A Survey on LLM Symbolic Reasoning

Jindong Li[1*], Yali Fu[2*], Yang Yang[1], Jiahong Liu[3], Hongce Zhang[1], Haoxuan Li[4], Yutao Yue[1], Menglin Yang[1†]

[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]Jilin University
[3]The Chinese University of Hong Kong
[4]Peking University
{jli839, yyang945}@connect.hkust-gz.edu.cn, fuyl23@mails.jlu.edu.cn
jiahong.liu21@gmail.com, hxli@stu.pku.edu.cn
{hongcezh, yutaoyue}@hkust-gz.edu.cn, menglin.yang@outlook.com

Abstract

Large Language Models (LLMs) have achieved impressive progress in reasoning and problem-solving, yet their logical consistency and verifiability remain limited. To enhance these capabilities, recent research has developed LLM symbolic reasoning, which integrates symbolic formalisms and structured computation into neural reasoning. This survey provides the first systematic overview of this rapidly growing field. We organize existing studies into a unified taxonomy from a progressive perspective, ranging from fundamental symbolic formalization, logic programming, theorem proving, neuro-symbolic integration, to practical planning and search-guided reasoning, program synthesis, and tabular reasoning. Each category reflects a distinct mechanism by which LLMs acquire, represent, or utilize symbolic knowledge to enhance reasoning reliability. Beyond taxonomy, we analyze interaction patterns between neural and symbolic components, summarize representative applications in domains such as mathematics, programming, and healthcare, and discuss challenges in alignment, consistency, and interpretability. This work aims to provide conceptual clarity and inspire future research toward trustworthy, generalizable, and cognitively grounded LLM symbolic reasoning. A continuously updated project is available at: https://github.com/jindongli-Ai/LLM-Symbolic-Reasoning-Survey.

Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding and generation. However, their reasoning capabilities often remain limited, such as making logical mistakes or failing to follow formal inference rules (Zhou et al. 2024; Cheng et al. 2025). A range of prompting and architectural techniques, such as Chain-of-Thought (CoT) prompting, self-verification, and tree search, have been introduced to improve logical consistency and multi-step reasoning. However, these methods typically rely on informal text language and do not guarantee correctness under rigorous verification.

[*]Equal Contribution
[†]Corresponding Author

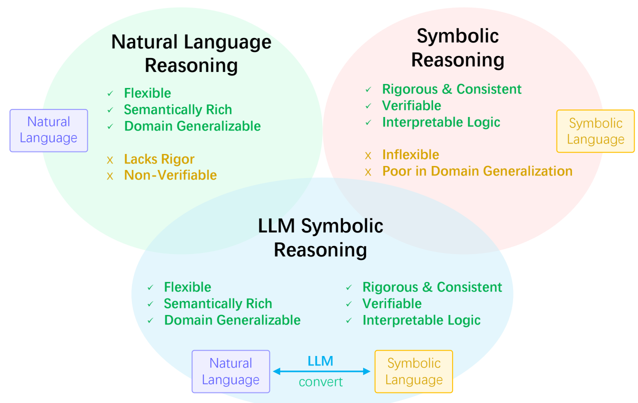

Figure 1: LLM Symbolic Reasoning bridges natural and symbolic reasoning, combining the flexibility and semantic richness of natural language with the rigor, consistency, and verifiability of symbolic logic.

To address these limitations, recent research has explored LLM symbolic reasoning, in which LLMs are augmented with symbolic reasoning mechanisms. This involves LLM-generated symbolic expressions (e.g., logic formulas, executable programs, proof sketches) being processed by symbolic solvers such as Prolog, Z3, Lean, or SMT (Zhang et al. 2023a; Yang et al. 2024; Hu et al. 2024). This hybrid design bridges the expressive flexibility of natural language with the rigor and verifiability of symbolic reasoning, enabling LLMs to produce not only plausible but also formally consistent reasoning outcomes, as shown in Fig. 1.

Despite rapid progress, none of the existing surveys systematically investigates LLM symbolic reasoning, where LLMs collaborate with or are guided by *symbolic reasoning mechanisms* (e.g., logic programs (Tan et al. 2024), theorem provers (Qi et al. 2025), or neural integration (Calanzone, Teso, and Vergari 2025)) to achieve interpretable and verifiable reasoning. Existing reviews on neuro-symbolic or logic-augmented models mainly address task-specific learning. A recent study (Cheng et al. 2025) focuses on logical question answering and consistency, rather than solver-

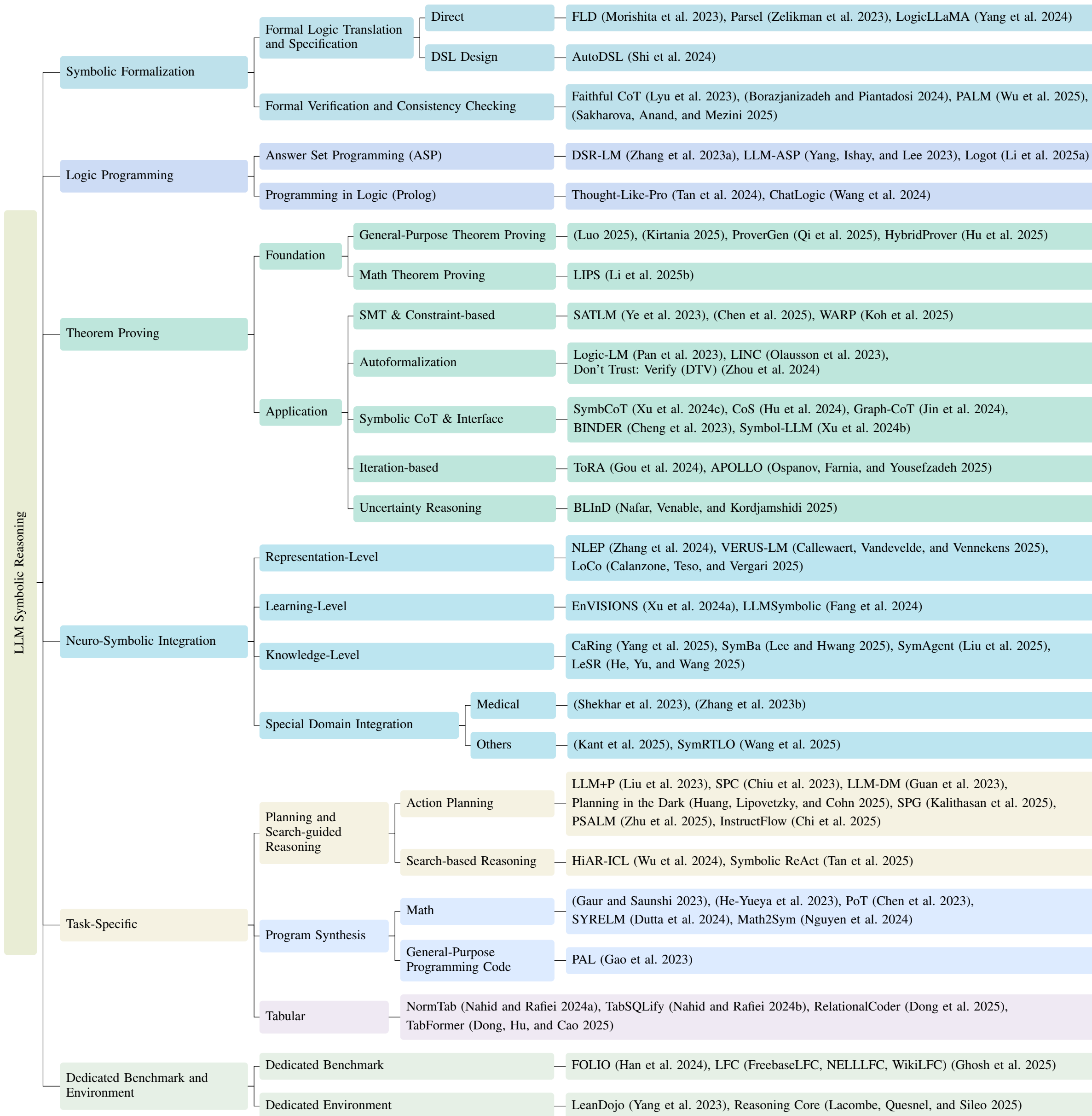

Figure 2: Taxonomy of *LLM Symbolic Reasoning* with representative works.

centric symbolic integration. It is worth noting that, while knowledge graph (KG)-based methods and probabilistic logic models can also be regarded as part of the broader symbolic reasoning paradigm, they represent relatively peripheral directions that are already well established. Therefore, we don't consider these works in this survey. This work mainly fills the gap by providing a unified and principled synthesis of how symbolic reasoning mechanisms are incorporated into LLMs to enhance logical consistency, reliability, and cognitive grounding.

Our main contributions are summarized as follows:

- We present the first comprehensive survey on LLM symbolic reasoning, providing a unified conceptual framework that connects symbolic reasoning paradigms with neural reasoning processes. Our taxonomy covers seven major research lines: *symbolic formalization, logic programming, theorem proving, neuro-symbolic integration, planning and search-guided reasoning, program synthesis, and tabular reasoning* (see Fig. 2).
- We introduce a mechanism-oriented taxonomy (*Sec.2-Sec.5*) that characterizes cooperative patterns between symbolic and neural components from a progressive perspective, ranging from fundamental language-to-symbol formalization, symbolic rule execution, and proof verification to neural integration, revealing general architectural principles across diverse reasoning systems.

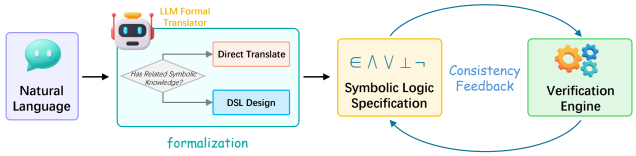

Figure 3: The general symbolic formalization pipeline, where LLMs translate natural language descriptions into formal logic specifications to verify their consistency through symbolic engines.

- We also summarize representative task-specific applications (*Sec.6*) across mathematics, programming, healthcare, robotics, etc., demonstrating how LLMs employ symbolic reasoning to support reliable and domain-specific decision making.
- We highlight open challenges in scalability, consistency, and interpretability, and propose future research directions toward building trustworthy, generalizable, and cognitively grounded LLM symbolic reasoning systems.

Symbolic Formalization

Symbolic formalization focuses on *transforming natural language into verifiable logical forms* (see Fig. 3). It bridges human semantics and machine-checkable symbolic structures, enabling LLMs to express reasoning in precise symbolic terms without executing the reasoning process itself.

Formal Logic Translation and Specification

Direct. Recent works enable LLMs to transform natural language (NL) into explicit logical or probabilistic specifications, providing machine-verifiable symbolic representations of reasoning semantics. e.g., FLD (Morishita et al. 2023) generates a logic-grounded corpus from formal axioms to improve deductive reasoning generalization. Parsel (Zelikman et al. 2023) converts natural language problems into compositional symbolic programs for verifiable algorithmic reasoning. LogicLLaMA (Yang et al. 2024) enhances natural language–to–first-order logic translation through iterative symbolic correction guided by logical consistency. Together, these approaches demonstrate a unified trend toward grounding linguistic semantics in formal logical structures, enabling LLMs to express reasoning in explicitly verifiable forms.

Domain-Specific Language (DSL) Design. When LLMs lack direct symbolic knowledge for formal translation, an intermediate language becomes necessary to encode procedural or domain-specific logic in a structured form. AutoDSL (Shi et al. 2024) automates this process by learning syntactic structures and semantic constraints from domain corpora, yielding dual-layer rules that guide LLMs toward structured and verifiable procedural reasoning.

Formal Verification and Consistency Checking

To ensure that reasoning is not only plausible but also logically sound, some works incorporate formal verification into

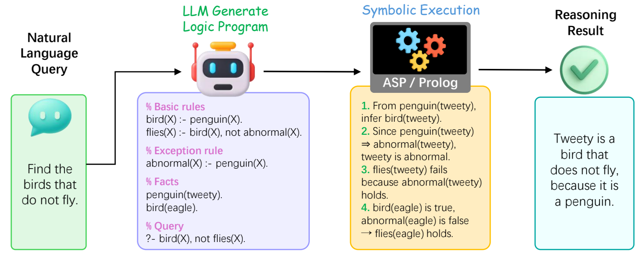

Figure 4: The general logic programming pipeline, where LLMs generate symbolic rules executed by ASP or Prolog solvers to derive reasoning results.

reasoning pipelines. By coupling LLM outputs with symbolic solvers or theorem provers, these works check the validity of reasoning chains, enforce internal consistency, and prevent spurious conclusions. For instance, Faithful CoT (Lyu et al. 2023) introduces a two-stage pipeline that translates natural language into symbolic reasoning chains and executes them deterministically to guarantee faithful reasoning. Reliable Reasoning beyond Natural Language (Borazjanizadeh and Piantadosi 2024) leverages Prolog execution to validate logical consistency and proposes the Non-Linear Reasoning dataset to test robustness. PALM (Wu et al. 2025) combines path-aware symbolic execution with LLM-based test generation, achieving higher path coverage and interpretability. Integrating Symbolic Execution into the Fine-Tuning of Code-Generating LLMs (Sakharova, Anand, and Mezini 2025) employs symbolic execution to generate unbiased test cases, improving RL and DPO training for code generation. Such efforts illustrate a shift from merely plausible reasoning traces to verifiable pipelines, where symbolic execution ensures correctness and consistency across different reasoning domains.

Logic Programming

Logic programming focuses on *enabling LLMs to generate and execute symbolic rules through declarative frameworks* such as ASP (Answer Set Programming) and Prolog (Programming in Logic) (see Fig. 4). It emphasizes rule-based reasoning, where relations are explicitly defined and solved by symbolic engines to derive interpretable conclusions.

Answer Set Programming (ASP)

LLMs have been combined with ASP to enable declarative, non-monotonic reasoning, where symbolic rules ensure logical consistency while natural language inputs are flexibly interpreted by neural models. e.g., DSR-LM (Zhang et al. 2023a) integrates differentiable symbolic modules with frozen LLMs, allowing gradient-based optimization of logical constraints. LLM-ASP (Yang, Ishay, and Lee 2023) leverages LLMs as few-shot semantic parsers that transform text into atomic facts for ASP solvers, achieving interpretable multi-hop reasoning without retraining. Logot (Li et al. 2025a) extends this line by letting the model dynamically construct and execute logic programs as intermediate thoughts, improving symbolic fidelity in natural-language reasoning. These approaches reveal how declarative logic programming serves as a stable reasoning substrate for LLMs, combining linguistic flexibility with explicit symbolic control.

Programming in Logic (Prolog)

Recent studies integrate LLMs with Prolog to perform predicate-based reasoning, where symbolic rules and facts enable explicit logical execution via backtracking. e.g., Thought-Like-Pro (Tan et al. 2024) introduces a Prolog-guided self-driven framework that aligns LLM reasoning traces with verifiable logical proofs. ChatLogic (Wang et al. 2024) combines LLMs with pyDatalog to translate natural language queries into executable logic programs for multi-step deductive reasoning. These approaches illustrate how Prolog-based integration provides a structured backbone that enhances the rigor and interpretability of LLM reasoning.

Theorem Proving

Theorem proving focuses on *integrating LLMs with formal proof systems to establish verifiable logical conclusions through Automated Theorem Proving (ATP) (e.g., Prover9, Z3, SMT) or Interactive Theorem Proving (ITP) (e.g., Lean, Coq, Isabelle/HOL)*. It enables LLMs to construct, guide, or verify proofs, bridging natural language reasoning with machine-checkable formal deduction (see Fig. 5).

Foundation

This part covers general-purpose theorem provers that serve as the core symbolic engines for formal deduction, where LLMs assist in proof generation, tactic prediction, or interactive verification within ATP or ITP systems such as Lean, Coq, and SMT.

General-Purpose Theorem Proving. General-purpose theorem proving integrates LLMs with formal systems to construct verifiable proofs across diverse logical settings. For example, Reinforced-LLM (Luo 2025) trains LLMs through reinforcement learning over Lean tactics, optimizing next-step prediction via reward feedback. Steering-LLMs (Kirtania 2025) enhances proof reliability by steering token generation through prompt-level supervision without additional training. ProverGen (Qi et al. 2025) links LLMs with Prover9 to generate FOL-based reasoning data and evaluate symbolic consistency in natural language proofs. HybridProver (Hu et al. 2025) combines whole-proof synthesis and tactic-based generation through proof sketch refinement, achieving state-of-the-art theorem proving in Isabelle/HOL with improved scalability and proof reliability. These approaches collectively move towards scalable, verifiable, and system-agnostic theorem proving with LLMs.

Math Theorem Proving. Math theorem proving integrates LLMs with symbolic solvers to construct verifiable proofs for algebraic or inequality reasoning within formal systems. e.g., LIPS (Li et al. 2025b) combines LLM-generated rewriting tactics with symbolic verification via CAD and SMT solvers, enabling stepwise proof synthesis for Olympiad-level inequalities.

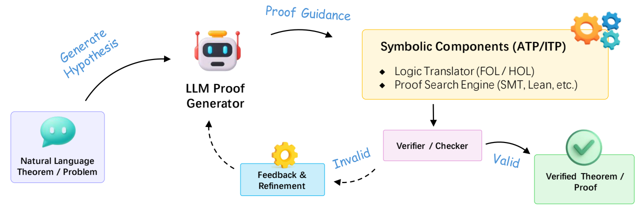

Figure 5: Iterative proof refinement of LLM-assisted theorem proving. LLMs generate proof hypotheses from natural language theorems, guide symbolic solvers in proof search, and iteratively refine invalid proofs through verifier feedback until a theorem is verified.

Application

This part primarily focuses on diverse downstream applications of theorem proving and symbolic reasoning, where LLMs closely cooperate with solvers for constraint satisfaction, verification, symbolic interfaces, iteration-based reasoning, and uncertainty modeling.

SMT & Constraint-based Reasoning. This line of work employs SMT and constraint solvers to verify logical relations and structured constraints, where LLMs convert natural specifications into symbolic forms for solver-based reasoning. e.g., SATLM (Ye et al. 2023) translates natural text into SAT/SMT constraints solved by Z3, reducing planning and execution errors in CoT reasoning. (Chen et al. 2025) applies satisfiability checking to string requirements by generating consistent data-checker pairs for SMT-based validation. WARP (Koh et al. 2025) formulates worst-case symbolic analysis as constraint solving, enabling robustness and generalization across boundary conditions. These methods strengthen logical reliability and extend symbolic verification to structured and adversarial settings.

Autoformalization. Recent works couple LLMs with symbolic solvers to ensure the faithfulness of reasoning and enable verifiable inference beyond textual chains. e.g., Logic-LM (Pan et al. 2023) formulates natural-language logical problems into symbolic representations (e.g., FOL, CSP, SAT) and delegates inference to external solvers such as Prover9 and Z3, with a self-refinement loop that leverages solver feedback to iteratively correct symbolic errors. LINC (Olausson et al. 2023) reformulates the problem space in symbolic FOL formulas and employs majority voting to ensure syntactic and semantic validity. Don't Trust (Zhou et al. 2024) applies autoformalization for quantitative reasoning by translating LLM-generated solutions into Isabelle/HOL proofs, verifying them through theorem proving rather than majority voting to filter unfaithful reasoning. These highlight a shift from heuristic textual reasoning to formally verifiable computation, showing that symbolic verification can transform LLM reasoning into a provably consistent and trustworthy process.

Symbolic CoT & Interface. This direction focuses on incorporating symbolic representations or communication interface mechanisms into the reasoning process of LLMs, either by embedding symbolic logic within CoT or enabling structured interaction between LLMs and external symbolic modules to achieve more faithful and verifiable reasoning. e.g., SymbCoT (Xu et al. 2024c) embeds symbolic expressions into CoT to realize faithful and verifiable logical reasoning within LLMs. Similarly, CoS (Hu et al. 2024) adopts chain-of-symbol prompting to express spatial relations and enhance multi-step geometric reasoning. Graph-CoT (Jin et al. 2024) performs iterative reasoning on text-attributed graphs by alternating LLM reasoning and graph function chain execution. BINDER (Cheng et al. 2023) builds a neural–symbolic interface linking programming languages with LM API calls for executable and interpretable reasoning. Symbol-LLM (Xu et al. 2024b) further unifies symbol-centric communication between LLM reasoning and various symbolic forms. These studies reveal that formalizing or externalizing reasoning steps through symbolic structures substantially improves the faithfulness and interpretability of LLM-based reasoning.

Iteration-based Reasoning. Iteration-based reasoning establishes a closed collaboration loop between LLMs and external provers or proof assistants, enabling models to receive feedback from compilers or solvers for iterative refinement and repair. e.g., ToRA (Gou et al. 2024) interleaves language reasoning with symbolic solvers over multiple rounds. APOLLO (Ospanov, Farnia, and Yousefzadeh 2025) enables Lean-based proof repair, where compiler feedback and modular agents iteratively fix and revalidate proofs with reduced sampling cost. These systems demonstrate that integrating formal tool feedback substantially enhances the reliability of LLM-based theorem proving.

Uncertainty Reasoning. Uncertainty reasoning addresses situations where input information or logical premises are incomplete, ambiguous, or probabilistic. Instead of producing deterministic conclusions, these methods estimate belief distributions and propagate uncertainty through the reasoning process. e.g., BLInD (Nafar, Venable, and Kordjamshidi 2025) focuses on Bayesian reasoning over uncertain text with generative LLMs. It designs three prompt strategies to map to symbolic solvers, enabling probabilistic conclusions under incomplete or conflicting evidence.

Neuro-Symbolic Integration

Neuro-symbolic integration focuses on unifying neural representation learning with symbolic reasoning, as shown in Fig. 6. It enables LLMs to effectively combine statistical generalization with logical structure, bridging continuous embeddings and discrete logic through coordinated or end-to-end learning mechanisms.

Representation-Level

Representation-level integration aligns continuous embeddings with symbolic constraints, allowing LLMs to internalize logic-consistent structures without relying on explicit solvers. e.g., NLEP (Zhang et al. 2024) embeds symbolic programs within the latent space of LLMs to enable unified

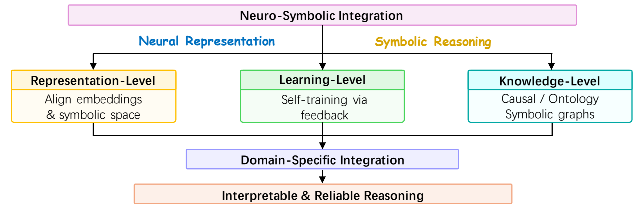

Figure 6: Hierarchical structure of Neuro-Symbolic Integration, bridging neural representations and symbolic reasoning toward interpretable reasoning.

neural–symbolic reasoning. VERUS-LM (Callewaert, Vandevelde, and Vennekens 2025) builds a two-phase framework integrating LLMs with an extended FOL engine (IDP-Z3) for versatile logical reasoning. LoCo (Calanzone, Teso, and Vergari 2025) enforces logical consistency in LLMs via a semantic loss derived from weighted model counting. These approaches bridge the gap between continuous and symbolic spaces, promoting logically faithful and interpretable reasoning within LLMs.

Learning-Level

Learning-level integration enables LLMs to improve symbolic reasoning through autonomous self-training and internal feedback without external supervision. e.g., ENVISIONS (Xu et al. 2024a) develops an environment-guided self-training loop that refines reasoning via self-exploration, self-correction, and self-rewarding. LLMSymbolic (Fang et al. 2024) views LLMs as implicit neurosymbolic reasoners, aligning neural representations with symbolic inference through iterative self-distillation. These approaches show that symbolic reasoning ability can emerge through self-evolution rather than external supervision.

Knowledge-Level

Knowledge-level integration combines neural reasoning with explicit causal, ontological, or knowledge-graph structures, aiming to enhance interpretability and traceable reasoning. e.g., CaRing (Yang et al. 2025) builds a Prolog-based framework where LLMs generate logical rules and symbolic solvers ensure causal and verifiable proofs. SymBa (Lee and Hwang 2025) employs symbolic backward chaining, letting a solver manage proofs while invoking LLMs only when additional information is required. SymAgent (Liu et al. 2025) develops a self-learning agent that integrates neural reasoning with symbolic rule induction for knowledge-graph reasoning. LeSR (He, Yu, and Wang 2025) combines LLM-generated rules with symbolic reasoning to enhance reliability and interpretability in knowledge base completion. These collectively emphasize causality, completeness, and reliable symbolic grounding in complex reasoning, bridging natural language understanding with structured knowledge reasoning.

Special Domain Integration

This part focuses on task-oriented frameworks where LLMs incorporate symbolic reasoning into specialized domains such as medicine, science, and law to enhance reasoning interpretability and reliability.

Medical. Neuro-symbolic approaches in the medical domain combine LLMs with symbolic knowledge bases and reasoning rules to support interpretable diagnosis and clinical decision-making. e.g., (Shekhar et al. 2023) couples symbolic reasoning with LLM-based temporal modeling to interpret longitudinal patient records and ensure consistency across heterogeneous clinical narratives. (Zhang et al. 2023b) integrates automated medical concept extraction with ontology-based symbolic reasoning for explainable recommendations. Together, these works illustrate how neuro-symbolic integration enhances reliability and transparency in medical understanding and decision support.

Others. Neuro-symbolic reasoning has also been extended to domains such as law and hardware engineering, where LLMs must reconcile natural-language understanding with formal symbolic constraints. e.g, In legal reasoning, (Kant et al. 2025) (Kant et al. 2025) pairs LLMs with Prolog to encode insurance contracts as computable rules for consistent, auditable decisions. In engineering optimization, SymRTLO (Wang et al. 2025) adds a neuron-inspired symbolic module for register-transfer-level (RTL) optimization, guiding LLM-generated code under logic constraints for correctness and efficiency. Together, these works show that symbolic integration enhances both reasoning transparency and domain reliability across diverse applications beyond traditional text understanding.

Task-Specific

This section reviews how LLMs can tailor symbolic reasoning workflows for task-specific domains, such as action planning, program synthesis, and tabular reasoning, highlighting their application-driven innovations and specialized design considerations.

Planning and Search-guided Reasoning

Planning and search-guided reasoning focuses on integrating LLMs with symbolic planners and search algorithms for structured, multi-step reasoning (see Fig. 7). In particular, planning refers to action planning, a core paradigm in embodied intelligence where LLMs collaborate with symbolic systems to generate, verify, and execute robot actions under logical constraints.

Action Planning. LLM-symbolic planning methods couple language-driven goal understanding with formal symbolic planning to achieve long-horizon, verifiable task generation. Across works, LLMs translate natural-language instructions into structured planning representations (e.g., PDDL files, action schemas, or symbolic constraints), while external planners search for optimal action sequences. This hybrid design retains LLM semantic flexibility and symbolic soundness. e.g., LLM+P (Liu et al. 2023) converts task descriptions into PDDL and uses a classical planner for optimal plans, then translates them back to language, achieving zero-shot planning without fine-tuning. SPC (Chiu et al. 2023) combines symbolic planning with code generation

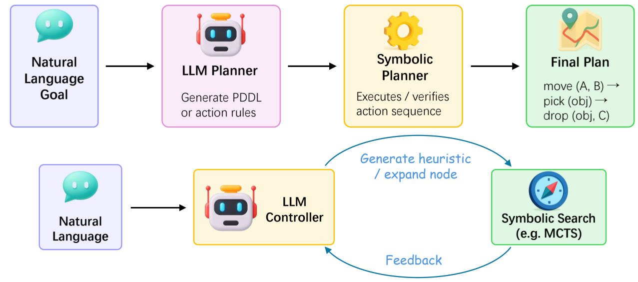

Figure 7: The general pipelines of action planning (top) and search-based reasoning (bottom), illustrating how LLMs generate symbolic plans or guide symbolic search processes.

for grounded dialogue, where LLMs generate Python actions and a symbolic planner selects goal-directed responses. LLM-DM (Guan et al. 2023) constructs and utilizes LLM-based world models to simulate state transitions for model-based task planning. Planning in the Dark (Huang, Lipovetzky, and Cohn 2025) builds a diverse action-schema library and semantic filter to eliminate expert intervention in LLM-symbolic planning. SPG (Kalithasan et al. 2025) integrates neuro-symbolic programmatic representations to generalize planning over inductive spatial concepts. PSALM (Zhu et al. 2025) infers action semantics from environment feedback, learning symbolic pre/post-conditions for planners without manual domain files. InstructFlow (Chi et al. 2025) employs a symbolic constraint-guided multi-agent flow for adaptive code generation and long-horizon plan repair. Together, these methods progress from LLM-generated symbolic plans to feedback-driven constraint-guided reasoning, marking the evolution from language-to-symbol translation toward adaptive symbolic execution in LLM-based action planning.

Search-based Reasoning. LLM-symbolic reasoning with search algorithms leverages structured exploration to refine multi-step reasoning trajectories. By integrating Monte Carlo Tree Search (MCTS) and other symbolic search paradigms, these methods allow LLMs to plan, evaluate, and verify intermediate reasoning steps rather than relying solely on sequential text generation. e.g, HiAR-ICL (Wu et al. 2024) employs MCTS to construct high-level reasoning strategies ("thought cards") for adaptive in-context learning. Symbolic ReAct (Tan et al. 2025) integrates symbolic supervision into MCTS to align reasoning trajectories with logical consistency. Together, these works demonstrate that search-guided symbolic control can systematically improve reasoning reliability, bridging structured exploration and LLM-based generation.

Program Synthesis

Program synthesis focuses on enabling LLMs to generate executable programs that satisfy symbolic specifications or logical constraints. By combining neural generation with symbolic verification, LLMs can produce code or expressions that are both semantically correct and formally sound.

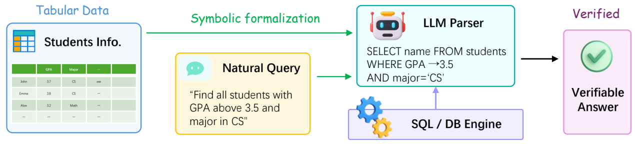

Figure 8: LLM Symbolic Reasoning over tabular data, where LLMs translate tabular data and natural queries into symbolic language (e.g., SQL) executed by symbolic engines for verifiable answers.

Math. LLM symbolic reasoning in mathematics aims to bridge natural-language understanding with formal symbolic computation. These methods translate math problems into executable equations or programs and employ external solvers for precise calculation and verification. e.g., (Gaur and Saunshi 2023) and (He-Yueya et al. 2023) leverage symbolic templates and solvers (e.g., SymPy) to ensure accurate equation generation and solution verification. PoT (Chen et al. 2023) separates reasoning from computation by letting LLMs produce executable Python programs. SYRELM (Dutta et al. 2024) reinforces compact models to invoke solvers efficiently through formal-language translation. Math2Sym (Nguyen et al. 2024) integrates symbolic formalization, execution, and explanation into a unified LLM–solver pipeline. These methods strengthen symbolic-assisted mathematical reasoning from equation parsing toward executable and verifiable solver-guided computation.

General-Purpose Programming Code. LLM-symbolic reasoning for general-purpose programming combines natural-language reasoning with executable program synthesis, allowing LLMs to delegate precise computation to external interpreters. e.g, PAL (Gao et al. 2023) enables LLMs to generate Python functions as intermediate reasoning steps, which are executed by a symbolic interpreter for final answers. By converting reasoning into executable programs, PAL achieves deterministic and verifiable results beyond text-only reasoning.

Tabular Reasoning

Recent studies enhance LLM reasoning on tabular data (i.e., table) by introducing an explicit symbolic layer (e.g., SQL or relational transformation) for structured and verifiable computation (see Fig. 8). All methods combine LLM-driven symbolic operations with neural reasoning, improving consistency and interpretability. e.g., NormTab (Nahid and Rafiei 2024a) performs LLM-guided normalization to unify table schemas and values, improving reasoning consistency. TabSQLify (Nahid and Rafiei 2024b) adopts SQL-based symbolic execution, decomposing large tables into subtables for interpretable reasoning. RelationalCoder (Dong et al. 2025) transforms tables via programmatic relational code, enabling procedural symbolic reasoning. TabFormer (Dong, Hu, and Cao 2025) uses reinforced relational transformations to jointly optimize normalization and reasoning accuracy. Together, they illustrate a unified trend: integrating symbolic representation and execution into LLM reasoning pipelines for structured, scalable table understanding.

Dedicated Benchmark and Environment

Dedicated benchmarks and environments form the foundation for evaluating LLM-symbolic reasoning, overcoming the limits of generic reasoning datasets by enabling solver-aware, formally grounded, and verifiable assessment.

Dedicated Benchmark

To enable systematic evaluation and data support for this emerging direction, several works have introduced benchmarks that provide standardized datasets for assessing LLM–symbolic reasoning across tasks such as theorem proving, logic programming, and program synthesis. FOLIO (Han et al. 2024) provides a first-order logic annotated benchmark for natural language reasoning, where conclusions are formally verified via an FOL inference engine to assess the reasoning ability of LLMs. (Ghosh et al. 2025) constructs three datasets (i.e., FreebaseLFC, NELLLFC, and WikiLFC) to evaluate the logical consistency of LLMs in fact-checking through logically equivalent and inverse queries.

Dedicated Environment

Establishing dedicated environments for LLM–symbolic reasoning is essential to enable controllable execution, systematic verification, and reproducible evaluation. LeanDojo (Yang et al. 2023) provides a large-scale interactive environment for Lean theorem proving, integrating proof retrieval, tactic generation, and kernel-level verification to support reliable LLM–solver interaction. Reasoning Core (Lacombe, Quesnel, and Sileo 2025) provides a scalable RL environment with verifiable rewards, procedurally generating symbolic reasoning tasks with external solver verification for rigorous LLM reasoning evaluation.

Challenges and Discussions

Despite rapid progress, LLM symbolic reasoning remains fragmented and lacks unified design principles. We discuss key challenges and corresponding future directions for its further advancement in this section.

Prompt Design. Effective symbolic reasoning heavily depends on how prompts define the interaction between LLMs and solvers. Ambiguous or poorly structured prompts can break logical consistency or produce invalid symbolic forms (e.g., DSR-LM (Zhang et al. 2023a)). Future research should explore adaptive and feedback-driven prompting strategies that encode symbolic constraints and stabilize reasoning across solver types (e.g., LLM-ASP (Yang, Ishay, and Lee 2023), BINDER (Cheng et al. 2023)).

Syntax Correction. LLM-generated symbolic expressions frequently contain syntactic or structural errors that prevent solver execution and verification. Ensuring well-formed outputs requires grammar-constrained decoding (e.g., SymbCoT (Xu et al. 2024c)), syntax-aware training objectives, and dynamic correction guided by formal parsers (e.g., Math2Sym (Nguyen et al. 2024)). Developing the robust syntax enforcement mechanisms remains a key challenge for executable and verifiable reasoning (e.g., LeanDojo (Yang et al. 2023)).

Lack of Symbolic Knowledge Acquisition. Most current approaches assume LLMs can directly map natural language to formal logic, restricting reasoning to domains with predefined or well-structured symbolic systems (e.g., SymbCoT (Xu et al. 2024c)). In many real-world cases, however, such symbolic foundations are absent or incomplete (e.g., AutoDSL (Shi et al. 2024)). Future work should enable LLMs to autonomously construct symbolic systems, such as DSLs, from unstructured data, allowing formal reasoning in domains lacking predefined symbolic foundations.

Architectural Coherence. Most existing frameworks are loosely coupled pipelines rather than end-to-end systems. Symbolic modules are often appended to pretrained LLMs without shared objectives or consistent intermediate states (e.g., SymbCoT (Xu et al. 2024c)). Future architectures should support differentiable or policy-based coordination between neural and symbolic components (e.g., BINDER (Cheng et al. 2023)), allowing mutual supervision and adaptive reasoning control (see also Reasoning-Core (Lacombe, Quesnel, and Sileo 2025)).

Representation Alignment. LLMs encode meaning in continuous embedding spaces, whereas symbolic solvers operate over discrete and explicitly structured forms. This fundamental gap makes it difficult to ensure that neural representations faithfully correspond to symbolic entities and logical relations. Future research should explore richer hybrid representations, such as symbolic latent spaces, structure-aware embeddings, or bidirectional mapping modules, that preserve both formal soundness and semantic generalization (e.g., NLEP (Zhang et al. 2024), LoCo (Calanzone, Teso, and Vergari 2025)). Achieving such alignment across these representational modalities will be key to reliable LLM symbolic reasoning.

Learning under Sparse Supervision. Symbolic reasoning tasks often involve discrete decision spaces and sparse supervision signals, where only a few intermediate steps yield verifiable rewards. These approaches rely heavily on handcrafted data, few-shot examples, or synthetic reasoning traces. To overcome these limitations, future work may integrate reinforcement learning with solver feedback (e.g., LLMSymbolic (Fang et al. 2024)), self-verifying training loops (e.g., ENVISIONS (Xu et al. 2024a)), or curriculum-style task synthesis, enabling LLMs to efficiently acquire symbolic reasoning capabilities with minimal supervision.

Evaluation Consistency. The lack of standardized evaluation protocols makes it difficult to compare symbolic reasoning systems across domains and solver types. Existing benchmarks differ in problem formulation, metric definition, and reasoning granularity (e.g., FOLIO (Han et al. 2024)), often emphasizing linguistic correctness while overlooking formal validity or proof coherence. A unified evaluation framework which incorporates both symbolic executability and reasoning trace consistency (e.g., Logic-LM (Pan et al. 2023)), will be essential for reproducible progress. Developing large-scale, solver-interactive benchmarks could further quantify the fidelity and robustness of LLM-based symbolic reasoning (see also LFC (Ghosh et al. 2025)).

Cognitive Grounding and Interpretability. Despite growing integration with formal logic, the connection between symbolic reasoning and human cognitive abstraction remains largely unexplored. Current systems often treat solvers as external modules, without internalizing symbolic structure within the model's own reasoning process. Future research should investigate cognitively grounded frameworks where LLMs can emulate the compositional, hierarchical (e.g., SymBa (Lee and Hwang 2025)), and causal aspects of human reasoning (e.g., CaRing (Yang et al. 2025)). Such integration may lead to more interpretable and cognitively aligned symbolic reasoning systems (see also LeSR (He, Yu, and Wang 2025)).

Conclusion and Limitation

This survey provides a unified synthesis of LLM symbolic reasoning from a progressive perspective that connects mechanisms to applications. It systematically categorizes existing research into the following aspects: fundamental symbolic formalization, logic programming, theorem proving, neuro-symbolic integration, practical planning and search-guided reasoning, program synthesis, and tabular reasoning. By linking the rigor and verifiability of symbolic reasoning with the expressive flexibility of language models, this work elucidates how verifiable and reliable reasoning can be achieved across domains such as mathematics, programming, and healthcare. Moreover, it highlights that the growing trend of embedding symbolic reasoning principles into LLM architectures marks a promising step toward trustworthy and cognitively grounded intelligence. Finally, we discuss potential challenges and possible future directions for advancing the development of LLM symbolic reasoning.

Limitation. Owing to space constraints, this survey includes only representative works and omits detailed preliminaries, assuming readers have basic knowledge of symbolic reasoning and formal systems. We focus primarily on solver-centric paradigms and illustrative examples rather than exhaustive coverage. Moreover, systematic comparisons across symbolic paradigms remain limited due to varying environments and evaluation criteria. Future research should develop unified benchmarks and reproducible toolchains that enable fairer empirical assessment. Despite these limitations, we anticipate continued progress in unifying logic, learning, and cognition, advancing symbolic reasoning toward generalizable and verifiable intelligence.

Ethical Statement

This study provides a comprehensive and in-depth survey of existing literature on LLM symbolic reasoning. No new models, human subjects, deployment in real-world systems, or sensitive information were involved. We hope this work can serve as a reliable reference for future research while encouraging ethical and interpretable AI systems. Therefore, this work does not raise ethical or societal concerns.

References

Borazjanizadeh, N.; and Piantadosi, S. T. 2024. Reliable reasoning beyond natural language. *arXiv preprint arXiv:2407.11373*.

Calanzone, D.; Teso, S.; and Vergari, A. 2025. Logically Consistent Language Models via Neuro-Symbolic Integration. In *The Thirteenth International Conference on Learning Representations*.

Callewaert, B.; Vandevelde, S.; and Vennekens, J. 2025. Verus-lm: a versatile framework for combining llms with symbolic reasoning. *arXiv preprint arXiv:2501.14540*.

Chen, B.; Babikian, A. A.; Feng, S.; Varró, D.; and Mussbacher, G. 2025. LLM-based Satisfiability Checking of String Requirements by Consistent Data and Checker Generation. *arXiv preprint arXiv:2506.16639*.

Chen, W.; Ma, X.; Wang, X.; and Cohen, W. W. 2023. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. *TMLR*.

Cheng, F.; Li, H.; Liu, F.; van Rooij, R.; Zhang, K.; and Lin, Z. 2025. Empowering LLMs with Logical Reasoning: A Comprehensive Survey. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*.

Cheng, Z.; Xie, T.; Shi, P.; Li, C.; Nadkarni, R.; Hu, Y.; Xiong, C.; Radev, D.; Ostendorf, M.; Zettlemoyer, L.; Smith, N. A.; and Yu, T. 2023. Binding Language Models in Symbolic Languages. In *The Eleventh International Conference on Learning Representations*.

Chi, H.; Feng, Z.; Lyu, Y.; Zheng, C.; Luo, L.; Ong, Y.-S.; Tsang, I.; Chen, H.; Chang, Y.; and Yin, H. 2025. InstructFlow: Adaptive Symbolic Constraint-Guided Code Generation for Long-Horizon Planning. In *ICML 2025 Workshop on Programmatic Representations for Agent Learning*.

Chiu, J.; Zhao, W.; Chen, D.; Vaduguru, S.; Rush, A. M.; and Fried, D. 2023. Symbolic Planning and Code Generation for Grounded Dialogue. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 7426–7436.

Dong, H.; Hu, Y.; and Cao, Y. 2025. Reasoning and retrieval for complex semi-structured tables via reinforced relational data transformation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1382–1391.

Dong, H.; Hu, Y.; Peng, H.; and Cao, Y. 2025. Relational-Coder: Rethinking Complex Tables via Programmatic Relational Transformation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1771–1784.

Dutta, S.; Pandey, I.; Singh, J.; Manchanda, S.; Chakrabarti, S.; and Chakraborty, T. 2024. Frugal lms trained to invoke symbolic solvers achieve parameter-efficient arithmetic reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 17951–17959.

Fang, M.; Deng, S.; Zhang, Y.; Shi, Z.; Chen, L.; Pechenizkiy, M.; and Wang, J. 2024. Large language models are neurosymbolic reasoners. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 17985–17993.

Gao, L.; Madaan, A.; Zhou, S.; Alon, U.; Liu, P.; Yang, Y.; Callan, J.; and Neubig, G. 2023. PAL: program-aided language models. In *Proceedings of the 40th International Conference on Machine Learning*, 10764–10799.

Gaur, V.; and Saunshi, N. 2023. Reasoning in Large Language Models Through Symbolic Math Word Problems. In *Findings of the Association for Computational Linguistics: ACL 2023*, 5889–5903.

Ghosh, B.; Hasan, S.; Arafat, N. A.; and Khan, A. 2025. Logical Consistency of Large Language Models in Fact-Checking. In *The Thirteenth International Conference on Learning Representations*.

Gou, Z.; Shao, Z.; Gong, Y.; yelong shen; Yang, Y.; Huang, M.; Duan, N.; and Chen, W. 2024. ToRA: A Tool-Integrated Reasoning Agent for Mathematical Problem Solving. In *The Twelfth International Conference on Learning Representations*.

Guan, L.; Valmeekam, K.; Sreedharan, S.; and Kambhampati, S. 2023. Leveraging pre-trained large language models to construct and utilize world models for model-based task planning. *Advances in Neural Information Processing Systems*, 36: 79081–79094.

Han, S.; Schoelkopf, H.; Zhao, Y.; Qi, Z.; Riddell, M.; Zhou, W.; Coady, J.; Peng, D.; Qiao, Y.; Benson, L.; et al. 2024. FOLIO: Natural Language Reasoning with First-Order Logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 22017–22031.

He, Q.; Yu, J.; and Wang, W. 2025. Large language model-enhanced symbolic reasoning for knowledge base completion. *arXiv preprint arXiv:2501.01246*.

He-Yueya, J.; Poesia, G.; Wang, R.; and Goodman, N. 2023. Solving Math Word Problems by Combining Language Models With Symbolic Solvers. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*.

Hu, H.; Lu, H.; Zhang, H.; Song, Y.-Z.; Lam, W.; and Zhang, Y. 2024. Chain-of-symbol prompting for spatial reasoning in large language models. In *First Conference on Language Modeling*.

Hu, J.; Zhang, J.; Zhao, Y.; and Ringer, T. 2025. HybridProver: Augmenting Theorem Proving with LLM-Driven Proof Synthesis and Refinement. *arXiv preprint arXiv:2505.15740*.

Huang, S.; Lipovetzky, N.; and Cohn, T. 2025. Planning in the Dark: LLM-Symbolic Planning Pipeline without Experts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 26542–26550.

Jin, B.; Xie, C.; Zhang, J.; Roy, K. K.; Zhang, Y.; Li, Z.; Li, R.; Tang, X.; Wang, S.; Meng, Y.; et al. 2024. Graph Chain-of-Thought: Augmenting Large Language Models by Reasoning on Graphs. In *Findings of the Association for Computational Linguistics ACL 2024*, 163–184.

Kalithasan, N.; Sachdeva, S.; Panjeta, G. S.; Vora, H. H.; Singh, H. G.; Bindal, V.; Tuli, A.; Agarwal, D.; Paul, R.; and Singla, P. 2025. Sketch-Plan-Generalize: Learning and

Planning with Neuro-Symbolic Programmatic Representations for Inductive Spatial Concepts. In *ICML 2025 Workshop on Programmatic Representations for Agent Learning*.

Kant, M.; Nabi, S.; Kant, M.; Scharrer, R.; Ma, M.; and Nabi, M. 2025. Towards robust legal reasoning: Harnessing logical llms in law. *arXiv preprint arXiv:2502.17638*.

Kirtania, S. 2025. Activation Steering in Neural Theorem Provers. *arXiv preprint arXiv:2502.15507*.

Koh, D.; Noller, Y.; Pasareanu, C. S.; Skapars, A.; and Sun, Y. 2025. Worst-Case Symbolic Constraints Analysis and Generalisation with Large Language Models. *arXiv preprint arXiv:2506.08171*.

Lacombe, V.; Quesnel, V.; and Sileo, D. 2025. Reasoning Core: A Scalable RL Environment for LLM Symbolic Reasoning. *arXiv preprint arXiv:2509.18083*.

Lee, J.; and Hwang, W. 2025. SymBa: Symbolic Backward Chaining for Structured Natural Language Reasoning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2468–2484.

Li, N.; Liu, P.; Liu, Z.; Dai, T.; Jiang, Y.; and Xia, S.-T. 2025a. Logic-of-Thought: Empowering Large Language Models with Logic Programs for Solving Puzzles in Natural Language. *arXiv preprint arXiv:2505.16114*.

Li, Z.; Li, Z.; Tang, W.; Zhang, X.; Yao, Y.; Si, X.; Yang, F.; Yang, K.; and Ma, X. 2025b. Proving Olympiad Inequalities by Synergizing LLMs and Symbolic Reasoning. In *The Thirteenth International Conference on Learning Representations*.

Liu, B.; Jiang, Y.; Zhang, X.; Liu, Q.; Zhang, S.; Biswas, J.; and Stone, P. 2023. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*.

Liu, B.; Zhang, J.; Lin, F.; Yang, C.; Peng, M.; and Yin, W. 2025. Symagent: A neural-symbolic self-learning agent framework for complex reasoning over knowledge graphs. In *Proceedings of the ACM on Web Conference 2025*, 98–108.

Luo, Z. 2025. Reinforced Large Language Model is a formal theorem prover. *arXiv preprint arXiv:2502.08908*.

Lyu, Q.; Havaldar, S.; Stein, A.; Zhang, L.; Rao, D.; Wong, E.; Apidianaki, M.; and Callison-Burch, C. 2023. Faithful Chain-of-Thought Reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 305–329.

Morishita, T.; Morio, G.; Yamaguchi, A.; and Sogawa, Y. 2023. Learning deductive reasoning from synthetic corpus based on formal logic. In *International Conference on Machine Learning*, 25254–25274. PMLR.

Nafar, A.; Venable, K. B.; and Kordjamshidi, P. 2025. Reasoning over uncertain text by generative large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 24911–24920.

Nahid, M.; and Rafiei, D. 2024a. NormTab: Improving Symbolic Reasoning in LLMs Through Tabular Data Normalization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 3569–3585.

Nahid, M.; and Rafiei, D. 2024b. TabSQLify: Enhancing Reasoning Capabilities of LLMs Through Table Decomposition. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 5725–5737.

Nguyen, M. P.; Pham, M. P.; Ngo, M.; and Minh, K. T. 2024. Math2Sym: A System for Solving Elementary Problems via Large Language Models and Symbolic Solvers. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Olausson, T.; Gu, A.; Lipkin, B.; Zhang, C.; Solar-Lezama, A.; Tenenbaum, J.; and Levy, R. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 5153–5176.

Ospanov, A.; Farnia, F.; and Yousefzadeh, R. 2025. APOLLO: Automated LLM and Lean Collaboration for Advanced Formal Reasoning. *arXiv preprint arXiv:2505.05758*.

Pan, L.; Albalak, A.; Wang, X.; and Wang, W. 2023. Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 3806–3824.

Qi, C.; Ma, R.; Li, B.; Du, H.; Hui, B.; Wu, J.; Laili, Y.; and He, C. 2025. Large Language Models Meet Symbolic Provers for Logical Reasoning Evaluation. In *The Thirteenth International Conference on Learning Representations*.

Sakharova, M.; Anand, A.; and Mezini, M. 2025. Integrating Symbolic Execution into the Fine-Tuning of Code-Generating LLMs. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, 271–278.

Shekhar, S.; Tiwari, S.; Rensink, T.; Eskander, R.; and Salloum, W. 2023. Coupling symbolic reasoning with language modeling for efficient longitudinal understanding of unstructured electronic medical records. *arXiv preprint arXiv:2308.03360*.

Shi, Y.-Z.; Hou, H.; Bi, Z.; Meng, F.; Wei, X.; Ruan, L.; and Wang, Q. 2024. AutoDSL: Automated domain-specific language design for structural representation of procedures with constraints. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12177–12214.

Tan, X.; Deng, Y.; Qiu, X.; Xu, W.; Qu, C.; Chu, W.; Xu, Y.; and Qi, Y. 2024. Thought-Like-Pro: Enhancing Reasoning of Large Language Models through Self-Driven Prolog-based Chain-of-Thought. *arXiv preprint arXiv:2407.14562*.

Tan, X.; Valentino, M.; Akhter, M.; Liakata, M.; and Aletras, N. 2025. Enhancing Logical Reasoning in Language Models via Symbolically-Guided Monte Carlo Process Supervision. *arXiv preprint arXiv:2505.20415*.

Wang, Y.; Ye, W.; Guo, P.; He, Y.; Wang, Z.; Tian, B.; He, S.; Sun, G.; Shen, Z.; Chen, S.; et al. 2025. SymRTLO: Enhancing RTL Code Optimization with LLMs and Neuron-Inspired Symbolic Reasoning. *arXiv preprint arXiv:2504.10369*.

Wang, Z.; Liu, J.; Bao, Q.; Rong, H.; and Zhang, J. 2024. Chatlogic: Integrating logic programming with large language models for multi-step reasoning. In *2024 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Wu, J.; Feng, M.; Zhang, S.; Che, F.; Wen, Z.; Liao, C.; and Tao, J. 2024. Beyond examples: High-level automated reasoning paradigm in in-context learning via mcts. *arXiv preprint arXiv:2411.18478*.

Wu, Y.; Zhou, X.; Humayun, A.; Gulzar, M. A.; and Kim, M. 2025. Generating and Understanding Tests via Path-Aware Symbolic Execution with LLMs. *arXiv preprint arXiv:2506.19287*.

Xu, F.; Sun, Q.; Cheng, K.; Liu, J.; Qiao, Y.; and Wu, Z. 2024a. Interactive evolution: A neural-symbolic self-training framework for large language models. *arXiv preprint arXiv:2406.11736*.

Xu, F.; Wu, Z.; Sun, Q.; Ren, S.; Yuan, F.; Yuan, S.; Lin, Q.; Qiao, Y.; and Liu, J. 2024b. Symbol-LLM: Towards Foundational Symbol-centric Interface For Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 13091–13116.

Xu, J.; Fei, H.; Pan, L.; Liu, Q.; Lee, M.-L.; and Hsu, W. 2024c. Faithful Logical Reasoning via Symbolic Chain-of-Thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 13326–13365.

Yang, K.; Swope, A.; Gu, A.; Chalamala, R.; Song, P.; Yu, S.; Godil, S.; Prenger, R. J.; and Anandkumar, A. 2023. Leandojo: Theorem proving with retrieval-augmented language models. *Advances in Neural Information Processing Systems*, 36: 21573–21612.

Yang, S.; Li, X.; Cui, L.; Bing, L.; and Lam, W. 2025. Neuro-Symbolic Integration Brings Causal and Reliable Reasoning Proofs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 5732–5744.

Yang, Y.; Xiong, S.; Payani, A.; Shareghi, E.; and Fekri, F. 2024. Harnessing the Power of Large Language Models for Natural Language to First-Order Logic Translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6942–6959.

Yang, Z.; Ishay, A.; and Lee, J. 2023. Coupling Large Language Models with Logic Programming for Robust and General Reasoning from Text. In *Findings of the Association for Computational Linguistics: ACL 2023*, 5186–5219.

Ye, X.; Chen, Q.; Dillig, I.; and Durrett, G. 2023. Satlm: Satisfiability-aided language models using declarative prompting. *Advances in Neural Information Processing Systems*, 36: 45548–45580.

Zelikman, E.; Huang, Q.; Poesia, G.; Goodman, N.; and Haber, N. 2023. Parsel: Algorithmic Reasoning with Language Models by Composing Decompositions. *Advances in Neural Information Processing Systems*, 36: 31466–31523.

Zhang, H.; Huang, J.; Li, Z.; Naik, M.; and Xing, E. 2023a. Improved Logical Reasoning of Language Models via Differentiable Symbolic Programming. In *Findings of the Association for Computational Linguistics: ACL 2023*, 3062–3077.

Zhang, H.; Li, J.; Wang, Y.; and Song, Y. 2023b. Integrating automated knowledge extraction with large language models for explainable medical decision-making. In *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 1710–1717. IEEE.

Zhang, T.; Ge, J.; Luo, H.; Chuang, Y.-S.; Gao, M.; Gong, Y.; Kim, Y.; Wu, X.; Meng, H.; and Glass, J. 2024. Natural Language Embedded Programs for Hybrid Language Symbolic Reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, 4131–4155.

Zhou, J. P.; Staats, C. E.; Li, W.; Szegedy, C.; Weinberger, K. Q.; and Wu, Y. 2024. Don't Trust: Verify – Grounding LLM Quantitative Reasoning with Autoformalization. In *The Twelfth International Conference on Learning Representations*.

Zhu, W. B.; Singh, I.; Jia, R.; and Thomason, J. 2025. Language Models Can Infer Action Semantics for Symbolic Planners from Environment Feedback. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 8751–8773.