Enabling Transparent Problem Solving in Thermodynamics with Ontologies and Knowledge Graphs

Luisa Vollmer[1,*], Rébecca Loubet[2], Fabian Jirasek[2], Sophie Fellenz[3], Hans Hasse[2] and Heike Leitte[1]

[1]*Visual Information Analysis Research Group (VIA), RPTU Kaiserslautern, 67663 Kaiserslautern, Germany*
[2]*Laboratory of Engineering Thermodynamics (LTD), RPTU Kaiserslautern, 67663 Kaiserslautern, Germany*
[3]*Machine Learning Research Group (ML), RPTU Kaiserslautern, 67663 Kaiserslautern, Germany*

Abstract

Solving physics problems, particularly in thermodynamics, often requires navigating complex reasoning processes and selecting appropriate equations, making transparency and interpretability essential for ensuring correct solutions and clear explanations. This paper explores how domain-specific ontologies and knowledge graphs can address these challenges in the context of eXplainable AI (XAI). We introduce an ontology of thermodynamics that encodes essential concepts, attributes, and equations, forming the basis for a dynamic and transparent reasoning process. Based on the ontology and problem-specific user input, a knowledge graph is dynamically constructed which captures the dependencies between concepts and equations, enabling flexible problem-solving across diverse thermodynamic scenarios. The subsequent two-step reasoning process–first identifying computable variables through reachability analysis, and second, filtering the graph to obtain a solution–ensures that the problem-solving steps are traceable and verifiable. The resulting pruned reasoning graph not only holds the computed values, but also provides an interpretable, human-understandable path to the solution. By representing the solution process in a directed acyclic graph, we enable visualization that aids in understanding the model's decision-making and its underlying logic. Experimental results show that the proposed system, KnowTD, efficiently handles important classes of thermodynamic problems, providing accurate and interpretable solutions.

Keywords

Knowledge Graphs, Ontology-Based Reasoning, Explainable AI, Symbolic Computation, Thermodynamic Problem Solving, Physics Education, Intelligent Tutoring Systems, Mathematical Reasoning, Knowledge-aware AI

1. Introduction

Thermodynamics plays a critical role in a wide range of scientific and engineering domains; e.g. it is essential for the design of new energy systems needed to mitigate climate change and, more generally, for developing sustainable production processes. While its principles are well-established, applying them to solve real-world problems often requires expert knowledge to set up mathematical models and solve their equations correctly. To assist with this, a variety of tools have been developed, ranging from engineering equation solvers (e.g., EES [1]) to simulation software (e.g., Aspen Plus [2]) and expert systems designed for specific applications such as power cycles or refrigeration systems. While these tools provide accurate and reliable solutions, they often require manual setup of the mathematical model or have only a limited scope.

More recently, machine learning models and large language models (LLMs) have been explored as potential alternatives. While LLMs demonstrate impressive language understanding capabilities, they often produce unreliable or unverifiable results when applied to scientific and engineering problems that require precise mathematical reasoning [3, 4, 5]. These limitations highlight the need for a knowledge-driven approach that leverages structured domain knowledge to guide problem-solving.

ESWC 2025 Workshops and Tutorials Joint Proceedings

*Corresponding author.
lvollmer@rptu.de (L. Vollmer); loubet@rptu.de (R. Loubet); jirasek@rptu.de (F. Jirasek); fellenz@cs.uni-kl.de (S. Fellenz); hasse@rptu.de (H. Hasse); leitte@cs.uni-kl.de (H. Leitte)
0009-0007-6698-9578 (L. Vollmer); 0009-0003-3633-0749 (R. Loubet); 0000-0002-2502-5701 (F. Jirasek); 0000-0002-5385-39265 (S. Fellenz); 0000-0003-4612-5995 (H. Hasse); 0000-0002-7112-2190 (H. Leitte)

Such an approach should not only compute solutions accurately but also provide transparent, traceable explanations, a core requirement for explainable AI (XAI) in scientific applications.

In response to these challenges, we present KnowTD, a knowledge-driven problem-solving system that leverages ontologies and knowledge graphs (KGs) to solve thermodynamic problems in a flexible, explainable, and traceable manner. At its core, KnowTD employs a newly developed ontology of thermodynamics that encodes key concepts, relationships, and equations. This ontology allows KnowTD to construct a problem-specific KG from structured, manually provided user input, representing a given problem in a thermodynamically valid and machine-interpretable way. A key feature of KnowTD is its ability to extract a structured reasoning graph from the instantiated KG. This directed graph encodes the logical flow of computations, linking input variables, applied equations, and computed variables. Unlike traditional tools that rely on predefined models or manually specified equations, KnowTD provides a transparent, step-by-step explanation of the solution. By formalizing domain knowledge in a machine-readable format and integrating symbolic reasoning, KnowTD enhances both accuracy and interpretability, making it a robust framework for thermodynamic problem-solving and advancing explainable AI in scientific domains. Our main contributions can be summarized as follows:

- **Ontology-Driven Problem Solving and Dynamic Knowledge Graph Construction**: We present an ontology of thermodynamics that formalizes key concepts, equations, and relationships, allowing the dynamic construction of a knowledge graph for solving thermodynamic problems.
- **Automated and Verifiable Reasoning**: KnowTD extracts a structured reasoning graph from the knowledge graph, offering step-by-step, explainable solutions.
- **Generalizable and Adaptive Framework**: KnowTD adapts to a broad range of thermodynamic problems, demonstrating how a flexible knowledge graph framework can support various problem types in scientific and engineering contexts.
- **Advancing XAI in Scientific Problem-Solving**: KnowTD demonstrates how combining structured symbolic reasoning with machine-readable knowledge enhances XAI. By leveraging knowledge graphs, it enables dynamic problem-solving while ensuring correctness and transparency in scientific reasoning.

2. Preliminaries

Thermodynamics is a fundamental discipline in science and engineering. It encompasses some of the most fundamental physical laws and concepts (including energy and entropy), and it is therefore applied in a very wide range of domains, from biology to astronomy. Despite its theoretical foundations being well-established, solving thermodynamic problems often requires expert knowledge to correctly identify relevant concepts, select appropriate equations, and ensure the solution aligns with physical constraints. This complexity makes thermodynamic problem-solving a challenging task for automated systems, especially those seeking to provide transparent and explainable solutions.

Thermodynamics Fundamentals At the core of thermodynamic analysis is the concept of a *thermodynamic system* (or short *system*), which serves as the fundamental framework for describing energy transformations and interactions. A *system* is defined as the part of the universe under study, separated from its surroundings by a boundary. Systems are characterized by *state variables* such as temperature, pressure, volume, and internal energy. A system's behavior is often analyzed by comparing its *initial state* and *final state* after undergoing a *thermodynamic process* of different nature, e.g. isothermal (constant temperature) or adiabatic (no heat exchange).

Thermodynamic Problems Reasoning about such systems is framed in terms of thermodynamic problems, often presented as word problems. These problems provide known values (e.g., initial temperature or volume), unknowns to be calculated, and governing conditions such as the information on the process (e.g. adiabatic, isobaric). A simple example from an introductory thermodynamics course is given in Problem 1.

Problem 1

A gas in a cylinder is compressed reversibly from $v_1 = 0.05\,\mathrm{m}^3/\mathrm{kg}$ *to* $v_2 = 0.02\,\mathrm{m}^3/\mathrm{kg}$. *The initial temperature is* $T_1 = 298\,\mathrm{K}$. *The process is adiabatic. What is the work supplied per kilogram of gas? The gas is ideal with* $R = 287\frac{J}{kgK}$ *and* $c_v = 1010\frac{J}{kgK}$.

Such problems require both identifying the relevant knowledge embedded in the description and deriving the correct solution strategy.

Thermodynamic Problem Solving The first step in solving thermodynamic problems is to extract structured knowledge from natural language text. This involves identifying key concepts such as the system, its states, applicable conditions, and given variable values. Representing this information in a structured format enables automated reasoning in subsequent steps. Once the data is structured, thermodynamic principles are applied to identify appropriate equations for calculating unknown variables. Guided by the system type and identified state variables, this process involves selecting relevant equations such as the *First Law of Thermodynamics* for energy conservation or the *Equation of State* for relating state variables. The resulting system of nonlinear equations can then be solved using modern numerical solvers.

3. Related Work

The integration of knowledge graphs (KGs) and symbolic reasoning with AI has been shown to significantly enhance the interpretability and transparency of machine learning models [6]. While KGs have been successfully employed in various domains to support XAI, the application of such approaches to complex technical fields like thermodynamics remains limited. Thermodynamic problem solving requires not only domain-specific knowledge, but also the ability to reason mathematically and semantically in a coherent manner. In this section, we explore existing work in the respective fields.

Providing knowledge for XAI Integrating KG with machine learning enhances AI transparency and interpretability [6]. While machine learning excels at extracting entities, features, and relationships, KGs provide structured, semantic representations that support reasoning and explanation [7]. Unlike data-driven XAI, which derives explanations from data and model behavior, knowledge-based XAI leverages external domain knowledge and symbolic rules to improve explanations and user understanding [8]. This knowledge can be integrated through human-in-the-loop methods or curated corpora. Task-specific KGs are widely applied: common sense KGs aid in classification, recommendation, and image recognition; factual KGs support prediction tasks; and domain-specific KGs enhance rule-based systems and natural language understanding [6].

KGs and ontologies also explain complex processes by describing phenomena, their influences, and potential effects. Jihen et al. [9] combine a plant disease ontology with concept explainability methods to clarify deep learning decisions. Mäkelburg et al. [10] model invoice terminology in an ontology, representing invoices as KGs and using SHACL constraints for data validation, reducing manual effort while ensuring correctness. Violations of these constraints provide interpretable explanations for validation issues. Tailhardat et al. [11] introduce an ontology for modeling infrastructure, events, and diagnostics in ICT systems, helping to detect anomalies and analyze root causes.

Ontology-based problem solving The Ontology Rela-Model [12] is a knowledge model for integrating knowledge-based systems using ontologies as the knowledge kernel. It enables reasoning across multiple domains, such as linear algebra and graph theory, producing explainable solutions in educational applications. Building on this foundation, the Rela-Funcs Model [13] extends the approach by incorporating functional knowledge. MathGraph [14] focuses on analytical problems, while other systems address discrete and geometric problems [12, 15, 16], with geometric solutions represented graphically [15]. This integration generates explainable solutions, tracking which knowledge was used and how it contributed to the final solution, mirroring human problem-solving and enhancing transparency [12]. Existing ontologies, such as EngMath [17] and PhysSys2 [18], formalize mathematical and engineering concepts, but do not address general thermodynamic problem solving, leaving a gap for an ontology-based solution.

Using LLMs for problem-solving LLMs like ChatGPT and Google Bard have been studied for mathematical use cases [4, 3, 19]. Wardat et al. [19] find that ChatGPT struggles with geometry and complex problems. Plevris et al. [3] show mixed results and note AI hallucinations, stressing the need for more reliable responses. Frieder et al. [4] conclude that ChatGPT and GPT-4 are good for querying facts but fail with graduate-level math problems. Venkatasubramanian [20] critique LLMs' limitations in understanding and reasoning, suggesting integrating geometric and algebraic knowledge for better AI capabilities. In previous work, we examined the thermodynamic problem-solving skills of state-of-the-art LLMs [5] and could confirm the general weakpoints of LLMs also for thermodynamics. In addition we found that newer models are gaining more reasoning abilities but still fail to apply thermodynamic laws reliably.

4. Methods

In the following, we present the design and methodological innovations of our knowledge-driven problem-solving system, KnowTD, which leverages a novel ontology and dynamic KGs (section 4.1) to solve thermodynamic problems in a transparent and explainable manner. KnowTD follows a four-step process, as illustrated in Fig. 1. The first step, problem specification (section 4.2), ensures the user-defined problem aligns with the ontology. Based on this input, KnowTD constructs a KG that contains relevant instances of concepts, attributes, variables, equations, and rules (section 4.2). This graph serves as the foundation for the system's reasoning process, ensuring KnowTD can dynamically adapt to diverse problem contexts. In the mathematical reasoning step (section 4.3), KnowTD analyzes a subset of the knowledge graph – specifically variables, equations, and rules – to compute additional variables. Finally, the system visualizes the solution as a flow diagram, a directed subset of the knowledge graph that outlines the steps required to compute the desired results (section 4.4).

4.1. Ontology and Knowledge Graph Design

In close collaboration with domain experts, we developed the KnowTD ontology following a user-centered design methodology combined with ontology engineering methods. The KnowTD ontology is not only a structured representation of domain knowledge but also a mechanism to guide reasoning, validate input, and bridge semantic and mathematical problem-solving approaches. To enable its application in the various steps of the KnowTD-pipeline, we made the following design choices:

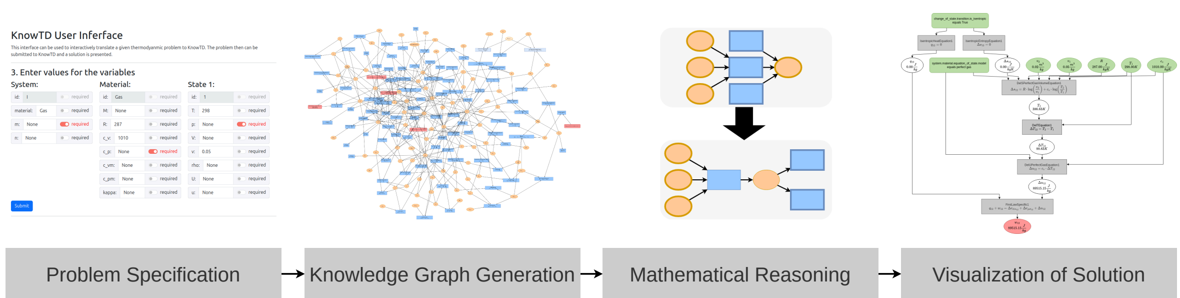

Figure 1: Pipeline underlying KnowTD: (i) The problem specification is supported via knowledge derived from the ontology. (ii) A knowledge graph is dynamically generated based on the user input. (iii) Additional values are computed using a combination of graph traversal and symbolic solving. (iv) The obtained solution is represented as a directed graph.

- *Modularity:* The ontology models thermodynamic concepts, variables, and equations as independent entities, allowing flexible recombination to match diverse problem scenarios.
- *Extensibility:* KnowTD starts with a very limited scope of thermodynamics, namely problems related to a change of state of a closed system containing an ideal gas, but is designed to be extensible and cover additional knowledge areas as the scope grows.
- *Separation of Declarative and Procedural Knowledge:* By distinguishing between factual knowledge (e.g., system definitions, thermodynamic laws) and procedural steps for deriving unknowns, the ontology improves traceability and supports step-by-step explanations.
- *Context-Driven Rule Definition:* Thermodynamic principles are linked to specific conditions, ensuring that KnowTD dynamically applies relevant equations based on the problem context.
- *Alignment with Standards:* The ontology integrates established thermodynamic standards such as SI units and aligns key concepts with Wikidata to ensure consistency and interoperability.

Components The KnowTD ontology is composed of several key components that represent both factual and procedural thermodynamic knowledge:

- *Concepts* form the foundation of the ontology, representing the primary entities involved in thermodynamics, such as *system*, *state*, and *process*. These concepts define the core structure of the knowledge representation, dictating how different elements of a thermodynamic problem are interconnected.
- *Variables* represent measurable properties that are associated with concepts, such as *temperature*, *pressure*, and *volume*. Each variable is defined by its name, unit, symbol, and value, ensuring that every element in a problem can be quantified appropriately.
- *Attributes* characterize non-numeric aspects of concepts, such as whether a system is in *equilibrium* or whether a process is *adiabatic*. These attributes play a crucial role in determining the behavior of the system and influencing the applicability of specific laws and equations.
- *Equations* describe the relationships between variables, linking concepts and defining how properties of a system change under different conditions. For example, the ideal gas law links pressure, volume, and temperature and is applied depending on whether the system is an ideal gas.
- *Rules and Constraints* are integral to the ontology, ensuring that thermodynamic laws are applied correctly based on the context of the problem. These rules govern when specific equations can be used and define conditions such as whether a process is isothermal, adiabatic, or isochoric.

Reuse and Adaptation of Ontologies and Design Patterns Following best practices in ontology development [21, 22], we aim to reuse existing data models, vocabularies, and design patterns as a foundation, adapting and extending them to represent domain-specific classes and properties. Drawing from the *Rela-Ops Model* [23], we distinguish between relations and operations: relations link ontology elements, while operations such as *derive*, *apply*, and *transform* are modeled through attributes, rules, or dedicated concepts (e.g., concept transition). Inspired by *OntoMath* [24], we represent only classes in the ontology; individuals such as specific values or occurrences in problem statements are treated as instances during reasoning. Constant variables such as the absolute zero temperature (T_0), are modeled as classes. Similar to *OntoKin* [25], we distinguish between data properties (e.g., variables and attributes) and object properties (e.g., conceptual references). Where possible, we align elements of our ontology with existing domain ontologies and thermodynamic standards such as *SI units* to ensure interoperability and reuse. We reuse vocabulary and definitions from *Wikidata*[1] by mapping the concepts, attributes, and variables of our ontology to their related entry in Wikidata where possible.

While several physics- and chemistry-related ontologies exist [26, 27, 28, 29, 30], they lack thermodynamic theorems or equations and are not designed for thermodynamic reasoning. Other ontologies [30, 25, 31] focus on adjacent fields, such as reaction mechanisms or material properties, which exceed our current scope but may be considered in future extensions. We introduced this ontology to

[1]https://www.wikidata.org/

the thermodynamics community in [32], where we focused on its ability to model the complex structure of thermodynamics theory. In this paper, we detail its role in computing and explaining KG-based solutions.

Facets The ontology is organized into key facets that are essential for dynamic problem solving: (i) *Problem Definition Facet*, which structures the problem from user inputs by identifying relevant concepts, variables, and conditions—avoiding reliance on fixed templates; (ii) *Inference Facet*, which enables reasoning over the problem and infers missing concepts or equations based on given inputs; (iii) *Mathematical Solving Facet*, which connects equations to variables and laws, providing a framework for computing unknowns using both declarative (thermodynamic laws) and procedural knowledge (steps to apply these laws).

Implementation The thermodynamics ontology is encoded using the LinkML schema language [33] and can be converted to a variety of formats, including OWL [34], RDF [35], Python data classes, and schemas for databases using the tools in the LinkML ecosystem. KnowTD is implemented in Python and is available online[2]. LinkML offers a native schemaviewer for Python which allows to parse the ontology and use information on classes, inheritance, and relations.

4.2. Ontology-based KG Building

KnowTD utilizes the ontology as a schema (T-Box) to guide problem definition and the dynamic generation of a problem-specific knowledge graph. This graph instantiates concrete individuals from the given problem (A-Box) while adhering to the structural definitions provided by the T-Box.

Problem Definition and Validation The base class for problem definitions is *ThermodynamicProblem* which defines the schema for valid problem formulations. To ease usability, we introduce three specialized subclasses with preconfigured base processes: *SteadyStateProcess* (representing systems in steady-state conditions), *SequentialStepProcess* (representing systems undergoing a sequence of one or more state changes), and *CyclicProcess* (a *SequentialStepProcess* that returns to its initial state).

Problems can be specified manually by the user in any format supported by LinkML (cf. section 4.1), with YAML being the preferred format due to its machine readability and accessibility for non-programmers. Additionally, an interactive user dialogue is available, as demonstrated in the KnowTD system [32]. For input validation, we utilize the LinkML ecosystem's built-in validator [33], which provides comprehensive error feedback to ensure correctness. An LLM-supported input dialogue is planned future work.

Dynamic Knowledge Graph Building To populate the KG with instances for the specified problem, we adopt a systematic instance generation process that ensures completeness, structural integrity, and consistency across interlinked classes. The process creates instances for all mandatory classes defined in the ontology, including those not explicitly specified in the input data to obtain a valid system specification. This is achieved by traversing class dependencies and instantiating referenced entities to satisfy cardinality constraints and maintain referential integrity. For each instantiated concept, available variables and attributes are populated with provided values where specified. Otherwise, defaults from the ontology schema, such as constants or derived attributes, are applied. Elements without defaults are initialized as None and may be resolved during mathematical reasoning.

A crucial step in the instance generation process is the *concept-scoped indexing* of variables, ensuring alignment with thermodynamic conventions. In our model, variables are defined as distinct classes linked to their associated concepts. Each concept is assigned an index, which is then inherited by its related variables. For example, the temperature of the initial state is labeled T_1, while the temperature of the final state is labeled T_2. To automate this process, we employ a *concept-scoped indexing mechanism*

[2]https://gitlab.rhrk.uni-kl.de/knowtd/knowtd/

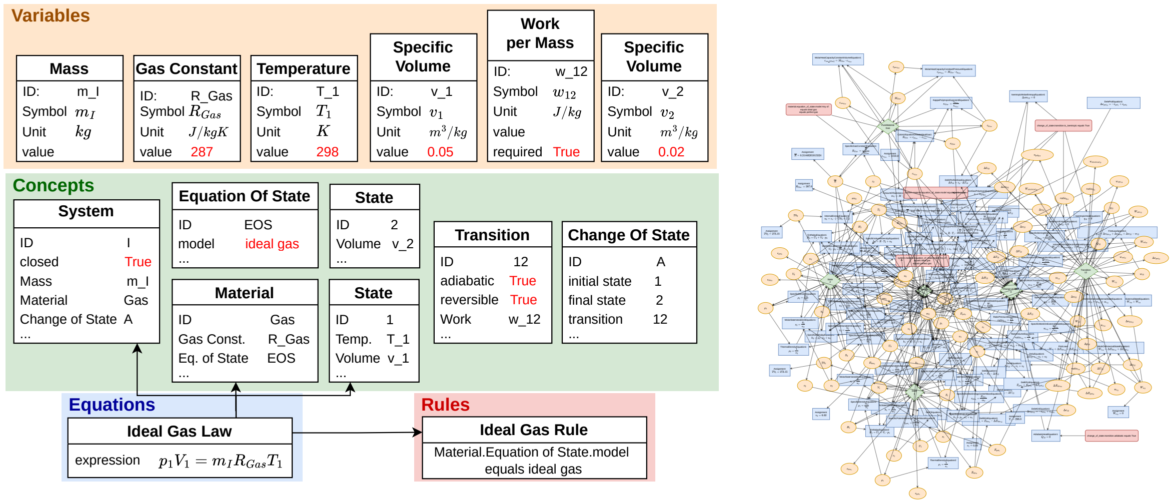

Figure 2: Representation of the knowledge graph: (left) Class diagram of nodes as defined in the ontology, attributes are directly encoded in the concept nodes. Given values are marked red. (right) Knowledge graph for a given problem containing concepts, variables, equations, and rules.

that systematically assigns indices to variables based on the index of their associated concept, ensuring consistency and adherence to domain-specific conventions.

While the user specifies only the system under consideration, the instantiation of applicable equations is crucial for the reasoning process and must be performed dynamically. The set of valid thermodynamic equations is represented as distinct classes within the ontology. Each equation class is annotated with metadata specifying concepts for which it is applicable and the preconditions that must be met. These preconditions are formalized as ontology rules using the LinkML rule language.

To integrate equations, the ontology is queried for the linked concepts and rules. For each combination of instantiated concepts, the associated rules are checked. If the rules are satisfied, the equation is instantiated by adjusting its expression to match the variables of the corresponding concepts. In the knowledge graph, the equation node is then linked to the relevant concepts, variables, and rules. The resulting KG contains all relevant concepts, variables, and equations as illustrated in fig. 2. Critical relations for further processing are the ones connecting *concepts* to *variables* and *equations* to *variables*.

4.3. Mathematical Reasoning for Problem Solving

The mathematical reasoning process determines the solvability of a thermodynamic problem based on the available information. It operates on a dedicated subgraph of the KG, referred to as the reasoning graph, which comprises variables and equations with directed edges representing information flow. Unlike the full KG, which encodes comprehensive thermodynamic knowledge, the reasoning graph is structured to facilitate equation selection and computation.

The solution process involves two steps: constructing a reachability graph to identify computable variables and extracting the minimal subgraph required to compute the target variables. KnowTD employs a custom heuristic inspired by structural analysis methods for large equation systems. This single equation traversal method follows a greedy strategy that prioritizes the shortest solution graph.

In the first phase, the reachability graph is constructed via a breadth-first traversal starting from known variables. Equations with exactly one unknown (free) variable are progressively added, with identified unknowns marked as computable. Directed edges are reoriented to reflect information flow from inputs to outputs. SymPy is used to simplify equations, identify unknown variables, and compute values. This process continues iteratively until no further variables can be computed.

The second phase extracts the minimal subgraph required to compute the requested variables. After confirming that all required variables are present in the reachability graph, a backward traversal identifies the relevant ancestor nodes. The resulting directed subgraph links known inputs to target outputs through the necessary intermediate steps. We refer to this subgraph as the solution graph.

4.4. Explaining the Solution

The solution graph, generated in the previous step, contains both the computed values for new variables and the instructions for reproducing the solution. This directed acyclic graph (DAG) has given attributes, the satisfied rules and known variables as root nodes. Its bipartite structure alternates between equations and variables, indicating dependencies and computable values. Equations are also linked to the rules that justify their application. For visualization, we use the DOT layout algorithm [36], which arranges the nodes in layers with edges flowing from top to bottom. As shown in Fig. 3, color-coded node types help distinguish given, computed, and required elements. Each node displays relevant information from the knowledge graph, including names, values, and units for variables, and ontology-defined names and expressions for equations.

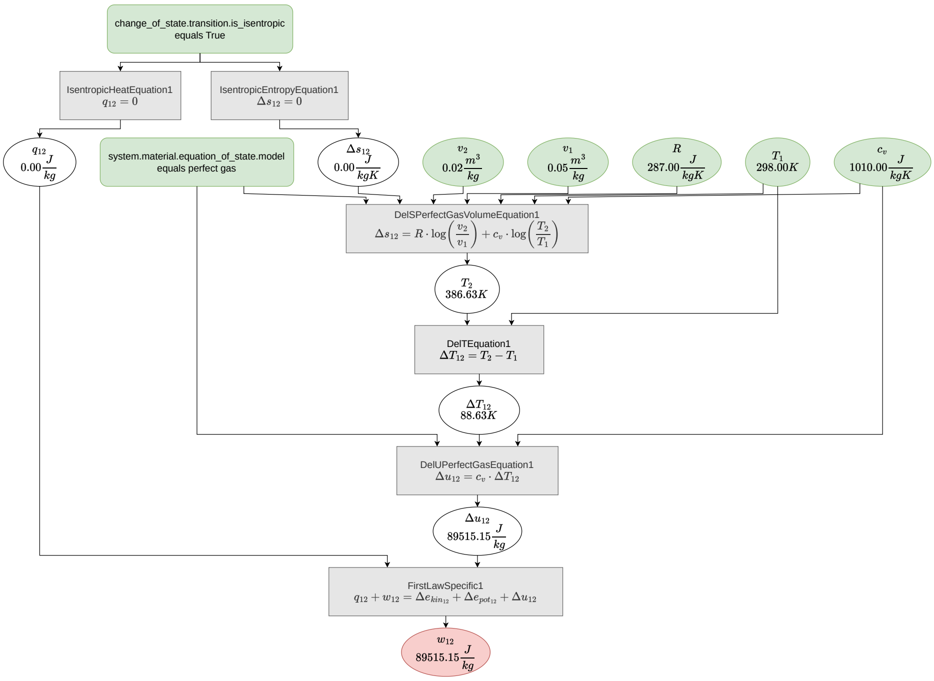

Figure 3: Solution graph of a prototypical problem: The solution is read from top to bottom. Green indicates given information and gray marks equations that are used. Following the paths down new variables (white) are computed until the required variable (red) is reached.

5. Evaluation

In this chapter, we evaluate KnowTD's performance in solving thermodynamic benchmark problems and compare it with LLM-based approaches.

5.1. Dataset

In this study, we evaluate KnowTD and LLM-based approaches using a dataset of 13 thermodynamic benchmark problems. These problems, carefully curated by domain experts, are representative of introductory engineering thermodynamics courses, requiring the systematic application of multiple thermodynamic principles to compute numerical values. While they are relatively simple for trained individuals, they still demand structured reasoning and precise calculations, making them suitable for assessing problem-solving capabilities.

Each problem is defined by text with a well-specified solution and maintains a fixed structure, unlike real-world problems that often vary in wording, numerical values, or context. Although infinite variants could be generated by modifying numerical values or rewording descriptions using LLMs, the underlying thermodynamic scope and required solution steps remain consistent. In this study, we focus on the 13 prototypical questions without generating additional variants. For KnowTD, we provide ontology-conformant YAML representations of the problem to ensure correct input for the reasoning process which is the focus of this study. The YAML files are included in the source code (4.1).

Key challenges in solving these problems include their multi-step nature, which requires combining multiple thermodynamic laws and equations, and the need for numerical precision, where minor computational errors can lead to incorrect results.

5.2. Evaluation Metric

To evaluate the performance of the KnowTD and LLM-based approaches, we used a structured evaluation method conducted by experienced thermodynamics experts. Solutions were assessed in an exam-like setting based on four key criteria: *numerical accuracy*, where correct numerical results were important; *solution graph correctness*, which examined the logical progression of steps with partial credit for correctly executed steps; *thermodynamic validity*, ensuring that the applied equations adhered to established laws and principles; and *appropriateness of equation application*, which verified that selected equations were correctly applied based on the problem's constraints and conditions.

5.3. Baseline: Problem Solving with LLMs

We previously used this problem set and evaluation metric to evaluate the thermodynamic problem-solving abilities of GPT-3.5, GPT-4, GPT-4o (OpenAI), LLaMA 3.1 (Meta), and Le Chat (MistralAI) [5]. To date, no LLM system has been developed that is specifically tailored to this use case. While GPT-4 and GPT-4o achieved the highest scores (percentage of trials with full score: GPT-4: 74%, GPT-4o: 64%), none of the models consistently produced correct results across multiple attempts. Frequent errors included incorrect physical assumptions, the use of invalid equations, inconsistent signs, and numerical errors. While the latter two may be mitigated with external solvers, the former highlight a lack of deep domain understanding. Full evaluation details and prompts are provided in [5].

5.4. Problem Solving, Explanation, and Analysis Using KGs

The problems were deliberately designed to align with the implemented scope of KnowTD and its underlying thermodynamics ontology. Since all problems were correctly formulated and translated into an ontology-compliant input format by domain experts, KnowTD successfully achieved full scores across all problems.

Beyond its problem-solving capabilities, KnowTD offers detailed insight into the complexity of various problems and the corresponding reasoning required for their solution, which can now be quantitatively analyzed. Table 1 presents an overview of the sizes of different graph structures employed in KnowTD. For each problem, we report the number of instances (i.e., nodes) in both the Knowledge Graph (KG) and the Solution Graph (SG), with further breakdowns for each ontology base class.

The initial KG comprises all node types (see Total / KG for overall node count), with individual counts provided for each class. Additionally, for each class, we specify the number of given instances (giv) that were provided as input. Our analysis reveals that users are required to identify between one and four attributes and eight to eleven variables per problem. Each problem involves a fixed set of seven concepts as part of a single-step process (not listed in the table), which is characterized by 27 attributes distributed across relevant ontology classes (also not listed in the table). This results in an overall KG size ranging between 205 and 216 nodes. The corresponding SG, identified by KnowTD, includes all the attributes, variables, equations, and rules necessary to derive the correct solution. As reported under Total / SG, the solution graphs are notably smaller, averaging 29 nodes. This demonstrates a significant reduction in the reasoning space, shrinking from an average of 210 nodes in the KG to 29 nodes in the SG which highlights KnowTD's effective reasoning capabilities, efficiently isolating the critical elements required for correct thermodynamic problem-solving.

Table 1
Overview of graph sizes in KnowTD for each evaluated problem. The table details the number of nodes in the Knowledge Graph (KG) and the Solution Graph (SG), with additional breakdowns by ontology base classes. Given input values are marked as giv.

		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	Avg.
Attributes	giv	4	2	2	1	2	2	2	2	2	2	2	2	3	**2**
Variables	giv	9	10	8	11	9	11	9	10	9	9	8	11	9	**9**
	KG	84	84	84	84	84	84	84	84	84	84	84	84	84	**84**
	SG	13	22	14	17	17	15	11	14	13	17	15	18	13	**15**
Equations	KG	89	86	93	83	88	83	88	88	88	84	89	84	91	**87**
	SG	8	13	9	8	11	8	5	7	8	10	12	11	6	**9**
Rules	KG	6	5	5	4	5	4	5	5	5	5	5	5	6	**5**
	SG	3	4	4	3	4	3	4	3	3	3	3	5	3	**3**
Total	KG	213	209	216	205	211	205	211	211	211	207	212	207	216	**210**
	SG	28	41	29	31	34	30	26	28	28	32	32	36	25	**29**

6. Conclusion and Future Work

This paper introduces and details the development of a novel ontology and dynamic knowledge graph framework that enables KnowTD to ensure correctness, verifiability, and interpretability in thermodynamic problem-solving. By combining structured domain knowledge with symbolic reasoning, KnowTD effectively addresses complex scientific problem-solving while ensuring traceable and verifiable solutions.

Our evaluation shows that KnowTD accurately solves diverse thermodynamic problems while consistently applying thermodynamic laws with domain fidelity. Compared to large language models, KnowTD provides more reliable, context-aware solutions with clear, step-by-step reasoning paths. The resulting structured reasoning graph enhances interpretability, offering human-understandable explanations that align with core Explainable AI (XAI) principles.

KnowTD's dynamic knowledge graph construction directly supports the creation of context-aware, semantic explanations, improving user trust and understanding. The graph's structure encodes causal dependencies between thermodynamic concepts, enabling verifiable insights crucial for scientific domains.

To extend KnowTD's capabilities, future work will expand its ontology to include advanced thermodynamic concepts such as non-ideal systems, multi-phase processes, and transient phenomena. To improve usability and automation, we plan to incorporate large language models for natural language processing tasks such as extracting and structuring problem statements. This hybrid approach will bridge symbolic and neural reasoning while ensuring adherence to valid thermodynamic knowledge, reinforcing fairness and trustworthiness in AI systems. Additionally, we aim to evaluate the intuitiveness and practical utility of the explanations generated by KnowTD, assessing how well they support user understanding and transparency in problem-solving.

KnowTD exemplifies how knowledge graphs, enriched with ontological reasoning and symbolic logic, can advance transparent, context-aware, and human-centric explanations–contributing to best practices in building interpretable AI models using knowledge graphs.

Acknowledgments

We gratefully acknowledge the funding by Deutsche Forschungsgemeinschaft DFG in the context of the Priority Program 2331 "Machine Learning in Chemical Engineering" and Research Unit 5359 "KI-FOR:

Deep Learning on Sparse Chemical Process Data".

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT 4.o and Grammarly in order to: Improve writing style and Grammar and spelling check. After using this tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] S. A. Klein, F. S. Alvarado, Engineering Equation Solver (EES), F-Chart Software, Middleton, WI, USA, 2025. https://fchartsoftware.com/ees/.

[2] Aspen Technology, Inc., Aspen Plus, Aspen Technology, Inc., Bedford, MA, USA, 2025. https://www.aspentech.com/en/products/engineering/aspen-plus.

[3] V. Plevris, G. Papazafeiropoulos, A. Jiménez Rios, Chatbots put to the test in math and logic problems: a comparison and assessment of chatgpt-3.5, chatgpt-4, and google bard, AI 4 (2023) 949–969. doi:10.3390/ai4040048.

[4] S. Frieder, L. Pinchetti, A. Chevalier, R. Griffiths, T. Salvatori, T. Lukasiewicz, P. Petersen, J. Berner, Mathematical capabilities of ChatGPT, in: Advances in Neural Information Processing Systems 36, NeurIPS, 2023. doi:10.5555/3666122.3667327.

[5] R. Loubet, P. Zittlau, L. Vollmer, M. Hoffmann, S. Fellenz, F. Jirasek, H. Leitte, H. Hasse, Using large language models for solving thermodynamic problems, 2025. arXiv:2502.05195.

[6] I. Tiddi, S. Schlobach, Knowledge graphs as tools for explainable machine learning: A survey, Artificial Intelligence 302 (2022) 103627. doi:10.1016/j.artint.2021.103627.

[7] E. Rajabi, K. Etminani, Knowledge-graph-based explainable AI: A systematic review, Journal of Information Science 50 (2024) 1019–1029. doi:10.1177/01655515221112844.

[8] X.-H. Li, C. C. Cao, Y. Shi, W. Bai, H. Gao, L. Qiu, C. Wang, Y. Gao, S. Zhang, X. Xue, L. Chen, A Survey of Data-Driven and Knowledge-Aware eXplainable AI, IEEE Transactions on Knowledge and Data Engineering 34 (2022) 29–49. doi:10.1109/TKDE.2020.2983930, conference Name: IEEE Transactions on Knowledge and Data Engineering.

[9] J. Amara, S. Samuel, B. König-Ries, Integrating domain knowledge for enhanced concept model explainability in plant disease classification, in: The Semantic Web, Springer Nature Switzerland, Cham, 2024, pp. 289–306. doi:10.1007/978-3-031-60626-7_16.

[10] J. Mäkelburg, C. John, M. Acosta, Automation of electronic invoice validation using knowledge graph technologies, in: The Semantic Web, Springer Nature Switzerland, 2024, pp. 253–269. doi:10.1007/978-3-031-60626-7_14.

[11] L. Tailhardat, Y. Chabot, R. Troncy, Noria-o: An ontology for anomaly detection and incident management in ict systems, in: The Semantic Web, Springer Nature Switzerland, Cham, 2024, pp. 21–39. doi:10.1007/978-3-031-60635-9_2.

[12] N. V. Do, H. D. Nguyen, T. T. Mai, A method of ontology integration for designing intelligent problem solvers, Applied Sciences 9 (2019). doi:10.3390/app9183793.

[13] N. P. Tran, H. D. Nguyen, D. Nguyen, D. A. Tran, A. T. Huynh, T. T. Le, A Method for Integrating of Knowledge Model and Functional Component and Application in Intelligent Problem Solver, in: Advances and Trends in Artificial Intelligence. Theory and Applications, Springer Nature, Singapore, 2024, pp. 149–162. doi:10.1007/978-981-97-4677-4_13.

[14] T. Zhao, Y. Huang, S. Yang, Y. Luo, J. Feng, Y. Wang, H. Yuan, K. Pan, K. Li, H. Li, F. Zhu, Mathgraph: A knowledge graph for automatically solving mathematical exercises, in: Database Systems for Advanced Applications, Springer International Publishing, 2019, pp. 760–776. doi:10.1007/978-3-030-18576-3_45.

[15] H. D. Nguyen, D. Nguyen, V. T. Pham, Design an intelligent problem solver in solid geometry

based on knowledge model about relations, in: Eighth International Conference on Knowledge and Systems Engineering (KSE), 2016, pp. 150–155. doi:10.1109/KSE.2016.7758045.
[16] S. S. Kurbatov, I. B. Fominykh, A. B. Vorobyev, Ontology-controlled geometric solver, in: Artificial Intelligence, 2020, p. 262–273. doi:10.1007/978-3-030-59535-7_19.
[17] T. R. Gruber, G. R. Olsen, An ontology for engineering mathematics, in: Principles of Knowledge Representation and Reasoning, 1994, pp. 258–269. doi:10.1016/B978-1-4832-1452-8.50120-2.
[18] P. Borst, H. Akkermans, J. Top, Engineering ontologies, International Journal of Human-Computer Studies 46 (1997) 365–406. doi:https://doi.org/10.1006/ijhc.1996.0096.
[19] Y. Wardat, M. A. Tashtoush, R. AlAli, A. M. Jarrah, Chatgpt: A revolutionary tool for teaching and learning mathematics, Eurasia Journal of Mathematics, Science and Technology Education 19 (2023). doi:10.29333/ejmste/13272.
[20] V. Venkatasubramanian, What's in an embedding? would a rose by any embedding smell as sweet?, 2024. URL: https://arxiv.org/abs/2406.06870. arXiv:2406.06870.
[21] T. R. Gruber, A translation approach to portable ontology specifications, Knowledge Acquisition 5 (1993) 199–220. doi:10.1006/knac.1993.1008.
[22] D. Calvanese, G. De Giacomo, M. Lenzerini, A framework for ontology integration, volume 75 of *Frontiers in Artificial Intelligence and Applications*, 2002, p. 14. doi:10.5555/2956602.2956623.
[23] H. D. Nguyen, N. V. Do, V. T. Pham, Rela-Ops Model: A Method for Knowledge Representation and Application, IOS Press, 2018. doi:10.3233/978-1-61499-900-3-825.
[24] O. A. Nevzorova, N. Zhiltsov, A. Kirillovich, E. Lipachev, Ontomath pro ontology: a linked data hub for mathematics, in: Knowledge Engineering and the Semantic Web, Springer International Publishing, 2014, pp. 105–119. doi:10.1007/978-3-319-11716-4_9.
[25] F. Farazi, J. Akroyd, S. Mosbach, P. Buerger, D. Nurkowski, M. Salamanca, M. Kraft, Ontokin: An ontology for chemical kinetic reaction mechanisms, Journal of Chemical Information and Modeling 60 (2020) 108–120. doi:10.1021/acs.jcim.9b00960.
[26] R. Borst, H. Akkermans, A. Pos, J. Top, The physsys ontology for physical systems, Proceedings Workshop Qualitative Reasoning '95, Amsterdam, NL: 11-21 (1995).
[27] B. Norton, M. J. Normann, Digital twins and semantic data fusion for security in a healthcare environment, in: SeDiT@ESWC, volume 2615, 2020.
[28] K. Degtyarenko, P. de Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, R. Alcantara, M. Darsow, M. Guedj, M. Ashburner, ChEBI: a database and ontology for chemical entities of biological interest, Nucleic Acids Research 36 (2007) D344–D350. doi:10.1093/nar/gkm791.
[29] D. L. Cook, F. L. Bookstein, J. H. Gennari, Physical properties of biological entities: An introduction to the ontology of physics for biology 6 (2011) 1–8. doi:10.1371/journal.pone.0028708.
[30] J. Morbach, A. Yang, W. Marquardt, Ontocape — a large-scale ontology for chemical process engineering, Engineering Applications of Artificial Intelligence 20 (2007) 147–161. doi:10.1016/j.engappai.2006.06.010.
[31] D. Mrdjenovich, M. K. Horton, J. H. Montoya, C. M. Legaspi, S. Dwaraknath, V. Tshitoyan, A. Jain, K. A. Persson, Propnet: A knowledge graph for materials science, Matter 2 (2020) 464–480. doi:10.1016/j.matt.2019.11.013.
[32] L. Vollmer, S. Fellenz, F. Jirasek, H. Leitte, H. Hasse, KnowTD-an actionable knowledge representation system for thermodynamics, Journal of Chemical Information and Modeling 64 (2024) 5878–5887. doi:10.1021/acs.jcim.4c00647.
[33] S. Moxon, H. Solbrig, D. Unni, D. Jiao, R. Bruskiewich, J. Balhoff, G. Vaidya, W. Duncan, H. Hegde, M. Miller, M. Brush, N. Harris, M. Haendel, C. Mungall, The linked data modeling language (linkml): A general-purpose data modeling framework grounded in machine-readable semantics, CEUR Workshop Proceedings 3073 (2021) 148–151. URL: https://ceur-ws.org/Vol-3073/paper24.pdf.
[34] G. Antoniou, F. van Harmelen, Web Ontology Language: OWL, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 67–92. doi:10.1007/978-3-540-24750-0_4.
[35] P. J. Hayes, P. F. Patel-Schneider, Resource description framework 1.1, 2014. URL: https://www.w3.org/TR/rdf11-mt/.
[36] E. Gansner, E. Koutsofios, S. North, Drawing graphs with dot, 2006.