Automating Reasoning in Chemical Science and Engineering

Tyler R. Josephson

University of Maryland, Baltimore County

Abstract

Step-by-step thinking is essential in all domains of chemical sciences and engineering. While machine learning tools are broadly used, algorithms that automate reasoning are far less common. We elaborate on seven categories of human reasoning activities and connect each to applications in chemical science and engineering: inductive, deductive, abductive, probabilistic, causal, analogical, and informal reasoning. For each, we describe approaches and tools for implementing in computational settings. We find the traditional categories of logic (induction, deduction, abduction) to be useful for interpreting a wide range of research activities in the computational sciences: curve fitting and supervised learning are induction, predicting from a curve or a supervised model is deduction, first-principles simulations are deduction, molecular structure elucidation is abduction, and inverse design is isomorphic to abduction. Experimental design incorporates themes of all three, but also stands apart because it modifies the data. We also survey the "reasoning" capabilities of large language models, and illustrate some challenges and opportunities in building systems that learn to reason. Central concepts that emerge include the dual nature of reasoning as propositional and step-by-step (the "what" and the "how"), the importance of appreciating syntax and semantics, and the centrality of abstraction in formal and informal contexts. Throughout, we highlight untapped opportunities, including bug-free scientific computing software, predictive modeling approaches to generalize outside training data, inverse design, and automated hypothesis generation and evaluation.

1 Introduction

Machines have been emulating scientists since the dawn of artificial intelligence. DENDRAL, the first expert system, was designed for analyzing mass spectra (1). BACON automated the discovery of scientific equations from data (2). "Adam" automated hypothesis formation and testing via robotic experiments in yeast genetics (3). As capabilities for AI have grown, a grand challenge

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

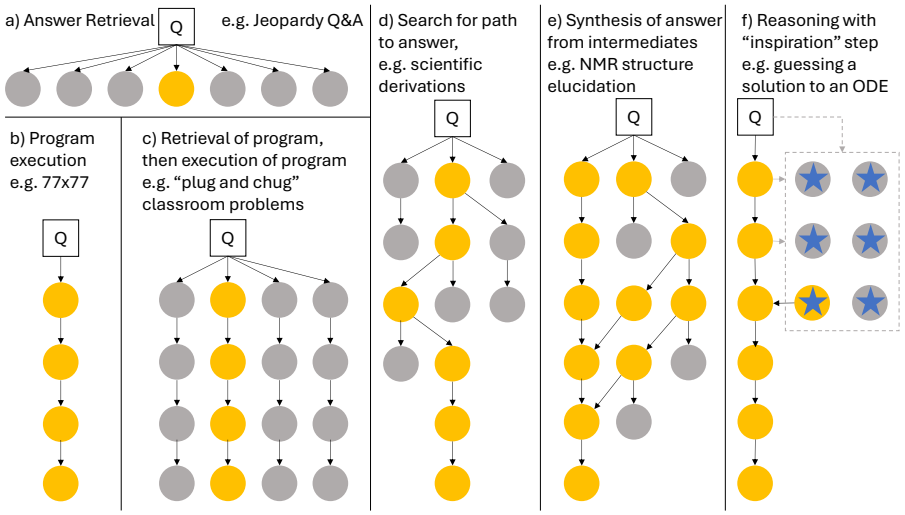

Figure 1: Types of different reasoning chains; yellow circles indicate path to final answer, while gray circles indicate unneeded steps or dead ends (these imply backtracking). Shallow problems only require retrieval of correct answer among many candidates. Deep problems can be narrow, only requiring deterministic program execution, or incorporate different elements of width, whether by retrieving the right narrow program to execute, searching for a path to an answer, or searching for multiple pieces of intermediate information needed to synthesize of final answer. The role of inspiration is roughly illustrated as a need to "come up with" ideas, one of which is critical for continuing the chain, and that these are somehow "inspired" by the context of the problem and solution thus far.

has been proposed – could an AI make scientific discoveries worthy of a Nobel Prize (4; 5)? While the 2024 Nobel Prizes recognized the impact of artificial neural networks and AlphaFold (6) as *tools* enabling humans to advance science, fully autonomous discovery of such impact is beyond current systems (7). Typically, computer algorithms acting as "artificial scientists" suggest experiments, generate hypotheses, and discover properties, but still require "human input, interpretation, and integration with existing knowledge." (8).

Nonetheless, in arena after arena, as machines' capabilities increase, humans have adopted them in ever-increasing ways. Machines have long bested humans at memory and arithmetic (which is why we store data on hard drives and use calculators to calculate tips). Competitive games like chess (9), Jeopardy (10), Go (11), StarCraft (12), and Diplomacy (13) have seen machines beating or competitive against top humans. Each milestone was made possible by smarter algorithms and more powerful hardware. The latest AI systems use large languages models (LLMs) to excel over many (but not top) humans at standardized tests like the LSAT exam (14; 15) and the International Math Olympiad (16; 17; 18).

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

What underscores the more challenging milestones is the role of reasoning in complex problem solving. Reasoning can be thought of as thinking through a step-by-step path from what is known to what is unknown, resulting in a reasoning chain with some amount of "depth." Jeopardy questions are generally "shallow," involving memorization and retrieval of facts, while chess is "deep" (can you think 4 moves ahead?). Fig. 1 illustrates the diversity of chains involved in solving different kinds of problems. Executing a program (Fig. 1b) is a primitive form of reasoning (multiplying large numbers requires a deep path, but applying the steps is monotonous and repetitive). Some problems may involve identifying a program that's suitable for the question at hand, and then simply applying it (Fig. 1c) – an example would be "plug and chug" problems in engineering education that require students to match the context of the problem to an equation from the textbook before substituting values and solving for the unknown. Some scientific derivations are examples of Fig. 1d, where the path forward might not be clear at the start, but trying different steps and backtracking can illuminate the path to the solution. Solving NMR spectra is an example of Fig. 1e, where one gleans bits of insight by studying each group of peaks, then incrementally consolidates that information until the solution can be constructed. Finally, "inspiration" can play a role, as well (Fig. 1f). Consider how one solves certain differential equations – an important step might be "guess that the solution has the form $y = A\sin(x) + B\cos(x)$." Whether the guess is useful will be verified later, but *where* the guess comes from can be mysterious; it's often drawn from some combination of experience, appreciating the context of the problem, and learning from earlier unsuccessful attempts.

Reasoning is also compositional, in that a reasoning chain may itself contain reasoning chains. This helps to clarify the role of external tool use in problem solving. The query, "What is the vapor pressure of n-hexane at 78 °C?" illustrates this in Fig. 2. Neither shallow retrieval (Fig. 1a) nor simple program execution (Fig. 1b) can solve this problem; the intrinsic information processing required is simply too complicated. Nonetheless, along the reasoning chain, smaller sub-problems arise that can be solved in these ways. Thus a human may use search tools and a calculator along their path to the final answer – these are specialized tools with "narrow" intelligence. The "extended mind thesis" (19) is a neat concept to consider here. With a tool capable of generalized problem solving, the user-facing complexity is completely abstracted away. Nonetheless, the intrinsic information processing required for this question remains; in the case of an LLM solving this using chain of thought, this processing occurs in pre-training and while generating the chain of thought (see Section 3.3).

Shallowness and depth do not necessarily correlate with problem difficulty. Multiplying two eight-digit numbers is a very deep problem, made trivial by calculators. Some shallow questions are "hard" in the sense that, most people wouldn't get it correct on a standardized test, because most people wouldn't have it memorized, but let them use a search engine, and they'll find it much easier. Benchmarks like Google-Proof Q&A (20) acknowledge this, where human performance is measured in a context where they have a long time to solve the problem and access to Google.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

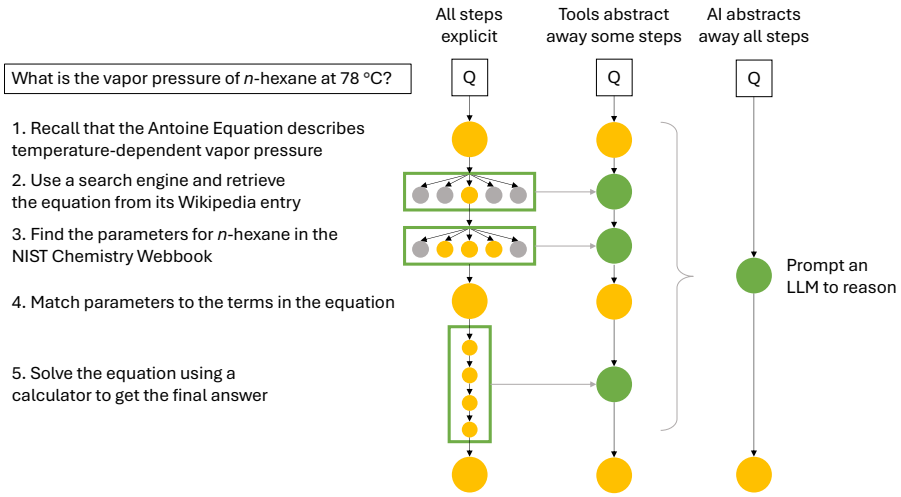

Figure 2: Composition and modularity in a reasoning chain. The chain on the left represents "all the information processing involved in solving the problem," with steps handled by specialized, external tools shown in green boxes. The center chain abstracts away the details of the external tools, and evokes how a human might solve the problem with Internet access and a calculator. In the chain on the right, the human uses an AI tool that abstracts away all the steps and delivers the answer (which may or may not explain the steps it took).

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

"Machine learning" (ML) is often conflated with "artificial intelligence" (AI). While definitions and are difficult to pin down and vary among experts (21; 22) and throughout history (23), here, we define ML as the class of algorithms that improve automatically through experience, without being explicitly programmed to do so, while AI is the broader category study of machines mimicking "cognitive" functions that humans associate with the mind. ML is essentially about identifying and memorizing patterns in data. Predictions about new observations are based on connecting new observations to patterns in prior experience (the training data). In particular, AI includes reasoning, a collection of functions associated with step by step thinking. Planning, logic, decision-making, verification, and mathematical problem solving all involve reasoning.[1]

Daniel Kahneman popularized a 2-system theory of human decision-making in his book "Thinking, fast and slow" (26). Humans use System 1 for rapid and imprecise pattern recognition in intuitive and reflexive tasks, and System 2 for conscious, deliberative tasks like reasoning and planning. To illustrate, consider multiplication. Many of us memorized "times tables" as a child. We can quickly answer "7 x 7 = 49" because the answer is memorized. But for 77x77, we must switch from recall to reasoning, break the problem down into pieces, solve them one by one, and synthesize the answer. In chemistry students, "Chemical intuition" and heuristic reasoning have been interpreted in light of System 1 and System 2 (27).

Recently, AI researchers have been explicitly aspiring to build machines that "think fast and slow" (28; 25). Sometimes these are separate computational modules with different functions (these systems are often categorized as "neurosymbolic AI") (29) and sometimes in a single integrated system, such as large language models (LLMs) optimized to think step by step (See Section 3.3).

Machine learning has exploded in the chemical sciences. The primary driver for adoption of these pattern-recognition algorithms is that they prove to be profoundly useful for analyzing high-dimensional datasets in the chemical sciences and engineering. Nonetheless, pattern recognition alone isn't enough to solve the hardest challenges we face; memorization is not enough. Reasoning greatly expands the range of problems we can solve. The future of artificial intelligence in the chemical sciences will see increased adoption of algorithms that go beyond basic pattern matching (29; 30).

This perspective aims to provide an accessible introduction to reasoning in logic and computer science for chemical scientists and engineers. We highlight the importance of these in engineering problem solving, highlight historical and modern developments in the AI community around automated reasoning, identify recent work developing and applying these tools for problem solving in science and engineering, and suggest opportunities for the future. Section 2 covers the central categories of formal reasoning (induction, deduction, and

[1] In humans, learning and reasoning are intertwined (24), after all, humans improve their decision-making, planning, and math abilities with practice and experience. In artificial systems, these functions have typically been distinct; for example, image recognition uses purely ML algorithms while math calculations are executed using pre-programmed calculators. Merging these is an active area of research (25). LLMs apparently do this, as well (Section 3.3).

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

abduction), as well as probabilistic and causal reasoning. Section 3 covers aspects of informal reasoning relevant to chemical science, especially analogical reasoning and chain-of-thought "reasoning" by LLMs. Finally, Section 4 discusses broader perspectives on reasoning as "being" vs. "doing", syntax and semantics, and the role of abstraction.

2 Formal reasoning

Induction and deduction are two kinds of logic that have been studied for millennia. The third pillar of reasoning, abduction, was formulated in the 20th century by chemistry and philosopher Charles Peirce. Such reasoning approaches are described as *formal* when they can be expressed in precise mathematical terms. Humans also engage in informal reasoning; analogical reasoning, spatiotemporal reasoning, and commonsense reasoning also involve step-by-step thinking, but are usually not expressed precisely.

2.1 Inductive reasoning

Inductive reasoning starts from specific observations and reasons toward general conclusions. "Swan #1 is white" & "Swan #2 is white" & "Swan #3 is white" $\rightarrow$ "All swans are white." Induction is unreliable, in that even when the premises are true, the conclusions are only probabilistically true. After all, no number of observed white swans preclude the existence of a black swan. "The sun will rise tomorrow" is an inductive conclusion born from observations (the sun rose today, yesterday, etc.). In fact, every statement about the future is an inductive leap; David Hume famously showed that such conclusions rely on a hidden assumption (the world will be the same tomorrow as it was today) that itself can only be proved using inductive reasoning about the future, which becomes a circular argument (31).

Nonetheless, induction is critical for everyday life, and indeed serves as the foundation for all statistical learning algorithms. Consider curve fitting (Fig. 3): when observations in the training set are used to fit a model, one is obtaining a general rule (the model parameters) from specific observations. This rule is henceforth used to make predictions about unseen data in the test set (this is deduction, discussed shortly). The general rule only approximately follows from the observations – this logical framework is the foundation of all supervised learning algorithms that fit a model to data (32).

Some AI systems prior to the 2000s incorporated algorithms for induction, in order to teach rules to expert systems with less human expert support. Meta-DENDRAL was an early example of this, which used mass spectrometry data to reverse-engineer rules for DENDRAL to use in structure elucidation (1). Another example included automated construction of rules for an expert system for HPLC column selection in enantioseparations (33).

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

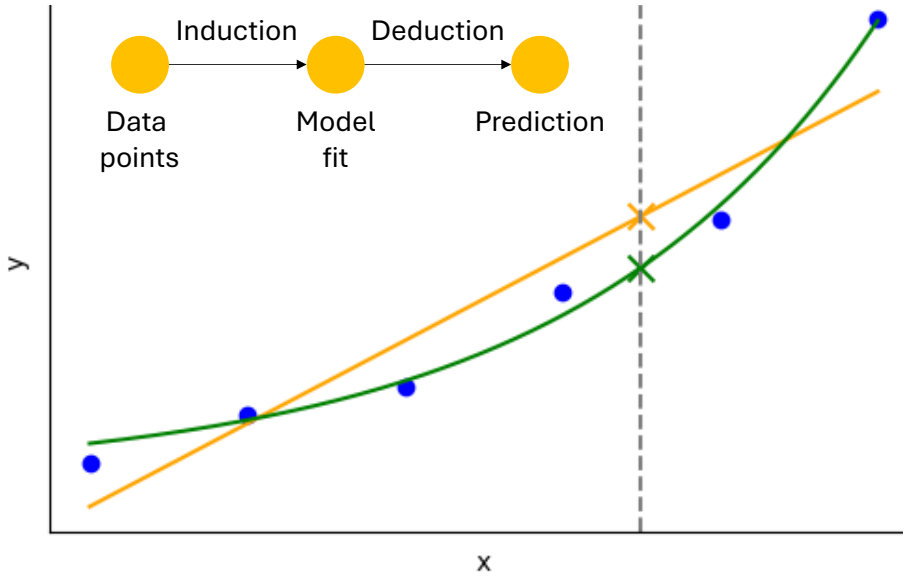

Figure 3: Fitting models (in this case, linear and exponential functions) to data is inductive reasoning (form a general conclusion from specific observations), while making model predictions (marked by x) is deductive reasoning (use a general principle to derive a specific conclusion).

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

2.2 Deductive reasoning

Deductive reasoning starts with general premises (assumed to be true) and proves that specific conclusions follow. "All men are mortal" & "Socrates is a man" imply "Socrates is mortal." Deduction is a reliable logical framework, in the sense that, if the premises are true, then the conclusions are guaranteed to be true, as well. More precisely, arguments using correct deduction are always valid (the step-by-step reasoning process is correct), but they are only sound when the premises are true. Sound arguments reason correctly from true premises, and always produce true conclusions.

Mathematics is fundamentally deductive. After assuming a foundation of mathematical axioms (axiomatic systems include set theory (34) and type theory (35)), repeated application of deduction enables construction of an ever-growing edifice of mathematical knowledge. The question of "are the premises true in the real world?" doesn't typically concern a pure mathematician. Part of the beauty of this "world of math" is that deductions are proved valid without concern for their soundness; mathematical progress continues independently of whether new math is immediately useful.

In this way, every derivation in science and engineering can be seen as a logical deduction. One assumes the premises of some physical model, then proceeds through various transformations of the equations until one arrives at the conclusion. Fig. 4 illustrates this for Langmuir's theory of adsorption (36). The isotherm equation is rigorously deduced from the model assumptions. When the isotherm equation doesn't describe a real-world system very well, we don't question the derivation (which is logically valid so long as the math is correct) – instead, we question whether the premises of the model adequately represent the real-world system.[2]

Planning is a sub-field of deductive reasoning, concerning actions, time, and events. "If this valve in a chemical plant is closed at time t, then X, Y, and Z will occur" or "If precursors A and B react under these conditions, then product C will be formed." A network with dozens or hundreds of steps can be reasoned over to make predictions about cause and effect. Such networks can also be the subject of optimization, to improve process safety, reliability, or profitability. In fact, "expert systems" (the predominant form of "artificial intelligence" from the 1970s to the early 2000s), use logical deduction to reason about a knowledge base of facts to derive new facts or evaluate hypotheses.

2.2.1 Expressivity in deduction

Deduction is categorized into different kinds of logic based on the complexity and expressivity of the statements they reason about. This drives the availability and effectiveness of computational tools, so here we summarize a few important kinds of logic and the computational tools for solving them.

[2]This is about semantics (which refers to the meaning behind the scientific model), which is distinct from the question of logical syntax.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

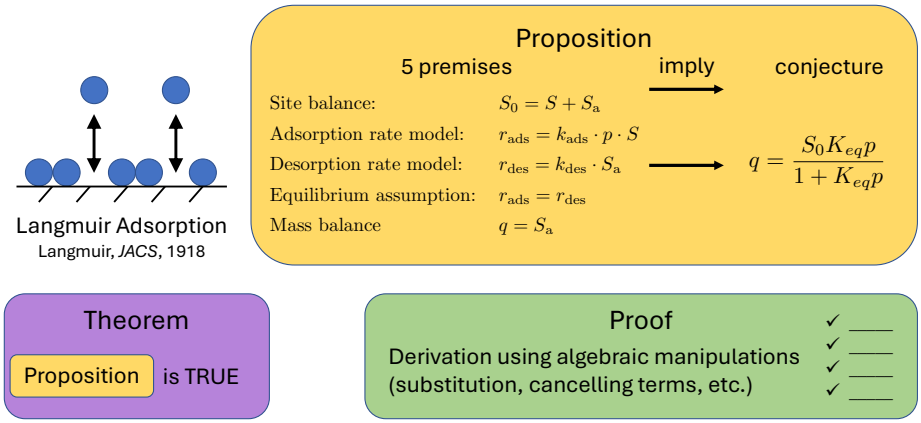

Figure 4: Derivations in science and engineering are isomorphic to theorems in mathematics, as illustrated by Langmuir's theory of adsorption (37). Such derivations can be formalized in a theorem proving system (36).

Propositional logic is the simplest[3], using only the logical operators AND, OR, NOT, IF ... THEN, and IF AND ONLY IF. First-order logic (FOL) includes these operators, and adds FOR ALL and THERE EXISTS operators, as well as the equality relation and functions that can return true or false depending on the properties of entities. For example, imagine a function ContainsHydrogen(molecule) that returns FALSE for CO_2 and TRUE for H_2O – this function can be expressed in FOL, but not in propositional logic. Higher-order logics like set theory and type theory generalize further, and can be used to define and reason about real numbers (like π and $\sqrt{2}$), limits and calculus, infinity, etc.

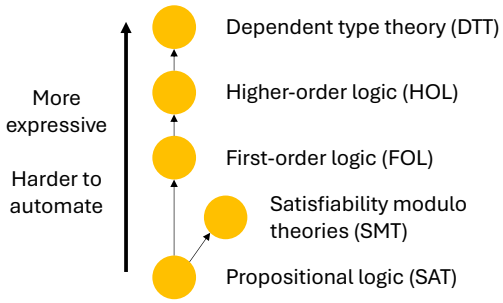

Figure 5: Hierarchy of logics and their solvers.

More expressive logics enable more compact representations of scientific knowledge. For example, a first-order logic term ContainsH(molecule) can generalize an infinite number of statements in propositional logic: "H_2O contains hydrogen," "n-octane contains hydrogen," "ibuprofen contains hydrogen," etc. This compactness is critical for representing abstract ideas, but this generality comes at a cost: it affects the kinds of computational tools that can be leveraged to solve such problems and the efficiency of automated search for solutions (Fig. 5).

[3]Even simpler logics can be constructed, for example, by restricting propositional logic to simpler classes of clauses like Horn clauses, but these are typically only of academic interest.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

SAT solvers address the Boolean satisfiability problem (SAT) in propositional logic, and aim to determine whether these exists an assignment of truth values (true or false) to variables in a propositional logic formula. While SAT problems in general have been proven to be NP-complete (the computational cost for solving a problem scales exponentially with problem size), commercial and open-source SAT solvers can efficiently solve many practical SAT problems. SMT solvers (Satisfiability Modulo Theories) expand SAT solvers to address problems somewhat between propositional and first-order logic, adding theories, like arithmetic and arrays, to propositional logic. SMT solvers like Z3 (38) are capable of expressing a function CountHydrogens(molecule) that returns an Integer (not a Boolean true/false), and representing arrays as variables, which are not single entities. These still operate efficiently, by reducing the problem to SAT at the core and relying on efficient SAT solving techniques.

SMT solvers approach first-order logic, but they cannot express arbitrary first-order logic statements, such as reasoning about "there exists" over arbitrary sets, not just Integers. Solvers are available for full first-order logic (FOL solvers), but this crosses a theoretical threshold: SAT is decidable while full first-order logic is undecidable. This means that we have a method to solve every SAT / SMT problem; even if the computational cost is exponential, the procedure will eventually finish in finite time. FOL is undecidable, meaning no single algorithm is capable of solving every FOL problem (39; 40). FOL solvers employ algorithms and heuristics that automatically solve many problems, but their success is not guaranteed. Moving to higher-order logics adds further expressivity and complexity; tools like Isabelle (higher-order logic, HOL) and Rocq / Lean (dependent type theory, DTT) can express and prove virtually any mathematical statement, but importantly, they don't operate automatically like SAT, SMT, and FOL solvers. Instead, these are interactive theorem provers that primarily facilitate verification of proofs written by humans (or coding LLMs (16; 41)). Even though higher-order logics are undecidable, this doesn't mean proofs aren't possible, only that no single algorithm can solve every statement.

Overall, we recommend weighing these considerations when formulating the logic for your scientific problem. If you can represent all the important facets of your domain of study using exclusively propositional logic, then you'll benefit from efficient SAT solvers. If you require first-order logic, then an interactive theorem prover like Lean might be overkill in terms of expressivity, with a comparative loss in automated solving capability compared to a focused FOL solver like Vampire (43). In certain domains, specialized tools have been developed with built-in heuristics to facilitate problems in that domain. For example, KeYmaera X (44) is a FOL solver optimized for symbolically reasoning about cyber-physical systems characterized by ODEs, including chemical reactors (45). That said, higher-order logics subsume lower-order logics; FOL solver Vampire also solves SMT and SAT problems automatically, and Isabelle has the Sledgehammer tactic (46) which attempts to convert a type theory problem into a first-order logic problem, then solve it automatically using an external, automated FOL solver. We are personally invested in learning Lean 4 (47), the newest of the tools mentioned thus far, which maximizes expressivity using de-

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

Formalized derivation of Langmuir adsorption

```
import Mathlib

-- Name the theorem and define the variable types as Real
theorem LangmuirAdsorption {θ K P r_ad r_d k_ad k_d A S_tot S : ℝ}
  -- Premises
  (h1 : r_ad = k_ad * P * S)  -- Adsorption rate expression
  (h2 : r_d = k_d * A)        -- Desorption rate expression
  (h3 : r_ad = r_d)           -- Equilibrium assumption
  (h4 : K = k_ad / k_d)       -- Definition of adsorption constant
  (h5 : S_tot = S + A)        -- Site balance
  (h6 : θ = A / S_tot)        -- Definition of fractional coverage
  (h7 : S + A ≠ 0)            -- Constraint to avoid division by zero
  (h8 : k_d + k_ad * P ≠ 0)   -- Constraint to avoid division by zero
  (h9 : k_d ≠ 0) :            -- Constraint to avoid division by zero
  -- Conjecture
  θ = K * P / (1 + K*P) := by -- Langmuir adsorption equation
  -- Proof starts here
  -- Goal: h1-h9 --> θ = K * P / (1 + K * P)
  -- Perform substitutions using the rw tactic
  rw [h1, h2] at h3
  rw [h6, h5, h4]
  -- Goal: h1-h9 --> A / (S + A) = k_ad / k_d * P / (1 + k_ad / k_d * P)
  -- Remove denominators and simplify using field_simp tactic
  field_simp
  -- Goal: h1-h9 --> A * (k_d + k_ad * P) = k_ad * P * (S + A)
  -- Use a 'calculation' block to close goal with ring and rw tactics
  calc
    A * (k_d + k_ad * P) = k_d * A + k_ad * P * A := by ring
    _ = k_ad * P * S + k_ad * P * A := by rw[h3]
    _ = k_ad * P * (S + A) := by ring
```

Figure 6: Formal proof of Langmuir's theory of adsorption in Lean 4, updated from the Lean 3 proof in (36) and a Lean 4 proof in (42). Comments (in green) annotate the premises, conjecture, and proof steps. Nine premises (h1–h9) imply the conjecture ($\theta = K * p/(1 + K * P)$). or here for a stable version.

pendent type theory while facilitating low-level automation via tactics. Fig. 6 shows how Langmuir's theory (Fig. 4) can be derived in Lean 4.

For example, the recent work by Clymo, et al. used an SMT solver (Z3, (38)) to navigate expert-specified constraints on the space of possible material compositions (48). The authors chose to incorporate constraints like charge neutrality, which can be expressed in linear arithmetic (and thus accommodated efficiently by Z3). But they were selective about the constraints they chose, either because Z3 cannot support them, or because Z3 supports them but their inclusion would dramatically affect efficiency. They describe their reasoning (48):

> "There are other constraints that might be desirable but cannot be expressed in this theory. For example, we could not constrain the variance of ionic radii using linear constraints, since calculat-

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

ing the variance requires the non-linear square-root function. By restricting attention to queries that can be expressed as a combination of linear constraints we have balanced flexibility and solving time. It would be possible to include more constraints by using a more expressive theory, as the SMT solver Z3 we have used supports non-linear arithmetic of different types, including polynomial real arithmetic; however this would substantially increase solving time in a problem-dependent manner that is hard to predict."

2.2.2 Deduction, predictive modeling, and statistical learning

Applications in predictive modeling have been the mainstay of applications of machine learning in the chemical sciences. Neural network- and Gaussian process-based tools are powerful and flexible predictors when high quality "big data" is available, but often, chemical data acquisition is the bottleneck (49), whether synthesis (50), characterization (51), or performance evaluation (52). Machine learning algorithms learn to interpolate within the convex hull of the data, and are unreliable for extrapolation (albeit, for high-dimensional deep learning systems, the story is more nuanced (53)). However, logical deduction is not limited by the boundaries of the data in the same way that typical machine learning approaches are. The boundaries of deduction are given by the "deductive closure," which refers to the set of all possible statements (which can often be infinite) that can be deduced from a given set of assumptions.

For example, machine-learned interatomic potentials (MLIPs) perform best near their training data, and are generally not trusted to extrapolate (54; 55; 56). An MLIP trained on molecules containing only H, C, N, and O atoms cannot be trusted for compounds with sulfur or lead (or even HCNO molecules with out-of-distribution geometries). Researchers have developed approaches to dynamically expand the training data (e.g. (57; 58; 59)) and have sought to build ML potentials with training data from the whole periodic table (e.g. (60)). Such strategies arise from a paradigm of *data-driven interpolation*, where the problem is seen as a gap in the data, and the solution is to obtain such data and fit a new model that doesn't have such a gap. Essentially, one aims to extend the training domain so nothing is out-of-distribution. One can also develop ML architectures or feature representations that use the data more efficiently, but the underlying paradigm is the same – design the domain so the anticipated use case is in-distribution.

Deduction does not depend on data in this way. Consider electronic structure calculations as a form of deduction (Fig. 7). To compute an energy E, one assumes A) a geometry of atoms, B) the principles of quantum mechanics, especially Schrödinger's equation, and C) various approximations that make this theory computable and efficient (for instance, those behind density functional theory, and selection of a particular model and basis set). The energy of the system follows from these assumptions, or in other words, A & B & C $\rightarrow$ E. Given A, B, and C, the "deductive closure" here would include the results of quantum chemical calculations for systems across the whole periodic table, at any level of

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

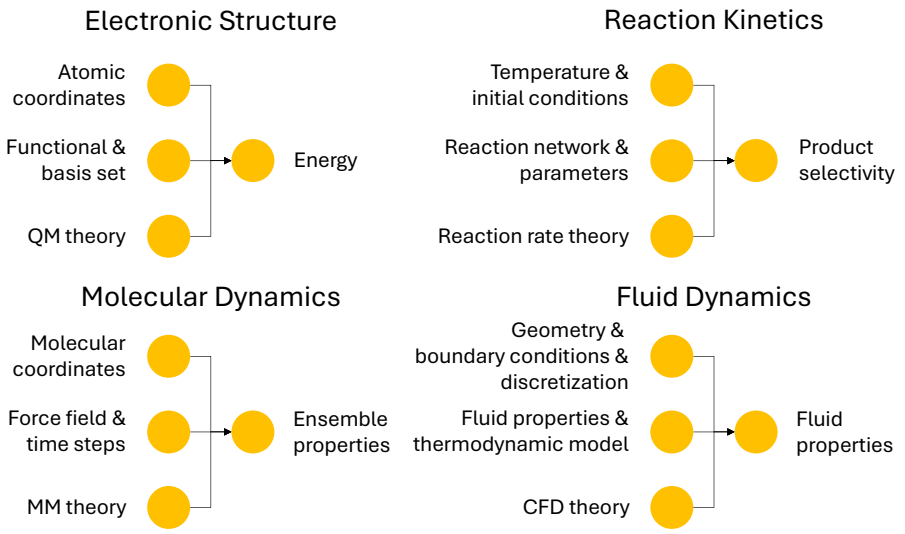

Figure 7: Predictive modeling as deduction. First-principles modeling and simulations at any scale can be seen as performing a form of deduction, in which the inputs & the theory "imply" the outputs.

theory encompassed in B & C.[4] Furthermore, A & B & C can also imply many things beyond an energy, including forces, frequencies, band gaps, etc. – these are in the "deductive closure" if they can be derived from the assumptions.

The deductive approach does have gaps, but of a different kind – gaps lie in either the *theory* or in the *algorithms*. A gap in theory means that the deductive closure for A & B & C does not contain the answer; an gap in the algorithms means, given A & B & C, obtaining E is not efficiently computable. An example of a theory gap would be relativistic effects – special techniques are required for modeling for heavy atoms (61); A & B & C do not imply accurate values for E when the theory is incomplete. Algorithmic gaps in the deductive approach manifest in the poor computational scaling and stability of quantum chemical calculations. One can define geometries and theory levels for which the electronic structure doesn't converge in a reasonable amount of time

[4]More precisely, A, B, and C are each sets (of geometries, and of propositions about quantum mechanics) and E would be a set of Real-valued conclusions. The "size" of this deductive closure would be the number of valid deductions to members of E, or the size of E. Certain geometry/theory combinations within the set that do not lead to a valid deduction are not included in the deductive closure. If the sets of geometries and QM models are countable, then E is countable, and we could potentially discuss the "size" of the deductive closure, like "the deductive closure for this set of geometries/theories is smaller than the deductive closure for this other set of geometries/theories." If any of the sets A, B, or C have real-valued members (e.g. if the set of atomic coordinates includes all possible positions, represented with infinite precision), then the deductive closure is uncountably infinite, raising tricky questions around cardinality. But in practice, computational chemists work with finite-precision floating-point numbers, which would always lead to countable deductive closures.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

or at all; this illustrates an algorithmic gap that can be addressed by better task-specific algorithms. But since the theory here is far more expressive than even first-order logic (see Section 2.2.1), the gap cannot be closed in general, as the question is undecidable. MLIPs address both of these gaps. If theory is missing and data is available (e.g. experimental measurements of systems not yet described by known theories), such data can be added to the training set. Furthermore, the underlying architectures of MLIPs ensure scalability and efficiency in deployment – for many applications, this is the chief reason these are favored over expensive DFT calculations.

This perspective essentially characterizes any form of predictive modeling across fields of science (Fig. 7). A practitioner who sets up a simulation is implicitly setting up the premises of a logical deduction problem; the completed simulation reveals something that is implied by those premises. Purpose-built software for modeling atomistic-, molecular-, kinetic-, or fluid-scale systems constrain the scope of the deductive task and provide algorithms to navigate otherwise complex and expressive theories.[5] Theory-based deduction is the philosophical foundation behind modeling and simulation, and enables generalization to anything (computable) within the deductive closure. Data-driven machine learning enables generalization within the domain of available training data. Novel approaches integrating learning and reasoning may bring the best of both worlds; we think neurosymbolic approaches show the most promise (62; 29). Nonetheless, no current tools appear to perform logic with the expressivity needed to fully encompass scientific computing applications (calculus, matrices, etc.), likely because automating such higher-order logic is challenging, in the general case (Section 2.2.1).

2.2.3 Bug-free software through automated deduction

Another benefit of more deeply integrating formal deduction into scientific computing is the prospect of writing software with formal correctness guarantees. In the 1990s, a bug in the floating-point division (FDIV) assembly code in Pentium 4 processors prompted a multi-million dollar recall (63). A mistake in the equipment that etched arrays into a chip caused 5 bits to be mis-written, leading to occasional errors in FDIV operations, typically in the 9th or 10th decimal place (64). In the following years, Intel researchers developed methods to formally verify the arithmetic operations performed on their chips, ensuring that these issues never arise in future chip designs (65).

Methods for verifying software are more sophisticated now, and can be used to verify complex properties of programs. For example, Selsam, Liang, and Dill built a variational autoencoder (a machine learning model) using the Lean programming language and proved high-level properties about its statistical behavior (66). Our research group is also using Lean to build scientific computing

[5] Even probabilistic simulations, such as those using Monte Carlo or evolutionary algorithms, are a form of deduction (though of a probabilistic form, see Section 2.4). Though the outputs are not deterministic, the process from inputs to outputs is still about logical *implication*, with conclusions, such as ensemble properties, holding in expectation.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

software with mathematical correctness guarantees, for instance, for computing the energy of Lennard-Jones particles in periodic boundaries (67). By linking formalized derivations in science and engineering (e.g. Fig. 4) to executable code in the same software environment, we can prove that the executed code satisfies the theory. For instance, we can formally state a theorem, that the minimum image distance between two particles, anywhere in space, is independent of whether those particles are wrapped into the simulation box, and prove that our implemented code satisfies this high-level theorem.

Another application is safety verification in systems of ODEs. Simple systems of ODEs can be integrated analytically, to obtain closed-form expressions for their behavior. When symbolic integration isn't feasible, numerical integration can provide concrete predictions for individual scenarios, such as for a specific reaction network, with specific rate parameters, and specific input conditions. A third way is differential dynamic logic (68), in which one can reason directly about stability, equilibrium, and reachability *without* integrating the ODEs. Researchers in the cyber-physical systems field use such proofs in domains such as self-driving vehicles (69). Because the analysis is done symbolically and not numerically, such proofs are valid across general sets of parameters. Rose Bohrer's "Chemical Case Studies in KeYmaera X" is an interesting example of this in chemical reaction engineering (45). We also explored similar problems, studying isothermal batch and continuous-stirred tank reactors (70). We were able to prove properties about whether certain concentrations were reachable given ranges of input concentrations and system parameters. For example, for A→B in a batch reactor, we could prove that $B \leq A_0 + B_0$ for all time. We scaled up the reaction network complexity to Michaelis-Menten kinetics (where analytical integration is impossible without extra assumptions), and could prove some rather generous bounds on molecule concentrations. We ultimately found the logic style difficult to learn and explain, and the solver, KeYmaera X, to not be supportive of the mathematics we'd need for more interesting reaction chemistry (e.g. the exp function for Arrhenius temperature dependence). Nonetheless, further development of this approach would be interesting.

2.3 Abductive reasoning

Abduction is a lesser-known form of reasoning, originally proposed by Charles Peirce, a chemist and philosopher unsatisfied with the ability of induction and deduction to describe the kind of thinking that a scientist does. Abduction reasons from observations and background knowledge to explanations of phenomena; it has been described as "inference to the best explanation" (71). What Sherlock Holmes famously described as "deduction" more closely resembles abduction, as he worked backwards from effects to plausible causes (72).

Consider a forensic chemist who has characterized a mysterious white powder, observing A from NMR and B from IR. Her aim is to identify what molecule gives the best explanation of the data. She'll analyze the data, hypothesize that the substance is a chemical species (e.g. ibuprofen), then argue (verify) that

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

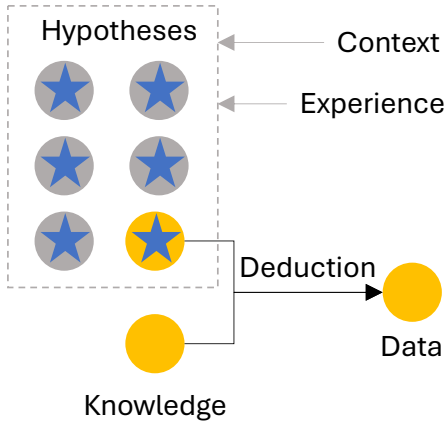

Figure 8: Abduction, also known as "inference to the best explanation," combines deduction with hypothesis formation. A hypothesis is plausible if it implies the observations.

this hypothesis is consistent with the data and standard background knowledge that dictates how this data should be interpreted. Or, she will find a mismatch between the hypothesis and the data, and try again.

Formally, abductive reasoning includes a bit of deduction, along with a unique aspect of hypothesis formation (Fig. 8).

1. D is a collection of data / observations
2. K is a collection of background knowledge (facts, givens)
3. H, a hypothesis, explains D in light of K ("explain" means a deduction to prove that H & K $\rightarrow$ D)
4. No other hypothesis explains D as well as H does
5. Therefore, H is probably correct.

In the above example, H would be "ibuprofen" and D would include the observations A and B from spectroscopy, as well as background chemistry knowledge (let's call this K) that includes insights like "a molecule containing C=O $\rightarrow$ observation of IR peak near 1700 cm^{-1}." If "ibuprofen" & K $\rightarrow$ A & B (and "other molecules" & K don't imply A & B), then ibuprofen is probably the detected compound; ibuprofen is the "best explanation" of the data in light of the knowledge. This problem setting is naturally multimodal; notice that A is from NMR and B is from IR. Moreover, multiple hypotheses can imply the same data (especially in the presence of noise); the data might not constrain the space of possible hypotheses to a single candidate, and indeed might constrain it too far, such that no hypothesis is supported. Jin, et al. helpfully describe

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

this essential feature as a "one-to-many, non-functional" mapping between data and predictions (73).[6]

While abduction includes deduction, it is a distinct kind of reasoning from deduction because it also requires effective hypothesis construction. Abduction doesn't prescribe *how* hypotheses should originate, though these are generally informed by prior experience, present context, and even iteration in light of previously-evaluated hypotheses. One could construct an abductive reasoning system by combining an independent "hypothesis formation" module with any of the deductive tools described in Section 2.2.1. The original expert system, DENDRAL (1), can be understood as performing abduction to solve mass spectroscopy problems. It integrated expert-crafted logical rules with CONGEN, a system for generating conformers (hypotheses), in a manner that integrated both rules and data.

2.3.1 Analogy with inverse design

In inverse design, one aims to generate a material with a target property. It is "inverse" because having a method for predicting the properties of a specific material is not enough, one needs to go "backwards" from the property to the material. Both Clymo, et al. (48) and Jin, et al. (74) identified a role for deduction to play in the context of inverse material design. Following their lead, we noticed an analogy between inverse design and abductive reasoning:

1. T is a target property (or properties)
2. K describes a known structure-property relationship (whether data-driven of from first principles)
3. C, a candidate, has property T in light of K (C & K $\rightarrow$ T)
4. No other candidate achieves property T as well as C does
5. Therefore, C is a suitable design.

This logical syntax is *equivalent* to that of abductive reasoning; they differ only in semantics. Indeed, Clymo, et al. named their system Comgen (because it generates *compositions* of materials); we couldn't help but notice an association with DENDRAL's CONGEN (which generated *conformers* of molecules). This also helps to crystallize the point that deduction *alone* won't suffice for certain applications in chemical science and engineering; when candidate generation or hypothesis formation is important, additional tools will need to be brought in.

2.3.2 Automating the scientific method

With its emphasis on hypothesis formation, abductive reasoning is closely related to the scientific method: hypothesize $\rightarrow$ experiment $\rightarrow$ analyze $\rightarrow$ support/refute hypothesis $\rightarrow$ repeat. Much work in the automation of science

[6] Though they, like Sherlock Holmes, refer to this abductive reasoning task as deduction.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

instantiates different versions of this, usually without reference to abductive reasoning *per se* (we noted it as future work in (75)). In traditional cases, humans craft a well-formed "space" of hypotheses (either implicitly or explicitly), which is navigated via automated experimentation and evaluation. For instance, the robot scientist "Adam" generated genomics hypotheses (in a symbolic, computable representation crafted by the human designers), then tested these hypotheses using automated laboratory instrumentation and analysis (3). In materials science, structured models for system behaviors can be designed by humans, in which adjustable parameters are tested and evaluated automatically (76). Likewise, reaction mechanism discovery can be formulated with human-crafted and validated building blocks, which are combined and remixed automatically to form hypotheses for testing and evaluation (77). In each of these systems, abduction (Fig. 8 plays a role for discriminating different hypotheses according to how well they explain the available data. The deduction step for evaluating K & H $\rightarrow$ D is typically execution / fitting of models (which is a form of deduction, see Section 2.2.2) to obtain scores for ranking hypotheses (abduction). However, note that the scientific method (automated or not) goes beyond abduction and includes an experimental design component: assessing the hypothesis landscape and collecting new data. As such, abduction *alone* is not sufficient in environments where data is not static. Experimental design goes beyond the traditional reasoning categories.[7]

Human scientists practicing the scientific method don't generally predefine a "space" of hypotheses amenable to automated testing, but construct free-form hypotheses in natural language. The capabilities of modern large language models have inspired researchers to task LLMs to generate research hypotheses in natural language (78; 79). However, even when efforts are taken to control the structure and quality of generated hypotheses, evaluation is challenging. Given a simple problem domain, hypotheses in the form of LLM-generated code can be executed on-the-fly (and simply rejected if code doesn't compute) (80). Given a fixed repository of curated data, automated agents capable of data analysis can evaluate natural language hypotheses about the content of the databases (78; 81). For applications in chemical and materials science, automated agents capable of seeking out new data are needed, especially tools for experimental synthesis, characterization, and performance evaluation (82; 83) or simulation setup and execution (84; 85; 86). All of these tools rely on LLM-based planning and reasoning, which we discuss further in Section 3.3.

To summarize, abduction as formalized by Peirce considers data to be a given, and aims for generation of hypotheses that can explain the data. It integrates deduction with hypothesis generation, with the goal of generating high-quality hypotheses (a.k.a. explanations). A single pass of the scientific method considers the hypotheses to be a given, and aims for generation of data

[7] We considered classifying experimental design as analogous to abduction, because it invokes hypothesis formation (which new data points to collect?) followed by deduction to predict the expected properties or information gain. But on further reflection, we've decided to emphasize the distinction that experimental design ultimately adds to the data, thus going beyond the traditional reasoning pillars, which work with the data available.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

that can best discriminate among the hypotheses. It integrates abduction with experimental design, with the goal of generating high-quality data. Recursive application of the scientific method leads to recursive improvements in data, hypotheses, and explanations.

2.4 Probabilistic reasoning

2.4.1 Deduction with uncertain premises

In practical settings with uncertainty, formal logic can be too rigid. One answer to this is probabilistic logic, in which statements are assigned truth values between 0 and 1, instead of binary false (0) and true (1). Thus, uncertainty propagates through deductive inference to obtain approximate conclusions. This is a bit like error propagation in engineering calculations, where rules for addition, multiplication, etc. are applied to obtain the uncertainty in a calculated output. Probabilistic reasoning applies this idea to logical operations, such as AND, OR, and IMPLIES.

Consider a chemical reactor that shuts down if the temperature is too high. In case a faulty sensor fails to report a high temperature, two redundant sensors have been installed, and emergency shutoff commences if sensor A or sensor B is triggered. A classical logic problem could be set up with three premises:

1. $T > T_{\text{alarm}} \rightarrow$ Sensor A
2. $T > T_{alarm} \rightarrow$ Sensor B
3. Sensor A OR Sensor B $\rightarrow$ shutoff.

With binary truth values, the statement "$T > T_{\text{alarm}} \rightarrow$ shutoff" follows from 1, 2, & 3 above (indeed, it also follows from 1 & 3 only, and from 2 & 3, only). Probabilistic logic enables one to incorporate probabilities on truth values, such as knowledge from the manufacturer about each sensor's false positive and false negative rates. If the sensors fail sometimes (1 and 2 have probability 0.99), the likelihood that "$T > T_{\text{alarm}} \rightarrow$ shutoff" is between 0.99 and 0.9999, depending on how independent the sensor errors are. If the shutoff condition (premise 3) is switched from OR to AND, the likelihood of shutoff becomes between 0.98 and 0.99, depending on sensor independence. Logical tasks for this scenario include 1) working forward through the implication to determine the likelihood of triggering a shutdown (deduction), 2) determining the likelihood of different hypothesized scenarios in the event of a shutdown (abduction), and 3) fitting unknown probabilities to data (induction). Tools for probabilistic logic include ProbLog (which extends the Prolog logic programming language) (87) and Markov Logic Networks (88).

2.4.2 Causal reasoning

A special case of probabilistic reasoning is causal reasoning, the study of cause and effect (89; 90; 91). This uses elements of probabilistic forms of induc-

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

tion, deduction, and abduction (applying these to causes and effects), and also includes counter-factual reasoning ("What If?" scenarios) and interventions.

Causal reasoning is especially prominent in the biomedical field and in epidemiology (92), where statements like "smoking causes cancer" need to be evaluated. Consider one patient, Sam, who smokes. Induction would be inferring a rule, "smoking causes cancer," from statistical data about patients over time. Deduction would be following a causal chain of events to determine the likelihood of Sam getting cancer as an effect of their behavior. Abduction would be explaining an observed effect (Sam gets cancer) by hypothesizing the causes of that effect and showing the effect can be deduced from those causes. Counterfactual reasoning would be evaluating the predictions of a causal model when Sam does not smoke ("What if Sam hadn't smoked?").

Methods for modeling causal reasoning include structural equation modeling (93), Bayesian causal networks (94), and directed acyclic graphs (89). Causal modeling has contributed to fault detection and diagnosis in chemical processes (95; 96), as well as in drug discovery (97).

3 Informal reasoning

For our purposes, informal reasoning describes step-by-step thinking processes that are not explicitly expressed in formal terms. These can include spatial and temporal reasoning, analogical reasoning, dialectic reasoning, commonsense reasoning, and moral reasoning[8]. While we won't discuss all of the forms of informal reasoning, we describe a case study in reasoning in uncertainty and describe analogical reasoning in more detail. Then, we move to the most informal realm of all – that of natural language, where large language models (LLMs) are demonstrating informal reasoning capabilities.

3.1 Reasoning about underspecified problems

Informal reasoning enables problem solving in diverse contexts and in the presence of uncertainty and ambiguity. This is especially critical in the real world. Consider the difference between the operation of a chemical plant in process simulation software vs. in the real world. A catalytic reactor in a process simulation might be modeled by a neat set of equations based on ODEs. A real-world reactor has dozens of additional complexities, including feed streams with trace impurities that may accumulate on the catalyst; catalyst variation across the reactor bed, over time on stream, and between manufacturer batches; reactor walls that may corrode over time; perturbations from an ideal cylinder due to inlet and outlet and sensor ports. Reasoning about the complex system requires qualitatively different skills than solving the ODEs.

Scientists and engineers need to reason informally when facing underspecified problems. Ask a student of thermodynamics, "What happens to a gas when it

[8]Since the 80's, researchers have tried to formalize these, with the Cyc project for commonsense reasoning being one of the most prominent.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

is compressed by a piston?" and they may answer, "The pressure will increase, and maybe it'll get hotter, depending on whether it can exchange heat with the surroundings." Conclusions emerge despite details being absent from the question (How much is the gas compressed? Is the gas real or ideal? What about phase changes?). They make be starting from an ideal gas as a base case, then thinking about Boyle's Law, where pressure and volume are inversely proportional ($P \propto 1/V$ when T is constant), then recognizing that we don't know whether T is constant, so there could be multiple answers worth exploring. Furthermore, in the case of Boyle's Law, the notion of "inversely proportional to" is hard to express mathematically; in formal logic, it is $\exists k, P = k/V$ – there exists some constant k such that $P = k/V$ (36). Students *don't* go through this route, because reasoning more loosely is easier, and is usually sufficient.

Adding details about the gas properties, heat transfer, and initial/final conditions can shift this problem into a formal one (perhaps of the plug-and-chug variety, i.e. Fig. 1c). But such a formulation doesn't test the student's ability to think at a more abstract level, which is required for more general problem-solving, as most real-world problems don't come fully-specified. In our view, a significant purpose of chemical education is to develop and refine "chemical intuition" (learned heuristics, both explicit and implicit) that enable chemists and chemical engineers to reason through underspecified problems (98), using both System 1 and System 2 reasoning patterns (27).

3.2 Reasoning with analogy

One important informal reasoning pattern is analogy. These play a role in the major paradigms in chemistry, such as the greenhouse effect in climate change, the lock and key in enzyme catalysis, and planetary orbits in the Bohr model of the atom relate scientific concepts to everyday ideas. Analogy also plays an important role in everyday problem solving: one can use an analogy between a novel polyfluorinated alkyl substance (PFAS) and a well-characterized one, like polyfluorinated octanoic acid (PFOA), and thus estimate the properties of the novel one. Or consider halogen chemistry – the concept "halogen" is an abstraction that places fluorine, chlorine, and bromine into the same category. This allows one to predict the properties of NaBr by way of analogy to the properties of NaCl, and predict the reaction products of an acyl bromide following an observation of the products of an acyl chloride.

Analogical reasoning relates something in a familiar source domain to something in a less familiar target domain. Formally, it is a form of inductive reasoning, "If X and Y share properties a, b, and c, they may also share property d" (99). But in practice, people don't usually articulate their reasoning in this way; analogies are developed *informally*. Atoms and the solar system each have many properties, after all – details such as "a, b, and c are the most relevant properties for grounding the comparison (and why)" are rarely written down. This has made computational implementations of analogical reasoning difficult, though progress has been made in developing computational analogical reasoning (100; 99).

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

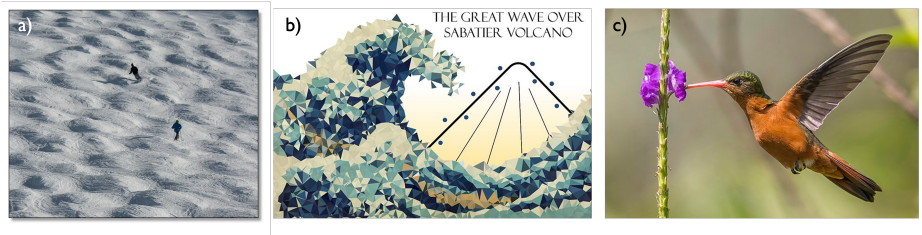

Figure 9: Analogies in catalytic resonance theory. a) In static catalysis, molecules "fall down" a free energy gradient, akin to downhill skiers, while b) in dynamic catalysis, molecules are promoted by a local free energy gradient, akin to surfers riding ocean waves. c) At the right frequency, a hummingbird's wings can appear static to the human eye; similarly, the coverage and rate of a catalytic surface appears statics near the resonance frequency. Image credits: a) The House Staff, b) From (103), reused with permission, c) Charles J. Sharp.

.

Nonetheless, analogies play a important role in education, as well as in scientific discovery. Kepler leaned on an analogy between light and the *vis motrix* (a precursor to gravity) to inspire his model of the solar system (101). Rutherford developed an analogy between planetary orbits and electrons orbiting the nucleus, which helped Bohr develop his eponymous model of the atom (102). To pick just one contemporary example, the development of catalytic resonance theory, a new paradigm in catalysis (103), leaned on many analogies through its conception. Sabatier first suggested that a catalyst shouldn't bind too strongly or too weakly (104); Balandin demonstrated this experimentally and first described "volcano-shaped" data (105), hence the ubiquitous "volcano plots" in heterogeneous catalysis (106). Ardagh, Abdelrahman, and Dauenhauer imagined an "inverted volcano" when conceiving catalytic resonance theory, and later developed more analogies when refining details, including a hummingbird flapping its wings, a band pass filter in electronics, and downhill skiing vs. surfing (Fig. 9).[9]

Computational approaches for analogical reasoning are usually applied in linguistics and natural language processing. In the chemical sciences, we know of two papers that explore this, using the term "transduction" to distinguish the approach from "induction" (107; 108). As we showed in Fig. 3, if x describes the material features (input) and y the a material properties (output), then induction reasons about data to find a general rule (the model fit) that is used to predict the property. Instead, transduction involves learning *contrasts* between input/output pairs in the training data. When a prediction is being made (at test time) for an unknown molecule, the algorithm identifies the input/output pair in the training set that is most similar to an input/output pair that includes the unknown molecule. It then uses the learned contrast to

[9]Personal communication.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

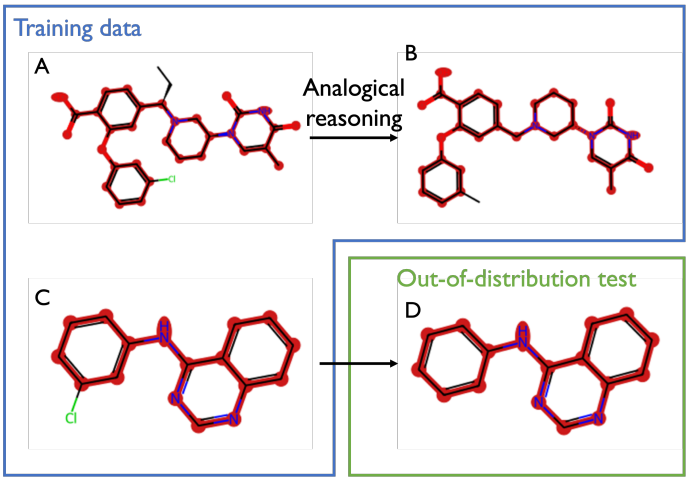

Figure 10: Analogical reasoning for molecular property prediction, adapted from (108). The property of an unknown, out of distribution molecule, is predicted using an analogy (A : B :: C : D) between anchor molecules (on the left) and target molecules (on the right).

compute the unknown property (108). Segal, et al. adopted a new, bilinear algorithm for transduction (109) that's simple and efficient, though transduction has previously been explored for quantitative structure/property relationships (107). Approaches that *combine* induction and transduction (110) may prove to be especially powerful, as they have shown promise on challenging analogical reasoning tasks like the Abstraction and Reasoning Corpus (21).

3.3 Large Language Models for Informal Reasoning

Human communication through natural language is much more variable, ambiguous, and contextual than the executable code or precise logical statements of formal languages. Historically, automating informal reasoning was generally not possible; reasoning about real-world contexts required definitions of facts and logical rules be first expressed in precise terms before a computer could reason about them. However, large language models (LLMs) operating on natural language have been breaking this paradigm. Furthermore, designing and building LLMs to excel at reasoning tasks has led to large reasoning models (LRMs), which are currently at the forefront of automated reasoning research.

LLMs are trained on a large corpus of text to generate the next word in a sequence of text. Various post-training approaches align the models to produce "desirable" responses, so they don't strictly follow probabilities in the training corpus (111). Fundamentally, these tools are *learning* (from the corpus during training and from reward models in post-training) and *predicting* what word is statistically likely to follow; they are thus not expected to be particularly ro-

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

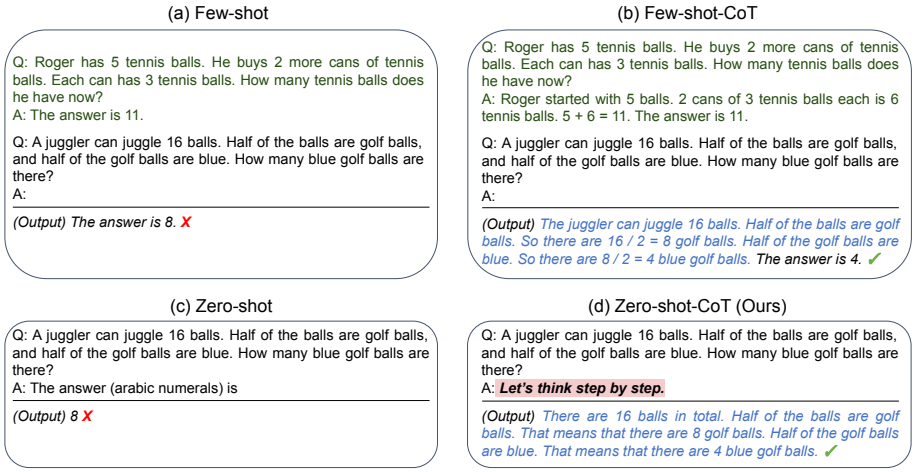

Figure 11: Example input and outputs of GPT-3 with different prompting strategies. Taken from (114).

bust or rigorous for solving reasoning tasks (112). Nonetheless, LLMs generate answers when given questions! They can be given questions that require reasoning, and they can be (post)trained on text that includes informal and/or formal reasoning patterns. Consequently, LLMs can be evaluated in their ability to engage in reasoning, and their performance has been remarkable.

3.3.1 Chain of thought: emergent "reasoning" in LLMs

This has most thoroughly been studied in the realm of solving word problems in math (113). Consider the examples in Fig. 11, from (114). Compare a prompt to an LLM that gives an example problem and answer, along with an unsolved problem. When the problem requires multistep reasoning, the LLM generates incorrect answers; it isn't capable of processing all the steps. Alternatively, if the prompt illustrates a step-by-step reasoning process, the generated text follows a similar step-by-step pattern of reasoning, before arriving at the answer (115). This prompting strategy dramatically improves LLMs' final answers in reasoning tasks. Shortly thereafter, researchers found that illustrating reasoning steps in the prompt is not necessary to induce the step-by-step solution sequence; simply including phrases like "Let's think step by step" is sufficient (114).

Such "chains of thought" form a neat bridge from "shallow" to "deep" problems (Fig. 1, as a singular algorithmic process (next-token prediction) can be deployed in service of both). These have also revealed a certain generality about reasoning, that *some problems simply require more work to solve than others*. Multiplying 8345 $\times$ 2317 requires more computation steps than multiplying 25 $\times$ 18. A large maze is harder to navigate than a small maze. NMR structure elucidation is harder for a complex molecule than for a simple one. These are

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

all hallmarks of problems requiring System 2 thinking.

John Kitchin draws a helpful analogy between LLM text generation and numerical solutions of ODEs (116): a system of ODEs can be integrated forward one step at a time, each new point depending on the model and prior states; an LLM propagates a string of tokens one at a time, each new token depending on the model and the previous tokens in the context. To find "what value will the ODEs arrive at after 1000 steps," one must visit the intermediate states; likewise, some questions to an LLM simply require generation of many intermediate tokens. A short chain of thought may be insufficient for solving a problem that intrinsically requires many steps.

This has led to novel paradigm in machine learning: "test-time compute." LLMs are trained using massive amounts of compute (even billions of dollars); the cost of running the LLM for an individual inference is a tiny fraction of that. The question becomes, "should fewer computational resources be put into training the LLM before its deployed, and more into generating its answers as it reasons about specific prompts, and by how much?" (117). Smaller models trained to generate more intermediate tokens at "test-time" can be more efficient to build and deploy than larger models. Running these models as a user is consequently more expensive (since they generate longer responses), but longer responses are intrinsically needed for problems that require reasoning.

After observing this behavior in LLMs, researchers have found ways to optimize this. For instance, chains of thought can be generalized to "trees of thought" (such as that shown in Fig. 1e), which more efficiently enable non-linear reasoning chains (118). Furthermore, reinforcement learning during training can improve the quality of reasoning chains. This was first demonstrated by models fine-tuned on data distinguishing good reasoning chains from bad (119). Later developments showed such "process reward models" are unnecessary; only question/answer pairs can be sufficient to optimize the models (120; 121). Recent approaches are also exploring ways to performing reasoning in latent space, rather than generating discrete tokens for the reasoning chain (122) (this leads to capability improvements, but we lament the necessary loss of interpretability). Aviary, from Future House, is a related approach leveraging reinforcement learning-trained LLMs to excel in scientific tasks (123).

Applying these strategies at scale has led to reasoning-optimized LLMs (sometimes called LRMs, or large reasoning models), which include OpenAI's o1, o3, and o4; Google's Gemini 2.0 Flash and 2.5 Flash; Anthropic's Claude and Opus; and DeepSeek's R1 and V3 models (124). LRMs can be traditional transformer-based LLMs that generate a linear sequences of tokens, whose parameters have been optimized for reasoning tasks. LRMs can also go further than next-token prediction, and include some form of non-linear search process for the best chain of tokens (like a tree of thoughts).

3.3.2 LLMs "reason" using learned vector programs

How are LLMs learning to reason? A helpful interpretation is that LLMs learn *vector programs* during training, which are retrieved and executed when

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

prompted by a user (125). When LLMs are trained on a large corpus of text, they effectively perform data compression (like a blurry JPEG, but for paragraphs (126)) – learning patterns in the source text, storing them in the weights of the model. Writing a prompt and using the LLM to generate the continuation of the text is a manner of retrieving this data, similar to how an entry in a database is retrieved by a query. Because the LLM transforms the training data and the prompt into vectors of numbers, the "database" and the "query" are no longer discrete, but continuous.[10]

But an LLM is even more than a continuous kind of lookup database for facts, it stores programs, as well. Some LLM prompts can be seen of as providing two things: 1) the "lookup key" for retrieving one of these programs, and 2) the inputs on which to run the program. Chollet illustrates with "Rewrite the following poem in the style of Shakespeare: ... my poem..." (125). In this example,"Rewrite the following poem in the style of" is the key used to retrieve a program, and "Shakespeare" and "... my poem..." are the program inputs. The program itself is an inscrutable collection of vectors stored in the LLM. LLMs consequently memorize millions of programs, which are stored in a continuous latent space, enabling retrieval of programs in an interpolative way. A prompt to an LLM points to one of these programs, which is "run" on inputs provided in the prompt. Those training LLMs aim to improve the reliability and generalizability of program learning, retrieval, and execution. Perhaps adding "Let's think step by step" is an effective prompting strategy for the InstructGPT LLM (114) because it points the LLM to a part of its latent space that stores programs that generate chains of thought. Likewise, perhaps the training strategies used by LRMs refine and improve the manner in which "programs" are stored and/or retrieved by the LLMs.[11] Researchers in the "mechanistic interpretability" field (129), or MechInterp, are trying to frame and answer questions like "What are the programs?", "What are the variables?", "How do they propagate through the network during inference?", "How are they learned?", and "Can these be controlled?" Systematically dissecting LLM behavior is challenging and expensive, but some approaches have shown promise at putting a "microscope" to LLM behavior (130).

3.3.3 Unreliable LLM reasoning

Nonetheless, LLMs (and LRMs) are far from a perfect solution to informal reasoning, exhibiting unreliability in many dimensions.

Firstly, chain of thought may not help some forms of reasoning tasks, and

[10] This plays a role in the tendency of LLMs to "hallucinate," or generate text whose form is plausible, even convincing, but whose content is false. For example, many have observed ChatGPT generate academic text with citations to works that don't exist (127). Approaches like Retrieval-Augmented Generation (RAG) can reduce hallucinations by connecting LLMs directly to source texts, improving accuracy and traceability of information (128). But, addressing factual accuracy is fundamentally a different question from reasoning.

[11] While a useful interpretation, this is far from the whole story. After all, a prompt is an undifferentiated string of tokens; there is no explicit distinction between the key and the program inputs.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 ORCID: https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. License: CC BY 4.0

may even hinder others. Sprague, et al. helpfully position informal reasoning tasks on a spectrum from symbolic to non-symbolic. Symbolic tasks are described in natural language, but ultimately can be converted into a purely symbolic problem describe in standard mathematical or logical terms. Non-symbolic tasks cannot be thus converted; "Where on a river can you hold a cup upright to catch water on a sunny day?" is such an example (from CommonsenseQA (131)). Tasks can also be in between, or semi-symbolic, which may describe most engineering problems – requiring non-symbolic reasoning (such as commonsense intuition and expert judgement) to "set up a problem," and symbolic reasoning to solve the resulting math problem. Evaluating LLM performance across diverse reasoning tasks shows that chain of thought helps better on symbolic reasoning tasks, and less well on non-symbolic reasoning tasks (132).

Second, reasoning processes may not be generalizable. For instance, in the grade school math questions in the popular GSM8k benchmark (113), changing the names of characters or objects in a question, or perturbing numbers in examples, caused variability in answers (133). LLMs are likewise not robust in analogical reasoning tasks (134). However, the latest LRMs appear to generalize much better than their LLM counterparts.

Third, even when generated chains of thought lead to improvements in performance, such chains might be plausible, but not faithful to rigorous reasoning processes (135; 136; 137; 138; 139; 140; 141; 142; 143). For example, DeepSeek's initial reasoning model output hard-to-interpret, multilingual chains of thought, but this was alleviated by adjusting training to produce monolingual chains of thought (121). Stechly and Valmeekam, et al. systematically studied chains of thought in a domain of planning/search problems (141). This domain allowed them to not just check whether answers are correct, but also evaluate whether the generated intermediate reasoning traces are valid. Training on correct reasoning traces improved LLMs performance, but LLMs still generated invalid traces that led to correct answers. Training on corrupted reasoning traces *also* improved LLM performance on generating correct solutions, and by a comparable amount. The authors interpret this through the lens of semantics: the intermediate tokens are certainly *useful* for the LLM to arrive at the correct solution (consider again the above analogy with ODEs), but their *meaning* is ultimately inconsistent with the semantics that the verifier (and the humans) are expecting. They suggest that "chains of thought" and "reasoning traces" are inappropriate anthropomorphisms, and that "intermediate tokens" would be a more appropriate term. In another recent paper testing LLMs and LRMs solving puzzles (143), internal reasoning traces were found to be unreliable, and neither LLMs and LRMs could generalize to the problems that required the most number of steps.

If LLMs are unreliable in reasoning, and perform best on symbolic reasoning tasks, then why use LLMs for reasoning at all? After all, hard-coded algorithms *don't* exhibit this sort of brittleness; algorithms for sorting lists or multiplying numbers don't suddenly fail for large lists and large numbers. An alternative would be to use an LLM to translate the informal reasoning task into a formal

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

problem, use a symbolic solver to rigorously execute the reasoning, then use the LLM to translate the result back into informal language (144). If a word problem requires multiplication of 8-digit numbers, instead of relying on an unreliable LLM for the multiplication, task the LLM to prepare an input to a calculator, which can rigorously execute the reasoning steps. This strategy may be especially relevant for reasoning problems akin to Fig. 1c, where "program execution" can be delegated to something reliable (and is indeed part of many commercial AI systems that support Python function execution). However, problems which require interleaving between informal and formal/symbolic reasoning steps would require multiple calls to reasoning tools; for simple reasoning steps (e.g. $1 < 2$) tool use might be overly tedious, and LLM's native reasoning skills might be sufficiently reliable. For more complex problems, LLMs can be integrated with solvers in nonlinear ways to combine the solvers' rigorous reasoning with the creativity and knowledge retrieval capabilities of LLMs (145).

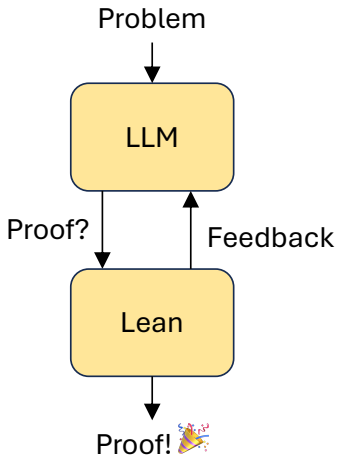

Figure 12: Workflow illustrating how LLM-generated proofs can be verified with Lean.

In other cases, the LLM may generate reasoning chains in the form of formal code, like Lean (16; 41; 146; 147), whose validity can be trusted after validation with an external verifier. Fig. 12 illustrates how this leverages the creativity of the LLM and the rigor of the theorem prover. When Lean accepts or rejects an LLM-generated proof to a problem, it can provide feedback for improving the LLM's performance, either via training or in-context learning. In such a framework, an LLM can generate thousands of incorrect proofs and one correct proof, and Lean will verify beyond doubt which is correct.

Fig. 12 is by no means the only workflow studied; variations include different ways of training on Lean proof libraries and feedback from the Lean compiler (16; 41), as well as multi-step workflows, such as drafting a proof strategy in natural language, sketching a multi-step proof in Lean, then proving the steps one at a time (148). DeepMind's AlphaProof system (18), which achieved a silver medal in the 2024 International Math Olympiad (IMO), used reinforcement learning to train an LLM to solve hundreds of millions of synthetic Lean proofs. Then, as the LLM attempted to solve a very hard problem, the problem synthesizer created variants of that problem, which were used to fine-tune the LLM at test time. As simpler variants were solved, and the LLM's weights updated, the LLM learned how to tackle the hard problem (the fine-tuning being discarded each time a new hard problem is attempted). "Neural theorem proving" is an active area of research in pure mathematics, which will likely benefit the sciences and engineering, as formal methods makes an impact there. Nonetheless, for the 2025 IMO compe-

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

tition, DeepMind's AI wrote its proofs in natural language, without Lean, and achieved a Gold medal (149).

4 Discussion

4.1 Being and doing

While we've organized these approaches into formal and informal categories, we can alternatively position them in two paradigms: propositional vs. procedural, "what" vs. "how", thermodynamics vs. kinetics. $77 \times 77 = 5929$ is a proposition that *is* true or false, and this logical view does not depend on *how* the truth value is obtained. Executing a multiplication algorithm, whether by hand, with a calculator, or via a "program" learned by a neural network, is principally about the step-by-step procedure.[12] Furthermore, in Section 2.2.2, we illustrate how a density functional theory program (indeed, any piece of scientific computing software) is a computable instantiation of a scientific theory; the theory is the "what" and the code is the "how." These perspectives intersect, as algorithms are ultimately employed to solve formal logic problems, and the computational complexity of an algorithm is itself subject to math proofs. Nonetheless, the scientist and engineer can benefit from keeping both of these in mind, and take care not to neglect the propositional perspective (and all the formal methods tools we have available) as chain-of-thought reasoning by LLMs grows ever more popular.

Which perspective is more fruitful may depend on the application. Algorithm execution may play an important role in developing next generation machine learning approaches that generalize in ways that current tools cannot. Representing chemistry knowledge using formal logic can help us make progress in simply thinking more precisely about familiar concepts (when such precision is possible), as well as enable rigorous verification of scientific derivations (36) and bug-free software for scientific computing (150). Planning can be seen as both algorithmic as well as formal, when one needs to construct and test plans alongside proofs to verify their properties.

Consider how these paradigms play out in the abductive reasoning task of NMR structure elucidation (see Section 2.3). The "procedural" paradigm positions this as a human or LLM undergoing a non-linear *process* of forming, evaluating, revising, and validating structural hypotheses. The "propositional" paradigm begins with a space of hypotheses that *exists* for this problem class, and is organized and/or navigated by humans and LLM agents in different ways (see Section 2.3.2). In other words, to study the "thermodynamics" of the task is to characterize the underlying hypothesis space (e.g. "What kinds of wrong answers are 'nearest' to the right answer?" (151)); to study the "kinetics" is to observe how different reasoners navigate that space (e.g. LLMs using chain of thought (152) and LLMs in structured workflows (153)).

[12] The functional vs. imperative programming paradigms (Lean 4 vs. Python), illustrate this dichotomy, as well.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

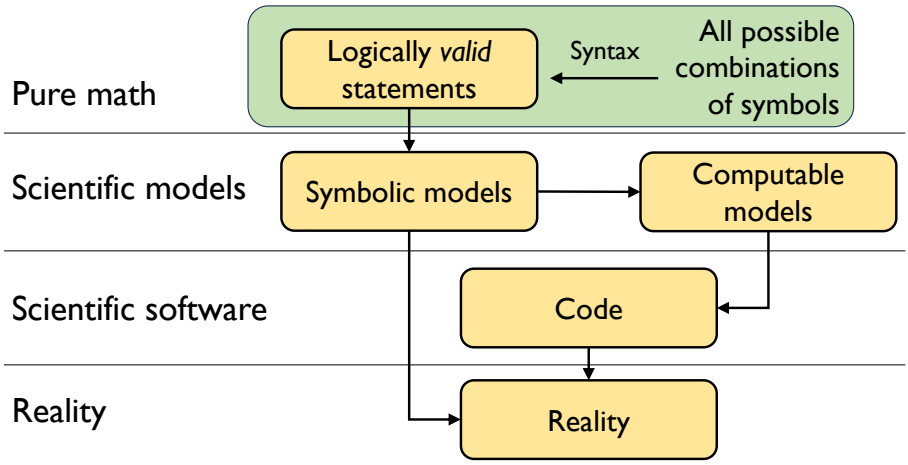

Figure 13: Syntax and semantics in scientific computing. In the realm of pure math, correct syntax isolates all logically valid statements. Semantics is requires to connect mathematics to the real world, in three bridges: from valid mathematics to meaningful scientific models, from these to computable models for scientific software, and once again when outputs of software are compared to experimental measurements. Image from Lecture 3, Lean for Scientists and Engineers, 2024 (42).

4.2 Syntax and semantics

In formal logic, *syntax* refers not just to whether a sentence is grammatically correct (the linguistic and programming senses of syntax), but also to whether it is logically valid (if the premises are true, then the conclusion is true). *Semantics* are about how such sentences connect to the real world (Fig. 13). Reasoning in mathematics is grounded in symbolic logic, where syntax is (nearly) everything, and connection to the real world is secondary. But for science in general, and for the chemical sciences in particular, semantics are essential. Let us compare and contrast repositories of "knowledge" in chemistry and in mathematics.

Stores of chemical knowledge, from the periodic table of the elements, to the NIST Chemistry Webbook (154) and ChemBL (155), are exercises in semantics. However, these are essentially structured *databases*, designed for facilitating information storage and retrieval, but not reasoning about their contents. Knowledge graphs, like Wikidata (156), are designed to facilitate reasoning about knowledge at scale, and can support deduction, induction, and abduction (157). Wikidata is a neat resource for encyclopedic knowledge; purpose-built knowledge graphs for materials science (158) and chemistry (159) applications have also been explored. For example, link prediction in a knowledge graph is a fundamentally logical task; when reactants, catalysts, and products comprise the knowledge graph, link prediction discovers missing reactions (160).

In contrast, the "data" involved in defining pure mathematics (for example,

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

Lean's math library Mathlib) is predominantly concepts defined and proofs establishing relationships among them. As of this writing, Mathlib has >110,000 definitions and >220,000 theorems expressed in just under 2 million lines of code. This comprises most of undergraduate mathematics, as well as much graduate material – it's dense, formally-verified, and far more intricate than any knowledge graph. Knowledge graphs just aren't expressive enough. Mathematics in Wikidata is semi-formal (161) and only somewhat computable (162).

Thus, ChemBL and Mathlib illustrate fundamental differences in how they approach syntax and semantics. ChemBL's >1.7 million assays are digital entires of real-world experiments – semantics is the purpose. Syntax is mostly relevant for harmonizing, structuring, and querying via controlled vocabulary – "formalizing" the data into a database. In Mathlib, syntax features far more prominently – it drives the precision of the library and is the basis of its logical verification. Semantics aren't absent; they show up in names of objects and in the human choices behind what belongs in the library and how the hierarchy of concepts is built. Mathematics isn't connected to the "real world" like chemistry is, but it is semantically grounded in human understanding of mathematics. For instance, renaming `Prime` to `Rose` throughout Mathlib wouldn't break the proofs, but it would distance the definition from the human understanding of primality.

At least three implications follow: First, the content of such libraries simply cannot be verified with as much confidence as Mathlib can be, because they primarily serve semantics, not syntax. A typo in the periodic table is a semantic error; Lean's type-checking rules out syntax errors. Second, scientific knowledge is not static; we will likely revise the atomic mass of ruthenium in the next century, at least some minor refinement. Updating knowledge in science is the norm; in mathematics, what is established is rarely overturned. Third, solving chemical problems really does require a great degree of memorization / retrieval. Statistical machine learning methods have made so much headway over the years, precisely because these effectively navigate these quantities of data. Yet, chemical knowledge is highly structured and often tied to theory; reasoning *is* important for navigating this structure.

4.3 Abstraction

Perhaps the most central, yet most mysterious, principle in reasoning is that of abstraction. Abstraction is forming categories that unite different entities according to what they share in common. In informal reasoning, it enables both analogical reasoning as well as reasoning about underspecified problems. This is the principal activity of analogical reasoning – obviously, a pair or molecules, or an enzyme/drug and a lock/key are *different*, but at some abstract level, they share enough in common for the analogy to be useful. Solving underspecified problems is impossible when the solution method requires specification of unknown details; by thinking about a problem at a higher, more abstract level, the unknown details can be set aside to focus on higher-level ideas. Scientists and engineers structure our knowledge and understanding with layers upon layers

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

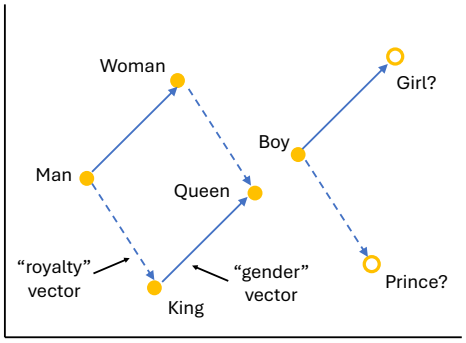

Figure 14: Learned "abstractions" in Word2vec, as similarity vectors between words. After learning "gender" and "royalty" vectors from *man*, *woman*, *king*, and *queen*, the location of "girl" can be predicted from that for "boy."

of abstractions; cognitive scientists have found that this ability to grasp "deep structure" is what makes experts superior to novices in solving physics problems (163). "Unit operation" is the abstraction that made chemical engineering possible; "functional group" is the basis of organic chemistry.

Abstraction plays out more concretely in formal reasoning: a theorem prover's math library doesn't need 5 versions of $(a + b) + c = a + (b + c)$ for Natural, Integer, Rational, Real, and Complex numbers. Instead, the mathematicians identify the most abstract type that subsumes the others (in this case, a semigroup), prove it for this most general case, and get the proofs for all the special cases automatically. Abstraction is in the very name of this field of study: *abstract* algebra. Authors of math libraries use abstraction abundantly, because it's necessary for complexity, efficiency, compactness, and scale.

Algorithms solve problems about abstract entities represented in the computer – this is the very mechanism by which computational researchers apply their skills in diverse domains. Genetic algorithms may be inspired by biology, but they find applications far beyond as a platform for gradient-free optimization. A materials scientist applying genetic algorithms to discover new metal-organic frameworks (MOFs) thinks about both MOFs and genetic algorithms at a more abstract level, and identifies the concepts about MOFs which map to "genes," "populations," and "fitness" in order to craft an appropriate implementation of the algorithm (164; 165). "Gradient-free optimization" is itself a more abstract category of algorithm, that includes particle swarm optimization (inspired by analogy to flocks of birds and schools of fish) (166) and simulated annealing (inspired by analogy to metallurgy) (167).

In each of these cases, humans are engaging in the effort of creating, identifying, adapting, and deploying abstractions. They may include computers in many steps of the process (e.g. executing a genetic algorithm or verifying a math library), but the computers are not practicing the art of abstraction *themselves*. However, this may be emerging, at least to some extent, in LLMs.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

The core challenge of this is answering the questions "Can a computer *learn* an abstraction (on its own)?" and "How can we test this?" Some have proposed that language models discovering similarities among words is a form of learning abstractions. The first of these, Word2vec, assigns a vector of numbers to each word according to the statistics of their proximity to other words in text (168). With words represented as vectors, properties such as distance and direction between pairs of words can be calculated (Fig. 14). Curiously, the vector between "man" and "woman" ends up being rather parallel to the vector between "king" and "queen"; the vectors between these can be interpreted as encoding the concepts of "gender" and "royalty." Thus, "word arithmetic" becomes possible, enabling the location of "boy" to be predicted by adding this vector to the position of "girl." When applied to a database of materials science abstracts, Word2vec finds associations among concepts in chemistry and materials science (169), with "word arithmetic" examples such as $(ferromagnetic - \mathrm{NiFe}) + \mathrm{IrMn} \approx antiferromagnetic$.

Linear relationships are quite inadequate for describing complex subjects. Rogers, Drozd, and Li nicely review the limitations of these "word analogies" (99) – for instance, they point out that a vector offset could theoretically solve the question "blood":"red" :: "snow":"white", but for the question "snow":"white" :: "sugar":"white", the correct answer is *a priori* excluded, since the vectors are necessarily not parallel. Large language models learn *nonlinear* relationships among words, and models like GPT-3 and GPT-4 have shown strong performance on some benchmarks for solving abstract reasoning problems that rely on analogies (170). However, properly evaluating whether systems are capable of learning abstract relationships isn't so simple – after all, they may learn shortcuts instead of rigorous reasoning, and still perform well on benchmarks. Subsequent work found LLM performance on these problems is not robust; making a small perturbation to a question, like modifying the alphabet, degrades LLM performance significantly while not affecting human performance (134). This suggests that the problem-solving techniques learned by LLMs from their training data are not generalizable in the way human problem-solving is. Novel approaches designed for learning and reasoning on-the-fly (171) are showing promise for tackling previously recalcitrant benchmarks (21). These may point the way to AI systems that can develop novel abstractions, but until then, computers will only deal in the abstractions that humans found first.

5 Conclusions

While statistical machine learning approaches are powerful techniques for "shallow" aspects of science and engineering problems, humans address the "depth" of these problems with a wide range of reasoning techniques. Several computational approaches for reasoning have been highlighted here, with most of our emphasis on deductive reasoning with formal logic and chain-of-thought "reasoning" with LLMs. Traditional, simulation-based scientific computing approaches are a form of deduction (albeit a narrow, specialized one),

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

while supervised learning-based approaches combine narrow forms of induction and deduction. Abductive and analogical reasoning feature prominently in human scientific reasoning, but are less common in computational settings, and are exciting subjects for future work. Applications of reasoning are as varied as topics in science and engineering, from verified bug-free software for scientific computing, to out-of-distribution predictive modeling, to inverse design and automated scientific discovery. Integrating symbolic and neural reasoning approaches could leverage advantages of each, with the creativity of generative AI reigned in by verification with symbolic AI.

6 Acknowledgements

Many individuals have contributed, directly or indirectly, to the perspective I've developed on automated reasoning in the chemical sciences. Cristina Cornelio introduced me to formal logic and theorem provers during our collaboration at IBM. Jeremy Avigad, P.S. Thiagarajan, and Katrin Erk have deepened and broadened my perspective through constructive conversations. The Zulip community has been supportive as I and my group have learned Lean, and textbooks written by Heather Macbeth and David Thrane Christianson have been invaluable for learning and teaching Lean to non-mathematicians. Sriram Sankaranarayanan, Chris Myers, and Balaji Rajagopalan organized the NSF workshop "Rigorous and Reproducible Scientific Reasoning," which broadened my perspective on formal logic beyond theorem provers.

This material is based on research that was in part supported by the NSF CTMC CAREER Award #2236769, by DARPA for the SciFy program under agreement number HR00112520301, by the Department of Energy's Bioenergy Technologies Office through the MSI STEM Research & Development Consortium, under contract number W911SR22F0104 (OR#64), and by the Army Research Office/Army Research Laboratory via grant #W911-NF-24-1-0399 to the University of Maryland, Baltimore County. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are that of the author and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of DARPA, the Army Research Office, the Department of Defense, or the U.S. Government.

Generative AI Disclosure

All text in this perspective was written by the author. PaperQA v1 (172) was used to find some of the cited references. ChatGPT was used to format some citations into Bibtex.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

References

[1] Robert K. Lindsay, Bruce G. Buchanan, Edward A. Feigenbaum, and Joshua Lederberg. DENDRAL: A case study of the first expert system for scientific hypothesis formation. *Artificial Intelligence*, 61(2):209–261, 1993. URL: https://www.sciencedirect.com/science/article/pii/000437029390068M, doi:https://doi.org/10.1016/0004-3702(93)90068-M.

[2] Pat Langley. Data-driven discovery of physical laws. *Cognitive Science*, 5(1):31–54, 1981. doi:10.1016/S0364-0213(81)80025-0.

[3] Ross D. King, Jem Rowland, Stephen G. Oliver, Michael Young, Wayne Aubrey, Emma Byrne, Maria Liakata, Magdalena Markham, Pinar Pir, Larisa N. Soldatova, Andrew Sparkes, Kenneth E. Whelan, and Amanda Clare. The Automation of Science. *Science*, 324(5923):85–89, April 2009. URL: https://www.science.org/doi/10.1126/science.1165620, doi:10.1126/science.1165620.

[4] Hiroaki Kitano. Artificial intelligence to win the Nobel Prize and beyond: Creating the engine for scientific discovery. *AI Magazine*, 37(1):39–49, 2016. doi:10.1609/aimag.v37i1.2642.

[5] Hiroaki Kitano. Nobel Turing Challenge: creating the engine for scientific discovery. *npj Systems Biology and Applications*, 7(1):29, June 2021. URL: https://www.nature.com/articles/s41540-021-00189-3, doi:10.1038/s41540-021-00189-3.

[6] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. URL: https://www.nature.com/articles/s41586-021-03819-2, doi:10.1038/s41586-021-03819-2.

[7] Nisheeth Vishnoi. What Counts as Discovery?, June 2025. URL: https://nisheethvishnoi.substack.com/p/what-counts-as-discovery.

[8] Kristian G. Barman. Is Automated Discovery Expanding Human Understanding? On the Role of Machine Learning in the Increase of Scientific Understanding. *Journal for General Philosophy of Science*, August 2025. URL: https://link.springer.com/10.1007/s10838-024-09716-2, doi:10.1007/s10838-024-09716-2.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[9] Murray Campbell, A.Joseph Hoane, and Feng-hsiung Hsu. Deep Blue. *Artificial Intelligence*, 134(1):57–83, 2002. URL: https://www.sciencedirect.com/science/article/pii/S0004370201001291, doi:https://doi.org/10.1016/S0004-3702(01)00129-1.

[10] D. A. Ferrucci. Introduction to "This is Watson". *IBM Journal of Research and Development*, 56(3.4):1:1–1:15, 2012. doi:10.1147/JRD.2012.2184356.

[11] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. doi:10.1038/nature16961.

[12] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, November 2019. URL: https://www.nature.com/articles/s41586-019-1724-z, doi:10.1038/s41586-019-1724-z.

[13] Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sasha Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of Diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, December 2022. URL: https://www.science.org/doi/10.1126/science.ade9097, doi:10.1126/science.ade9097.

[14] Siyuan Wang, Zhongkun Liu, Wanjun Zhong, Ming Zhou, Zhongyu Wei, Zhumin Chen, and Nan Duan. From LSAT: The Progress and Challenges of Complex Reasoning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2201–2216, 2022. URL: https://ieeexplore.ieee.org/document/9747955/, doi:10.1109/TASLP.2022.3164218.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[15] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders,

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 Technical Report, March 2024. arXiv:2303.08774 [cs]. URL: http://arxiv.org/abs/2303.08774.

[16] Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. Formal Mathematics Statement Curriculum Learning. *arXiv:2202.01344 [cs]*, February 2022. arXiv: 2202.01344. URL: http://arxiv.org/abs/2202.01344.

[17] Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, January 2024. URL: https://www.nature.com/articles/s41586-023-06747-5, doi:10.1038/s41586-023-06747-5.

[18] Google DeepMind. AI solves IMO problems at silver medal level, July 2024. URL: https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/.

[19] Andy Clark and David Chalmers. The Extended Mind. *Analysis*, 58(1):7–19, January 1998. tex.eprint: https://academic.oup.com/analysis/article-pdf/58/1/7/359060/58-1-7.pdf. doi:10.1093/analys/58.1.7.

[20] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A Graduate-Level Google-Proof Q&A Benchmark, November 2023. arXiv:2311.12022 [cs]. URL: http://arxiv.org/abs/2311.12022, doi:10.48550/arXiv.2311.12022.

[21] François Chollet. On the Measure of Intelligence, 2019. URL: http://arxiv.org/abs/1911.01547.

[22] M. Mitchell. *Artificial intelligence: a guide for thinking humans*. Pelican books. Penguin Books, Limited, 2019. URL: https://books.google.com/books?id=uOUIwQEACAAJ.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[23] A. M. Turing. Computing machinery and intelligence. *Mind*, LIX(236):433–460, October 1950. URL: https://academic.oup.com/mind/article/LIX/236/433/986238, doi:10.1093/mind/LIX.236.433.

[24] Leslie Valiant. *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. Basic Books, Inc., 2013.

[25] Amit Sheth, Kaushik Roy, and Manas Gaur. Neurosymbolic AI – Why, What, and How, May 2023. arXiv:2305.00813 [cs]. URL: http://arxiv.org/abs/2305.00813, doi:10.48550/arXiv.2305.00813.

[26] Daniel Kahneman. *Thinking, fast and slow*. Farrar, Straus and Giroux, New York: New York, 2011.

[27] Jenine Maeyer and Vicente Talanquer. The role of intuitive heuristics in students' thinking: Ranking chemical substances. *Science Education*, 94(6):963–984, November 2010. URL: https://onlinelibrary.wiley.com/doi/10.1002/sce.20397, doi:10.1002/sce.20397.

[28] Grady Booch, Francesco Fabiano, Lior Horesh, Kiran Kate, Jonathan Lenchner, Nick Linck, Andreas Loreggia, Keerthiram Murgesan, Nicholas Mattei, Francesca Rossi, and Biplav Srivastava. Thinking Fast and Slow in AI. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):15042–15046, May 2021. URL: https://ojs.aaai.org/index.php/AAAI/article/view/17765, doi:10.1609/aaai.v35i17.17765.

[29] Jennifer J. Sun, Megan Tjandrasuwita, Atharva Sehgal, Armando Solar-Lezama, Swarat Chaudhuri, Yisong Yue, and Omar Costilla-Reyes. Neurosymbolic Programming for Science, November 2022. arXiv:2210.05050 [cs]. URL: http://arxiv.org/abs/2210.05050, doi:10.48550/arXiv.2210.05050.

[30] Venkat Venkatasubramanian. Do large language models "understand" their knowledge? *AIChE Journal*, 71(3):e18661, March 2025. URL: https://aiche.onlinelibrary.wiley.com/doi/10.1002/aic.18661, doi:10.1002/aic.18661.

[31] Leah Henderson. The Problem of Induction. In Edward N. Zalta and Uri Nodelman, editors, *Stanford Encyclopedia of Philosophy*. Winter 2024 edition, 2022. URL: https://plato.stanford.edu/archives/win2024/entries/induction-problem/.

[32] Davor Lauc. Machine Learning and the Philosophical Problems of Induction. In Sandro Skansi, editor, *Guide to Deep Learning Basics*, pages 93–106. Springer International Publishing, Cham, 2020. URL: http://link.springer.com/10.1007/978-3-030-37591-1_9, doi:10.1007/978-3-030-37591-1_9.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[33] C.H. Bryant, A.E. Adam, D.R. Taylor, and R.C. Rowe. Towards an expert system for enantioseparations: induction of rules using machine learning. *Chemometrics and Intelligent Laboratory Systems*, 34(1):21–40, August 1996. URL: https://linkinghub.elsevier.com/retrieve/pii/0169743996000160, doi:10.1016/0169-7439(96)00016-0.

[34] Joan Bagaria. Set Theory. In Edward N. Zalta and Uri Nodelman, editors, *Stanford Encyclopedia of Philosophy*. Spring 2023 edition, 2023. URL: https://plato.stanford.edu/archives/spr2023/entries/set-theory.

[35] Thierry Coquand. Type Theory. In Edward N. Zalta and Uri Nodelman, editors, *Stanford Encyclopedia of Philosophy*. Fall 2022 edition, 2022. URL: https://plato.stanford.edu/archives/fall2022/entries/type-theory/.

[36] Maxwell P. Bobbin, Samiha Sharlin, Parivash Feyzishendi, An Hong Dang, Catherine M. Wraback, and Tyler R. Josephson. Formalizing chemical physics using the Lean theorem prover. *Digital Discovery*, 3(2):264–280, 2024. URL: https://xlink.rsc.org/?DOI=D3DD00077J, doi:10.1039/D3DD00077J.

[37] Irving Langmuir. The adsorption of gases on plane surfaces of glass, mica and platinum. *Journal of the American Chemical Society*, 40(9):1361–1403, 1918. doi:10.1021/ja02242a004.

[38] Leonardo De Moura and Nikolaj Bjørner. Z3: an efficient SMT solver. In *Proceedings of the theory and practice of software, 14th international conference on tools and algorithms for the construction and analysis of systems*, TACAS'08/ETAPS'08, pages 337–340, Berlin, Heidelberg, 2008. Springer-Verlag. Number of pages: 4 Place: Budapest, Hungary.

[39] Alonzo Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2):345–363, 1936. Publisher: The Johns Hopkins University Press. URL: http://www.jstor.org/stable/2371045.

[40] A. M. Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937. URL: http://doi.wiley.com/10.1112/plms/s2-42.1.230, doi:10.1112/plms/s2-42.1.230.

[41] Jesse Michael Han, Jason Rute, Yuhuai Wu, Edward W. Ayers, and Stanislas Polu. Proof Artifact Co-training for Theorem Proving with Language Models, March 2022. arXiv:2102.06203 [cs]. URL: http://arxiv.org/abs/2102.06203.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[42] Tyler R. Josephson, Tomas Skrivan, Ejike D. Ugwuanyi, Colin Jones, Alan Vithayathil, Oscar Matemb, and Shashane Anderson. Lean for Scientists and Engineers, 2024, 2024. URL: https://github.com/ATOMSLab/LFSE2024.

[43] Laura Kovács and Andrei Voronkov. First-Order Theorem Proving and Vampire. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Natasha Sharygina, and Helmut Veith, editors, *Computer Aided Verification*, volume 8044, pages 1–35. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. Series Title: Lecture Notes in Computer Science. URL: http://link.springer.com/10.1007/978-3-642-39799-8_1, doi:10.1007/978-3-642-39799-8_1.

[44] Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Volp, and Andre Platzer. KeYmaera X: An Axiomatic Tactical Theorem Prover for Hybrid Systems. In *Automated Deduction - CADE-25*, pages 527–538, 2015.

[45] Rose Bohrer. Chemical Case Studies in KeYmaera X. In Jan Friso Groote and Marieke Huisman, editors, *Formal Methods for Industrial Critical Systems*, volume 13487, pages 103–120. Springer International Publishing, Cham, 2022. Series Title: Lecture Notes in Computer Science. URL: https://link.springer.com/10.1007/978-3-031-15008-1_8, doi:10.1007/978-3-031-15008-1_8.

[46] Sascha Böhme and Tobias Nipkow. Sledgehammer: Judgement Day. In Jürgen Giesl and Reiner Hähnle, editors, *Automated Reasoning*, pages 107–121, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[47] Leonardo de Moura and Sebastian Ullrich. The Lean 4 Theorem Prover and Programming Language. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction – CADE 28*, volume 12699, pages 625–635. Springer International Publishing, Cham, 2021. Series Title: Lecture Notes in Computer Science. URL: https://link.springer.com/10.1007/978-3-030-79876-5_37, doi:10.1007/978-3-030-79876-5_37.

[48] Judith Clymo, Christopher M. Collins, Katie Atkinson, Matthew S. Dyer, Michael W. Gaultois, Vladimir V. Gusev, Matthew J. Rosseinsky, and Sven Schewe. Exploration of Chemical Space Through Automated Reasoning. *Angewandte Chemie International Edition*, page e202417657, January 2025. URL: https://onlinelibrary.wiley.com/doi/10.1002/anie.202417657, doi:10.1002/anie.202417657.

[49] David A. C. Beck, James M. Carothers, Venkat R. Subramanian, and Jim Pfaendtner. Data science: Accelerating innovation and discovery in chemical engineering. *AIChE Journal*, 62(5):1402–1416, 2016. ISBN:

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

9513862380. URL: http://doi.wiley.com/10.1002/aic.15192, doi:10.1002/aic.15192.

[50] Jarosław M. Granda, Liva Donina, Vincenza Dragone, De Liang Long, and Leroy Cronin. Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature*, 559(7714):377–381, 2018. doi:10.1038/s41586-018-0307-8.

[51] Sergei V. Kalinin, Maxim Ziatdinov, Jacob Hinkle, Stephen Jesse, Ayana Ghosh, Kyle P. Kelley, Andrew R. Lupini, Bobby G. Sumpter, and Rama K. Vasudevan. Automated and Autonomous Experiments in Electron and Scanning Probe Microscopy. *ACS Nano*, 15(8):12604–12627, August 2021. URL: https://pubs.acs.org/doi/10.1021/acsnano.1c02104, doi:10.1021/acsnano.1c02104.

[52] Claudia Houben and Alexei A Lapkin. Automatic discovery and optimization of chemical processes. *Current Opinion in Chemical Engineering*, 9:1–7, August 2015. URL: https://linkinghub.elsevier.com/retrieve/pii/S2211339815000404, doi:10.1016/j.coche.2015.07.001.

[53] Randall Balestriero, Jerome Pesenti, and Yann LeCun. Learning in High Dimension Always Amounts to Extrapolation, October 2021. arXiv:2110.09485 [cs]. URL: http://arxiv.org/abs/2110.09485, doi:10.48550/arXiv.2110.09485.

[54] Albert P. Bartók, James Kermode, Noam Bernstein, and Gábor Csányi. Machine Learning a General-Purpose Interatomic Potential for Silicon. *Physical Review X*, 8(4):041048, December 2018. URL: https://link.aps.org/doi/10.1103/PhysRevX.8.041048, doi:10.1103/PhysRevX.8.041048.

[55] Xiang Fu, Zhenghao Wu, Wujie Wang, Tian Xie, Sinan Keten, Rafael Gomez-Bombarelli, and Tommi Jaakkola. Forces are not Enough: Benchmark and Critical Evaluation for Machine Learning Force Fields with Molecular Simulations, August 2023. arXiv:2210.07237 [physics]. URL: http://arxiv.org/abs/2210.07237, doi:10.48550/arXiv.2210.07237.

[56] Bowen Deng, Yunyeong Choi, Peichen Zhong, Janosh Riebesell, Shashwat Anand, Zhuohan Li, KyuJung Jun, Kristin A. Persson, and Gerbrand Ceder. Systematic softening in universal machine learning interatomic potentials. *npj Computational Materials*, 11(1):9, January 2025. URL: https://www.nature.com/articles/s41524-024-01500-6, doi:10.1038/s41524-024-01500-6.

[57] Tobias Morawietz, Andreas Singraber, Christoph Dellago, and Jörg Behler. How van der Waals interactions determine the unique properties of water. *Proceedings of the National Academy of Sciences*, 113(30):1–18, 2016. ISBN: 0027-8424. URL:

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

http://www.pnas.org/content/suppl/2016/07/07/1602375113.DCSupplemental/pnas.1602375113.sapp.pdf%5Cnpapers3://publication/uuid/322B9EB6-BC55-4C7D-B6D6-BC5505527FB1, doi:10.1073/pnas.1602375113.

[58] Jonathan Vandermause, Steven B. Torrisi, Simon Batzner, Yu Xie, Lixin Sun, Alexie M. Kolpak, and Boris Kozinsky. On-the-fly active learning of interpretable Bayesian force fields for atomistic rare events. *npj Computational Materials*, 6(1):20, March 2020. URL: https://www.nature.com/articles/s41524-020-0283-z, doi:10.1038/s41524-020-0283-z.

[59] Chenru Duan, Aditya Nandy, Gianmarco G. Terrones, David W. Kastner, and Heather J. Kulik. Active Learning Exploration of Transition-Metal Complexes to Discover Method-Insensitive and Synthetically Accessible Chromophores. *JACS Au*, 3(2):391–401, February 2023. URL: https://pubs.acs.org/doi/10.1021/jacsau.2c00547, doi:10.1021/jacsau.2c00547.

[60] Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M. Elena, Dávid P. Kovács, Janosh Riebesell, Xavier R. Advincula, Mark Asta, William J. Baldwin, Noam Bernstein, Arghya Bhowmik, Samuel M. Blau, Vlad Cărare, James P. Darby, Sandip De, Flaviano Della Pia, Volker L. Deringer, Rokas Elijošius, Zakariya El-Machachi, Edvin Fako, Andrea C. Ferrari, Annalena Genreith-Schriever, Janine George, Rhys E. A. Goodall, Clare P. Grey, Shuang Han, Will Handley, Hendrik H. Heenen, Kersti Hermansson, Christian Holm, Jad Jaafar, Stephan Hofmann, Konstantin S. Jakob, Hyunwook Jung, Venkat Kapil, Aaron D. Kaplan, Nima Karimitari, Namu Kroupa, Jolla Kullgren, Matthew C. Kuner, Domantas Kuryla, Guoda Liepuoniute, Johannes T. Margraf, Ioan-Bogdan Magdău, Angelos Michaelides, J. Harry Moore, Aakash A. Naik, Samuel P. Niblett, Sam Walton Norwood, Niamh O'Neill, Christoph Ortner, Kristin A. Persson, Karsten Reuter, Andrew S. Rosen, Lars L. Schaaf, Christoph Schran, Eric Sivonxay, Tamás K. Stenczel, Viktor Svahn, Christopher Sutton, Cas van der Oord, Eszter Varga-Umbrich, Tejs Vegge, Martin Vondrák, Yangshuai Wang, William C. Witt, Fabian Zills, and Gábor Csányi. A foundation model for atomistic materials chemistry, December 2023. arXiv:2401.00096 [cond-mat, physics:physics]. URL: http://arxiv.org/abs/2401.00096.

[61] Christoph van Wüllen. Relativistic density functional theory. In Maria Barysz and Yasuyuki Ishikawa, editors, *Relativistic methods for chemists*, pages 191–214. Springer Netherlands, Dordrecht, 2010. doi:10.1007/978-1-4020-9975-5_5.

[62] Monireh Ebrahimi, Aaron Eberhart, Federico Bianchi, and Pascal Hitzler. Towards bridging the neuro-symbolic gap: deep deductive reasoners. *Applied Intelligence*, 51(9):6326–6348, September 2021. URL:

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

https://link.springer.com/10.1007/s10489-020-02165-6, doi:10.1007/s10489-020-02165-6.

[63] D. Price. Pentium FDIV flaw-lessons learned. *IEEE Micro*, 15(2):86–88, April 1995. URL: http://ieeexplore.ieee.org/document/372360/, doi:10.1109/40.372360.

[64] Vaughan Pratt. Anatomy of the Pentium bug. In Gerhard Goos, Juris Hartmanis, Jan Leeuwen, Peter D. Mosses, Mogens Nielsen, and Michael I. Schwartzbach, editors, *TAPSOFT '95: Theory and Practice of Software Development*, volume 915, pages 97–107. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995. Series Title: Lecture Notes in Computer Science. URL: http://link.springer.com/10.1007/3-540-59293-8_189, doi:10.1007/3-540-59293-8_189.

[65] Roope Kaivola and Naren Narasimhan. Formal Verification of the Pentium 4 Floating–Point Multiplier. In *Proceedings of the 2002 Design, Automation and Test in Europe Conference and Exhibition (DATEi02)*, page 8. IEEE Computer Society, 2002. doi:10.5555/882452.874523.

[66] Daniel Selsam, Percy Liang, and David L Dill. Developing Bug-Free Machine Learning Systems With Formal Mathematics. *ICML*, page 10, 2017.

[67] Ejike D. Ugwuanyi, Colin T. Jones, John Velkey, and Tyler R. Josephson. Benchmarking energy calculations using formal proofs. *Molecular Physics*, page e2539421, August 2025. URL: https://www.tandfonline.com/doi/full/10.1080/00268976.2025.2539421, doi:10.1080/00268976.2025.2539421.

[68] André Platzer. Differential Dynamic Logic for Hybrid Systems. *Journal of Automated Reasoning*, 41(2):143–189, August 2008. URL: https://link.springer.com/10.1007/s10817-008-9103-8, doi:10.1007/s10817-008-9103-8.

[69] David W. Renshaw, Sarah M. Loos, and André Platzer. Distributed Theorem Proving for Distributed Hybrid Systems. In Shengchao Qin and Zongyan Qiu, editors, *Formal Methods and Software Engineering*, volume 6991, pages 356–371. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. Series Title: Lecture Notes in Computer Science. URL: http://link.springer.com/10.1007/978-3-642-24559-6_25, doi:10.1007/978-3-642-24559-6_25.

[70] Parivash Feyzishendi, Sophia Hamer, Jinyu Huang, and Tyler R. Josephson. Formal Verification of Isothermal Chemical Reactors, September 2025. arXiv:2509.01130 [cs]. URL: http://arxiv.org/abs/2509.01130, doi:10.48550/arXiv.2509.01130.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[71] Atocha Aliseda. The logic of abduction: An introduction. In Lorenzo Magnani and Tommaso Bertolotti, editors, *Springer handbook of model-based science*, pages 219–230. Springer International Publishing, Cham, 2017. doi:10.1007/978-3-319-30526-4_10.

[72] Emmanuel J. Genot. Strategies of inquiry: The 'Sherlock Holmes sense of deduction' revisited. *Synthese*, 195(5):2065–2088, May 2018. URL: http://link.springer.com/10.1007/s11229-017-1319-x, doi:10.1007/s11229-017-1319-x.

[73] Tianfan Jin, Qiyuan Zhao, Andrew B. Schofield, and Brett M. Savoie. Deductive machine learning models for product identification. *Chemical Science*, 15:11995–12005, 2024. doi:10.1039/d3sc04909d.

[74] Tianfan Jin and Brett M Savoie. Deductive Machine Learning Challenges and Opportunities in Chemical Applications. *Annual Review of Chemical and Biomolecular Engineering*, 15:343–360, 2024. doi:10.1146/annurev-chembioeng-100722-111917.

[75] Cristina Cornelio, Sanjeeb Dash, Vernon Austel, Tyler R. Josephson, Joao Goncalves, Kenneth L. Clarkson, Nimrod Megiddo, Bachir El Khadir, and Lior Horesh. Combining data and theory for derivable scientific discovery with AI-Descartes. *Nature Communications*, 14(1):1777, April 2023. URL: https://www.nature.com/articles/s41467-023-37236-y, doi:10.1038/s41467-023-37236-y.

[76] Maxim A. Ziatdinov, Yongtao Liu, Anna N. Morozovska, Eugene A. Eliseev, Xiaohang Zhang, Ichiro Takeuchi, and Sergei V. Kalinin. Hypothesis Learning in Automated Experiment: Application to Combinatorial Materials Libraries. *Advanced Materials*, 34(20), May 2022. Publisher: Wiley. URL: https://onlinelibrary.wiley.com/doi/10.1002/adma.202201345, doi:10.1002/adma.202201345.

[77] Arijit Chakraborty, Abhishek Sivaram, and Venkat Venkatasubramanian. AI-DARWIN: A first principles-based model discovery engine using machine learning. *Computers & Chemical Engineering*, 154:107470, November 2021. Publisher: Elsevier BV. URL: https://linkinghub.elsevier.com/retrieve/pii/S0098135421002489, doi:10.1016/j.compchemeng.2021.107470.

[78] Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. DiscoveryBench: Towards Data-Driven Discovery with Large Language Models, July 2024. arXiv:2407.01725 [cs]. URL: http://arxiv.org/abs/2407.01725, doi:10.48550/arXiv.2407.01725.

[79] Atilla Kaan Alkan, Shashwat Sourav, Maja Jablonska, Simone Astarita, Rishabh Chakrabarty, Nikhil Garuda, Pranav Khetarpal, Maciej Pióro,

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

Dimitrios Tanoglidis, Kartheik G. Iyer, Mugdha S. Polimera, Michael J. Smith, Tirthankar Ghosal, Marc Huertas-Company, Sandor Kruk, Kevin Schawinski, and Ioana Ciucă. A Survey on Hypothesis Generation for Scientific Discovery in the Era of Large Language Models, April 2025. arXiv:2504.05496 [cs]. URL: http://arxiv.org/abs/2504.05496, doi:10.48550/arXiv.2504.05496.

[80] Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah D. Goodman. Hypothesis Search: Inductive Reasoning with Language Models, May 2024. arXiv:2309.05660 [cs]. URL: http://arxiv.org/abs/2309.05660, doi:10.48550/arXiv.2309.05660.

[81] Kexin Huang, Ying Jin, Ryan Li, Michael Y. Li, Emmanuel Candès, and Jure Leskovec. Automated Hypothesis Validation with Agentic Sequential Falsifications, February 2025. arXiv:2502.09858 [cs]. URL: http://arxiv.org/abs/2502.09858, doi:10.48550/arXiv.2502.09858.

[82] Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, and Andrew D White. Augmenting large language models with chemistry tools. In *NeurIPS 2023 AI for Science Workshop*. 2023. URL: https://openreview.net/forum?id=wdGIL6lx3l.

[83] Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, December 2023. URL: https://www.nature.com/articles/s41586-023-06792-0, doi:10.1038/s41586-023-06792-0.

[84] Quintina Campbell, Sam Cox, Jorge Medina, Brittany Watterson, and Andrew D. White. MDCrow: Automating Molecular Dynamics Workflows with Large Language Models, February 2025. arXiv:2502.09565 [cs]. URL: http://arxiv.org/abs/2502.09565, doi:10.48550/arXiv.2502.09565.

[85] Yunheng Zou, Austin H. Cheng, Abdulrahman Aldossary, Jiaru Bai, Shi Xuan Leong, Jorge Arturo Campos-Gonzalez-Angulo, Changhyeok Choi, Cher Tian Ser, Gary Tom, Andrew Wang, Zijian Zhang, Ilya Yakavets, Han Hao, Chris Crebolder, Varinia Bernales, and Alán Aspuru-Guzik. El Agente: An autonomous agent for quantum chemistry. *Matter*, 8(7):102263, July 2025. Publisher: Elsevier BV. URL: https://linkinghub.elsevier.com/retrieve/pii/S2590238525003066, doi:10.1016/j.matt.2025.102263.

[86] Ziqi Wang, Hongshuo Huang, Hancheng Zhao, Changwen Xu, Shang Zhu, Jan Janssen, and Venkatasubramanian Viswanathan. DREAMS: Density Functional Theory Based Research Engine for Agentic Materials Simulation, July 2025. arXiv:2507.14267 [cs]. URL: http://arxiv.org/abs/2507.14267, doi:10.48550/arXiv.2507.14267.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[87] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. ProbLog: a probabilistic prolog and its application in link discovery. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI'07, pages 2468–2473. Morgan Kaufmann Publishers Inc., San Francisco, CA, 2007. URL: https://dl.acm.org/doi/10.5555/1625275.1625673.

[88] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, February 2006. URL: http://link.springer.com/10.1007/s10994-006-5833-1, doi:10.1007/s10994-006-5833-1.

[89] Judea Pearl. Causal inference in statistics: An overview. *Statist. Surv.*, 3:96–149, 2009. doi:10.1214/09-SS057.

[90] Judea Pearl. Theoretical Impediments to Machine Learning With Seven Sparks from the Causal Revolution. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 3–3, Marina Del Rey CA USA, February 2018. ACM. URL: https://dl.acm.org/doi/10.1145/3159652.3176182, doi:10.1145/3159652.3176182.

[91] Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Basic Books, Inc., USA, 1 edition, 2018.

[92] Johannes Textor and Maciej Li. Robust causal inference using directed acyclic graphs: the R package 'dagitty'. *International Journal of Epidemiology*, 45(6), 2016.

[93] Joseph F. Hair, G. Tomas M. Hult, Christian M. Ringle, Marko Sarstedt, Nicholas P. Danks, and Soumya Ray. An introduction to structural equation modeling. In *Partial least squares structural equation modeling (PLS-SEM) using R: a workbook*, pages 1–29. Springer International Publishing, Cham, 2021. doi:10.1007/978-3-030-80519-7_1.

[94] David Heckerman. A Tutorial on Learning With Bayesian Networks, January 2022. arXiv:2002.00269 [cs]. URL: http://arxiv.org/abs/2002.00269, doi:10.48550/arXiv.2002.00269.

[95] Chung Chien Chang and Cheng Ching Yu. On-line fault diagnosis using the signed directed graph. *Ind. Eng. Chem. Res.*, 29:1290–1299, 1990. doi:10.1021/ie00103a031.

[96] Mano Ram Maurya, Raghunathan Rengaswamy, and Venkat Venkatasubramanian. Fault diagnosis using dynamic trend analysis: A review and recent developments. *Engineering Applications of Artificial Intelligence*, 2007.

[97] Tom Michoel. Causal inference in drug discovery and development. *Drug Discovery Today*, 28(10), 2023.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[98] Nicole Graulich, Henning Hopf, and Peter R. Schreiner. Heuristic thinking makes a chemist smart. *Chem. Soc. Rev.*, 39(5):1503–1512, 2010. URL: https://xlink.rsc.org/?DOI=B911536F, doi:10.1039/B911536F.

[99] Anna Rogers, Aleksandr Drozd, and Bofang Li. The (too Many) Problems of Analogical Reasoning with Word Vectors. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 135–148, Vancouver, Canada, 2017. Association for Computational Linguistics. URL: http://www.aclweb.org/anthology/S17-1017, doi:10.18653/v1/S17-1017.

[100] Rogers P. Hall. Computational approaches to analogical reasoning: A comparative analysis. *Artificial Intelligence*, 39(1):39–120, 1989. URL: https://www.sciencedirect.com/science/article/pii/0004370289900039, doi:https://doi.org/10.1016/0004-3702(89)90003-9.

[101] Dedre Gentner. Analogy in Scientific Discovery: The Case of Johannes Kepler. In Lorenzo Magnani and Nancy J. Nersessian, editors, *Model-Based Reasoning*, pages 21–39. Springer US, New York, NY, 2002. URL: https://link.springer.com/10.1007/978-1-4615-0605-8_2, doi:10.1007/978-1-4615-0605-8_2.

[102] Manfred Bucher. Rise and fall of the old quantum theory. *arXiv: History and Philosophy of Physics*, 2008. URL: https://api.semanticscholar.org/CorpusID:15422334.

[103] M. Alexander Ardagh, Omar A. Abdelrahman, and Paul J. Dauenhauer. Principles of Dynamic Heterogeneous Catalysis: Surface Resonance and Turnover Frequency Response. *ACS Catalysis*, 9(8):6929–6937, August 2019. URL: https://pubs.acs.org/doi/10.1021/acscatal.9b01606, doi:10.1021/acscatal.9b01606.

[104] Paul Sabatier. *La catalyse en chimie organique*, volume 3. C. Béranger, 1920.

[105] A.A. Balandin. Modern state of the multiplet theory of heterogeneous catalysis. volume 19 of *Advances in catalysis*, pages 1–210. Academic Press, 1969. ISSN: 0360-0564. URL: https://www.sciencedirect.com/science/article/pii/S0360056408600292, doi:https://doi.org/10.1016/S0360-0564(08)60029-2.

[106] Andrew J. Medford, Aleksandra Vojvodic, Jens S. Hummelshøj, Johannes Voss, Frank Abild-Pedersen, Felix Studt, Thomas Bligaard, Anders Nilsson, and Jens K. Nørskov. From the Sabatier principle to a predictive theory of transition-metal heterogeneous catalysis. *Journal of Catalysis*, 328:36–42, August 2015. URL: https://linkinghub.elsevier.com/retrieve/pii/S0021951714003686, doi:10.1016/j.jcat.2014.12.033.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[107] Jason Weston, Fernando Pérez-Cruz, Olivier Bousquet, Olivier Chapelle, André Elisseeff, and Bernhard Schölkopf. Feature selection and transduction forprediction of molecular bioactivity for drug design. *Bioinformatics*, 19(6):764–771, April 2003. URL: https://academic.oup.com/bioinformatics/article/19/6/764/234339, doi:10.1093/bioinformatics/btg054.

[108] Nofit Segal, Aviv Netanyahu, Kevin P. Greenman, Pulkit Agrawal, and Rafael Gomez-Bombarelli. Known Unknowns: Out-of-Distribution Property Prediction in Materials and Molecules, February 2025. arXiv:2502.05970 [cs]. URL: http://arxiv.org/abs/2502.05970, doi:10.48550/arXiv.2502.05970.

[109] Aviv Netanyahu, Abhishek Gupta, Max Simchowitz, Kaiqing Zhang, and Pulkit Agrawal. Learning to Extrapolate: A Transductive Approach, April 2023. arXiv:2304.14329 [cs]. URL: http://arxiv.org/abs/2304.14329, doi:10.48550/arXiv.2304.14329.

[110] Wen-Ding Li, Keya Hu, Carter Larsen, Yuqing Wu, Simon Alford, Caleb Woo, Spencer M. Dunn, Hao Tang, Michelangelo Naim, Dat Nguyen, Wei-Long Zheng, Zenna Tavares, Yewen Pu, and Kevin Ellis. Combining Induction and Transduction for Abstract Reasoning, December 2024. arXiv:2411.02272 [cs]. URL: http://arxiv.org/abs/2411.02272, doi:10.48550/arXiv.2411.02272.

[111] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, March 2022. arXiv:2203.02155 [cs]. URL: http://arxiv.org/abs/2203.02155.

[112] Gary Marcus. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence, 2020. URL: http://arxiv.org/abs/2002.06177.

[113] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, November 2021. arXiv:2110.14168 [cs]. URL: http://arxiv.org/abs/2110.14168.

[114] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large Language Models are Zero-Shot Reasoners, June 2022. arXiv:2205.11916 [cs]. URL: http://arxiv.org/abs/2205.11916.

[115] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain of Thought

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

Prompting Elicits Reasoning in Large Language Models, June 2022. arXiv:2201.11903 [cs]. URL: http://arxiv.org/abs/2201.11903.

[116] John Kitchin. *A practical introduction to large language models in Python*. Point Breeze Publishing, LLC, 2025. URL: https://pointbreezepubs.gumroad.com/l/llm.

[117] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters, August 2024. arXiv:2408.03314 [cs]. URL: http://arxiv.org/abs/2408.03314, doi:10.48550/arXiv.2408.03314.

[118] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate Problem Solving with Large Language Models, December 2023. arXiv:2305.10601 [cs]. URL: http://arxiv.org/abs/2305.10601, doi:10.48550/arXiv.2305.10601.

[119] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step, May 2023. arXiv:2305.20050 [cs]. URL: http://arxiv.org/abs/2305.20050, doi:10.48550/arXiv.2305.20050.

[120] Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training Language Models to Self-Correct via Reinforcement Learning, October 2024. arXiv:2409.12917 [cs]. URL: http://arxiv.org/abs/2409.12917, doi:10.48550/arXiv.2409.12917.

[121] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan,

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, January 2025. arXiv:2501.12948 [cs]. URL: `http://arxiv.org/abs/2501.12948`, `doi:10.48550/arXiv.2501.12948`.

[122] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training Large Language Models to Reason in a Continuous Latent Space, December 2024. arXiv:2412.06769 [cs]. URL: `http://arxiv.org/abs/2412.06769`, `doi:10.48550/arXiv.2412.06769`.

[123] Siddharth Narayanan, James D. Braza, Ryan-Rhys Griffiths, Manu Ponnapati, Albert Bou, Jon Laurent, Ori Kabeli, Geemi Wellawatte, Sam Cox, Samuel G. Rodriques, and Andrew D. White. Aviary: training language agents on challenging scientific tasks, December 2024. arXiv:2412.21154 [cs]. URL: `http://arxiv.org/abs/2412.21154`, `doi:10.48550/arXiv.2412.21154`.

[124] Melanie Mitchell. Artificial intelligence learns to reason. *Science*, 387(6740):eadw5211, 2025. tex.eprint: https://www.science.org/doi/pdf/10.1126/science.adw5211. URL: `https://www.science.org/doi/abs/10.1126/science.adw5211`, `doi:10.1126/science.adw5211`.

[125] Francois Chollet. How I think about LLM prompt engineering, October 2023. URL: `https://fchollet.substack.com/p/how-i-think-about-llm-prompt-engineering`.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[126] Ted Chiang. ChatGPT is a blurry JPEG of the web. *The New Yorker*, February 2023. URL: https://www.newyorker.com/tech/annals-of-technology/chatgpt-is-a-blurry-jpeg-of-the-web.

[127] Courtni Byun, Piper Vasicek, and Kevin Seppi. This Reference Does Not Exist: An Exploration of LLM Citation Accuracy and Relevance. In *Proceedings of the Third Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 28–39, Mexico City, Mexico, 2024. Association for Computational Linguistics. URL: https://aclanthology.org/2024.hcinlp-1.3, doi:10.18653/v1/2024.hcinlp-1.3.

[128] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval Augmentation Reduces Hallucination in Conversation, April 2021. arXiv:2104.07567 [cs]. URL: http://arxiv.org/abs/2104.07567, doi:10.48550/arXiv.2104.07567.

[129] Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. Open Problems in Mechanistic Interpretability, January 2025. arXiv:2501.16496 [cs]. URL: http://arxiv.org/abs/2501.16496, doi:10.48550/arXiv.2501.16496.

[130] Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025. URL: https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

[131] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North*, pages 4149–4158, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. URL: http://aclweb.org/anthology/N19-1421, doi:10.18653/v1/N19-1421.

[132] Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To CoT or not to CoT? Chain-of-thought helps mainly

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

on math and symbolic reasoning, September 2024. arXiv:2409.12183 [cs]. URL: http://arxiv.org/abs/2409.12183.

[133] Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models, October 2024. arXiv:2410.05229 [cs]. URL: http://arxiv.org/abs/2410.05229.

[134] Martha Lewis and Melanie Mitchell. Evaluating the robustness of analogical reasoning in large language models. *Transactions on Machine Learning Research*, 2025. URL: https://openreview.net/forum?id=t5cy5v9wph.

[135] Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring Faithfulness in Chain-of-Thought Reasoning, July 2023. arXiv:2307.13702 [cs]. URL: http://arxiv.org/abs/2307.13702, doi:10.48550/arXiv.2307.13702.

[136] Rylan Schaeffer, Kateryna Pistunova, Samar Khanna, Sarthak Consul, and Sanmi Koyejo. Invalid Logic, Equivalent Gains: The Bizarreness of Reasoning in Language Model Prompting, July 2023. arXiv:2307.10573 [cs]. URL: http://arxiv.org/abs/2307.10573, doi:10.48550/arXiv.2307.10573.

[137] Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the Planning Abilities of Large Language Models : A Critical Investigation, November 2023. arXiv:2305.15771 [cs]. URL: http://arxiv.org/abs/2305.15771, doi:10.48550/arXiv.2305.15771.

[138] Saurabh Srivastava, Annarose M. B, Anto P. V, Shashank Menon, Ajay Sukumar, Adwaith Samod T, Alan Philipose, Stevin Prince, and Sooraj Thomas. Functional Benchmarks for Robust Evaluation of Reasoning Performance, and the Reasoning Gap, February 2024. arXiv:2402.19450 [cs]. URL: http://arxiv.org/abs/2402.19450, doi:10.48550/arXiv.2402.19450.

[139] Chirag Agarwal, Sree Harsha Tanneru, and Himabindu Lakkaraju. Faithfulness vs. Plausibility: On the (Un)Reliability of Explanations from Large Language Models, March 2024. arXiv:2402.04614 [cs]. URL: http://arxiv.org/abs/2402.04614, doi:10.48550/arXiv.2402.04614.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[140] Evelyn Yee, Alice Li, Chenyu Tang, Yeon Ho Jung, Ramamohan Paturi, and Leon Bergen. Dissociation of Faithful and Unfaithful Reasoning in LLMs, September 2024. arXiv:2405.15092 [cs]. URL: http://arxiv.org/abs/2405.15092, doi:10.48550/arXiv.2405.15092.

[141] Kaya Stechly, Karthik Valmeekam, Atharva Gundawar, Vardhan Palod, and Subbarao Kambhampati. Beyond Semantics: The Unreasonable Effectiveness of Reasonless Intermediate Tokens, May 2025. arXiv:2505.13775 [cs]. URL: http://arxiv.org/abs/2505.13775, doi:10.48550/arXiv.2505.13775.

[142] Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vlad Mikulik, Sam Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. Reasoning Models Don't Always Say What They Think. April 2025. URL: https://assets.anthropic.com/m/71876fabef0f0ed4/original/reasoning_models_paper.pdf.

[143] Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity, 2025. URL: https://ml-site.cdn-apple.com/papers/the-illusion-of-thinking.pdf.

[144] Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning, October 2023. arXiv:2305.12295 [cs]. URL: http://arxiv.org/abs/2305.12295, doi:10.48550/arXiv.2305.12295.

[145] Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. LLMs Can't Plan, But Can Help Planning in LLM-Modulo Frameworks, June 2024. arXiv:2402.01817 [cs]. URL: http://arxiv.org/abs/2402.01817, doi:10.48550/arXiv.2402.01817.

[146] Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models, October 2023. arXiv:2306.15626 [cs, stat]. URL: http://arxiv.org/abs/2306.15626.

[147] Huajian Xin, Z. Z. Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, Wenjun Gao, Qihao Zhu, Dejian Yang, Zhibin Gou, Z. F. Wu, Fuli Luo, and Chong Ruan. DeepSeek-Prover-V1.5: Harnessing Proof Assistant Feedback for Reinforcement Learning and Monte-Carlo Tree Search, August 2024. arXiv:2408.08152 [cs]. URL: http://arxiv.org/abs/2408.08152, doi:10.48550/arXiv.2408.08152.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[148] Albert Q. Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timothée Lacroix, Yuhuai Wu, and Guillaume Lample. Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs, February 2023. arXiv:2210.12283 [cs]. URL: http://arxiv.org/abs/2210.12283, doi:10.48550/arXiv.2210.12283.

[149] Google DeepMind. Advanced Version of Gemini with Deep Think Officially Achieves Gold-Medal Standard at the International Mathematical Olympiad, 2025. URL: https://deepmind.google/discover/blog/advanced-version-of-gemini-with-deep-think-officially-achieves-gold-medal-standard-at-

[150] Ejike D. Ugwuanyi, Colin T. Jones, John Velkey, and Tyler R. Josephson. Benchmarking Energy Calculations Using Formal Proofs, May 2025. arXiv:2505.09095 [cond-mat]. URL: http://arxiv.org/abs/2505.09095, doi:10.48550/arXiv.2505.09095.

[151] Zuzana Osifová, Ondřej Socha, and Martin Dračínský. NMR-Challenge.com: Exploring the Most Common Mistakes in NMR Assignments. *Journal of Chemical Education*, page acs.jchemed.4c00092, May 2024. URL: https://pubs.acs.org/doi/10.1021/acs.jchemed.4c00092, doi:10.1021/acs.jchemed.4c00092.

[152] Samiha Sharlin, Fariha Agbere, Kevin Ishimwe, Zuzana Osifová, Ondřej Socha, Martin Dračínský, and Tyler Josephson. NMR-Challenge for LLMs: Evaluating Chemical Reasoning in Humans and AI, August 2025. URL: https://chemrxiv.org/engage/chemrxiv/article-details/689cccf6728bf9025e4831c3, doi:10.26434/chemrxiv-2025-x8h36-v2.

[153] Martin Priessner, Anna Tomberg, Jon Paul Janet, Richard Lewis, Magnus Johansson, and Jonathan Goodman. Enhancing Molecular Structure Elucidation with Reasoning-Capable LLMs, May 2025. URL: https://chemrxiv.org/engage/chemrxiv/article-details/6810775de561f77ed447d9ab, doi:10.26434/chemrxiv-2025-8l64w.

[154] P.J. Linstrom and W.G. Mallard. *NIST Chemistry WebBook, NIST Standard Reference Database Number 69*. National Institute of Standards and Technology, Gaithersburg, MD, 2025. URL: https://doi.org/10.18434/T4D303.

[155] Barbara Zdrazil, Eloy Felix, Fiona Hunter, Emma J Manners, James Blackshaw, Sybilla Corbett, Marleen de Veij, Harris Ioannidis, David Mendez Lopez, Juan F Mosquera, Maria Paula Magarinos, Nicolas Bosc, Ricardo Arcila, Tevfik Kizilören, Anna Gaulton, A Patrícia Bento, Melissa F Adasme, Peter Monecke, Gregory A Landrum, and Andrew R Leach. The ChEMBL Database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic Acids Research*, 52(D1):D1180–D1192, January 2024. URL: https://academic.oup.com/nar/article/52/D1/D1180/7337608, doi:10.1093/nar/gkad1004.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[156] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, September 2014. Publisher: Association for Computing Machinery (ACM). URL: https://dl.acm.org/doi/10.1145/2629489, doi:10.1145/2629489.

[157] Ricardo Guimarães and Ana Ozaki. Reasoning in Knowledge Graphs (Invited Paper). *OASIcs, Volume 99, AIB 2022*, 99:2:1–2:31, 2022. Artwork Size: 31 pages, 1029096 bytes ISBN: 9783959772280 Medium: application/pdf Publisher: Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/OASIcs.AIB.2022.2, doi:10.4230/OASICS.AIB.2022.2.

[158] Xiaoming Zhang, Chongchong Zhao, and Xiang Wang. A survey on knowledge representation in materials science and engineering: An ontological perspective. *Computers in Industry*, 73:8–22, October 2015. URL: https://linkinghub.elsevier.com/retrieve/pii/S0166361515300233, doi:10.1016/j.compind.2015.07.005.

[159] Angiras Menon, Nenad B Krdzavac, and Markus Kraft. From database to knowledge graph — using data in chemistry. *Current Opinion in Chemical Engineering*, 26:33–37, December 2019. Publisher: Elsevier BV. URL: https://linkinghub.elsevier.com/retrieve/pii/S2211339819300322, doi:10.1016/j.coche.2019.08.004.

[160] Marwin H. S. Segler and Mark P. Waller. Modelling Chemical Reasoning to Predict and Invent Reactions. *Chemistry – A European Journal*, 23(25):6118–6128, May 2017. URL: https://chemistry-europe.onlinelibrary.wiley.com/doi/10.1002/chem.201604556, doi:10.1002/chem.201604556.

[161] Philipp Sharpf, Moritz Schubotz, and Bela Gipp. Mathematics in Wikidata. In *Proceedings of the 2nd Wikidata Workshop*, volume 2982, pages 1–14, 2021. URL: https://ceur-ws.org/Vol-2982/paper-1.pdf.

[162] André Greiner-Petter, Moritz Schubotz, Corinna Breitinger, Philipp Scharpf, Akiko Aizawa, and Bela Gipp. Do the Math: Making Mathematics in Wikipedia Computable. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4384–4395, April 2023. Publisher: Institute of Electrical and Electronics Engineers (IEEE). URL: https://ieeexplore.ieee.org/document/9847017/, doi:10.1109/tpami.2022.3195261.

[163] Michelene T. H. Chi, Paul J. Feltovich, and Robert Glaser. Categorization and Representation of Physics Problems by Experts and Novices. *Cognitive Science*, 5(2):121–152, April 1981. URL: https://onlinelibrary.wiley.com/doi/10.1207/s15516709cog0502_2, doi:10.1207/s15516709cog0502_2.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[164] Yongchul G. Chung, Diego A. Gómez-Gualdrón, Peng Li, Karson T. Leperi, Pravas Deria, Hongda Zhang, Nicolaas A. Vermeulen, J. Fraser Stoddart, Fengqi You, Joseph T. Hupp, Omar K. Farha, and Randall Q. Snurr. In silico discovery of metal-organic frameworks for precombustion CO_2 capture using a genetic algorithm. *Science Advances*, 2(10):e1600909, October 2016. URL: https://www.science.org/doi/10.1126/sciadv.1600909, doi:10.1126/sciadv.1600909.

[165] Sean P. Collins, Thomas D. Daff, Sarah S. Piotrkowski, and Tom K. Woo. Materials design by evolutionary optimization of functional groups in metal-organic frameworks. *Science Advances*, 2(11):e1600954, November 2016. URL: https://www.science.org/doi/10.1126/sciadv.1600954, doi:10.1126/sciadv.1600954.

[166] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, Perth, WA, Australia, 1995. IEEE. URL: http://ieeexplore.ieee.org/document/488968/, doi:10.1109/ICNN.1995.488968.

[167] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. tex.eprint: https://www.science.org/doi/pdf/10.1126/science.220.4598.671. URL: https://www.science.org/doi/abs/10.1126/science.220.4598.671, doi:10.1126/science.220.4598.671.

[168] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space, September 2013. arXiv:1301.3781 [cs]. URL: http://arxiv.org/abs/1301.3781, doi:10.48550/arXiv.1301.3781.

[169] Vahe Tshitoyan, John Dagdelen, Leigh Weston, Alexander Dunn, Ziqin Rong, Olga Kononova, Kristin A. Persson, Gerbrand Ceder, and Anubhav Jain. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763):95–98, July 2019. URL: https://www.nature.com/articles/s41586-019-1335-8, doi:10.1038/s41586-019-1335-8.

[170] Taylor Webb, Keith J. Holyoak, and Hongjing Lu. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9):1526–1541, September 2023. doi:10.1038/s41562-023-01659-w.

[171] Francois Chollet, Mike Knoop, Gregory Kamradt, and Bryan Landers. ARC Prize 2024: Technical Report, January 2025. arXiv:2412.04604 [cs]. URL: http://arxiv.org/abs/2412.04604, doi:10.48550/arXiv.2412.04604.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

[172] Jakub Lála, Odhran O'Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G. Rodriques, and Andrew D. White. PaperQA: Retrieval-Augmented Generative Agent for Scientific Research, December 2023. arXiv:2312.07559 [cs]. URL: http://arxiv.org/abs/2312.07559.

https://doi.org/10.26434/chemrxiv-2025-q9bb1 **ORCID:** https://orcid.org/0000-0002-0100-0227 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0