high-dimensional feature spaces (dILP-HDS [126]), matrix-based semantics (DFOL [127]), and structured symbolic inputs under noise (Structured ILP [128]). All such approaches define an explicit hypothesis space over logical clauses and incorporate differentiable mechanisms for structure learning. In addition to existing methods, DFORL

(Differentiable First-Order Rule Learner) [129] introduces a scalable differentiable ILP approach capable of learning first-order Horn clauses from both small and large datasets, including knowledge graphs. Notably, DFORL requires only the number of variables in the target logic programs as input, eliminating the need for strong language biases and enhancing flexibility and usability.

(2) Differentiable Inference and Proof Engines. These methods emphasize differentiable realizations of logical reasoning, including forward chaining and proof construction. Neural Theorem Provers (NTP) [130] model soft unification and recursive proof composition in vector space, while TensorLog [131] compiles logic into sparse matrix operations. Subsequent extensions include DLM [132] for symbolic relaxation in forward chaining, and NEUMANN [133], which achieves memory-efficient reasoning via graph-based message passing. Takemura et al. [134] further demonstrate how rule grounding can be encoded differentiably under distant supervision.

(3) Declarative Logic as Differentiable Constraints. A distinct line of work embeds logical formulas into neural networks using differentiable logic semantics. Logic Tensor Networks (LTNs) [135] and Logical Neural Networks (LNNs) [136] implement fuzzy logic operations (e.g., conjunction, disjunction, implication) via t-norms and real-valued logic gates. NeuraLogic [137] compiles symbolic logic into graph-based computations within neural message passing frameworks. The dPASP system [138] extends probabilistic answer set programming with neural predicates and interval-valued probabilistic facts, supporting learning under complex semantic regimes, including multiple stable-model interpretations. Scallop [139] presents a general-purpose neurosymbolic programming language that combines relational data modeling, a Datalog-based declarative logic programming language supporting recursion, aggregation, and negation, and an efficient differentiable reasoning framework based on provenance semirings. Scallop facilitates the development of a wide range of neurosymbolic applications, providing a succinct interface for integrating logical domain knowledge into machine learning workflows.

(4) Vision-Oriented DLP. A growing subset of DLP systems are designed for integration with visual perception. αILP [140] formulates visual scene understanding as logical rule induction, using differentiable programs to capture object-level reasoning and visual relations. NEUMANN, while technically general, is instantiated primarily for visual scene graphs and offers scalable reasoning over structured perceptual inputs using GNNs.

Despite its progress, Differentiable Logic Programming still faces several fundamental challenges. First, most DLP methods rely on predefined rule templates or symbolic skeletons, limiting their capacity for open-ended rule discovery and transfer across domains. Second, scaling DLP systems to large symbolic spaces or long reasoning chains remains difficult due to the computational cost of soft unification, large matrix operations, or recursive rule grounding. Third, while DLP enhances interpretability compared to purely neural models, it still lacks fine-grained symbolic transparency, especially when embeddings encode ambiguous or entangled semantics. Additionally, generalization from limited supervision—particularly when symbols are sparse or grounded in perceptual data—remains an open question. Future work may explore hybrid optimization (e.g., integrating discrete search with gradient-based learning), task-specific symbolic priors, or unifying DLP with neural module composition to improve scalability, transparency, and robustness.

3.2. Abductive Learning

Abductive learning is designed to perform symbolic explanation induction by jointly optimizing perceptual grounding and logic-level hypothesis search. The core reasoning foundation is based on *abduction*, originally proposed by Charles Sanders Peirce [141] as a form of logical inference distinct from deduction and induction. In artificial intelligence, abduction was formalized by Reiter [114] and Poole [142] as the task of finding a hypothesis H such that $B \cup H \models O$, where B is the background knowledge and O is an observation. Unlike DLP, which assumes that both the knowledge base and the reasoning engine are differentiable, abductive learning frameworks are designed for settings in which (i) some symbolic representations are latent or ungrounded, and (ii) logic-based consistency must be satisfied through explicit search. The overarching goal is to jointly identify missing symbolic facts and induce logical structures that best explain the observed data.

A defining characteristic of abductive learning is its decoupled architecture, consisting of (i) a neural perception module that maps raw inputs to candidate symbolic atoms and (ii) a symbolic reasoning module that searches for hypotheses consistent with background logic. Optimization is typically achieved through alternating or nested search loops, where symbolic consistency constraints guide the learning of neural components, and neural predictions constrain the symbolic search space. This design enables the system to perform symbolic structure search under weak supervision and to support partial observability and symbolic ambiguity.

Let O denote a set of observations, B the background knowledge (e.g., a logic program), and H a hypothesis (e.g., a set of abduced facts or a latent logical program). Abductive learning seeks to find H such that the entailment condition $B \cup H \models O$ holds. In neural–symbolic contexts, O often arises from neural perception models (e.g., image classifiers or object detectors), and H is constructed over a latent symbolic space. Formally, given a neural module f_ϕ parameterized by ϕ and a symbolic reasoning engine $\mathcal{E}$, the abductive learning objective is

$$\min_{\phi, H} \mathcal{L}_{\text{task}}\big(\mathcal{E}(B \cup H), O\big) + \lambda \cdot \mathcal{L}_{\text{symbolic}}(H), \tag{10}$$

where $\mathcal{L}_{\text{task}}$ measures the discrepancy between the inferred outcomes and the observations, and $\mathcal{L}_{\text{symbolic}}$ encodes logical priors (e.g., minimality, consistency, interpretability). H may include missing facts, learned rule templates, or logic programs constructed via symbolic search. The challenge is that H is not differentiable with respect to ϕ, so gradient updates are typically restricted to the neural module, with symbolic components optimized through discrete search.

Example 2. *Suppose a vision system observes a scene with two objects A and B, and through visual classifiers it detects*

$$O = \{\mathit{onTop}(A, B),\ \mathit{color}(A, \mathit{red}),\ \mathit{color}(B, \mathit{blue})\}, \tag{11}$$

but it fails to identify whether the two objects are aligned. The symbolic background knowledge B includes a rule:

$$\mathit{stacked}(X, Y) \leftarrow \mathit{onTop}(X, Y),\ \mathit{aligned}(X, Y). \tag{12}$$

To determine whether A and B are `stacked`*, the system performs abductive reasoning. It observes that the visual classifier cannot confidently output aligned(A, B). Thus, it proposes a hypothesis,*

$$H = \{\mathit{aligned}(A, B)\}, \tag{13}$$

which allows the symbolic engine to entail when combined with the observed facts and background rule:

$$B \cup O \cup H \models \mathit{stacked}(A, B). \tag{14}$$

Now, the system assigns a symbolic goal $y = \mathit{stacked}(A, B)$ as supervision, and optimizes both the classifier f_ϕ (for predicting aligned) and the logic module to satisfy symbolic consistency. Here, *abduction infers missing intermediate predicates (like aligned) necessary to complete a symbolic explanation, bridging the gap between perception and abstract logic reasoning. Unlike differentiable logic systems, the hypothesis H is constructed via discrete symbolic search, and the consistency of $\mathcal{E}(B \cup O \cup H)$ with y is used as a training signal.*

In recent years, several advances have been made in abductive learning. Neural Abductive Learning [70] is a seminal framework that combines perception-driven predicate grounding with logic-based abductive search. It assumes a latent symbol space and employs iterative refinement to jointly update perceptual classifiers and symbolic hypotheses. ABL-KG [84] extends this idea by introducing differentiable neural operators into abductive search, improving efficiency and allowing for partial gradient propagation. MetaAbd [143] explores meta-abductive learning, using higher-order logic templates and abductive meta-interpretation to support generalized rule learning across tasks. ABIL (Abductive Imitation Learning) [144] applies the abductive paradigm to policy learning by constructing logical explanations of expert demonstrations and enabling long-horizon planning through conflict resolution between perception and reasoning.

Several recent works have further advanced the capabilities of abductive reasoning. ABL-Sym [145] introduces abductive–symbolic interaction for mathematical reasoning tasks, while ABL-Refl [146] proposes an abductive reflection mechanism to detect and correct inconsistencies during neuro-symbolic inference. Rel-SAR [147] leverages diverse relation representations in vector-symbolic architectures for systematic abductive reasoning, achieving strong performance in visual abstraction benchmarks. ARLC [148] proposes an abductive rule learner with context-awareness, attaining state-of-the-art accuracy on abstract visual reasoning datasets like I-RAVEN [149]. Jin et al. [150] developed a meta-rule pretraining strategy to improve visual abductive learning by reducing the hypothesis search space. Additionally, Yang et al. [151] present a formal analysis of neural–symbolic reasoning shortcuts, identifying failure modes in generalization and proposing mitigations through hybrid inference.

Despite its flexibility, abductive learning faces several challenges: (i) the search space for candidate explanations grows combinatorially, necessitating efficient symbolic solvers and heuristics; (ii) the lack of full differentiability complicates integration with gradient-based learning frameworks; and (iii) the quality of learned models is sensitive to symbol grounding errors and misalignment between perception and logic. Addressing these issues may involve hybrid solvers, symbolic structure priors, and tighter neural–symbolic co-training. Nevertheless, abductive learning offers a compelling approach for tasks requiring interpretable structure induction, commonsense reasoning, and neuro-symbolic bootstrapping in low-data regimes.

3.3. Program Induction and Neural–Symbolic Concept Learners

Program induction has emerged as a compelling framework for neural–symbolic reasoning, particularly in tasks that require compositional generalization, interpretable decision-making, and grounded execution of abstract logic. The central goal is to synthesize executable programs—often in a domain-specific language (DSL)—that explain perceptual observations or solve complex reasoning tasks. In contrast to purely symbolic program synthesis, neural–symbolic program induction integrates perception modules with symbolic abstraction, enabling systems to interpret inputs such as images, language, or trajectories and induce symbolic programs that are executable, generalizable, and semantically coherent. Among these methods, Neural–Symbolic Concept Learners (NSCLs) form an important subclass that focuses on concept-centric reasoning in structured environments. These models extract discrete concepts (e.g., object attributes) from perceptual inputs and reason over them using symbolic programs. NSCLs are typically used in tasks like visual question answering and require grounding symbolic queries (derived from language) in structured visual representations. While general program induction frameworks target open-ended DSL construction or program synthesis from examples, NSCLs emphasize compositional alignment between language, perception, and symbolic reasoning over grounded concept spaces.

Program induction methods often rely on modular neural architectures that map raw inputs to symbolic tokens or operators, which are then composed into structured programs. These include (i) program sketches or templates filled by neural predictors, (ii) differentiable neural modules aligned with DSL operations, (iii) symbolic execution engines with neural-guided search, and (iv) curriculum learning strategies to handle sparse supervision and long-horizon reasoning. The symbolic layer not only serves as an interpretable scaffold but also enforces inductive biases and compositional structure. In the case of NSCLs, the symbolic vocabulary is tied to visual semantics (e.g., object shape, color, spatial relation), and the system learns to ground symbolic operators in perceptual features and execute logic-based queries.

Given an input x (e.g., an image, a question, or an example trajectory), the goal of program induction is to predict a symbolic program P that, when executed over an environment E, produces a result r matching the ground truth answer y. Here, E denotes the symbolic environment or world model—such as a visual scene, a knowledge base, or a stateful simulator—on which the program P is executed. Formally, the objective is

$$\min_{\theta} \mathcal{L}(\mathcal{E}(P_{\theta}(x), E), y), \tag{15}$$

where $P_{\theta}(x)$ denotes a program predicted by a neural model (e.g., sequence-to-sequence or Transformer-based), $\mathcal{E}$ is a symbolic program executor, and $\mathcal{L}$ is a task loss (e.g., cross-entropy or execution error). In some variants, the executor may provide intermediate supervision or execution traces to guide the learning process.

Example 3. *Consider a visual question answering task where the system is given an image depicting several geometric objects and a natural language question: "How many red cubes are there?" The input x consists of the image I and the question q. The symbolic environment E is a structured scene graph extracted from I, containing object attributes such as shape, color, and position. The program induction model P_{θ} predicts a symbolic program based on q, such as the following:*

$$P = \texttt{count(filter(shape=cube, filter(color=red, scene)))} \tag{16}$$

This program is executed by a symbolic executor $\mathcal{E}$ over environment E. The output of $\mathcal{E}(P, E)$ is an integer r indicating the number of red cubes in the scene. The training objective is to optimize θ to minimize the difference between r and the ground truth answer y:

$$\min_{\theta} \mathcal{L}(\mathcal{E}(P_{\theta}(x), E), y),$$

where $\mathcal{L}$ is typically a classification or regression loss. This framework enforces compositional structure: the model must learn to decompose the linguistic query into functional operators (e.g., `filter`, `count`) and to bind symbolic tokens (e.g., `red`, `cube`) to their perceptual counterparts in E. Unlike DLP or abduction, here the symbolic program is explicitly constructed, executed, and supervised end-to-end.

Program induction methods vary in how they define the program search space (explicit vs. learned), the executor (neural vs. symbolic), and the level of supervision (ground truth programs vs. weak or end-task supervision). These methods can be categorized based on their integration strategy:

(1) Neural–Symbolic Execution over Perception and Language. These systems map perceptual inputs and natural language to executable symbolic programs. For instance, NS-CL [22] parses natural language questions into logic programs and executes them over scene graphs extracted from visual inputs. NS-VQA [113] and CLEVR-CoGenT [72] further extend the NSCL paradigm with modular and disentangled reasoning architectures. VISPROG [152] leverages large language models to generate compositional visual programs without task-specific training. ViperGPT [83] similarly combines frozen vision-language models and code generation to synthesize programs for open-ended image understanding tasks. Recent work such as NeSyCoCo [153] presents a compositional concept learner that mitigates symbolic vocabulary bottlenecks in neural–symbolic reasoning. By introducing generalizable predicate functions and leveraging syntactic program structures, it improves generalization to out-of-distribution visual-language tasks. Other methods such as Code-as-Policies [154] extend this paradigm to embodied control. Here, natural language instructions or observations are translated into executable Python policies, enabling LLMs to act as symbolic planners for robotic agents. This framework bridges perception, language, and structured program generation for real-world decision-making tasks.

(2) Program Synthesis from Examples. These methods learn to generate symbolic programs from structured input-output pairs. Neuro-Symbolic Program Synthesis (NSPS) [155] combines symbolic search over DSLs with neural scoring functions. DreamCoder [156] incrementally induces both the DSL and the programs via wake–sleep-style learning and Bayesian structure search. More recently, LLM-Guided Compositional Program Synthesis [157] proposes a hybrid framework combining top-down propagation and bottom-up program sketch enumeration, using LLMs to guide program construction under weak supervision.

(3) Program Execution as Reasoning Architecture. NMNs [71] instantiate symbolic programs as dynamically composed neural modules that mimic function composition. PiNet [158] builds differentiable program graphs to support visual reasoning. DeepLogic [159] unifies neural encoders with logic program generation, providing theoretical guarantees for symbolic inference over structured environments. In parallel, Scallop [139] presents a differentiable declarative programming language that integrates logic rules, recursion, and neural prediction into a unified end-to-end reasoning framework. It compiles symbolic logic into differentiable provenance semirings, enabling efficient neural–symbolic execution over graphs and structured data.

Despite significant progress, neural–symbolic program induction faces several open challenges. First, the search space of programs grows combinatorially, making symbolic synthesis intractable without strong inductive biases or neural priors. Second, grounding symbolic tokens in perceptual inputs (e.g., objects or actions) remains fragile in complex environments. Third, robustness to ambiguity, multi-turn interactions, and spurious programs requires improved reasoning consistency and symbolic alignment. Future directions include incorporating large language models for program scaffolding, learning symbolic grammars from data, and developing hybrid architectures that combine neural generativity with symbolic verifiability.

3.4. LLM-Based Reasoning

While recent LLM-guided program induction approaches (e.g., Code-as-Policies [154], VISPROG [152]) leverage large language models to synthesize executable programs, another paradigm focuses on using LLMs as direct reasoners. In this section, we review LLM-based reasoning methods that perform inference purely through language modeling, without external program execution. LLMs such as GPT-4 [73] and PaLM [160] have demonstrated remarkable capabilities in solving complex reasoning tasks across diverse domains including mathematics, programming, natural language inference, and planning. While LLMs are trained as autoregressive sequence predictors, recent advances show that their internal representations can support multi-step symbolic and procedural reasoning when guided by appropriate prompting strategies and tool interaction mechanisms. This has led to a new paradigm we refer to as *LLM-based reasoning*, in which LLMs serve as high-level cognitive agents orchestrating reasoning processes through chain-of-thought inference, tool invocation, and code generation.

Several key mechanisms have emerged to enable structured reasoning within or guided by LLMs:

- **Chain-of-Thought (CoT) Prompting:** Introduced to enable intermediate reasoning steps in LLMs [92], CoT encourages models to produce step-by-step symbolic or numeric inferences, improving accuracy on arithmetic, commonsense, and logic tasks.
- **Program-Aided Reasoning (PAL):** LLMs generate intermediate code (e.g., in Python) that is executed externally to perform symbolic or procedural computation [127]. This extends program induction to tool-based execution.
- **Tool-Augmented Planning:** ReAct [75] combines CoT and tool use by interleaving thoughts and actions, while Toolformer [76] learns to insert API calls into generation. Models like Gorilla [161] extend this idea to structured tool APIs.
- **Symbolic Planner Interfaces:** Frameworks like DSPy [77] and AutoGPT [162] treat LLMs as symbolic planners that generate code or instructions, optionally grounded in formal logic or task graphs.

Given a natural language query x, the goal of LLM-based reasoning is to produce a reasoning trajectory $\tau = [s_1, s_2, \ldots, s_n]$ that leads to a correct answer y. Each step s_i may involve internal reasoning (e.g., a chain-of-thought sentence), external computation (e.g., calculator or Python call), or environment interaction (e.g., querying a knowledge base). The final answer is derived by executing or summarizing this trajectory. Formally, the model aims to learn or generate

$$\hat{y} = \mathcal{E}_{\text{LLM}}(x) = f(s_1, s_2, \ldots, s_n), \quad s_i \in \text{internal or tool-executed step}, \tag{17}$$

where $\mathcal{E}_{\text{LLM}}$ denotes the LLM-driven reasoning executor.

Example 4. *Consider a math problem: "A shop sells pencils at \$2 each and pens at \$5 each. If you buy 3 pencils and 2 pens, how much will you spend in total?" An LLM-based reasoning system processes this problem as a sequence of reasoning steps, each expressed either in natural language (internal CoT) or as tool-executable code:*

- s_1: *"Each pencil costs \$2, and you buy 3 pencils."*
- s_2: *"Total cost of pencils: \$2 × 3 = \$6."* *(CoT)*
- s_3: *"Each pen costs \$5, and you buy 2 pens."*
- s_4: *"Total cost of pens: \$5 × 2 = \$10."* *(CoT)*
- s_5: *"Total cost = \$6 + \$10 = \$16."* *(Final answer)*

Here, the reasoning trajectory $\tau = [s_1, s_2, \ldots, s_5]$ *is either produced directly by the LLM via chain-of-thought prompting, or composed into a code snippet executed by an external Python interpreter. The final answer* $\hat{y} = 16$ *is obtained via* $\mathcal{E}_{LLM}(x)$*, where* $\mathcal{E}_{LLM}$ *interleaves generation and tool-based evaluation. Unlike symbolic logic systems or Inductive Logic Programming, this framework emphasizes the use of pretrained language models as generative agents of stepwise inference, with tools acting as semantic anchors or verifiers.*

We organize representative methods into the following categories:

(1) CoT-based Inference with Natural Language Only. These methods use natural language as the sole medium for reasoning. CoT prompting [92] improves performance by guiding LLMs to generate intermediate reasoning steps in arithmetic, symbolic, and commonsense tasks. Self-Consistency [163] enhances reliability by sampling multiple CoT trajectories and aggregating their outcomes. Least-to-Most Prompting [164] decomposes complex queries into subproblems solved sequentially. Tree of Thoughts [165] models reasoning as a tree search with LLMs generating and evaluating multiple branches.

(2) Tool-Augmented Reasoning. These approaches embed tool use within the reasoning loop. ReAct [75] integrates reasoning and action, allowing LLMs to alternately think (via CoT) and act (via tool calls), with feedback-based refinement. Toolformer [76] self-supervises API call insertion into prompts, improving factual and symbolic accuracy. Gorilla [161] formulates API calls as code generation tasks, enabling precise tool binding through instruction tuning. AutoGPT [162] and AgentBench [166] extend tool use to long-horizon planning, modeling agents that autonomously decompose and solve tasks via sequences of tool-enhanced decisions.

(3) Program Execution Frameworks. These methods guide LLMs to produce executable programs for reasoning. PAL [127] converts questions into Python programs that are externally executed to derive answers, achieving high accuracy on math and logic tasks. Minerva [167] fine-tunes PaLM on scientific reasoning with CoT and symbolic execution. PyCoT [168] reformulates CoT reasoning as code snippets, leveraging program traces for supervision.

(4) Modular and Declarative Controller Systems. These systems treat LLMs as symbolic planners or controllers. DSPy [77] compiles declarative task specifications into executable plans that combine tools and subprograms. LangChain [169] and MetaGPT [170] provide composable agent APIs for orchestrating reasoning and tool interactions. CAMEL [171] models collaborative multi-agent systems where each LLM-agent follows role-specific instructions while coordinating over symbolic goals.

While LLMs show emergent reasoning capabilities, several key challenges remain in structured and symbolic guidance:

- **Tool orchestration:** Determining when and how to invoke external tools remains non-trivial, especially under uncertainty;
- **Reasoning faithfulness:** LLM-generated reasoning paths may be logically inconsistent or hallucinated, even if answers are correct;
- **Symbol grounding:** Aligning abstract reasoning steps with executable semantics requires fine-grained interface design;
- **Reproducibility and verifiability:** Unlike formal methods, LLM reasoning traces can vary stochastically, complicating reliability.

Unlike formal symbolic systems, LLM-based reasoning trades off rigor and verifiability for flexibility and task generality—posing new challenges in alignment, abstraction control, and trace consistency. Future work could explore modular controller architectures, reasoning-aligned instruction tuning, and hybrid neuro-symbolic loops, leveraging symbolic consistency constraints to guide generation and decision-making, paving the way toward robust and verifiable reasoning systems.

3.5. Logic-Aware Transformers

Logic-aware Transformers integrate explicit logical structures or symbolic constraints directly into Transformer-based neural architectures, aiming to improve reasoning capabilities, interpretability, and robustness. These methods bridge the gap between symbolic logic and Transformer models, enabling structured reasoning tasks that benefit from logical consistency and compositional generalization.

Formally, a Logic-aware Transformer can be modeled as a tuple:

$$\mathcal{M}LT = (\mathcal{X}, \mathcal{T}_\theta, \mathcal{C}, \mathcal{L}), \tag{18}$$

where

- $\mathcal{X}$ are input data, typically structured data, text, or symbolic representations;
- $\mathcal{T}_\theta$ is a Transformer model parameterized by θ, augmented with logic-aware mechanisms such as constrained attention or logical embedding;
- $\mathcal{C}$ are symbolic logic constraints or logical structure templates integrated within the Transformer, guiding attention mechanisms or loss functions;
- $\mathcal{L}$ is a loss function combining task-specific predictive losses and logic constraint losses.

The logic-aware reasoning process is defined as follows:

$$\hat{y} = \mathcal{T}_\theta(\mathcal{X}, \mathcal{C}) \approx \text{LogicalInfer}(\mathcal{X}, \mathcal{C}), \tag{19}$$

where $\text{LogicalInfer}(\mathcal{X}, \mathcal{C})$ denotes the logical inference outcomes based on inputs $\mathcal{X}$ and logic constraints $\mathcal{C}$. The training objective can be expressed as

$$\min_\theta \mathcal{L}_{\text{task}}(\hat{y}, y) + \lambda \cdot \mathcal{L}_{\text{logic}}(\mathcal{C}), \tag{20}$$

where

- $\mathcal{L}_{\text{task}}$ is task-specific loss (e.g., cross-entropy);
- $\mathcal{L}_{\text{logic}}$ is a logic-based regularization term enforcing adherence to symbolic constraints;
- λ is a hyperparameter controlling constraint regularization strength.

Example 5. *Consider a semantic parsing task, where the model is asked to translate natural language queries into structured logical forms. We define each component of the Logic-aware Transformer $\mathcal{M}LT = (\mathcal{X}, \mathcal{T}_\theta, \mathcal{C}, \mathcal{L})$ as follows:*

- ***Input Data ($\mathcal{X}$):*** *A natural language question, e.g., "Which movies directed by Christopher Nolan have a rating above 8?";*
- ***Logical Constraints ($\mathcal{C}$):*** *Domain-specific logical form constraints, e.g., a logical template ensuring valid SQL-like outputs:* `SELECT Movie WHERE Director = X AND Rating > Y`*;*
- ***Logic-aware Transformer ($\mathcal{T}_\theta$):*** *Transformer-based encoder–decoder architecture with attention mechanisms guided by logical form templates and constraints, ensuring output validity;*
- ***Supervision Target:*** *Ground-truth logical form* `SELECT Movie WHERE Director = "Christopher Nolan" AND Rating > 8`*;*
- ***Training Objective:*** *$\mathcal{L} = \mathcal{L}_{task}(\hat{y}, y) + \lambda \cdot \mathcal{L}_{logic}(\mathcal{C})$, balancing predictive accuracy and logical consistency.*

In this example, the Transformer generates outputs conditioned explicitly on logical templates to enforce structural validity. Logical constraints $\mathcal{C}$ guide the Transformer's decoder attention, preventing logically invalid token sequences. During training, the task loss (cross-entropy against ground-truth logical forms) and logic-based regularization jointly ensure outputs adhere strictly to specified logical structures.

Logic-aware Transformers have recently emerged as a powerful method for enhancing Transformer-based architectures by explicitly embedding symbolic logic constraints. Representative methods can be classified into the following categories:

(1) Logical Constraint Integration in Transformers. LogicBench [79] introduces logic constraints into the Transformer training process via constraint-based loss functions, ensuring output logical consistency in semantic parsing tasks. Transformer-based Type Inference has been explored by Miranda et al. [172], who propose applying Transformer architectures to perform type inference in the simply typed lambda calculus. Instead of relying on post hoc symbolic validation, their model integrates typed lambda calculus constraints directly into the Transformer decoding process, offering structural guidance for well-formed expression generation.

(2) Constrained Attention Mechanisms. Models such as NeuroLogic Decoding [137] leverage explicit symbolic constraints to modulate Transformer attention distributions, ensuring token selection respects logical coherence and domain-specific semantics.

(3) Compositional and Modular Transformers. Architectures like Modular Transformer (ModuTrans) [173] and Compositional Transformers [174] decompose complex logical tasks into modular subtasks, each handled by Transformer submodules under symbolic constraints, enhancing compositional generalization and interpretability.

(4) Formal Logic Reasoning with Transformers. These studies explore the capabilities of Transformer-based models in formal logical reasoning tasks, emphasizing their ability to handle complex logical structures. For example, DELTA$_D$ [175] develops a comprehensive dataset based on the description logic ALCQ, comprising 384K examples that vary in reasoning depth and linguistic complexity. Evaluations reveal that fine-tuned DeBERTa models and GPT variants can effectively perform entailment checking within this framework. Mechanistic Analysis of Transformers [176] conducts an in-depth analysis of how Transformers solve symbolic multi-step reasoning tasks. By constructing synthetic logic tasks, the study uncovers specific attention patterns and reasoning circuits within Transformer models, shedding light on their internal logical reasoning processes.

Despite their strengths, Logic-aware Transformers face several fundamental challenges. Firstly, efficiently embedding complex logical constraints without sacrificing Transformer scalability remains difficult. Secondly, maintaining computational efficiency while integrating symbolic reasoning into high-dimensional attention mechanisms is non-trivial, especially for large-scale tasks. Thirdly, ensuring generalization and robustness of logic-aware mechanisms across diverse and noisy real-world datasets requires further exploration. Lastly, while logic-awareness enhances interpretability compared to standard Transformers, achieving fine-grained symbolic transparency is still challenging.

Future research could explore more scalable integration methods, hybrid architectures combining differentiable logic modules with Transformer layers, and adaptive logic embedding strategies tailored to specific domains or reasoning tasks. Advances in these areas promise improved scalability, transparency, and robustness for Transformer-based symbolic reasoning.

3.6. Knowledge-Augmented Reasoning

Knowledge-augmented reasoning integrates structured external knowledge bases into neural networks, enabling models to explicitly reason with symbolic facts and relations [20,81,177,178]. This approach enhances neural architectures with structured domain knowledge, improving reasoning accuracy, interpretability, and generalization, particularly for tasks requiring explicit knowledge grounding.

Formally, knowledge-augmented reasoning models can be described as

$$\mathcal{M}KR = (\mathcal{D}, \mathcal{K}, \mathcal{N}\phi, \mathcal{I}, \mathcal{L}), \tag{21}$$

where

- $\mathcal{D}$ are input data (e.g., natural language queries, perceptual inputs);
- $\mathcal{K}$ is a structured external knowledge base (e.g., knowledge graphs, logical rules);
- $\mathcal{N}_\phi$ is a neural encoder–decoder model parameterized by ϕ;
- $\mathcal{I}$ is a knowledge integration mechanism, e.g., embedding lookups, GNNs, or differentiable reasoning layers;
- $\mathcal{L}$ is the loss function combining prediction accuracy and knowledge grounding losses.

The reasoning output is given by the following:

$$\hat{y} = \mathcal{N}_\phi(\mathcal{D}, \mathcal{I}(\mathcal{K})) \approx \text{Infer}(\mathcal{D}, \mathcal{K}), \tag{22}$$

The training objective typically combines task-specific loss and a knowledge regularization loss:

$$\min_\phi \mathcal{L}_{\text{task}}(\hat{y}, y) + \lambda \cdot \mathcal{L}_{\text{knowledge}}(\mathcal{K}). \tag{23}$$

Example 6. *Consider a knowledge-enhanced question-answering task. We define the knowledge-augmented reasoning model $\mathcal{MKR} = (\mathcal{D}, \mathcal{K}, \mathcal{N}\phi, \mathcal{I}, \mathcal{L})$ as follows:*

- ***Input Data ($\mathcal{D}$):*** *A natural language query, e.g., "Who is the author of 'The Lord of the Rings'?";*
- ***Knowledge Base ($\mathcal{K}$):*** *a structured knowledge graph with entities (authors, books) and relationships (`authoredBy`);*
- ***Neural Model ($\mathcal{N}\phi$):*** *Transformer-based encoder–decoder that encodes the question and decodes the answer;*
- ***Integration Mechanism ($\mathcal{I}$):*** *GNN that retrieves relevant entities and relations from the knowledge graph, producing embeddings integrated into the Transformer's hidden states;*
- ***Supervision Target:*** *ground-truth answer `J.R.R. Tolkien`;*
- ***Training Objective:*** *$\mathcal{L} = \mathcal{L}_{task}(\hat{y}, y) + \lambda \cdot \mathcal{L}_{knowledge}(\mathcal{K})$, optimizing accuracy and knowledge grounding consistency.*

The integration mechanism identifies relevant nodes and edges in the knowledge graph, producing embeddings that augment Transformer inputs. Training jointly optimizes answer accuracy and coherence with the knowledge structure, enhancing both interpretability and performance.

Knowledge-augmented reasoning models can be categorized based on their knowledge integration approaches:

(1) Graph-based Knowledge Integration. Methods such as GNNs [65,179] explicitly leverage knowledge graphs to encode structured relations into neural representations. Models like K-BERT and KEPLER integrate graph-structured knowledge into Transformer-based language models, significantly improving performance on commonsense reasoning and entity linking tasks [20,177].

(2) Probabilistic Logic-based Integration. Approaches like DeepProbLog [80] and DeepStochLog [178] integrate probabilistic logic programs with neural modules, enabling reasoning under uncertainty. These models embed probabilistic logical inference within neural architectures, facilitating flexible and robust reasoning capabilities.

(3) Answer Set Programming (ASP)-based Integration. Systems such as NeurASP [81] explicitly encode symbolic constraints as logic programs solved by ASP solvers, integrating results into neural decision processes. This enables structured and explainable reasoning, particularly beneficial in domains with explicit symbolic constraints.

(4) Graph-constrained and Multimodal Knowledge Integration. Recent research has explored the integration of structured knowledge through constrained decoding or multimodal grounding, enabling large language models to reason faithfully over external symbolic contexts. Graph-Constrained Reasoning (GCR) [180] introduces a decoding framework that constrains the output space of LLMs to paths grounded in a knowledge graph. By building a trie-based decoding index over entities and relations, GCR ensures that generated reasoning trajectories strictly adhere to the symbolic structure of the underlying knowledge graph, improving faithfulness and factual consistency in reasoning tasks. KAM-CoT [181] presents a framework that augments chain-of-thought prompting with structured knowledge and multimodal information. KAM-CoT integrates visual concepts and symbolic relations into the CoT reasoning path, enabling more accurate and interpretable inference across vision-language tasks, particularly under low-resource supervision.

Knowledge-augmented reasoning faces several ongoing challenges. First, scalability and computational efficiency remain critical, especially when reasoning with large-scale knowledge graphs or complex logic constraints. Second, effectively grounding neural representations in structured symbolic knowledge without losing expressiveness or flexibility is difficult. Third, handling uncertainty and noisy knowledge remains a significant challenge, requiring robust probabilistic reasoning methods. Lastly, ensuring generalization across diverse tasks and datasets demands adaptive knowledge integration techniques.

Future research directions may explore dynamic knowledge retrieval mechanisms, hybrid reasoning architectures combining symbolic inference and neural computation, and methods for robust integration of uncertain or incomplete knowledge. These advances promise to further enhance reasoning capabilities, interpretability, and generalization in knowledge-augmented neural systems.

3.7. Multimodal Neuro-Symbolic Reasoning

Multimodal neuro-symbolic reasoning integrates symbolic reasoning and neural computation across multiple data modalities such as vision, language, and speech. This approach aims to enable coherent and structured reasoning over multimodal inputs, addressing complex tasks like visual question answering (VQA), multimodal dialogue systems, video understanding, and robotic control. By explicitly combining symbolic reasoning processes with neural multimodal perception, these methods significantly enhance interpretability, compositional generalization, and robustness.

Formally, multimodal neuro-symbolic reasoning models can be represented as

$$\mathcal{M}_{MNS} = (\mathcal{X}_{mod}, \mathcal{S}, \mathcal{N}_{\psi}, \mathcal{R}, \mathcal{L}), \tag{24}$$

where

- $\mathcal{X}_{mod}$ are input data across multiple modalities (e.g., images, text, audio, video);
- $\mathcal{S}$ is symbolic representation derived from multimodal inputs, such as scene graphs, logical forms, or semantic frames;
- $\mathcal{N}_{\psi}$ is a neural multimodal encoder–decoder parameterized by ψ;
- $\mathcal{R}$ is a symbolic reasoning module, often implemented through differentiable symbolic reasoning engines or logic-based inference mechanisms;
- $\mathcal{L}$ is the loss function balancing prediction accuracy, modality alignment, and symbolic consistency.

The multimodal reasoning output is

$$\hat{y} = \mathcal{R}(\mathcal{N}_{\psi}(\mathcal{X}_{mod}), \mathcal{S}) \approx \text{SymbolicInfer}(\mathcal{X}_{mod}, \mathcal{S}), \tag{25}$$

with the training objective expressed as

$$\min_{\psi} \mathcal{L}_{\text{task}}(\hat{y}, y) + \lambda \cdot \mathcal{L}_{\text{symbolic}}(\mathcal{S}). \tag{26}$$

Example 7. *Consider a multimodal task of video-based question answering. The multimodal neuro-symbolic reasoning model $\mathcal{M}_{MNS} = (\mathcal{X}_{mod}, \mathcal{S}, \mathcal{N}_{\psi}, \mathcal{R}, \mathcal{L})$ is instantiated as follows:*

- ***Multimodal Inputs ($\mathcal{X}_{mod}$):** A video clip and a natural language question, e.g., "Why did the person enter the room?";*
- ***Symbolic Representation ($\mathcal{S}$):** A structured event graph representing entities, actions, and temporal–causal relationships extracted from the video;*
- ***Neural Multimodal Model ($\mathcal{N}_{\psi}$):** A Transformer-based multimodal encoder that encodes video frames and linguistic queries into unified multimodal embeddings;*
- ***Symbolic Reasoning Module ($\mathcal{R}$):** A differentiable reasoning engine that performs logical inference on the event graph to infer the causal reason;*
- ***Supervision Target:** Natural language answer, e.g., "`to attend a meeting`";*
- ***Training Objective:** $\mathcal{L} = \mathcal{L}_{task}(\hat{y}, y) + \lambda \cdot \mathcal{L}_{symbolic}(\mathcal{S})$, optimizing answer correctness and symbolic reasoning coherence.*

Here, the symbolic reasoning module processes the structured event graph extracted by neural perception modules, enabling explicit reasoning over causal and temporal relations, resulting in interpretable and accurate answers.

Multimodal neuro-symbolic reasoning approaches can be broadly classified into several categories:

(1) Multimodal Symbolic Parsing and Reasoning. These approaches explicitly parse multimodal inputs into symbolic structures such as scene graphs or event graphs, followed by logical inference. For example, VideoCoT [82] performs chain-of-thought reasoning over event-centric video graphs. NeuSyRE [182] builds enriched scene graphs by integrating neuro-symbolic modules with visual grounding, enabling symbolic concept enrichment and reasoning over image regions, and demonstrating strong performance on visual understanding tasks.

(2) Differentiable Symbolic Reasoning over Multimodal Representations. Models such as ViperGPT [83] integrate frozen vision-language models with symbolic toolchains to generate Python-like reasoning code from natural language queries and image features. This framework bridges perceptual understanding and symbolic computation for open-ended multimodal tasks.

(3) Neuro-symbolic Modular Architectures. These systems dynamically assemble reasoning modules guided by symbolic queries. For instance, NMNs [71] use learned parser trees to assign submodules to query components. Similarly, NS-CL [22] executes compositional programs over grounded visual scenes, enabling robust compositional generalization.

Despite significant progress, multimodal neuro-symbolic reasoning faces critical challenges. First, effectively aligning and grounding symbolic representations across different modalities is non-trivial due to modality-specific semantics and noise. Second, computational complexity grows rapidly with symbolic reasoning over rich multimodal data, demanding efficient reasoning mechanisms. Third, ensuring robustness against ambiguous or incomplete multimodal inputs requires advanced uncertainty modeling and reasoning techniques. Lastly, achieving comprehensive interpretability remains challenging, particularly when neural modules produce ambiguous intermediate symbolic representations.

Future research directions include developing robust multimodal alignment and grounding methods, efficient differentiable symbolic reasoning engines capable of handling complex multimodal data, and scalable neuro-symbolic architectures to improve reasoning interpretability, accuracy, and generalization across diverse multimodal tasks.

3.8. Conclusion and Emerging Trends

This section systematically explores various paradigms of AI reasoning in the deep learning era, highlighting the significant progress and existing challenges across Differentiable Logic Programming, abductive learning, program induction and Neural–Symbolic Concept Learning, LLM-based reasoning, Logic-aware Transformers, knowledge-augmented reasoning, and multimodal neuro-symbolic reasoning. Collectively, these paradigms illustrate the transition from traditional symbolic AI toward increasingly sophisticated neural–symbolic integrations, combining the interpretability and structured reasoning capabilities of symbolic methods with the flexibility and scalability of neural networks.

Additionally, several emerging and influential subfields warrant brief mention due to their potential to significantly shape future research directions:

Neuro-symbolic Reinforcement Learning (RL) merges symbolic reasoning with reinforcement learning frameworks to enhance decision-making, interpretability, and sample efficiency in sequential tasks. For instance, Programmatically Interpretable Reinforcement Learning (PIRL) [183] learns neural policies that are distilled into symbolic programs, enabling interpretable and verifiable control. More recent works such as LOA [184] employ logic-based action pruning in text-based environments, while NUDGE [185] leverages symbolic abstractions guided by neural modules to learn explainable policies. These frameworks illustrate how symbolic priors and logic rules can guide exploration, enforce constraints, and enhance generalization in decision-making systems.

Causal Neuro-symbolic Reasoning combines neural networks with causal inference frameworks to enable structured, interpretable, and robust reasoning over cause-effect relationships. By integrating symbolic causal models into deep architectures, these approaches move beyond correlational learning to support counterfactual reasoning, intervention-based predictions, and causal discovery. For instance, CausalVAE [186] incorporates structural causal models into variational autoencoders, enabling disentangled representation learning and causal intervention. CM-CRL [187] enhances contrastive representation learning by injecting symbolic causal graphs, improving vision-language navigation under distribution shifts. More recently, Causal Component Analysis [188] introduces a neuro-symbolic model that identifies latent generative factors causally aligned with observed outcomes, supporting causal reasoning in high-dimensional settings. These works highlight the potential of neuro-symbolic approaches to advance causal understanding in AI systems, especially in environments with ambiguity, uncertainty, or intervention needs.

In summary, the integration of symbolic reasoning within deep learning continues to evolve rapidly, driven by the demand for enhanced interpretability, robustness, and structured reasoning. Future advancements in these emerging areas promise further breakthroughs, bridging the gap between human-like cognitive reasoning capabilities and artificial intelligence systems.

4. Application View: Benchmarks, Datasets, and Reasoning-Oriented AI Systems

4.1. Reasoning-Centric Tasks and Benchmarks

Reasoning-oriented AI systems have demonstrated significant impact across a range of domains, from natural language understanding and commonsense inference to planning in embodied agents and multimodal cognition. In this section, we categorize representative reasoning tasks and summarize the associated methodologies and real-world applications, with a focus on how symbolic or neural–symbolic approaches enhance interpretability, generalization, and causal understanding.

(1) Question Answering. QA is a long-standing benchmark for evaluating machine reasoning, spanning deductive, commonsense, abductive, causal, and explanatory inference. These tasks require systems to move beyond surface-level text matching and instead construct, apply, and verify reasoning chains from explicit premises or implicit world knowledge.

- **Deductive QA.** Tasks such as ProofWriter [189] and FOLIO [190] involve formal logic reasoning, requiring models to infer conclusions from natural language premises using entailment, conjunction, and implication rules. These benchmarks emphasize the need for systematic generalization over formal logical forms.
- **Commonsense QA.** Benchmarks like CommonsenseQA [191], CosmosQA [192], and OpenBookQA [193] test models' ability to integrate background knowledge with contextual understanding. They target reasoning over latent knowledge, including naïve physics, social conventions, and intentionality.
- **Abductive and Causal QA.** Datasets such as AbductiveNLI [194], ART [195], CausalQA [196], and SituatedQA [197] evaluate models on inferring plausible causes or explanations behind observed scenarios. Such tasks reflect the importance of abductive reasoning and counterfactual analysis in explainable AI.
- **Explanatory QA.** Tasks like e-SNLI [198] and EntailmentBank [199] require not only answer prediction but also structured generation of reasoning chains, either as entailment trees or natural language justifications. These tasks are crucial for interpretability and educational applications.

(2) Planning, Tool Use, and Decision-Making. Symbolic and logical reasoning form the foundation of traditional AI planning. In contemporary systems, these paradigms are integrated with neural policies and tool-augmented agents to support long-horizon reasoning and explainable decision-making.

- **Symbolic Planning.** Symbolic planning focuses on solving goal-directed tasks by generating action sequences under symbolic representations of states, transitions, and constraints. In contrast to purely reactive control policies, symbolic planning requires agents to explicitly reason about future states, preconditions, and causality. This often entails the construction of intermediate symbolic structures such as logic programs, action graphs, or scene representations, enabling long-horizon planning, interpretability, and task compositionality. PUZZLES [200] present structured environments composed of discrete logic-based tasks (e.g., grid games, puzzles, combinatorial path planning), designed to test whether agents can generalize across symbolic domains and solve algorithmic reasoning problems under limited feedback. RSBench [201] introduces a suite of neuro-symbolic reasoning environments targeting concept-level evaluation. SCERL [202] adapts textual reinforcement learning environments for safe and interpretable planning, covering sub-domains like side-effect minimization and reward uncertainty. In the domain of physical and robotic environments, RLBench [203] serves as a high-dimensional benchmark featuring over 100 task variants ranging from simple object manipulation to multi-step tool use.
- **Tool-Augmented Agents.** Systems such as ReAct [75], AutoGPT [162], and DSPy [77] combine LLMs with external tools and APIs to perform chain-of-thought reasoning and actionable planning. These agents dynamically invoke tools, reason over intermediate outputs, and adaptively revise plans.
- **Multi-Agent and Interactive Planning.** Frameworks like CAMEL [171] leverage symbolic role assignments and structured dialogue policies to coordinate among

collaborative agents. They enable decentralized planning, intention modeling, and joint task execution in social or multi-agent settings.

(3) Multimodal Reasoning and Perception. Perception-driven reasoning tasks integrate visual or sensorimotor input with symbolic abstraction, supporting compositional, causal, and temporally grounded inference.

- **VQA.** Datasets such as CLEVR [72], GQA [204], and VQA-X [205] are designed to probe structured reasoning over visual scenes, testing capabilities like relational comparison, quantification, and spatial inference under varying degrees of language grounding and visual complexity.
- **Video and Event Reasoning.** Benchmarks including CLEVRER [206], NExT-QA [207], and VideoCoT [82] evaluate temporal and causal reasoning in video contexts, such as predicting future states, identifying event chains, and explaining dynamic processes. Symbolic modules or causal priors are often critical in modeling these temporal dependencies.
- **Embodied and Situated Reasoning.** In robotics and embodied environments, agents must perform goal-oriented reasoning from partial observations. Systems increasingly utilize symbolic representations (e.g., scene graphs or task logic) derived from perceptual inputs to support grounding, action abstraction, and generalizable planning. Recent approaches such as Embodied Chain-of-Thought Reasoning (ECoT) [208] and Inner Monologue [209] demonstrate how integrating structured reasoning steps and leveraging language model feedback can enhance robotic control and planning capabilities in complex environments.

(4) Program Induction and Semantic Parsing. Programmatic representations—ranging from SQL queries to domain-specific languages—serve as explicit reasoning artifacts, allowing verification, interpretability, and execution within structured environments.

- **Semantic Parsing.** Benchmarks such as Spider [210], ATIS [211], and ScienceBenchmark [212] evaluate the mapping of natural language to executable queries. These tasks often require understanding compositional semantics, coreference resolution, and domain schemas.
- **Program Synthesis.** Tasks like CODET [213], NL2Bash [214], and MathQA [215] involve generating symbolic code from language descriptions or problems. Success in these tasks depends on precise logical translation, error handling, and explanation capabilities, particularly in mathematical or command-line environments.

Beyond academic evaluation, these reasoning tasks are increasingly deployed in high-stakes domains such as healthcare, education, and human–computer interaction. Causal and abductive QA enables hypothesis testing in scientific discovery; tool-augmented reasoning powers assistive agents; multimodal reasoning enables situational understanding in autonomous systems; and programmatic reasoning supports transparent decision pipelines. Together, they form the backbone of AI systems aspiring toward generality, interpretability, and real-world reliability.

We summarize the benchmarks and datasets by domain and reasoning type in Table 5. These benchmarks differ across three critical dimensions: (i) the nature of the input modality (symbolic, visual, or hybrid), (ii) the type of reasoning targeted (deductive, abductive, causal, or procedural), and (iii) the structural complexity of the task formulation (single-step vs. multi-hop, static vs. dynamic environments). As the field matures, recent datasets increasingly emphasize alignment with real-world conditions, such as noisy perception (e.g., VideoCoT [82]), open-domain tool use (e.g., WebArena [216]), and multi-agent coordination (e.g., AgentBench [166]).

Table 5. Representative benchmarks and datasets for reasoning-centric AI tasks.

Domain	Dataset / Benchmark	Focus / Highlights
Question Answering	ProofWriter [189], FOLIO [190]	Deductive reasoning with formal logic entailment
	CSQA [191], CosmosQA [192], OBQA [193]	Commonsense reasoning with background knowledge
	AbductiveNLI [194], ART [195]	Hypothesis selection based on plausible explanation
	WhyQA [217], CausalQA [196]	Cause–effect inference and causal trace evaluation
Symbolic Reasoning	ToolBench [218], WebArena [216], AgentBench [166]	LLM-based reasoning with API tools and task orchestration
	HotPotQA [104], WebGPT [219]	Multi-hop tool-guided question answering
Multimodal Reasoning and Perception	CLEVR [72], GQA [204], VQA-X [205]	Visual relational and compositional reasoning
	CLEVRER [206], NExT-QA [207]	Temporal reasoning and event-based causal inference
	NLVR2 [220], Winoground [221]	Visual-linguistic grounding and referential ambiguity
Program Induction and Semantic Parsing	Spider [210], ATIS [211], NL2SQL [222]	Mapping questions to executable SQL/logical forms
	NL2Bash [214], MathQA [215], CODET [213]	Program synthesis and symbolic reasoning from examples
	CoSQL [223]	Conversational semantic parsing with symbolic schema linking

The diversity of benchmarks in symbolic and neural–symbolic reasoning reflects the richness of reasoning paradigms and their application scopes. The field has evolved from synthetic logic tasks (e.g., CLEVR [72], ProofWriter [189]) toward complex, real-world, multi-agent or tool-augmented scenarios (e.g., ToolBench [218], AgentBench [166]), presenting new opportunities for scalable and interpretable reasoning in AI systems.

4.2. Reasoning Frameworks and Toolkits

Alongside benchmarks and datasets, a growing ecosystem of toolkits, frameworks, and reasoning engines has been developed to support symbolic and Neural–Symbolic AI. These systems differ by their reasoning paradigms, supported modalities, and integration with learning-based models. Table 6 summarizes representative systems.

These frameworks offer varying levels of abstraction—from declarative logic interfaces (e.g., ASP [81]) to Python-integrated neural–symbolic environments (e.g., DeepProbLog [80]). Some (e.g., DSPy [77], LangChain [169]) enable LLM-centric tool chaining, while others emphasize differentiable semantics, probabilistic reasoning, or symbolic learning. The selection of toolkits depends on the desired balance between interpretability, learning capacity, and domain constraints.

Table 6. Representative toolkits and frameworks for symbolic and neural–symbolic reasoning.

Toolkit / Library	Paradigm	Modality	Key Capabilities	Example Use Cases
ProbLog [50], ProbLog2 [224]	Probabilistic Logic Programming	Symbolic	Probabilistic inference over logic programs	Knowledge base reasoning, uncertain facts
DeepProbLog [80], NeurASP [81]	Neural–Symbolic Integration	Symbolic + Perception	Combine logic rules with neural outputs	VQA, handwritten digit inference
LTN [69], LNN [136]	Differentiable Logic	Symbolic	Real-valued logic via fuzzy operators	Logic-constraint learning, semantic consistency

Table 6. *Cont.*

Toolkit / Library	Paradigm	Modality	Key Capabilities	Example Use Cases
AlphaILP [140], NEUMANN [133]	Neural ILP	Symbolic + Visual	Learning rules over graph/scene structure	Visual scene reasoning, object relation induction
DSPy [77], LangChain [169], AgentBench [166]	Tool-Augmented LLMs	Language + API Tools	Modular agent control with tools + LLMs	Autonomous planning, reasoning via tool calls
Alchemy [66], PSL [89]	Statistical Relational Learning	Symbolic	Soft logic + scalable inference	Entity resolution, joint inference tasks
ASP Tools (clingo [225], DLV [226])	Answer Set Programming	Symbolic	Non-monotonic rule-based reasoning	Diagnosis, planning, combinatorial search
MiniKanren [227], CLOG [228]	Functional Logic Programming	Symbolic	Symbolic search with functional constructs	Program synthesis, theorem proving

While reasoning-oriented AI systems have demonstrated strong theoretical advances, practical deployment in real-world applications remains a critical benchmark of their utility. We highlight representative deployment domains where symbolic or neural–symbolic reasoning frameworks have proven effective:

- **Question Answering and Explainable Search.** Neuro-symbolic systems such as DeepProbLog [80] and DSPy [77] have been integrated into QA pipelines to provide interpretable reasoning traces alongside factual answers. Logic-enhanced retrieval and reasoning has also shown promise in scientific QA and legal document analysis.
- **Automated Planning and Robotics.** ASP solvers like clingo [229] and symbolic planners such as DLV [226] are widely used in robotic task planning, where action constraints, resource dependencies, and failure recovery can be naturally expressed using logical rules.
- **Scene and Event Understanding.** Hybrid models like NEUMANN [133] and AlphaILP [140] have enabled compositional visual reasoning in scene graphs and multi-object tracking tasks. Their integration with visual detectors improves the accuracy and interpretability of symbolic queries over visual domains.
- **Tool-Augmented Agent Systems.** Toolchain frameworks such as LangChain [169], DSPy [77], and AgentBench [166] allow LLMs to invoke APIs, retrieve documents, and invoke solvers in complex reasoning workflows. These systems have been deployed in domains like software engineering, autonomous planning, and complex report generation.
- **Decision Support and Diagnosis.** Probabilistic logic systems like ProbLog [50] and PSL [89] have been applied in healthcare, recommender systems, and risk assessment settings, where uncertainty and rule-based policies must be jointly modeled.

These deployments highlight the increasing maturity and integration capability of modern AI reasoning systems. As reasoning modules become more modular and explainable, their adoption in domain-critical applications—such as law, finance, and scientific discovery—is likely to expand.

5. Closing Remarks and Future Directions

As this survey has illustrated, reasoning remains a cornerstone capability in the pursuit of Artificial General Intelligence. From symbolic logic systems rooted in the

Physical Symbol System Hypothesis to neural architectures excelling at perceptual tasks, the historical development of AI has reflected a persistent tension between structure and flexibility, interpretability and adaptability.

Neural–Symbolic AI emerges as a promising paradigm to bridge this divide, aiming to unify the structured inference of symbolic reasoning with the representation power and scalability of deep learning. Across this survey, we have categorized seven methodological directions—ranging from Differentiable Logic Programming and abductive learning to LLM-guided reasoning and multimodal neuro-symbolic integration—and analyzed how each contributes to reconciling classical reasoning principles with modern AI capabilities.

On the application side, we have witnessed a growing adoption of reasoning systems across diverse domains: question answering, robot planning, visual scene understanding, and agentic tool use. Benchmarks and toolkits have also matured, offering common grounds for evaluating inference consistency, generalization ability, and modular reasoning workflows.

Despite these advances, several open challenges remain at the heart of AI reasoning research:

- **Unified Architectures.** Existing systems are often task-specific or handcrafted. Achieving general-purpose, reusable reasoning modules remains an unsolved problem.
- **Symbol-Vector Bridging.** Seamlessly combining discrete symbolic structures with continuous neural representations requires more principled modeling and training strategies.
- **Reasoning under Uncertainty.** While probabilistic and fuzzy logic frameworks exist, efficient integration with deep perception remains limited in practice.
- **Explainability and Trust.** As reasoning systems are increasingly deployed in sensitive applications such as healthcare and law, their transparency, robustness, and ethical alignment become essential.

Looking forward, the distinction between symbolic and non-symbolic reasoning paradigms is likely to blur further. While symbolic logic provides interpretability, verifiability, and structured abstraction, emerging generative paradigms—particularly those enabled by large language models (LLMs)—demonstrate remarkable reasoning-like behavior through prompt-conditioned generation. Whether reasoning necessarily requires symbolic logic, or whether statistical generation alone can yield robust and generalizable reasoning, remains an open question. Rather than taking an exclusive stance, we advocate for a pluralistic view: symbolic, sub-symbolic, and generative paradigms offer complementary strengths, and their integration may be the most promising path toward general-purpose reasoning systems. We anticipate that the next generation of AI reasoning will involve hybrid designs that combine large language models, symbolic planners, differentiable theorem provers, and structured memory components. Progress in this direction will not only push the boundaries of interpretable and generalizable AI, but also lay the groundwork for cognitively inspired, human-aligned intelligent systems. Ultimately, advancing AI reasoning is not merely a technical pursuit—it is a conceptual imperative. It calls for rethinking the foundations of how machines perceive, abstract, and decide, and for constructing systems that reason not only with data, but also with knowledge, causality, and intent.

Author Contributions: Conceptualization, B.L.; Methodology, B.L.; Data curation, Y.W.; Writing—original draft, B.L.; Writing—review & editing, C.T. All authors have read and agreed to the published version of the manuscript. .

Funding: This study was partially supported by the National Natural Science Foundation of China (62176016, 72274127), National Key R&D Program of China (No. 2021YFB2104800), Guizhou Province

Science and Technology Project: Research on Q&A Interactive Virtual Digital People for Intelligent Medical Treatment in Information Innovation Environment (supported by Qiankehe[2024] General 058), Capital Health Development Research Project(2022-2-2013), Haidian innovation and translation program from Peking University Third Hospital (HDCXZHKC2023203).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Turing, A.M. Computing machinery and intelligence. *Mind* **1950**, *59*, 433–460. [CrossRef]
2. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 4th ed.; Pearson: London, UK, 2021.
3. Goertzel, B.; Pennachin, C. *Artificial General Intelligence*; Springer: Berlin/Heidelberg, Germany, 2007.
4. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
5. Marcus, G. Deep learning: A critical appraisal. *arXiv* **2018**, arXiv:1801.00631.
6. Lake, B.M.; Ullman, T.D.; Tenenbaum, J.B.; Gershman, S.J. Building Machines That Learn and Think Like People. *Behav. Brain Sci.* **2017**, *40*, e253. [CrossRef] [PubMed]
7. Pearl, J. *Causality: Models, Reasoning, and Inference*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2009.
8. Wang, X.; Zhang, S.; Cen, J.; Gao, C.; Zhang, Y.; Zhao, D.; Sang, N. CLIP-guided Prototype Modulating for Few-shot Action Recognition. *Int. J. Comput. Vis.* **2023**, *132*, 1899–1912. [CrossRef]
9. Brachman, R.J.; Levesque, H.J. *Knowledge Representation and Reasoning*; The Morgan Kaufmann Series in Artificial Intelligence; Morgan Kaufmann: San Francisco, CA, USA, 2004.
10. Newell, A.; Simon, H.A. Computer Science as Empirical Inquiry: Symbols and Search. *Commun. ACM* **1976**, *19*, 113–126. [CrossRef]
11. Shortliffe, E.H.; Buchanan, B.G. A model of inexact reasoning in medicine. *Math. Biosci.* **1975**, *23*, 351–379. [CrossRef]
12. Campbell, M.; Hoane, A.J.; Hsu, F.h. Deep Blue. *Artif. Intell.* **2002**, *134*, 57–83. [CrossRef]
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 4–9 December 2017; NIPS'17; p. 6000–6010.
15. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature* **2016**, *529*, 484–489. [CrossRef]
16. Garcez, A.d.; Lamb, L.C. Neurosymbolic AI: The 3rd Wave. *Artif. Intell. Rev.* **2022**, *55*, 2245–2274. [CrossRef]
17. Bengio, Y. From System 1 Deep Learning to System 2 Deep Learning. Available online: https://neurips.cc/virtual/2019/invited-talk/15488 (accessed on 11 December 2019).
18. Zhang, B.; Zhu, J.; Hang, S. Toward the third generation of artificial intelligence. *Sci. Sin. (Informationis)* **2020**, *9*, 1281–1302. [CrossRef]
19. Evans, R.; Grefenstette, E. Learning explanatory rules from noisy data. *J. Artif. Intell. Res.* **2018**, *61*, 1–64. [CrossRef]
20. Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; Wang, P. K-BERT: Enabling Language Representation with Knowledge Graph. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 2901–2908.
21. Bordes, A.; Usunier, N.; Garcia-Durán, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. In Proceedings of the 27th International Conference on Neural Information Processing Systems–Volume 2, Red Hook, NY, USA, 5–10 December 2013; NIPS'13; pp. 2787–2795.
22. Mao, J.; Gan, C.; Kohli, P.; Tenenbaum, J.B.; Wu, J. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv* **2019**, arXiv:1904.12584.
23. Besold, T.R.; d'Avila Garcez, A.; Bader, S.; Bowman, H.; Domingos, P.; Hitzler, P.; Kuehnberger, K.U.; Lamb, L.C.; Lowd, D.; Lima, P.M.V.; et al. Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. In *Neuro-Symbolic Artificial Intelligence: The State of the Art*; IOS Press: Amsterdam, The Netherlands, 2017.
24. Wan, Z.; Liu, C.K.; Yang, H.; Li, C.; You, H.; Fu, Y.; Wan, C.; Krishna, T.; Lin, Y.; Raychowdhury, A. Towards Cognitive AI Systems: A Survey and Prospective on Neuro-Symbolic AI. *arXiv* **2024**, arXiv:2401.01040.
25. Li, Z.Z.; Zhang, D.; Zhang, M.L.; Zhang, J.; Liu, Z.; Yao, Y.; Xu, H.; Zheng, J.; Wang, P.J.; Chen, X.; et al. From System 1 to System 2: A Survey of Reasoning Large Language Models. *arXiv* **2025**, arXiv:2502.17419.

26. Bock, H.H.; Diday, E. *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*; Springer: Berlin/Heidelberg, Germany, 2000.
27. Colmerauer, A.; Roussel, P. The birth of Prolog. In *History of Programming Languages—II*; Association for Computing Machinery: New York, NY, USA, 1996; pp. 331–367.
28. Newell, A.; Simon, H.A. GPS, a program that simulates human thought. In *Computation & Intelligence: Collected Readings*; American Association for Artificial Intelligence: Washington, DC, USA, 1995; pp. 415–428.
29. Michalski, R.S. Variable-valued logic and its applications to pattern recognition and machine learning. In *Computer Science and Multiple-Valued Logic*; Rine, D.C., Ed.; Elsevier: Amsterdam, The Netherlands, 1977; pp. 506–534. [CrossRef]
30. Buchanan, B.G.; Feigenbaum, E.A. Dendral and meta-dendral: Their applications dimension. *Artif. Intell.* **1978**, *11*, 5–24. [CrossRef]
31. Giarratano, J.C.; Riley, G. *Expert Systems: Principles and Programming*; Brooks/Cole Publishing Co.: Pacific Grove, CA, USA, 1989.
32. Forgy, C.L. Rete: A fast algorithm for the many pattern/many object pattern match problem. In *Readings in Artificial Intelligence and Databases*; Elsevier: Amsterdam, The Netherlands, 1989; pp. 547–559.
33. Reiter, R. A logic for default reasoning. *Artif. Intell.* **1980**, *13*, 81–132. [CrossRef]
34. McCarthy, J. Circumscription—A form of non-monotonic reasoning. *Artif. Intell.* **1980**, *13*, 27–39. [CrossRef]
35. Gelfond, M.; Lifschitz, V. Classical negation in logic programs and disjunctive databases. *New Gener. Comput.* **1991**, *9*, 365–385. [CrossRef]
36. Doyle, J. A truth maintenance system. *Artif. Intell.* **1979**, *12*, 231–272. [CrossRef]
37. Fikes, R.E.; Nilsson, N.J. STRIPS: A new approach to the application of theorem proving to problem solving. *Artif. Intell.* **1971**, *2*, 189–208. [CrossRef]
38. Nilsson, N.J. Shakey the Robot. In *SRI International AI Center Technical Note*; SRI International: Menlo Park, CA, USA, 1984.
39. Dung, P.M. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.* **1995**, *77*, 321–357. [CrossRef]
40. Modgil, S.; Prakken, H. The ASPIC+ framework for structured argumentation: A tutorial. *Argum. Comput.* **2014**, *5*, 31–62. [CrossRef]
41. Brachman, R.J.; Schmolze, J.G. An overview of the KL-ONE knowledge representation system. *Cogn. Sci.* **1985**, *9*, 171–216. [CrossRef]
42. Baader, F. *The Description Logic Handbook: Theory, Implementation and Applications*; Cambridge University Press: Cambridge, UK, 2003.
43. McGuinness, D.L.; Van Harmelen, F. OWL web ontology language overview. *W3C Recomm.* **2004**, *10*, 2004.
44. Hintikka, J. *Knowledge and Belief: An Introduction to the Logic of the Two Notions*; Texts in Philosophy; King's College Publications: London, UK, 2005.
45. Pnueli, A. The temporal logic of programs. In Proceedings of the 18th Annual Symposium on Foundations of a Computer Science (sfcs 1977), Providence, RI, USA, 31 October–2 November 1977; pp. 46–57.
46. Emerson, E.A. Temporal and modal logic. In *Formal Models and Semantics*; Elsevier: Amsterdam, The Netherlands, 1990; pp. 995–1072.
47. Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1988.
48. Koller, D.; Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*; MIT Press: Cambridge, MA, USA, 2009.
49. Richardson, M.; Domingos, P. Markov logic networks. *Mach. Learn.* **2006**, *62*, 107–136. [CrossRef]
50. De Raedt, L.; Kimmig, A.; Toivonen, H. ProbLog: A probabilistic Prolog and its application in link discovery. In Proceedings of the IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, 6–12 January 2007; pp. 2462–2467.
51. Sato, T.; Kameya, Y. PRISM: A language for symbolic-statistical modeling. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, Nagoya, Japan, 23–29 August 1997; Volume 97, pp. 1330–1339.
52. Riguzzi, F. Extended semantics and inference for the Independent Choice Logic. *Log. J. IGPL* **2009**, *17*, 589–629. [CrossRef]
53. Jaeger, M. Relational Bayesian networks. In Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI), Providence, RI, USA, 1–3 August 1997; pp. 266–273.
54. Getoor, L.; Taskar, B. *Introduction to Statistical Relational Learning*; MIT Press: Cambridge, MA, USA, 2007.
55. De Raedt, L.; Frasconi, P.; Kersting, K.; Muggleton, S. (Eds.) *Probabilistic Inductive Logic Programming: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2008.
56. Michalski, R.S. Knowledge Acquisition Through Conceptual Clustering: A Theoretical Framework and an Algorithm for Partitioning Data into Conjunctive Concepts. *J. Policy Anal. Inf. Syst.* **1980**, *4*, 219–244.
57. Michalski, R.S.; Stepp, R.E. Automated construction of classifications: Conceptual clustering versus numerical taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.* **1983**, *PAMI-5*, 396–410. [CrossRef]

58. Fisher, D.H. Knowledge Acquisition via Incremental Conceptual Clustering. *Mach. Learn.* **1987**, *2*, 139–172. [CrossRef]
59. Pearl, J.; Mackenzie, D. *The Book of Why: The New Science of Cause and Effect*; Basic Books: New York, NY, USA, 2018.
60. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
61. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
62. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
63. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
64. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
65. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *arXiv* **2018**, arXiv:1812.08434. [CrossRef]
66. Kok, S.; Sumner, M.; Richardson, M.; Singla, P.; Poon, H.; Lowd, D.; Domingos, P. *The Alchemy System for Statistical Relational AI*; Technical Report; Department of Computer Science and Engineering, University of Washington: Seattle, WA, USA, 2007. Available online: http://alchemy.cs.washington.edu (accessed on 20 April 2025).
67. Niu, F.; Zhang, C.; Ré, C.; Shavlik, J. Tuffy: Scaling up statistical inference in Markov Logic Networks using an RDBMS. *Proc. Vldb Endow.* **2011**, *4*, 373–384. [CrossRef]
68. Dong, H.; Mao, J.; Lin, C.G. Neural Logic Machines. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
69. Serafini, L.; d'Avila Garcez, A. Learning and Reasoning with Logic Tensor Networks. In Proceedings of the Conference of the Italian Association for Artificial Intelligence, Genova, Italy, 29 November 2016.
70. Zhou, Z.H. Abductive learning: Towards bridging machine learning and logical reasoning. *Sci. China Inf. Sci.* **2019**, *62*, 76101. [CrossRef]
71. Andreas, J.; Rohrbach, M.; Darrell, T.; Klein, D. Neural Module Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 39–48.
72. Johnson, J.; Hariharan, B.; van der Maaten, L.; Hoffman, J.; Fei-Fei, L.; Zitnick, C.L.; Girshick, R. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2901–2910.
73. OpenAI. GPT-4 Technical Report. 2023. Available online: https://openai.com/research/gpt-4 (accessed on 24 April 2025).
74. Anthropic. The Claude 3 Model Family: Opus, Sonnet, Haiku. 2024. Available online: https://www.anthropic.com/news/claude-3-family (accessed on 17 April 2025).
75. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. React: Synergizing reasoning and acting in language models. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
76. Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Hambro, E.; Zettlemoyer, L.; Cancedda, N.; Scialom, T. Toolformer: Language models can teach themselves to use tools. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 68539–68551.
77. Khattab, O.; Singhvi, A.; Maheshwari, P.; Zhang, Z.; Santhanam, K.; Vardhamanan, S.; Haq, S.; Sharma, A.; Joshi, T.T.; Moazam, H.; et al. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv* **2023**, arXiv:2310.03714.
78. Wang, B.R.; Huang, Q.Y.; Deb, B.; Halfaker, A.; Shao, L.Q.; McDuff, D.; Awadallah, A.H.; Radev, D.; Gao, J.F. Logical Transformers: Infusing Logical Structures into Pre-Trained Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, Toronto, ON, Canada, 9–14 July 2023.
79. Parmar, M.; Patel, N.; Varshney, N.; Nakamura, M.; Luo, M.; Mashetty, S.; Mitra, A.; Baral, C. LogicBench: Towards Systematic Evaluation of Logical Reasoning Ability of Large Language Models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand, 11–16 August 2024; pp. 13679–13707. [CrossRef]
80. Manhaeve, R.; Dumancic, S.; Kimmig, A.; Demeester, T.; De Raedt, L. DeepProbLog: Neural Probabilistic Logic Programming. In *Advances in Neural Information Processing Systems*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31.
81. Yang, Z.; Ishay, A.; Lee, J. NeurASP: Embracing Neural Networks into Answer Set Programming. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, Yokohama, Japan, 7–15 January 2020; Bessiere, C., Ed.; pp. 1755–1762. [CrossRef]
82. Wang, Y.; Zeng, Y.; Zheng, J.; Xing, X.; Xu, J.; Xu, X. VideoCoT: A Video Chain-of-Thought Dataset with Active Annotation Tool. In Proceedings of the 3rd Workshop on Advances in Language and Vision Research (ALVR), Bangkok, Thailand, 16 August 2024; Gu, J., Fu, T.J.R., Hudson, D., Celikyilmaz, A., Wang, W., Eds.; pp. 92–101. [CrossRef]
83. Surís, D.; Menon, S.; Vondrick, C. ViperGPT: Visual Inference via Python Execution for Reasoning. *arXiv* **2023**, arXiv:2303.08128.
84. Huang, Y.X.; Sun, Z.Q.; Li, G.Y.; Tian, X.B.; Dai, W.Z.; Hu, W.; Jiang, Y.; Zhou, Z.H. Enabling Abductive Learning to Exploit Knowledge Graph. In Proceedings of the 32th International Joint Conference on Artificial Intelligence (IJCAI), Macao SAR, China, 19–25 August 2023; pp. 2730–2736.
85. Saparov, A.; He, H. Language modeling via stochastic processes and data augmentation. *arXiv* **2022**, arXiv:2205.09310.

86. Marcus, G.; Davis, E. *Rebooting AI: Building Artificial Intelligence We Can Trust*; Pantheon: London, UK, 2019.
87. Russell, S.J.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 3rd ed.; Prentice Hall: Hoboken, NJ, USA, 2010.
88. Robinson, J.A. A machine-oriented logic based on the resolution principle. *J. ACM (JACM)* **1965**, *12*, 23–41. [CrossRef]
89. Bach, S.H.; Broecheler, M.; Huang, B.; Getoor, L. Hinge-loss markov random fields and probabilistic soft logic. *J. Mach. Learn. Res.* **2017**, *18*, 1–67.
90. Battaglia, P.; Hamrick, J.B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. Relational inductive biases, deep learning, and graph networks. *arXiv* **2018**, arXiv:1806.01261.
91. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv* **2023**, arXiv:2302.13971.
92. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D.; Ichter, B. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837.
93. Ganguli, D.; Hernandez, D.; Lovitt, L.; DasSarma, N.; Henighan, T.; Jones, A.; Joseph, N.; Kernion, J.; Mann, B.; Askell, A.; et al. Predictability and surprise in large generative models. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, 21–24 June 2022; pp. 1747–1764.
94. Abiteboul, S.; Hull, R.; Vianu, V. *Foundations of Databases*; Addison-Wesley: Boston, MA, USA, 1995.
95. Baader, F.; Horrocks, I.; Sattler, U. *The Description Logic Handbook*; Cambridge University Press: Cambridge, UK, 2010.
96. Rabiner, L.R. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*; IEEE: New York, NY, USA, 1989; Volume 77, pp. 257–286.
97. De Moura, L.; Bjørner, N. Z3: An Efficient SMT Solver. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 337–340.
98. Muggleton, S. Inverse entailment and Progol. *New Gener. Comput.* **1995**, *13*, 245–286. [CrossRef]
99. Cropper, A.; Muggleton, S. Learning efficient logic programs. In Proceedings of the ILP Conference, London, UK, 4–6 September 2016.
100. Falkenhainer, B.; Forbus, K.; Gentner, D. The structure-mapping engine: Algorithm and examples. *Artif. Intell.* **1989**, *41*, 1–63. [CrossRef]
101. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the 38th International Conference on Machine Learning (ICML), Virtual, 18–24 July 2021; pp. 8748–8763.
102. Kaelbling, L.P.; Littman, M.L.; Cassandra, A.R. Planning and acting in partially observable stochastic domains. *Artif. Intell.* **1998**, *101*, 99–134. [CrossRef]
103. Palmirani, M.; Governatori, G.; Rotolo, A.; Sartor, G.; Tabet, S.; Boley, H. LegalRuleML: XML-based rules and norms. In Proceedings of the Rule-Based Modeling and Computing on the Semantic Web (RuleML), America, Ft. Lauderdale, FL, USA, 3–5 November 2011; pp. 298–312.
104. Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.W.; Salakhutdinov, R.; Manning, C.D. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv* **2018**, arXiv:1809.09600.
105. Reddy, S.; Chen, D.; Manning, C.D. CoQA: A conversational question answering challenge. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 249–266. [CrossRef]
106. Taylor, R.; Kardas, M.; Cucurull, G.; Scialom, T.; Hartshorn, A.; Saravia, E.; Poulton, A.; Kerkez, V.; Stojnic, R. Galactica: A large language model for science. *arXiv* **2022**, arXiv:2211.09085.
107. Goodwin, T.; Demner-Fushman, D. Enhancing Question Answering by Injecting Ontological Knowledge through Regularization. In Proceedings of the Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, Online, 19–20 November 2020; pp. 56–63. [CrossRef]
108. Valmeekam, K.; Ellis, K.; Solar-Lezama, A.; Tenenbaum, J.B. DreamCoder: Growing generalizable, interpretable knowledge with wake-sleep Bayesian program learning. In Proceedings of the NeurIPS, New Orleans, LA, USA, 28 November–9 December 2022.
109. Morel, R.; Cropper, A.; Ong, C.-H.L. Typed meta-interpretive learning of logic programs. In Proceedings of the Logics in Artificial Intelligence (JELIA), Rende, Italy, 7–11 May 2019; pp. 198–213.
110. Saad, F.A.; Cusumano-Towner, M.F.; Schaechtle, U.; Rinard, M.C.; Mansinghka, V.K. Bayesian synthesis of probabilistic programs for automatic data modeling. *ACM Program. Lang.* **2019**, *3*, 1–32. [CrossRef]
111. Tai, K.S.; Socher, R.; Manning, C.D. Improved semantic representations from tree-structured long short-term memory networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL), Beijing, China, 26–31 July 2015; pp. 1556–1566.
112. Adrita, B.; Cara, W.; Pascal, H. Concept Induction Using LLMs: A User Experiment for Assessment. In Proceedings of the International Conference on Neural-Symbolic Learning and Reasoning (NeSy), Barcelona, Spain, 9–12 September 2024.

113. Yi, K.; Wu, J.; Gan, C.; Torralba, A.; Kohli, P.; Tenenbaum, J. Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding. In Proceedings of the NeurIPS, Montréal, QC, Canada, 3–8 December 2018.
114. Reiter, R. A theory of diagnosis from first principles. *Artif. Intell.* **1987**, *32*, 57–95. [CrossRef]
115. Ramírez, M.; Geffner, H. Plan Recognition as Planning. In Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI), Pasadena, CA, USA, 11–17 July 2009; pp. 1778–1783.
116. Glória-Silva, D.; Ferreira, R.; Tavares, D.; Semedo, D.; Magalhaes, J. Plan-Grounded Large Language Models for Dual Goal Conversational Settings. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Valletta, Malta, 21–22 March 2024; pp. 1271–1292.
117. Liang, B.; Su, Q.; Zhu, S.; Liang, Y.; Tong, C. VidEvent: A Large Dataset for Understanding Dynamic Evolution of Events in Videos. *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 5128–5136. [CrossRef]
118. Shutova, E.; Sun, L.; Korhonen, A. Metaphor identification using verb and noun clustering. In Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China, 23–27 August 2010; pp. 1002–1010.
119. Borgwardt, K.; Kriegel, H.P. Shortest-path kernels on graphs. In Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM), Houston, TX, USA, 27–30 November 2005; pp. 74–81.
120. Du, X.; Ji, H. Retrieval-augmented generative question answering for event argument extraction. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP), Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 4567–4578.
121. Webb, T.; Holyoak, K.J.; Lu, H. Emergent analogical reasoning in large language models. *Nat. Hum. Behav.* **2023**, *7*, 1526–1541. [CrossRef]
122. Bengio, Y. The consciousness prior. *arXiv* **2017**, arXiv:1709.08568.
123. LeCun, Y. A path towards autonomous machine intelligence. *arXiv* **2022**, arXiv:2206.06927.
124. Garcez, A.d.; Gori, M.; Lamb, L.C.; Serafini, L.; Spranger, M.; Tran, S.N. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *arXiv* **2019**, arXiv:1905.06088.
125. Yang, F.; Yang, Z.; Cohen, W.W. Differentiable learning of logical rules for knowledge base reasoning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 2319–2328.
126. Purgał, S.J.; Cerna, D.M.; Kaliszyk, C. Differentiable inductive logic programming in high-dimensional space. *arXiv* **2022**, arXiv:2208.06652.
127. Gao, K.; Inoue, K.; Cao, Y.; Wang, H. Learning first-order rules with differentiable logic program semantics. *arXiv* **2022**, arXiv:2204.13570.
128. Shindo, H.; Nishino, M.; Yamamoto, A. Differentiable inductive logic programming for structured examples. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 5034–5041. [CrossRef]
129. Gao, K.; Inoue, K.; Cao, Y.; Wang, H. A Differentiable First-Order Rule Learner for Inductive Logic Programming. *Artif. Intell.* **2024**, *331*, 104108. [CrossRef]
130. Rocktäschel, T.; Riedel, S. End-to-end differentiable proving. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3788–3800.
131. Cohen, W.W. Tensorlog: A differentiable deductive database. *arXiv* **2016**, arXiv:1605.06523.
132. Zimmer, M.; Feng, X.; Glanois, C.; Jiang, Z.; Zhang, J.; Weng, P.; Li, D.; Hao, J.; Liu, W. Differentiable logic machines. *arXiv* **2021**, arXiv:2102.11529.
133. Shindo, H.; Pfanschilling, V.; Dhami, D.S.; Kersting, K. Learning differentiable logic programs for abstract visual reasoning. *Mach. Learn.* **2024**, *113*, 8533–8584. [CrossRef]
134. Takemura, A.; Inoue, K. Differentiable Logic Programming for Distant Supervision. In Proceedings of the European Conference on Artificial Intelligence, Santiago de Compostela, Spain, 19–24 October 2024.
135. Serafini, L.; Garcez, A.d. Logic tensor networks: Deep learning and logical reasoning from data and knowledge. *arXiv* **2016**, arXiv:1606.04422.
136. Riegel, R.; Gray, A.; Luus, F.; Khan, N.; Makondo, N.; Akhalwaya, I.Y.; Qian, H.; Fagin, R.; Barahona, F.; Sharma, U.; et al. Logical Neural Networks. *arXiv* **2020**, arXiv:2006.13155.
137. Šourek, G.; Železný, F.; Kuželka, O. Beyond graph neural networks with lifted relational neural networks. *Mach. Learn.* **2021**, *110*, 1695–1738. [CrossRef]
138. Geh, R.L.; Gonçalves, J.; Silveira, I.C.; Mauá, D.D.; Cozman, F.G. dPASP: A probabilistic logic programming environment for neurosymbolic learning and reasoning. In Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, Hanoi, Vietnam, 2–8 November 2024; Volume 21, pp. 731–742.
139. Li, Z.; Huang, J.; Naik, M. Scallop: A Language for Neurosymbolic Programming. *Proc. ACM Program. Lang.* **2023**, *7*, 166:1–166:25. [CrossRef]
140. Shindo, H.; Pfanschilling, V.; Dhami, D.S.; Kersting, K. α ilp: Thinking visual scenes as differentiable logic programs. *Mach. Learn.* **2023**, *112*, 1465–1497. [CrossRef]
141. Peirce, C.S. *Collected Papers of Charles Sanders Peirce*; Harvard University Press: Cambridge, MA, USA, 1935; Volume 5.

142. Poole, D. A logical framework for default reasoning. *Artif. Intell.* **1988**, *36*, 27–47. [CrossRef]
143. Dai, W.Z.; Muggleton, S.H. Abductive Knowledge Induction from Raw Data. In Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI), Montreal, QC, Canada, 19–26 August 2021; pp. 2730–2736.
144. Shao, J.J.; Hao, H.R.; Yang, X.W.; Li, Y.F. Abductive Learning for Neuro-Symbolic Grounded Imitation. In Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1, Toronto, ON, Canada, 3–7 August 2025; KDD '25, pp. 1221–1232. [CrossRef]
145. Hu, Y.Y.; Yu, Y. Enhancing Neural Mathematical Reasoning by Abductive Combination with Symbolic Library. *arXiv* **2023**, arXiv:2203.14487.
146. Hu, W.C.; Dai, W.Z.; Jiang, Y.; Zhou, Z.H. Efficient rectification of neuro-symbolic reasoning inconsistencies by abductive reflection. *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 17333–17341. [CrossRef]
147. Sun, Z.H.; Zhang, R.Y.; Zhen, Z.; Wang, D.H.; Li, Y.J.; Wan, X.; You, H. Systematic Abductive Reasoning via Diverse Relation Representations in Vector-symbolic Architecture. *arXiv* **2025**, arXiv:2501.11896.
148. Camposampiero, G.; Hersche, M.; Terzić, A.; Wattenhofer, R.; Sebastian, A.; Rahimi, A. Towards Learning Abductive Reasoning Using VSA Distributed Representations. In *Neural-Symbolic Learning and Reasoning*; Besold, T.R., d'Avila Garcez, A., Jimenez-Ruiz, E., Confalonieri, R., Madhyastha, P., Wagner, B., Eds.; Springer Nature: Cham, Switzerland, 2024; pp. 370–385.
149. Hu, S.; Ma, Y.; Liu, X.; Wei, Y.; Bai, S. Stratified Rule-Aware Network for Abstract Visual Reasoning. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 1561–1569. [CrossRef]
150. Jin, Y.; Liu, J.; Luo, Z.; Peng, Y.; Qin, Z.; Dai, W.Z.; Ding, Y.X.; Zhou, K. Pre-Training Meta-Rule Selection Policy for Visual Generative Abductive Learning. *arXiv* **2025**, arXiv:2503.06427.
151. Yang, X.W.; Wei, W.D.; Shao, J.J.; Li, Y.F.; Zhou, Z.H. Analysis for Abductive Learning and Neural-Symbolic Reasoning Shortcuts. In Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024; Volume 235, pp. 56524–56541.
152. Gupta, T.; Kembhavi, A. Visual Programming: Compositional visual reasoning without training. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023.
153. Kamali, D.; Barezi, E.J.; Kordjamshidi, P. NeSyCoCo: A Neuro-Symbolic Concept Composer for Compositional Generalization. *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 4184–4193. [CrossRef]
154. Liang, J.; Huang, W.; Xia, F.; Xu, P.; Hausman, K.; Ichter, B.; Florence, P.; Zeng, A. Code as Policies: Language Model Programs for Embodied Control. In Proceedings of the 6th Conference on Robot Learning (CoRL), Auckland, New Zealand, 14–18 December 2022.
155. Shin, R.; Kant, N.; Gupta, K.; Bender, C.; Trabucco, B.; Singh, R.; Song, D. Synthetic Datasets for Neural Program Synthesis. In Proceedings of the 7th International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
156. Ellis, K.; Wong, L.; Nye, M.; Sable-Meyer, M.; Cary, L.; Anaya Pozo, L.; Hewitt, L.; Solar-Lezama, A.; Tenenbaum, J.B. DreamCoder: Growing generalizable, interpretable knowledge with wake–sleep Bayesian program learning. *Philos. Trans. R. Soc. A* **2023**, *381*, 20220050. [CrossRef] [PubMed]
157. Khan, R.M.; Gulwani, S.; Le, V.; Radhakrishna, A.; Tiwari, A.; Verbruggen, G. LLM-Guided Compositional Program Synthesis. *arXiv* **2025**, arXiv:2503.15540.
158. Liang, C.; Berant, J.; Le, Q.; Forbus, K.D.; Lao, N. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. *arXiv* **2016**, arXiv:1611.00020.
159. Duan, X.; Wang, X.; Zhao, P.; Shen, G.; Zhu, W. DeepLogic: Joint Learning of Neural Perception and Logical Reasoning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 4321–4334. [CrossRef]
160. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.* **2023**, *24*, 1–113.
161. Patil, S.G.; Zhang, T.; Wang, X.; Gonzalez, J.E Gorilla: Large Language Model Connected with Massive APIs. *arXiv* **2023**, arXiv:2305.15334.
162. Gravitas, S. AutoGPT: An Autonomous GPT-4 Experiment. GitHub Repository. 2023. Available online: https://github.com/Torantulino/Auto-GPT (accessed on 20 May 2025).
163. Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv* **2022**, arXiv:2203.11171.
164. Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.; et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv* **2022**, arXiv:2205.10625.
165. Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 11809–11822.
166. Liu, X.; Yu, H.; Zhang, H.; Xu, Y.; Lei, X.; Lai, H.; Gu, Y.; Ding, H.; Men, K.; Yang, K.; et al. Agentbench: Evaluating llms as agents. *arXiv* **2023**, arXiv:2308.03688.

167. Lewkowycz, A.; Andreassen, A.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V.; Slone, A.; Anil, C.; Schlag, I.; Gutman-Solo, T.; et al. Solving quantitative reasoning problems with language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 3843–3857.
168. AtlasUnified. PyCoT: A Pythonic Chain-of-Thought Dataset Series. Hugging Face. 2025. Comprehensive CoT Expansions Across Multiple Domains. Available online: https://huggingface.co/datasets/AtlasUnified/PyCoT-Collection_Main (accessed on 1 May 2025).
169. Chase, H. LangChain: Building Applications with LLMs Through Composability. 2023. Available online: https://www.langchain.com/ (accessed on 1 May 2025).
170. Hong, S.; Zheng, X.; Chen, J.; Cheng, Y.; Wang, J.; Zhang, C.; Wang, Z.; Yau, S.K.S.; Lin, Z.; Zhou, L.; et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv* **2023**, arXiv:2308.00352.
171. Li, G.; Hammoud, H.; Itani, H.; Khizbullin, D.; Ghanem, B. Camel: Communicative agents for "mind" exploration of large language model society. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 51991–52008.
172. Miranda, B.; Shinnar, A.; Pestun, V.; Trager, B. Transformer Models for Type Inference in the Simply Typed Lambda Calculus: A Case Study in Deep Learning for Code. *arXiv* **2023**, arXiv:2304.10500.
173. Zhou, W.; Le Bras, R.; Choi, Y. Modular Transformers: Compressing Transformers into Modularized Layers for Flexible Efficient Inference. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, Toronto, ON, Canada, 9–14 July 2023; Rogers, A., Boyd-Graber, J., Okazaki, N., Eds.; pp. 10452–10465. [CrossRef]
174. Herzig, J.; Berant, J. Span-based semantic parsing for compositional generalization. *arXiv* **2020**, arXiv:2009.06040.
175. Poulis, A.; Tsalapati, E.; Koubarakis, M. Transformers in the Service of Description Logic-Based Contexts. In *Knowledge Engineering and Knowledge Management*; Springer Nature: Cham, Switzerland, 2025; pp. 328–345.
176. Brinkmann, B.J.; Smith, J.D.; Lee, C.M. How Transformers Solve Propositional Logic Problems: A Mechanistic Analysis. In Proceedings of the Workshop on Mechanistic Interpretability of Neural Networks at ICLR, Vienna, Austria, 27 July 2024; Workshop paper.
177. Wang, X.; Gao, T.; Zhu, Z.; Zhang, Z.; Liu, Z.; Li, J.; Tang, J. KEPLER: A Unified Model for Knowledge Embedding and Pre-Trained Language Representation. *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 176–194. [CrossRef]
178. Winters, T.; Marra, G.; Manhaeve, R.; De Raedt, L. DeepStochLog: Neural Stochastic Logic Programming. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 10090–10100. [CrossRef]
179. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
180. Luo, L.; Zhao, Z.; Gong, C.; Haffari, G.; Pan, S. Graph-constrained Reasoning: Faithful Reasoning on Knowledge Graphs with Large Language Models. *arXiv* **2024**, arXiv:2410.13080.
181. Mondal, D.; Modi, S.; Panda, S.; Singh, R.; Rao, G.S. KAM-CoT: Knowledge Augmented Multimodal Chain-of-Thoughts Reasoning. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 18798–18806. [CrossRef]
182. Khan, M.J.; Breslin, J.G.; Curry, E. NeuSyRE: Neuro-Symbolic Visual Understanding and Reasoning Framework based on Scene Graph Enrichment. *Semant. Web* **2023**, *15*, 1389–1413. [CrossRef]
183. Verma, A.; Murali, V.; Singh, R.; Kohli, P.; Chaudhuri, S. Programmatically Interpretable Reinforcement Learning. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5045–5054.
184. Kimura, D.; Chaudhury, S.; Ono, M.; Tatsubori, M.; Agravante, D.J.; Munawar, A.; Wachi, A.; Kohita, R.; Gray, A. LOA: Logical Optimal Actions for Text-based Interaction Games. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, Online, 1–6 August 2021; pp. 227–231.
185. Delfosse, Q.; Shindo, H.; Dhami, D.S.; Kersting, K. Interpretable and Explainable Logical Policies via Neurally Guided Symbolic Abstraction. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 10–16 December 2023.
186. Yang, M.; Liu, F.; Chen, Z.; Shen, X.; Hao, J.; Wang, J. CausalVAE: Disentangled Representation Learning via Neural Structural Causal Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 9593–9602.
187. Wang, L.; He, Z.; Dang, R.; Shen, M.; Liu, C.; Chen, Q. Vision-and-Language Navigation via Causal Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024.
188. Liang, W.; Kekić, A.; von Kügelgen, J.; Buchholz, S.; Besserve, M.; Gresele, L.; Schölkopf, B. Causal Component Analysis. In Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023.
189. Tafjord, O.; Dalvi, B.; Clark, P. ProofWriter: Generating Implications, Proofs, and Abductive Statements over Natural Language. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 1–6 August 2021; Zong, C., Xia, F., Li, W., Navigli, R., Eds.; pp. 3621–3634. [CrossRef]

190. Han, S.; Schoelkopf, H.; Zhao, Y.; Qi, Z.; Riddell, M.; Zhou, W.; Coady, J.; Peng, D.; Qiao, Y.; Benson, L.; et al. FOLIO: Natural Language Reasoning with First-Order Logic. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024; Al-Onaizan, Y., Bansal, M., Chen, Y.N., Eds.; pp. 22017–22031. [CrossRef]
191. Talmor, A.; Herzig, J.; Lourie, N.; Berant, J. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; Burstein, J., Doran, C., Solorio, T., Eds.; pp. 4149–4158. [CrossRef]
192. Huang, L.; Le Bras, R.; Bhagavatula, C.; Choi, Y. Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; Inui, K., Jiang, J., Ng, V., Wan, X., Eds.; pp. 2391–2401. [CrossRef]
193. Mihaylov, T.; Clark, P.; Khot, T.; Sabharwal, A. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J., Eds.; pp. 2381–2391. [CrossRef]
194. Bhagavatula, C.; Bras, R.L.; Malaviya, C.; Sakaguchi, K.; Holtzman, A.; Rashkin, H.; Downey, D.; Yih, S.W.t.; Choi, Y. Abductive commonsense reasoning. *arXiv* **2019**, arXiv:1908.05739.
195. Du, L.; Ding, X.; Liu, T.; Qin, B. Learning Event Graph Knowledge for Abductive Reasoning. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August 2021; [CrossRef]
196. Bondarenko, A.; Wolska, M.; Heindorf, S.; Blübaum, L.; Ngonga Ngomo, A.C.; Stein, B.; Braslavski, P.; Hagen, M.; Potthast, M. CausalQA: A Benchmark for Causal Question Answering. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; Calzolari, N., Huang, C.R., Kim, H., Pustejovsky, J., Wanner, L., Choi, K.S., Ryu, P.M., Chen, H.H., Donatelli, L., Ji, H., et al., Eds.; pp. 3296–3308.
197. Zhang, M.; Choi, E. SituatedQA: Incorporating Extra-Linguistic Contexts into QA. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; Moens, M.F., Huang, X., Specia, L., Yih, S.W.t., Eds.; pp. 7371–7387. [CrossRef]
198. Camburu, O.M.; Rocktäschel, T.; Lukasiewicz, T.; Blunsom, P. e-SNLI: Natural Language Inference with Natural Language Explanations. In *Advances in Neural Information Processing Systems*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31.
199. Dalvi, B.; Jansen, P.; Tafjord, O.; Xie, Z.; Smith, H.; Pipatanangkura, L.; Clark, P. Explaining Answers with Entailment Trees. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; Moens, M.F., Huang, X., Specia, L., Yih, S.W.t., Eds.; pp. 7358–7370. [CrossRef]
200. Estermann, B.; Lanzendörfer, L.A.; Niedermayr, Y.; Wattenhofer, R. PUZZLES: A Benchmark for Neural Algorithmic Reasoning. In *Advances in Neural Information Processing Systems*; Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2024; Volume 37, pp. 127059–127098.
201. Bortolotti, S.; Marconato, E.; Carraro, T.; Morettin, P.; van Krieken, E.; Vergari, A.; Teso, S.; Passerini, A. A Neuro-Symbolic Benchmark Suite for Concept Quality and Reasoning Shortcuts. In Proceedings of the Thirty-Eighth Annual Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 9–15 December 2024; pp. 1–44.
202. Mirchandani, R.; Sundar, K.; Mohamed, S.; Krueger, D. SCERL: A Text-based Safety Benchmark for Reinforcement Learning Problems. In Proceedings of the NeurIPS Datasets and Benchmarks Track, Virtual, 28 November 2022.
203. James, S.; Ma, Z.; Rovick Arrojo, D.; Davison, A.J. RLBench: The Robot Learning Benchmark & Learning Environment. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3019–3026.
204. Hudson, D.A.; Manning, C.D. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6700–6709.
205. Park, D.H.; Hendricks, L.A.; Akata, Z.; Rohrbach, A.; Schiele, B.; Darrell, T.; Rohrbach, M. Multimodal Explanations: Justifying Decisions and Pointing to the Evidence. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
206. Yi, K.; Gan, C.; Li, Y.; Kohli, P.; Wu, J.; Torralba, A.; Tenenbaum, J.B. Clevrer: Collision events for video representation and reasoning. *arXiv* **2019**, arXiv:1910.01442.
207. Xiao, J.; Shang, X.; Yao, A.; Chua, T.S. Next-qa: Next phase of question-answering to explaining temporal actions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 9777–9786.
208. Zawalski, M.; Chen, W.; Pertsch, K.; Mees, O.; Finn, C.; Levine, S. Robotic Control via Embodied Chain-of-Thought Reasoning. *arXiv* **2024**, arXiv:2407.08693.

209. Huang, W.; Xia, F.; Xiao, T.; Chan, H.; Liang, J.; Florence, P.; Zeng, A.; Tompson, J.J.R.; Mordatch, I.; Chebotar, Y.; et al. InnerMonologue: Embodied Reasoning through Planning with Language Models. In Proceedings of the Conference on Robot Learning (CoRL), Auckland, New Zealand, 14–18 December 2022.
210. Yu, T.; Zhang, R.; Yang, K.; Yasunaga, M.; Wang, D.; Li, Z.; Ma, J.; Li, I.; Yao, Q.; Roman, S.; et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *arXiv* **2018**, arXiv:1809.08887.
211. Price, P.; Fisher, W.M.; Bernstein, J.; Pallett, D.S. The DARPA 1000-word resource management database for continuous speech recognition. In Proceedings of the ICASSP-88, International Conference on Acoustics, Speech, and Signal Processing, New York, NY, USA, 11–14 April 1988; IEEE Computer Society: New York, NY, USA, 1988; pp. 651–652.
212. Zhang, Y.; Deriu, J.; Katsogiannis-Meimarakis, G.; Kosten, C.; Koutrika, G.; Stockinger, K. ScienceBenchmark: A Complex Real-World Benchmark for Evaluating Natural Language to SQL Systems. In Proceedings of the VLDB Endowment, Vancouver, BC, Canada, 28 August–1 September 2023.
213. Chen, B.; Zhang, F.; Nguyen, A.; Zan, D.; Lin, Z.; Lou, J.G.; Chen, W. CodeT: Code Generation with Generated Tests. In Proceedings of the Eleventh International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
214. Lin, X.V.; Wang, C.; Zettlemoyer, L.; Ernst, M.D. NL2Bash: A Corpus and Semantic Parser for Natural Language Interface to the Linux Operating System. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018; Calzolari, N., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., et al., Eds.
215. Amini, A.; Gabriel, S.; Lin, P.; Chaturvedi, S.; Farhadi, A.; Hajishirzi, H. MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), Minneapolis, MN, USA, 2–7 June 2019.
216. Zhou, S.; Xu, F.F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; et al. Webarena: A realistic web environment for building autonomous agents. *arXiv* **2023**, arXiv:2307.13854.
217. Oh, J.H.; Kadowaki, K.; Kloetzer, J.; Iida, R.; Torisawa, K. Open-Domain Why-Question Answering with Adversarial Learning to Encode Answer Texts. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Korhonen, A., Traum, D., Màrquez, L., Eds.; pp. 4227–4237. [CrossRef]
218. Xu, Q.; Hong, F.; Li, B.; Hu, C.; Chen, Z.; Zhang, J. On the Tool Manipulation Capability of Open-source Large Language Models. *arXiv* **2023**, arXiv:2305.16504.
219. Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv* **2021**, arXiv:2112.09332.
220. Suhr, A.; Zhou, S.; Zhang, A.; Zhang, I.; Bai, H.; Artzi, Y. A corpus for reasoning about natural language grounded in photographs. *arXiv* **2018**, arXiv:1811.00491.
221. Thrush, T.; Jiang, R.; Bartolo, M.; Singh, A.; Williams, A.; Kiela, D.; Ross, C. Winoground: Probing vision and language models for visio-linguistic compositionality. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5238–5248.
222. Zhong, V.; Xiong, C.; Socher, R. Seq2SQL: Generating structured queries from natural language using reinforcement learning. *arXiv* **2017**, arXiv:1709.00103.
223. Yu, T.; Zhang, R.; Er, H.Y.; Li, S.; Xue, E.; Pang, B.; Lin, X.V.; Tan, Y.C.; Shi, T.; Li, Z.; et al. Cosql: A conversational text-to-sql challenge towards cross-domain natural language interfaces to databases. *arXiv* **2019**, arXiv:1909.05378.
224. Dries, A.; Kimmig, A.; Meert, W.; Renkens, J.; Van den Broeck, G.; Vlasselaer, J.; De Raedt, L. ProbLog2: Probabilistic Logic Programming. In *Machine Learning and Knowledge Discovery in Databases*; Bifet, A., May, M., Zadrozny, B., Gavalda, R., Pedreschi, D., Bonchi, F., Cardoso, J., Spiliopoulou, M., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 312–315.
225. Gebser, M.; Kaminski, R.; Kaufmann, B.; Schaub, T. clingo = ASP + Control: Preliminary Report. *arXiv* **2014**, arXiv:1405.3694.
226. Leone, N.; Pfeifer, G.; Faber, W.; Eiter, T.; Gottlob, G.; Perri, S.; Scarcello, F. The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Log. (TOCL)* **2006**, *7*, 499–562. [CrossRef]
227. Byrd, W.E.; Holk, E.; Friedman, D.P. miniKanren, live and untagged: Quine generation via relational interpreters (programming pearl). In Proceedings of the 2012 Annual Workshop on Scheme and Functional Programming, Copenhagen, Denmark, 9–15 September 2012; pp. 8–29.
228. Manandhar, S.; Džeroski, S.; Erjavec, T. Learning multilingual morphology with CLOG. In *International Conference on Inductive Logic Programming*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 135–144.
229. Gebser, M.; Kaminski, R.; Kaufmann, B.; Schaub, T. *Answer Set Solving in Practice*; Springer Nature: Cham, Switzerland, 2022.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.