UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

UNIVERSITY OF PADUA
DEPARTMENT OF INFORMATION ENGINEERING

MASTER'S DEGREE IN COMPUTER ENGINEERING

"Evidence-Grounded Evaluation of Materials Science Knowledge Graph"

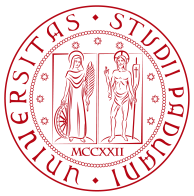
Supervisor: Prof. Gianmaria Silvello

Candidate: Zahra Rahgooy
Student ID: 2085445

Co-supervisor: Dr. Sören Auer

ACADEMIC YEAR 2025-2026

Graduation date 13.04.2026



University of Padova

Department of Information Engineering.

Master Thesis in Computer Engineering

Evidence-Grounded Evaluation of Materials Science Knowledge Graph

Supervisor

Prof. Gianmaria Silvello

University of Padova

Co-supervisor

Prof. Dr. Sören Auer

TIB – Leibniz Information Centre for Science and Technology

Master Candidate

Zahra Rahgooy

Student ID: 2085445

Academic Year

2025-2026

Dedication.

To my family, for their endless love and support.

And to my friends, who stood by me through every step of this journey.

Abstract

The exponential growth of scientific literature has created a pressing need for automated methods capable of extracting and validating domain knowledge reliably and transparently. This thesis presents an evidence-grounded framework for verifying structured scientific facts extracted from Atomic Layer Deposition (ALD) literature. Although large language models can extract candidate facts from scientific papers, these outputs must be validated against the original source text before they can be used in downstream scientific analysis. To support trustworthy verification, the proposed framework separates evidence extraction from evaluation and grounds each judgment in verbatim textual evidence retrieved from the corresponding paper.

The methodology combines dataset preparation with a two-stage retrieval-and-judgment pipeline. Full-text papers and GPT-generated factual annotations from the AWASES-ALD dataset are first normalized, validated, and aligned through a common identifier. Evidence extraction is then performed using a hybrid approach that ranks candidate sentences through TF–IDF similarity and verifies them with a large language model, returning either verbatim supporting evidence or an explicit "no direct evidence found" outcome. In the evaluation stage, a second model assesses each fact–evidence pair using a structured schema that assigns a support category (direct, partial, or none) together with relevance and accuracy scores. Four extractor–evaluator configurations were examined across proprietary and open-source model families: GPT-4/GPT-3.5 and Llama-3.1-8B/Qwen2.5-7B.

Experiments were conducted on a subset of 141 AWASES-ALD papers containing 2,195 extracted facts. Results show that all configurations can retrieve and evaluate evidence at scale but differ in calibration. The Llama–Qwen configuration achieved the highest evidence retrieval coverage, while proprietary configurations produced higher proportions of direct support and more stable evaluator behavior. Evaluator reliability was further assessed on a human-annotated ZnO dataset containing 327 fact–evidence pairs, where all models successfully recognized true support relationships, with GPT-3.5 achieving the highest direct agreement.

Overall, the thesis demonstrates that evidence-grounded verification improves the transparency and reliability of automated scientific fact validation. By combining sentence-level provenance, structured LLM judgments, controlled model comparisons, and external benchmarking, the proposed framework provides a scalable approach for evaluating materials-science facts and supporting knowledge-graph quality assurance.

Sommario

La crescita esponenziale della letteratura scientifica richiede metodi automatizzati capaci di estrarre e validare conoscenza di dominio in modo affidabile e trasparente. Questa tesi presenta un framework basato su evidenza testuale per la verifica di fatti scientifici strutturati estratti dalla letteratura sull'Atomic Layer Deposition (ALD). Sebbene i modelli linguistici di grandi dimensioni possano estrarre fatti candidati dagli articoli scientifici, tali risultati devono essere verificati rispetto al testo originale prima di poter essere utilizzati in analisi scientifiche successive. Per garantire una verifica affidabile, il framework proposto separa l'estrazione dell'evidenza dalla fase di valutazione e basa ogni giudizio su evidenze testuali riportate verbatim dai documenti di origine.

La metodologia combina la preparazione del dataset con una pipeline di recupero e valutazione articolata in due fasi. I testi completi degli articoli e le annotazioni fattuali generate automaticamente nel dataset AWASES-ALD vengono normalizzati, validati e allineati tramite un identificatore comune. L'estrazione dell'evidenza viene quindi effettuata mediante un approccio ibrido che classifica le frasi candidate tramite similarità TF–IDF e utilizza un modello linguistico per verificarne la rilevanza rispetto al fatto considerato, restituendo frasi di supporto riportate verbatim oppure una risposta esplicita di assenza di evidenza ("no direct evidence found"). Nella fase di valutazione, un secondo modello analizza ogni coppia fatto–evidenza attraverso uno schema strutturato che assegna una categoria di supporto (direct, partial oppure none) insieme a punteggi di rilevanza e accuratezza. Sono state analizzate quattro configurazioni di modelli estrattore–valutatore appartenenti sia alla famiglia proprietaria sia a quella open source: GPT-4/GPT-3.5 e Llama-3.1-8B/Qwen2.5-7B.

Gli esperimenti sono stati condotti su un sottoinsieme di 141 articoli del corpus AWASES-ALD contenente 2.195 fatti estratti. I risultati mostrano che tutte le configurazioni sono in grado di recuperare e valutare evidenze su larga scala, ma presentano differenze nella calibrazione delle decisioni. La configurazione Llama–Qwen ha raggiunto la copertura più elevata nel recupero dell'evidenza, mentre le configurazioni proprietarie hanno mostrato una maggiore proporzione di supporto diretto e un comportamento valutativo più stabile. L'affidabilità dei modelli valutatori è stata inoltre analizzata utilizzando un dataset ZnO annotato manualmente contenente 327 coppie fatto–evidenza, nel quale tutti i modelli hanno riconosciuto relazioni di supporto reali, con GPT-3.5 che ha ottenuto il più alto tasso di accordo diretto.

Nel complesso, la tesi dimostra che un approccio di verifica basato su evidenza testuale migliora la trasparenza e l'affidabilità della validazione automatica di fatti scientifici. Combinando provenienza testuale a livello di frase, giudizi strutturati LLM, confronti tra modelli e benchmark esterni, il framework proposto offre un approccio scalabile per la valutazione di fatti nella letteratura della scienza dei materiali e per il supporto alla qualità dei knowledge graph scientifici.

Contents

Listing of figures

Listing of tables

Listing of acronyms

ALD Atomic Layer Deposition

KG Knowledge Graph

LLM Large Language Model

ALE Atomic Layer Etching

GPC Growth Per Cycle

TF-IDF Term Frequency–Inverse Document Frequency

1 Introduction

1.1 Motivation for automating scientific knowledge verification

The pace of publication in materials science has accelerated to the point where even specialized subfields such as Atomic Layer Deposition (ALD) and plasma etching produce more primary literature than domain experts can feasibly read, verify, and curate[1]. New results arrive in journals, conference proceedings, and preprint servers with heterogeneous formats and notations. Facts central to process understanding, precursor and co-reactant identities, thermal versus plasma modality, operational windows in temperature and pressure, and quantitative outcomes such as growth-per-cycle, are dispersed across this expanding corpus. Converting such dispersed information into structured, trustworthy Knowledge Graphs (KGs) is therefore constrained not only by extraction quality but by the limited capacity of human verification.

Manual curation remains the gold standard for trust, yet it is intrinsically costly[2]. Verifying a single fact typically requires locating the relevant sections of a document, interpreting domain-specific notation (including chemical subscripts and unit conventions), checking that contextual qualifiers (substrate, reactor configuration, carrier gas) apply, and recording provenance at the level of exact sentences. Multiplied across thousands of papers and tens of thousands of facts, the effort becomes prohibitive. Moreover, purely manual workflows struggle to remain current as new publications continually alter the distribution of reported conditions

and materials.

Unchecked or weakly verified automatic extraction poses material risks for downstream discovery. Knowledge graphs are used to guide screening, model training, and experimental design; an error introduced at ingestion can silently propagate, shaping recommendations and conclusions. Typical failure modes include unit conversion mistakes (e.g., Torr versus Pascal), loss of chemical subscripts during PDF-to-text conversion, and conflation of processes or modalities reported in the same paper[3]. Without sentence-level provenance, these issues are difficult to diagnose or correct once embedded in a KG.

For these reasons, there is a clear need for automation that does more than increase throughput: it must preserve traceability and auditability. An evidence-grounded approach, one that explicitly links each asserted fact to the verbatim sentences that justify it, addresses both requirements[4, 5]. By insisting on exact textual support and encoding judgments in a constrained, machine-readable form, automated verification has the potential to scale with the literature while maintaining a level of transparency compatible with expert review. The present thesis adopts this stance: automation is justified not only by the volume and velocity of scientific output, but by the necessity of reliable, inspectable verification to safeguard downstream scientific inference and decision-making.

1.2 Challenges in evaluating materials-science knowledge graphs

Evaluating the correctness of facts encoded in materials-science KGs is made difficult by how information is reported in the literature and by how current curation pipelines record provenance. In domains such as ALD and etching, key statements concern precursor and co-reactant identity, process modality (thermal versus plasma), operating windows (temperature, pressure), and quantitative outcomes (e.g., growth-per-cycle). Establishing whether such statements are supported by the source text requires overcoming three recurring obstacles.

First, reporting is heterogeneous. Equivalent scientific content appears under different unit systems, symbols, or naming conventions, and textual extraction from PDFs often corrupts chemical notation. Temperatures may be reported in $^{\circ}\mathrm{C}$ or K, pressures in Pa, mbar, or Torr, and flow rates in sccm or $\mathrm{mL\ min^{-1}}$. Ligands and products are referenced by formulas with and without subscripts (e.g., $\mathrm{NH_3}$ versus NH3) or by aliases (e.g., TMA for trimethylaluminum). Contextual qualifiers that determine the truth of a statement, substrate, reactor configuration,

carrier gas, pulse and purge timing, are frequently distant from the numerical values they condition. This variability undermines simple string matching and creates ambiguity in normalization: aggressive canonicalization risks distorting meaning, whereas conservative strategies miss equivalences.

Second, evidence traceability is often incomplete. Many KGs store curated triples or tuples without the exact sentences that justify them, or link facts only at the document level (e.g., DOI) rather than at the passage level. When a single paper describes multiple processes, or when critical details reside in tables or figure captions, the absence of sentence-level grounding prevents auditors from determining precisely what was claimed, where, and under which assumptions. Preprocessing steps (e.g., citation removal, section trimming) can inadvertently strip qualifiers such as "remote" in plasma-enhanced ALD, further weakening provenance. Without durable links between facts and their verbatim support, it is difficult to reproduce curation decisions, to resolve disagreements, or to correct systematic errors once facts have propagated downstream.

Third, there is a lack of scalable, auditable evaluation protocols. Existing assessments are frequently ad hoc: they conflate retrieval of related text with verification of the specific ALD tuple, and they provide limited guidance on what constitutes sufficient support. Criteria such as relevance to the stated fact and compactness of the supporting span are seldom formalized, and inter-rater reliability is rarely reported. Variations in prompts, chunking strategies, truncation limits, and failure handling ("no evidence found") are often undocumented, which hinders reproducibility and comparability across studies. Manual spot-checking cannot keep pace with corpus growth, and evaluator outputs that are not machine-readable impede aggregation, auditing, and continuous improvement.

These challenges motivate an evaluation framework that (i) tolerates surface variability without altering scientific meaning, (ii) requires sentence-level, verbatim evidence for each asserted fact, and (iii) encodes judgments in a constrained, machine-readable format. The evidence-grounded approach pursued in this thesis addresses these needs by separating evidence retrieval from judgment, enforcing explicit criteria, and producing artifacts suitable for expert audit and large-scale deployment.

1.3 Importance of evidence-grounded evaluation

Evidence grounding requires that every verified fact be supported by an explicit span of text copied verbatim from the source paper. This constraint is more than a convenience for curation: it directly counteracts Large Language Model (LLM) failure modes. Without ground-

ing, LLMs may synthesize plausible but unfounded statements by interpolating across prior knowledge or by resolving ambiguity in ways that are not justified by the document. Requiring verbatim support narrows the model's action space to what the text actually states, reducing opportunities for hallucination and limiting judgments to falsifiable claims[6]. In the context of ALD and etching, where units, subscripts, and process qualifiers are brittle under PDF-to-text conversion, the insistence on exact wording preserves the semantic precision necessary for reliable interpretation.

The same requirement substantially improves transparency. When a KG stores a fact alongside the sentences that justify it, consumers can inspect the provenance in situ: they can see the formulation of chemical names, the exact units and ranges, and any contextual qualifiers (substrate, reactor configuration, plasma modality) that condition the claim. This visibility changes verification from an opaque, trust-me assertion into a traceable linkage between data and source[7]. It also enables principled disagreement: experts can debate whether a particular sentence suffices as evidence without first reconstructing where it came from.

Reproducibility benefits follow naturally. Verbatim spans act as stable artifacts that survive model updates, prompt revisions, and corpus growth. Given a fact and its cited sentences, an independent system can re-run the evaluation under identical criteria and expect the same outcome. This property is essential for longitudinal maintenance of scientific KGs, where facts may be corrected, superseded, or reinterpreted as new literature appears. It is equally important for benchmarking: comparable systems must reason over the same evidence to make performance differences meaningful.

Finally, grounding supports expert audit at scale. Structured outputs that pair facts with exact sentences and machine-readable judgments (e.g., relevant vs. not relevant, compact vs. not compact) allow curators to triage effectively. They can prioritize review of borderline or low-agreement cases, target systematic errors (such as recurrent unit mismatches), and integrate human-in-the-loop corrections without discarding prior work. In this thesis, the two-stage design, verbatim evidence extraction followed by criteria-based evaluation, operationalizes these advantages by producing artifacts that are simultaneously inspectable by experts and amenable to automated, repeatable scoring.

1.4 Research goals, hypothesis, and objectives

The overarching goal of this work is to evaluate facts about ALD and etching by grounding every judgment in explicit text spans drawn verbatim from the source literature. Rather than

relying on implicit model knowledge or document-level citations, the evaluation must hinge on what is actually written, preserving units, chemical notation, and contextual qualifiers. This goal reflects a practical need in KG curation: decisions should be auditable back to sentence-level evidence, enabling reproducibility and targeted expert review.

The central hypothesis is that, when constrained by verbatim evidence and a clear evaluation rubric, the aggregated judgments of LLMs approximate expert verification. Concretely, we expect multiple evaluators, applied under a shared schema of support classification and evaluator scoring behavior, to reach reliable fact–evidence verification performance when compared with expert-validated fact–evidence pairs. Under this hypothesis, separation of concerns (evidence retrieval versus evaluation) and the use of structured outputs reduce hallucination and improve consistency across models and documents.

To test this hypothesis, the thesis pursues the following objectives:

1. **Build an evidence-grounded pipeline** that links preprocessed full texts to extracted facts, retrieves candidate sentences using Term Frequency–Inverse Document Frequency (TF-IDF) similarity and verifies them with an LLM, and evaluates fact–evidence pairs under a JSON-constrained rubric.
2. **Compare model families and sizes** in both roles (retrieval and evaluation), assessing their performance on relevance and compactness, as well as their stability to prompt and chunking choices.
3. **Analyze reliability** via agreement with expert annotations and inter-model agreement, including audits of typical error modes (unit mismatches, notation loss, context drift).
4. **Quantify cost and throughput** by reporting latency, token usage, and cost per verified fact, and by examining trade-offs between precision (verbatim strictness) and coverage.
5. **Characterize failure modes and limits** of the approach, identifying when verbatim evidence is unavailable or ambiguous, and how such cases should be surfaced for human-in-the-loop resolution.

Together, these objectives operationalize the goal of evidence-grounded evaluation and provide measurable criteria to accept or reject the stated hypothesis in the context of ALD/etching KGs.

1.5 Contributions

This thesis contributes a practical and auditable methodology for verifying materials-science facts by anchoring every decision in verbatim textual evidence. The work combines robust data

preparation with a two-stage LLM workflow, and evaluates reliability across model families and sizes. The key contributions are:

1. **Evidence-grounded verification pipeline.** We design a two-stage architecture that separates retrieval from evaluation. Stage 1 extracts verbatim evidence sentences for each fact strictly from sentence-preserving chunks of the source paper; Stage 2 judges each fact–evidence pair under a JSON-constrained rubric of relevance and, if applicable, compactness. Carefully crafted prompts and schema validation bound model behavior to the document text and yield machine-auditable outputs.

2. **Robust preprocessing and safe alignment of sources.** We implement a data pipeline that normalizes and cleans full texts (removing front matter and references, preserving units and chemical notation), safely parses noisy fact records (JSON with `ast` fallback), enforces schema requirements, and aligns papers with extracted facts via a canonical `process_id`. The result is a reproducible joined dataset that supports deterministic downstream evaluation.

3. **Comparative study and audit artifacts.** We compare multiple LLM families and sizes in both roles (retrieval and evaluation), reporting evidence retrieval coverage, support-level distributions, evaluator scores, and agreement with expert-validated fact–evidence pairs. All intermediate and final artifacts, joined inputs, evidence spans, and structured judgments, are persisted for inspection, enabling expert audit, error analysis, and replication of results.

Together, these contributions provide a scalable, transparent approach to KG verification in ALD/etching, and establish reusable components, prompts, schemas, and data artifacts, that can be transferred to related scientific domains.

1.6 Structure of the thesis

The remainder of this thesis is organized into four substantive chapters that progressively develop, implement, and evaluate an evidence-grounded framework for verifying materials-science facts.

- **Chapter 2 — Background and Related Work:** situates the research within prior literature. It introduces the physical and chemical principles of ALD and etching, reviews the role of KGs in materials science, and surveys information-extraction and fact-verification methods for scientific text. The chapter then discusses the emerging *LLM-as-a-judge* paradigm, covering evaluator prompting, schema-constrained outputs, and evidence-grounded reasoning, and concludes by positioning this work within the broader landscape of scientific fact-checking and materials-knowledge representation.

- **Chapter 3 — Methodology and Experimental Setup:** details the design and implementation of the proposed framework. It first outlines the two-stage architecture separating evidence extraction from evaluation, then describes dataset preparation, including source acquisition, normalization, claim parsing, and schema alignment. Subsequent sections present the evidence extraction workflow, its conceptual basis, prompt design, and quality controls, followed by the evidence evaluation stage, which formalizes the LLM-as-a-judge approach through a structured judgment schema and scoring framework. The chapter concludes with implementation details covering model configuration and computational environment, and introduces the quantitative metrics and validation strategies used to assess factual support, evaluator scoring behavior, and consistency across model configurations.

- **Chapter 4 — Results and Discussion:** presents the empirical findings of the study. It reports performance across different extractor–evaluator model configurations, analyzing evidence retrieval coverage, support-level distributions, and evaluator scoring statistics. The chapter further compares the behavior of proprietary and open-source model families and interprets these results in terms of evaluator calibration and pipeline robustness. In addition, an external evaluation using a human-annotated ZnO dataset is presented to benchmark evaluator reliability against expert-validated fact–evidence pairs.

- **Chapter 5 — Conclusion:** summarizes the main contributions of the thesis and revisits the central research objectives in light of the experimental findings. It reflects on methodological limitations, discusses the implications of the results for automated fact verification in scientific literature, and outlines potential directions for future work, including hybrid retrieval strategies, improved evaluator calibration, and the extension of the framework to broader domains of materials science research.

2

Background and Related Work

2.1 Overview of Atomic Layer Deposition & Etching

Atomic Layer Deposition (ALD) is a cyclic thin-film growth technique based on pairs of self-limiting surface reactions. A typical cycle consists of (i) pulsing a gaseous precursor that chemisorbs to available surface sites until saturation, (ii) purging the reactor to remove excess precursor and byproducts, (iii) introducing a co-reactant that converts the chemisorbed layer to the desired material, and (iv) a second purge. Repetition of these half-cycles enables Ångström-level thickness control and excellent conformality on high-aspect-ratio features because surface chemistry, rather than gas-phase kinetics, limits the reaction [8, 9, 10, 11, 12]. In thermal ALD, activation is provided by temperature; in plasma-enhanced ALD (PEALD), radicals and ions generated by an RF plasma can lower effective reaction barriers, expand the accessible temperature window, and modify film properties [12, 9].

Atomic Layer Etching (ALE) adapts the same cyclic, self-limiting principle to controlled material removal. In a prototypical thermal or plasma ALE process, an adsorption step selectively modifies the surface (e.g., through chemisorption or fluorination) and a subsequent activation step (thermal, ligand exchange, or ion/radical exposure) removes a single or sub-monolayer of the modified layer, followed by purge steps. By decoupling modification and removal, ALE offers directional control, atomic-scale precision, and selectivity that are difficult to achieve with continuous etching chemistries [13]. Plasma assistance can again broaden the process space,

but also introduces additional variables related to ion energy and radical flux.

Across ALD and ALE, several reported parameters are central to interpreting and reproducing results. The temperature window must be high enough to activate surface reactions yet low enough to avoid precursor decomposition; it is usually specified together with reactor pressure (Pa, mbar, or Torr) and carrier-gas flow. Pulse and purge times (or exposure doses) control surface saturation and removal of gas-phase species, while in PEALD/PEALE the plasma power, frequency, exposure time, and source configuration (direct vs. remote) affect radical densities and ion bombardment. Film growth is quantified by the Growth Per Cycle (GPC), commonly reported in Å/cycle or nm/cycle, and may exhibit a nucleation delay that depends on substrate termination and initial surface chemistry. Substrate identity and topology (e.g., high-aspect-ratio trenches) are routinely discussed because conformality and selectivity depend on local transport and site availability. Finally, precursor and co-reactant choice, volatility, thermal stability, reactivity, and byproduct profiles, critically shape both the attainable temperature window and the resultant film quality [8, 9, 10, 12].

This brief overview highlights why subsequent sections emphasize verbatim, context-aware evidence. Reported values for temperature, pressure, GPC, and plasma conditions often appear alongside qualifiers such as substrate, reactor geometry, or exposure history. Reliable downstream use, whether for knowledge-graph construction or automated verification, requires preserving these qualifiers as part of the factual record, not merely the numerical values themselves.

2.2 Knowledge Graphs in materials science

Knowledge Graphs (KGs) have become a prominent abstraction for structuring heterogeneous materials knowledge into machine-interpretable entities and relations. Typical nodes include materials, precursors, processes, properties, and instruments; edges encode typed relations such as `has_precursor`, `processed_by`, `exhibits_property`, or `measured_under`. By aggregating facts across publications into a single graph, KGs enable querying at scale, link prediction, and downstream analytics and recommendation. Representative efforts span property-centric graphs and broad, literature-scale resources; for example, PropNet integrates computational and experimental quantities through derived-property relationships to support inference over materials properties [14], while MatKG assembles millions of entities and relations mined from the literature as a foundation for graph representation learning and large-scale link prediction [15]. Closely related initiatives in scholarly knowledge organization, such as structured sci-

entific summarization for research KGs, likewise emphasize explicit schema design and fine-grained linking of concepts and claims [16].

Despite their utility, materials KGs face persistent limitations rooted in how scientific facts are reported. Synonymy and polysemy (e.g., precursor aliases, trade names vs. IUPAC nomenclature) complicate entity resolution; units and symbols vary across papers (Pa, mbar, Torr; °C vs. K), and PDF-to-text conversion can degrade chemical notation (loss of subscripts), all of which challenge canonicalization without distorting meaning. Extraction noise arises from table parsing, figure caption text, and context fragmentation, while process-centric facts (e.g., ALD/ALE tuples combining precursors, modality, conditions, and quantitative outputs) are easily under-specified if context is not carried through the pipeline. Most critically for evaluation, provenance is often weak: facts are frequently linked to papers at the document level rather than to the exact sentences that justify them, limiting auditability and reproducibility once facts have been ingested.

These constraints motivate evaluation methods that retain both structure and verbatim evidence. For process-intensive domains such as ALD and etching, the usefulness of a KG depends not only on the presence of a triple but on its precise, context-aware support in the source text. The approach developed in this thesis addresses the above limitations by coupling fact representations compatible with KG schemas to sentence-level provenance, enabling scalable verification while preserving the qualifiers—units, chemical notation, substrates, and reactor configuration—that determine scientific validity.

2.3 Information extraction & fact verification in scientific literature

End-to-end pipelines for structuring scientific knowledge typically proceed from PDF ingestion to text segmentation, followed by entity and relation extraction, and finally fact verification. The PDF-to-text step remains a major source of noise: headers and footers are interleaved with body text; tables and figure captions are flattened; and chemical notation may lose subscripts during conversion. These artifacts complicate the downstream tagging of materials, precursors, and process parameters, and they motivate preprocessing strategies that normalize units and preserve domain-specific symbols wherever possible [17]. Beyond conversion, section detection and discourse cues (abstract, methods, results) influence where reliable facts reside, suggesting that structure-aware extraction can improve precision for process-centric state-

ments such as ALD/ALE tuples.

On top of basic information extraction, scientific fact verification requires linking candidate claims to the specific passages that support them. Classic verification paradigms distinguish between document-level evidence, which establishes that a supporting paper exists, and span-level evidence, which identifies the exact sentences or clauses that justify a claim. Surveys of fact extraction and verification emphasize the importance of explicit evidence selection, calibration of decision criteria, and careful handling of negation, hedging, and scope [18]. In scientific texts, these challenges are compounded by formulaic expressions, units, and cross-references that distribute the relevant context across multiple sections.

Recent perspectives argue that effective scientific fact-checking must be full-paper and structure-aware: evidence often resides in methods sections, tables, or figure captions rather than in abstracts, and provenance may be time-dependent through citations and versions [19]. For process data such as temperatures, pressures, growth-per-cycle values, and plasma conditions, span-level grounding is therefore critical to ensure that qualifiers (substrate, reactor configuration, exposure timing) travel with the numerical values. In this thesis, these considerations motivate a pipeline that retrieves candidate sentences using TF–IDF similarity and verifies them with an LLM, requiring verbatim spans for every verified fact.

2.4 LLM-as-a-judge paradigm

The *LLM-as-a-judge* paradigm treats a language model not as a generator of task outputs but as an evaluator of candidate outputs under explicit instructions. Recent surveys synthesize a rapidly growing body of methods that operationalize this idea across natural language generation, information retrieval, and factuality assessment, documenting both the promise of scalable evaluation and the pitfalls that arise when free-form models are used as judges [20, 21]. Central themes include how to formulate judging prompts, how to elicit consistent criteria application, and how to measure the reliability of model verdicts relative to human annotators.

A first design choice is whether to use a single judge or multiple judges. Single-judge setups rely on one model (or one call) to render a verdict, often with a structured rubric and explicit constraints. Multi-judge configurations increase robustness via repeated sampling (self-consistency), cross-model voting, or staged protocols where one model proposes and another critiques or verifies [20]. Empirically, repeated judging with aggregation reduces variance but does not by itself resolve systematic biases, particularly in settings where the judge has access to the candidates but not to strong references.

A second axis concerns how instructions are specified and enforced. Rubric- and schema-guided judging restricts outputs to a predefined format, commonly a compact JSON object with named fields and short justifications [21, 22]. Such constraints make judgments machine-auditable, support downstream aggregation, and reduce prompt drift. However, recent meta-evaluations show that judges can deviate from instructions or ignore criteria, and that models differ in their adherence depending on prompt wording and content domain [23, 22]. These findings motivate explicit validation of outputs (e.g., JSON parsing, range checks) and, when possible, automatic rejection or re-asking when schema violations occur.

A related line of work augments judging with references or evidence to mitigate hallucination and position effects. Reference-guided judging conditions the verdict on retrieved passages or gold references and asks the model to ground its decision in those sources, sometimes by returning the supporting spans [24]. In general-purpose benchmarking, MT-Bench and Chatbot Arena highlight risks such as position bias, where the relative placement of candidates influences the judge's preference, and verbosity bias, where longer answers are favored irrespective of quality; both require careful prompt design and randomized presentation to control [7]. Surveys recommend evidence grounding, blind evaluation protocols, and normalization of candidate presentation to counteract these biases [20, 21].

Beyond format control, several studies examine calibration and agreement. Systematic evaluations propose families of prompts and explainable metrics that separate dimension scores (e.g., correctness, relevance, style) and report human–model agreement with measures such as Cohen's κ or Krippendorff's α [22]. Others investigate instruction adherence explicitly, quantifying when judges honor requested criteria and when they revert to generic preferences [23]. These works collectively argue for reporting not only accuracy against a gold standard but also reliability indicators and ablations over prompt templates, ordering, and formatting.

For scientific applications, where claims are falsifiable against a source document, evidence grounding becomes central. By conditioning the judge on explicit, verbatim spans and forbidding inference outside those spans, the evaluation is narrowed to what the text actually states, reducing the opportunity for hallucination and making the decision auditable [24, 20]. In our setting, ALD and etching facts that hinge on units, subscripts, and contextual qualifiers, rubric-guided, evidence-conditioned judging offers a principled way to ask: (i) is the cited passage relevant to the specific fact tuple (precursors, modality, conditions, quantitative values), and (ii) is the passage compact enough to support the fact without extraneous context. The emphasis on compact, schema-constrained justifications addresses known risks of verbosity and instruction drift [7, 22].

Finally, the paradigm raises questions of scalability and reproducibility. Surveys emphasize documenting prompts, random seeds, and decoding parameters, publishing schema definitions, and persisting judge outputs to enable independent audit [20, 21]. Complementary frameworks propose adaptive pipelines that select judge strength, sample counts, and aggregation rules based on task difficulty [25]. Taken together, these threads inform the design choices in this thesis: we separate evidence extraction from evaluation, condition judgments on verbatim spans, enforce JSON schemas with short justifications, and report agreement and ablation studies alongside accuracy. This configuration aligns with best practices for transparent, evidence-grounded judging while addressing the domain-specific brittleness of materials-science facts.

2.5 Evidence-grounded reasoning and evaluation

Evidence grounding denotes the practice of conditioning every evaluation decision on explicit text spans copied verbatim from the source document. In scientific fact verification, grounding narrows the hypothesis space to what the paper actually states, reducing opportunities for hallucination and unverifiable interpolation. The distinction between document-level and span-level evidence is crucial: the former merely asserts that a supporting paper exists, whereas the latter identifies the exact sentences that license a claim, together with their units, symbols, and contextual qualifiers. Surveys of fact extraction and verification emphasize this difference and link span selection to downstream verifiability, especially in technical domains where claims are sensitive to notation and scope [18]. Recent perspectives on scientific fact-checking further argue for full-paper, structure-aware retrieval because methods sections, tables, and figure captions often contain the decisive details for process claims [19].

Grounded judging benefits reliability in at least two ways. First, by forcing the evaluator to decide with respect to a concrete span, it mitigates position and verbosity biases observed in general LLM judges, where salience or length can sway verdicts independent of quality [7, 20, 21]. Second, grounding supports transparent disagreement: when annotators or models differ, their cited sentences make the locus of disagreement inspectable (e.g., whether a temperature refers to the substrate or to a precursor line). Reference-guided judging formalizes this practice by presenting the judge with retrieved passages and requiring a verdict that is explicitly tied to those passages; empirical studies show improved robustness relative to free-form judging [24].

To translate these principles into machine-auditable outputs, rubric- and schema-guided evaluation has emerged as a best practice. In this approach, the judge must emit a compact,

typed structure, commonly JSON, whose fields correspond to criteria such as relevance (does the passage directly support the fact) and compactness (is the passage minimally sufficient) [21, 22]. Short justification fields encourage focus and reduce the likelihood of drifting into unsupported paraphrase. Meta-evaluations indicate that instruction adherence is nontrivial, judges sometimes ignore criteria or produce malformatted outputs, hence parsers, validators, and re-asking strategies are important components of a dependable pipeline [23, 22].

The granularity of grounding also affects the trade-off between precision and recall. Requiring verbatim sentences maximizes precision and auditability but may undercount paraphrastic or distributed evidence (e.g., a value in a table and its qualifier in the caption). Conversely, semantic evidence that is not copied verbatim can increase coverage but weakens reproducibility and complicates error diagnosis. For materials processes such as ALD and etching, where units, subscripts, and modality qualifiers are brittle under PDF-to-text conversion, the precision gains of verbatim spans typically outweigh potential recall losses; section-aware retrieval and layout-sensitive preprocessing can partially recover recall without relaxing the grounding requirement [19].

Calibration and agreement metrics are essential to characterize grounded evaluators. Beyond accuracy against a gold standard, studies recommend reporting inter-rater agreement (e.g., Cohen's κ, Krippendorff's α) between models and humans, as well as sensitivity analyses over prompt templates, ordering, and candidate presentation [20, 22]. Multi-judge aggregation (self-consistency or cross-model voting) can reduce variance, but its benefits are greatest when each individual judge is constrained by the same evidence and schema, ensuring that aggregation combines compatible decisions rather than compounding uncontrolled biases [20, 21].

Recent work on lightweight evaluator models complements these protocol choices. Compact "reasoning" judges trained to output evidence-grounded justifications achieve competitive balanced accuracy relative to larger models while maintaining strict formatting guarantees—an appealing property for high-throughput scientific verification [26]. Such models underscore that reliability is not solely a function of parameter count; adherence to grounding and schema can dominate raw model capacity in evaluator settings.

The framework adopted in this thesis operationalizes these insights. Evidence is extracted at sentence level and fed to the judge; verdicts are restricted to a minimal JSON schema with bounded justification fields; and reporting includes support-level distributions, evaluator scoring statistics, and agreement with expert-validated fact–evidence pairs. In addition, criteria reflect the domain's structure: relevance is defined with respect to an ALD/ALE tuple (precursors/co-reactants, process modality, conditions, and quantitative values), and compactness penalizes

spans that include unrelated context. Together, these design choices align with recommendations from the LLM-as-a-judge literature while addressing the domain-specific brittleness of materials-science facts [18, 20, 21, 22, 19].

2.6 Scope and contribution within prior work

This thesis sits at the intersection of process-centric materials science, information extraction from scientific literature, and the emerging practice of using Large Language Model (LLM) as evaluators. From the materials side, ALD and etching provide a well-studied but notation-sensitive domain in which facts are tightly coupled to process tuples, precursors and co-reactants, thermal versus plasma modality, operating windows, and quantitative outcomes such as growth-per-cycle [8, 9, 10]. From the knowledge representation side, materials KGs promise scalable querying and inference but continue to struggle with synonymy, unit variance, extraction noise, and weak sentence-level provenance [14, 15]. On the information extraction side, PDF-to-text conversion and structure recovery remain key sources of error, motivating careful preprocessing and explicit evidence selection in scientific settings [17, 18].

Against this backdrop, the LLM-as-a-judge literature argues for evaluator configurations that are instruction-following, bias-aware, and grounded, with outputs constrained by rubrics and schemas to support audit and reproducibility [20, 21, 22, 7]. Reference-guided judging further conditions decisions on retrieved passages, mitigating hallucination and position effects [24]. Recent perspectives on scientific fact-checking additionally stress full-paper, section-aware evidence retrieval as essential for technical claims whose decisive details often live outside abstracts [19]. Compact evaluator models trained to emit evidence-grounded justifications suggest that strict formatting and grounding can rival larger models in balanced accuracy while improving deployability [26].

Within this landscape, the present work advances three specific commitments:

1. First, it adopts a verbatim-only stance on evidence: every verified fact must be supported by exact sentences copied from the source document, preserving units, subscripts, and contextual qualifiers. This choice prioritizes precision, traceability, and auditability over broader recall, directly addressing known failure modes of ungrounded evaluators [20, 21, 7].

2. Second, it enforces a two-stage separation between evidence retrieval and judgment. evidence is retrieved from sentence-level candidates under strict prompts, and only then

evaluated under a minimal JSON schema with bounded justifications that operationalize relevance and compactness. This architectural separation curbs instruction drift, simplifies error diagnosis, and yields machine-checkable artifacts [22, 23].

3. Third, it tailors the rubric to the ALD/ALE fact structure. Relevance is defined with respect to a process tuple, precursors/co-reactants, modality, conditions, and quantitative values, so that judgments reflect domain semantics rather than generic textual relatedness [8, 12].

These commitments differentiate the thesis from prior work in three ways. Compared to general LLM judging studies, the evaluation target here is not free-form quality but falsifiable, unit-sensitive scientific facts tied to explicit spans. Compared to materials KGs, the contribution is not a new graph or linker, but an evaluation protocol and pipeline that produces sentence-level provenance and structured verdicts suitable for large-scale audit. Compared to traditional IE pipelines, the emphasis shifts from extraction breadth to verification with evidence, aligning with calls for structure-aware retrieval in scientific fact-checking [19].

The expected contributions are twofold. Methodologically, the thesis provides a reproducible, evidence-grounded evaluation pipeline with clearly specified prompts, schemas, and outputs, together with reliability reporting (accuracy, inter-model and human–model agreement) and ablations recommended by the LLM-as-a-judge literature [20, 21, 22]. Practically, it supplies audit-ready artifacts, joined inputs, evidence spans, and verdicts, that can be integrated into knowledge-graph quality assurance workflows, helping curators prioritize review and trace corrections in domains where small notation errors can invalidate downstream inference [14, 15].

3

Methodology and Experimental Setup

This chapter presents the methodological framework adopted to develop and evaluate the proposed evidence-grounded evaluation system for materials science facts, with a focus on Atomic Layer Deposition (ALD) and etching processes. The chapter outlines the full experimental pipeline, from data preparation and preprocessing of scientific literature, through evidence extraction and evaluation using Large Language Models (LLMs), to implementation details and validation strategies. Each component of the pipeline is described in detail, emphasizing the rationale behind its design, the models and prompts employed, and the steps taken to ensure reproducibility and interpretability. The methodology is structured to provide both conceptual clarity and technical transparency, supported by diagrams, pseudocode, and equations where appropriate to illustrate data flow, algorithmic logic, and evaluation computations.

3.1 Overview of the Approach

This study employs a multi-stage pipeline designed to evaluate the factual correctness of GPT-extracted knowledge about ALD and etching processes. The pipeline systematically connects scientific literature to structured facts by extracting, grounding, and evaluating evidence through large language models (LLMs). Figure 3.1 presents an overview of the workflow, which proceeds through six main stages: data acquisition, preprocessing, fact extraction, evidence retrieval, LLM-based evaluation, and result aggregation and analysis.

The process begins with a curated corpus of 141 research papers from the AWASES-ALD [27]

dataset. These papers, along with their associated GPT-generated annotations, form the primary input. Each annotation represents a candidate fact expressed in structured JSON format, describing process parameters, film properties, and process characteristics of ALD experiments. Because these facts were originally generated by GPT models, the purpose of the present work is to verify their accuracy and level of support within the corresponding source papers.

During the preprocessing stage, textual data from the papers are cleaned, normalized, and divided into manageable sentence or paragraph chunks suitable for LLM input. The evidence extraction stage identifies candidate supporting sentences for each fact using a hybrid retrieval strategy. First, the full text of each paper is segmented into individual sentences. A Term Frequency–Inverse Document Frequency (TF-IDF)–based similarity retrieval mechanism is then applied to select the top candidate sentences most semantically related to the given fact. These candidate sentences are subsequently verified by a large language model through a fact-conditioned prompt that determines whether the retrieved text explicitly supports the claim. If supporting evidence is identified, the exact sentence is returned verbatim; otherwise, the extractor reports that no direct evidence was found. This two-stage retrieval–verification process improves efficiency by reducing the search space presented to the language model while maintaining semantic interpretability of the extracted evidence.

In the LLM evaluation stage, a second, independent model, different from the one used for extraction, is prompted to assess each pair and return a structured judgment. The evaluator produces its verdict in a controlled JSON schema containing fields such as `support_level` ("direct," "partial," or "none"), `relevance_score`, `accuracy_score`, and an explanatory `comment`. This separation of extractor and evaluator models (e.g., using combinations of GPT-4 [28], GPT-3.5 [29], Qwen [30] and LLaMA [31]) reduces confirmation bias and provides a more robust test of cross-model reliability. The experimental design focuses on controlled intra-family comparison to ensure methodological clarity and reduce confounding effects introduced by heterogeneous training regimes.

Finally, the scoring and analysis stage aggregates the individual evaluations into fact-level and corpus-level metrics. These metrics quantify the degree of factual support present in the literature, enabling both quantitative comparison and qualitative error analysis.

The overall methodology is implemented in a modular fashion, with distinct components for data preparation, evidence retrieval, evaluation, and analysis. This design promotes reproducibility and allows for future substitution of models or prompt variants without altering the overall pipeline structure.

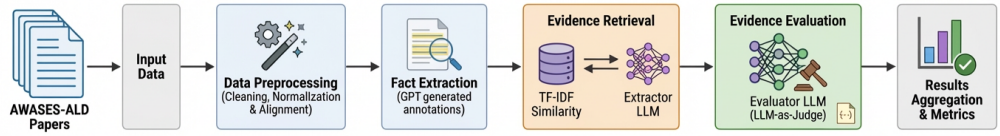


Figure 3.1: Overview of the evidence-grounded evaluation pipeline. The process begins with the AWASES-ALD papers and proceeds through data preprocessing, fact extraction, evidence retrieval, LLM evaluation, and result aggregation. Each stage produces intermediate artifacts that feed into subsequent steps, enabling systematic grounding and evaluation of extracted ALD facts.

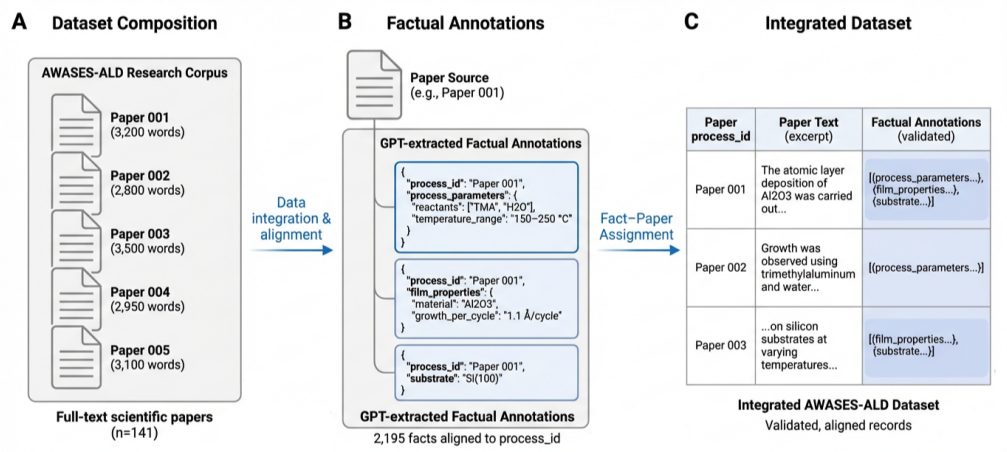


Figure 3.2: Integration of AWASES-ALD papers with GPT-generated factual annotations.

3.2 Dataset Preparation

The integrity of the verification pipeline relies on the quality and precise alignment of the source data. The initial dataset combined two complementary streams that required separate cleaning and subsequent integration: the full-text corpus of ALD research papers and the semi-structured factual extractions derived from these papers. The goal of this stage was to produce a unified, validated corpus suitable for downstream evidence retrieval and evaluation. Figure 3.2 illustrates how the paper corpus and extracted factual annotations are integrated into a unified dataset.

3.2.1 Acquisition and Characterization of Source Data

Two primary data sources were employed. First, a corpus of 141 full-text research papers obtained from the public AWASES-ALD dataset [27] served as the evidentiary foundation. Second, a companion file containing LLM-generated annotations represented the candidate fac-

tual assertions to be verified. Each annotation was stored as a serialized dictionary describing process parameters, film properties, and process characteristics, for example:

```
{
  "process_parameters": {"reactants": ["Al(NiPr2)3, H2O"],
                         "temperature_range": "Up to 325 °C"},
  "film_properties": {"material": "Al2O3",
                      "growth_per_cycle": "0.73 Å/cycle"}
}
```

Preliminary inspection revealed notable heterogeneity between these two sources. The document corpus exhibited inconsistent encodings and irregular whitespace patterns, while the factual dataset contained non-standard serialized strings that were not always valid JSON objects.

3.2.2 Text Normalization and Document Deduplication

To standardize the full-text corpus, all entries were first converted to UTF-8 encoding. Regular expressions were used to collapse contiguous whitespace characters into a single space, thereby ensuring uniform tokenization in subsequent LLM processing. Each paper was assigned a unique `process_id` to serve as an immutable key, allowing robust deduplication and ensuring that each document appeared exactly once in the corpus. Entries lacking valid or non-empty text fields were excluded to prevent downstream parsing errors. The resulting distribution of document lengths, expressed as word counts per `process_id`, is illustrated in Figure 3.3, which highlights variation in paper sizes and helps identify outliers or incomplete entries.

3.2.3 Robust Parsing and Validation of Factual Claims

The dataset of extracted facts required a fault-tolerant deserialization strategy. A two-stage parsing routine was implemented. In the first stage, each serialized string was parsed using a standard JSON loader. Entries that failed strict JSON parsing, often due to single quotes or other Pythonic syntax, were passed to a secondary parser, which safely evaluates string literals into their corresponding data structures. Entries that failed both parsing attempts were logged and removed. This strict filtering ensured that only syntactically valid and structurally consistent factual dictionaries were retained for verification.

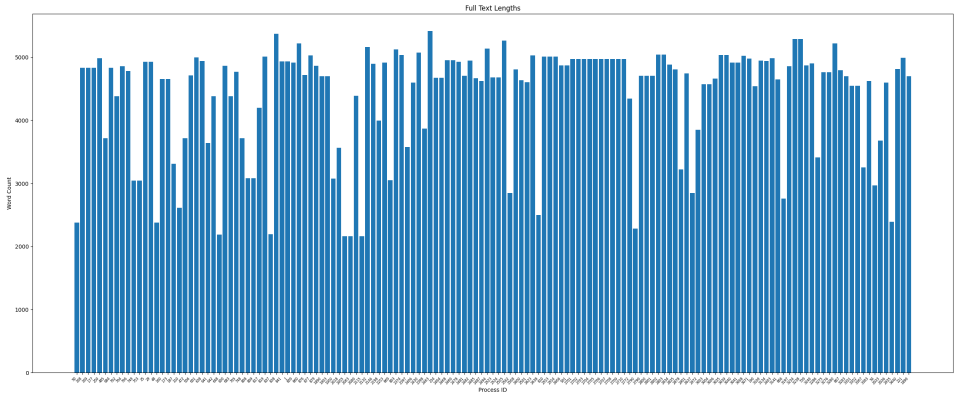


Figure 3.3: Distribution of word counts across the AWASES-ALD corpus.

3.2.4 Schema Alignment and Corpus Integration

After independent cleaning, the validated full-text corpus and the structured fact dataset were integrated into a unified corpus. A schema validation step confirmed that both data sources contained the necessary fields, namely the `process_id`, the preprocessed text, and the factual extractions. Before joining, the `process_id` field in each dataset was coerced to a consistent string type and stripped of any extraneous whitespace. The two datasets were then merged through an inner join on this standardized key, producing an aligned dataset in which each record contained one paper and its associated list of factual claims. Join statistics were computed to assess coverage and data retention, confirming that the majority of records were successfully paired.

The resulting integrated corpus serves as the canonical input for the subsequent evidence extraction and evaluation stages described in the following sections. The complete data preparation workflow, encompassing normalization of full texts, robust parsing of factual extractions, schema alignment, and corpus integration, is summarized in Algorithm 3.1.

3.3 Evidence Extraction

The evidence extraction stage bridges the curated corpus and the subsequent evaluation by identifying textual evidence that may substantiate each factual claim. Rather than relying solely on direct LLM retrieval across the entire document, the implemented pipeline employs a hybrid strategy combining classical information retrieval with language model verification.

For each factual statement derived from the annotated dataset, the corresponding paper is

Algorithm 3.1 Data preparation: curation, normalization, parsing, and integration

Require: Raw papers table D_{raw}, GPT annotations table F_{raw}
Ensure: Integrated corpus C with aligned full text and factual extractions

procedure PREPAREDATA(D_{raw}, F_{raw})
▷ Full-text cleaning
    $D \leftarrow$ UTF8NORMALIZE(D_{raw}.full_text)
    $D \leftarrow$ COLLAPSEWHITESPACE(D)
    $D \leftarrow$ ASSIGNKEY(D_{raw}, process_id)
    $D \leftarrow$ DROPINVALID(D, field=full_text)
    $D \leftarrow$ DEDUPLICATE(D, key=process_id)
    D.word_count $\leftarrow$ WORDCOUNT(D.full_text)
▷ Facts parsing with safe fallback
    for all $r \in F_{\text{raw}}$
        $p \leftarrow$ TRYJSONPARSE(r.extracted_info)
        if $p = \varnothing$
            $p \leftarrow$ TRYLITERALEVAL(r.extracted_info)
        end if
        if $p \neq \varnothing$ **and** VALIDATESCHEMA(p)
            Append $(r.\text{process\_id}, p)$ to F
        end if
    end for
▷ Schema alignment and integration
    D.process_id $\leftarrow$ NORMALIZEKEY(D.process_id)
    F.process_id $\leftarrow$ NORMALIZEKEY(F.process_id)
    $C \leftarrow$ INNERJOIN(D, F, key=process_id)
▷ Diagnostics and outputs
    REPORTJOINSTATS(D, F, C)
    SAVE(C, `corpus_integrated.jsonl`)
    return C
end procedure

first segmented into individual sentences. A lightweight retrieval step based on TF-IDF similarity is then used to identify a small set of candidate sentences most semantically related to the fact. These candidate sentences are subsequently provided to an LLM, which determines whether one or more of them explicitly support the claim and returns the supporting sentence verbatim.

This two-stage design significantly reduces the search space presented to the language model

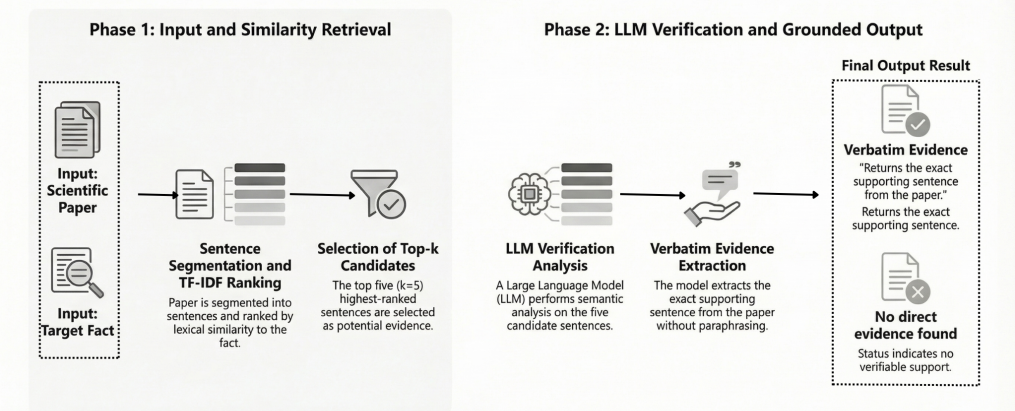


Figure 3.4: Evidence extraction workflow combining TF–IDF sentence retrieval with LLM-based verification of supporting evidence.

while preserving high recall, enabling efficient evidence identification across large scientific documents. Figure 3.4 illustrates the hybrid retrieval–verification process used to identify supporting evidence for each factual claim.

3.3.1 Conceptual Framework

The underlying assumption is that each factual statement within the knowledge representation has one or more supporting expressions distributed across the scientific text. Because manual alignment of facts and supporting evidence is infeasible at scale, an automated retrieval approach is adopted. The implemented method follows a hybrid information retrieval and verification paradigm. First, the full paper text is segmented into sentences using rule-based sentence boundary detection. Each factual claim is then compared against all candidate sentences using a TF-IDF vector representation combined with cosine similarity.

This retrieval stage ranks sentences according to their lexical similarity to the fact and selects the top k candidates (with $k = 5$ in the current implementation). These candidate sentences serve as a focused context window that is subsequently evaluated by an LLM. The model determines whether any of the retrieved sentences directly support the factual claim and returns the supporting text verbatim when such evidence exists. This formulation enables semantic verification of candidate sentences while the retrieval stage captures lexical similarity between

the fact and the document.

3.3.2 Prompt Design and Model Configuration

Prompt engineering plays a central role in ensuring that the extractor model identifies evidence that is both relevant and traceable to the source text. Rather than scanning entire documents directly with a language model, the implemented pipeline first retrieves a small set of candidate sentences using a TF-IDF similarity search. These candidate sentences are then provided to the LLM, together with the factual claim, for semantic verification.

The prompt template therefore includes the factual claim and the retrieved candidate sentences and instructs the model to determine whether any of them explicitly support the claim. To ensure transparency and prevent hallucinated evidence, the model is required to return supporting sentences verbatim from the provided candidates rather than generating new text.

An abbreviated form of the prompt is illustrated below:

```
You are verifying whether a scientific claim is supported.

Claim: {fact}
Candidate sentences: {retrieved_sentences}

If a sentence explicitly supports the claim,
return the supporting sentence verbatim.
Otherwise return:
"No direct evidence found."
```

This standardized format ensures consistent responses across different model backends while preserving traceability between extracted evidence and the original source text. The extraction stage was executed using both proprietary and open-source LLMs, including GPT-4, GPT-3.5, LLaMA-3.1-8B, and Qwen2.5-7B. Each model was integrated through a unified interface that allows interchangeable use of extractor backends within the same pipeline configuration. By combining deterministic retrieval with LLM-based verification, the system reduces the search space presented to the model while maintaining high recall for potential evidence sentences.

3.3.3 Operational Workflow

For each factual record, the extraction routine proceeds as follows. First, the associated paper text is segmented into individual sentences using rule-based sentence boundary detection. Very short fragments (fewer than 20 characters) are discarded to remove incomplete sentence fragments.

Second, a TF-IDF vectorization step is applied to both the factual claim and the set of candidate sentences extracted from the paper. Cosine similarity between the fact vector and each sentence vector is computed, and the top k most relevant sentences are selected as candidate evidence.

Third, the selected candidate sentences are concatenated and provided to the extractor LLM together with the factual claim. The model analyzes these candidate sentences and returns the sentence(s) that directly support the claim or a predefined negative response if no supporting evidence is present.

This retrieval–verification loop is executed for every fact extracted from each document in the corpus. The model response is parsed and stored as the supporting evidence for the corresponding fact. Algorithm 3.2 summarizes the high-level workflow of this process.

Algorithm 3.2 Hybrid evidence retrieval and verification

Require: Integrated corpus $C = \{(process\_id, text, facts)\}$, extractor model M, retrieval size k
Ensure: Evidence set $E = \{(fact, evidence)\}$
for all $(process\_id, text, facts) \in C$
  $sentences \leftarrow$ SegmentIntoSentences($text$)
  $sentences \leftarrow$ FilterShortSentences($sentences$)
  for all $f \in facts$
    $scores \leftarrow$ TF-IDFSimilarity($f, sentences$)
    $candidates \leftarrow$ TopK($sentences, scores, k$)
    $response \leftarrow$ QueryLLM(M, *Prompt*$(f, candidates)$)
    if $response$ contains supporting sentence
      $E[f] \leftarrow response$
    else
      $E[f] \leftarrow$ "No direct evidence found"
    end if
  end for
end for
return E

3.3.4 Output Structure and Quality Control

The output of the extraction process consists of structured JSON objects linking each fact to its retrieved evidence. Each output record links an extracted fact to the evidence identified in the corresponding paper. For each fact, the system stores the hierarchical fact category (parent), the factual value itself, and the extracted evidence sentence returned by the model. If no supporting evidence is identified, the system records the predefined response "No direct evidence found." The resulting dataset therefore forms a structured mapping between facts and their supporting textual evidence. To ensure quality and traceability, logs were maintained for all model calls, including timestamps, model versions, and prompt identifiers. Post-processing scripts validated the JSON structure and removed any malformed responses. Basic statistics, such as the average number of evidence sentences per fact and the proportion of facts with at least one retrieved evidence, were computed to monitor extraction coverage and stability.

The resulting evidence dataset constitutes the intermediary layer between raw documents and LLM-based evaluation, providing a structured and semantically aligned foundation for factual verification.

The hybrid retrieval–verification design offers several advantages. By restricting the LLM input to a small set of semantically relevant sentences, the approach significantly reduces computational cost while improving reliability. The TF-IDF retrieval stage provides deterministic candidate selection, whereas the language model focuses exclusively on semantic verification. This separation of retrieval and reasoning reduces the likelihood of hallucinated evidence and improves the interpretability of the verification process.

3.4 Evidence Evaluation (LLM-as-Judge)

The evidence evaluation stage operationalizes the concept of *LLM-as-judge*, in which a large language model assesses the factual validity of extracted knowledge claims based on retrieved evidence. This component transforms the problem of factual verification into a structured judgment task, allowing the model to express both categorical and graded assessments. The outcome is a machine-readable judgment that quantifies the extent to which the extracted facts are supported by the underlying scientific literature.

3.4.1 Conceptual Framework

The evaluation procedure is grounded in the idea that an LLM, when appropriately prompted, can serve as an analytical assessor that interprets natural-language evidence with respect to a factual statement. Each fact–evidence pair generated during the extraction stage is presented to the evaluator model, which determines the level of factual support and assigns quantitative scores reflecting the semantic relevance of the evidence and the factual correctness of the claim. To mitigate self-confirmation bias, the evaluator model is deliberately chosen to differ from the extractor model within each experimental configuration. Role assignments are alternated systematically within model families, ensuring epistemic separation between retrieval and judgment while allowing controlled robustness analysis.

3.4.2 Prompt Structure and Judgment Schema

The evaluation prompt is designed to elicit structured, transparent reasoning. It provides the model with a factual statement, one or more evidence sentences, and explicit instructions to return its assessment in a predefined JSON format. This format standardizes outputs across models and supports automatic post-processing. A condensed illustration of the evaluation prompt is shown below:

```
You are a scientific fact evaluator.
Given a FACT and the extracted EVIDENCE sentence,
assess whether the evidence supports the fact.
Return your answer as JSON:
{
  "support_level": "direct" | "partial" | "none",
  "relevance_score": 0-5,
  "accuracy_score": 0-5,
  "comments": "Brief explanation of your judgment"
}
```

The three main judgment dimensions are defined as follows: `support_level` captures the categorical relationship between the fact and the evidence, ranging from complete alignment (“direct”) to explicit contradiction or absence of support (“none”). `relevance_score` and `accuracy_score` quantify, on a discrete scale from 0 to 5, the semantic relatedness of the evidence to the fact and the factual correctness of the claim, respectively. The free-text `comments`

field provides qualitative reasoning that explains the assigned support category and scores, improving interpretability of the automated judgments.

3.4.3 Evaluation Workflow

For each fact–evidence pair, the evaluator model corresponding to the current experimental configuration is invoked with the formatted prompt. To ensure reliability, each judgment undergoes schema validation, and any malformed responses are discarded or corrected through minimal re-prompting. All evaluations are logged with metadata including model identifier, temperature setting, and timestamp, enabling reproducibility and later error analysis.

The evaluation process is executed separately for each extractor–evaluator configuration described in Chapter 4, allowing systematic comparison of evaluator behavior across proprietary and open-source model families.

Algorithm 3.3 summarizes the operational workflow of the evaluation stage.

Algorithm 3.3 LLM-based factual evaluation of extracted evidence

Require: Evidence set $E = \{(fact, evidence)\}$, evaluator model M
Ensure: Judgment set $J = \{(fact, support\_level, relevance, accuracy, comment)\}$

for all $(fact, evidence) \in E$
  $prompt \leftarrow$ FormatPrompt($fact, evidence$)
  $response \leftarrow$ QueryLLM($M, prompt$)
  $parsed \leftarrow$ ParseJSON($response$)
  if ValidateSchema($parsed$)
    Append $(fact, parsed.support\_level, parsed.relevance\_score,$
    $parsed.accuracy\_score, parsed.comments)$ to J
  end if
end for
return J

3.4.4 Scoring and Aggregation

Following evaluation, the judgments are aggregated to derive quantitative measures of factual support across the corpus. For each fact, the evaluator returns both categorical and numerical assessments, including the support level and the relevance and accuracy scores. These scores provide a graded indication of how strongly the retrieved evidence aligns with the factual claim.

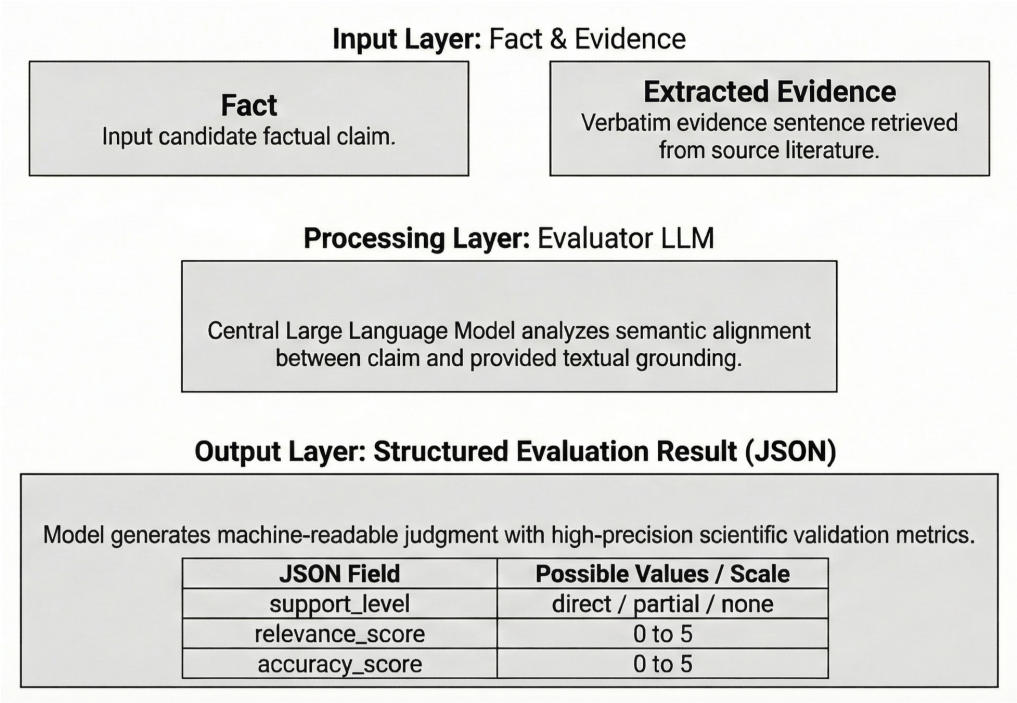


Figure 3.5: LLM-as-judge evaluation framework for assessing fact–evidence pairs and generating structured judgments.

Aggregating these per-fact judgments across the corpus yields summary statistics such as the distribution of support categories and the average relevance and accuracy scores. These aggregated metrics enable comparative evaluation of different extractor–evaluator model combinations.

3.4.5 Quality Assurance and Reliability

To assess the stability of evaluator behavior, results were compared across multiple extractor–evaluator configurations involving both proprietary and open-source models. Agreement rates across model pairs were measured to estimate inter-model consistency, providing an internal reliability measure analogous to inter-annotator agreement in human evaluation. Any systematic discrepancies were analyzed qualitatively to identify potential biases arising from prompt interpretation or model-specific tendencies.

The output of this stage is a structured and interpretable dataset of machine-generated factual judgments. Figure 3.5 presents the evaluation architecture in which the evaluator model assigns structured support judgments to fact–evidence pairs.

This dataset serves as the empirical foundation for the quantitative analysis presented in

Chapter 4, enabling systematic comparison of model configurations and evaluation of evidence-grounded factual support.

3.5 Implementation Details

The methodological framework described above was implemented as a modular and sequential pipeline to ensure transparency, reproducibility, and scalability. This section outlines the system architecture, technical environment, model configuration, and operational considerations that governed the execution of all experiments.

3.5.1 Model Selection and Configuration

The implementation employs multiple LLMs in complementary roles to decouple information retrieval from factual judgment. Extraction and evaluation are executed by distinct models, which mitigates self-confirmation bias and provides a test of cross-model robustness. Two primary model families were used: proprietary (closed-source) and open-source. The proprietary family includes GPT-4 and GPT-3.5, accessed via the OpenAI API. The open-source family includes Llama-3.1-8B and Qwen2.5-7B, deployed locally or through hosted inference endpoints.

Within each family, extractor and evaluator roles were systematically alternated to assess sensitivity to role assignment. Accordingly, four experimental configurations were executed:

- GPT-4 (extractor) → GPT-3.5 (evaluator)
- GPT-3.5 (extractor) → GPT-4 (evaluator)
- Llama-3.1-8B (extractor) → Qwen2.5-7B (evaluator)
- Qwen2.5-7B (extractor) → Llama-3.1-8B (evaluator)

Model choice in each role was guided by three main criteria: robustness of instruction following, stability of output formatting, and availability within the experimental infrastructure.

In the extraction stage, the system first applies a lightweight TF-IDF retrieval step to identify candidate sentences that are lexically similar to each factual claim. Only these top-ranked sentences are then provided to the extractor model for semantic verification. This hybrid design significantly reduces the amount of text processed by the language model while preserving high recall for potential evidence.

In setting parameters, the extraction stage was configured to favor more extensive coverage of candidate evidence, while the evaluation stage prioritized deterministic judgments. All prompts were standardized to enforce predictable output formats, enabling reliable parsing and downstream processing across different model backends. Because the retrieval stage restricts the input to a small set of candidate sentences, the prompts provided to the language models remain well within the context limits of all models used in this study. Consequently, the system does not require long-context document processing or document chunking during model inference. Each request consists only of the factual claim and the retrieved candidate sentences, ensuring efficient execution while respecting provider rate-limit constraints.

The dual-model design (one for extraction, another for evaluation) is justified by prior studies in multi-stage LLM pipelines, which warn against "self-prompting bias" when the same model is used for both retrieval and validation stages. Separating roles enables more robust introspection of errors and better generalizability of judgment behavior across model types.

3.5.2 Technical Stack and Environment

All experiments were implemented in Python 3.11 and executed in a cloud-based computational environment using Kaggle Notebooks. The experiments utilized Kaggle's GPU resources equipped with two NVIDIA Tesla T4 GPUs, along with the standard Kaggle runtime environment based on Ubuntu Linux. GPU acceleration was primarily used for local inference with open-source models, while interactions with proprietary models were executed through remote API calls. The use of a cloud-based execution environment ensured reproducibility and simplified access to GPU resources required for open-source model inference.

The main dependencies included `pandas`, `numpy`, and `tqdm` for data manipulation and progress tracking, as well as `openai` and `requests` for model interfacing. Data were stored as line-delimited JSON (`.jsonl`) files for scalability and easy parsing. Each experimental configuration and output was version-controlled using Git to maintain complete traceability of code and data revisions.

Access to hosted LLMs was managed through authenticated API endpoints. All API calls were executed through synchronous request interfaces. A lightweight scheduling loop sequentially processed documents and facts while respecting provider rate limits and handling transient API failures. Model responses were parsed and validated against expected output patterns, and requests were automatically retried when responses were incomplete or malformed. The number of tokens per request and per response was recorded to monitor cost and com-

putational efficiency. Open-source models were accessed through HTTP-based inference endpoints, allowing the same request interface to be used for both proprietary and locally hosted models. To simplify integration across heterogeneous model providers, the system implements a modular extractor interface that abstracts model-specific request logic. This interface standardizes prompt formatting, response handling, and error management, enabling the pipeline to switch between proprietary and open-source models without modifying the surrounding workflow.

Robust error handling was essential given the stochastic nature of remote model interactions. Each API call was wrapped in a retry mechanism with exponential back-off to recover from transient network or rate-limit errors. All system activities were logged with timestamps, process identifiers, and model metadata. Model outputs were additionally recorded together with prompt identifiers and response timestamps, enabling traceability and reproducibility of all experiments.

The combination of modular design, rigorous logging, and consistent API management ensured that the system operated reliably under varying model conditions and could be re-executed deterministically for future verification studies.

The full implementation is organized as a sequence of modular scripts corresponding to the major stages of the pipeline. Evidence extraction and evidence evaluation are implemented as separate processing stages, allowing intermediate results to be stored and inspected independently. This separation improves debugging transparency and enables re-evaluation of extracted evidence using different evaluator models without repeating the extraction stage. Core processing stages correspond to dedicated scripts for evidence extraction and evidence evaluation, which enables intermediate outputs to be reused and facilitates controlled experimentation across different model configurations.

3.6 Evaluation Metrics and Validation Strategy

To assess the reliability and interpretability of the evidence-grounded evaluation framework, a combination of quantitative metrics, cross-model consistency checks, and external validation experiments were conducted. The objective of this stage was to quantify the degree of factual support expressed across the corpus and to estimate the stability of model judgments under variation in model configuration and dataset composition.

3.6.1 Quantitative Metrics

Each evaluated fact–evidence pair produces three structured outputs: a categorical support label (`support_level`) and two numerical scores (`relevance_score` and `accuracy_score`). These outputs provide both categorical and graded indicators of factual grounding between the extracted claim and the retrieved evidence.

Let N denote the total number of evaluated facts. For each evaluated instance i, the evaluator assigns a relevance score $r_i \in [0, 5]$ and an accuracy score $a_i \in [0, 5]$. If valid numerical scores are returned, they are aggregated across all scored instances N_s to compute the average relevance and accuracy:

$$\bar{r} = \frac{1}{N_s} \sum_{i=1}^{N_s} r_i, \qquad \bar{a} = \frac{1}{N_s} \sum_{i=1}^{N_s} a_i,$$

where N_s denotes the number of facts for which valid evaluator scores were obtained.

In addition to the numerical scores, the evaluator assigns a categorical support label $s_i \in \{\texttt{direct}, \texttt{partial}, \texttt{none}\}$. The proportion of facts assigned to each support category is computed as

$$P_{\text{direct}} = \frac{N_{\text{direct}}}{N}, \qquad P_{\text{partial}} = \frac{N_{\text{partial}}}{N}, \qquad P_{\text{none}} = \frac{N_{\text{none}}}{N},$$

where $N_{\text{direct}}, N_{\text{partial}}, N_{\text{none}}$ denote the number of facts assigned to each support level.

To quantify the coverage of the evidence extraction stage, the evidence retrieval rate (referred to as extraction recall) is defined as

$$R_{\text{evidence}} = \frac{N_{\text{evidence}}}{N},$$

where N_{evidence} represents the number of facts for which the extraction stage successfully retrieved candidate evidence sentences.

Finally, to capture cases where the evaluator detects at least partial semantic alignment between a fact and the evidence, an additional metric is defined as

$$P_{\text{direct+partial}} = \frac{N_{\text{direct}} + N_{\text{partial}}}{N}.$$

These aggregated metrics provide a quantitative summary of factual support, evaluator confidence, and evidence retrieval coverage across the corpus.

3.6.2 Consistency and Agreement Analysis

To assess the robustness of the evaluation framework, results were compared across four extractor–evaluator configurations. Within each model family, extractor and evaluator roles were systematically swapped (e.g., GPT-4 extractor with GPT-3.5 evaluator and vice versa), enabling controlled intra-family comparisons while minimizing confounding differences in training data and architectural design.

Rather than computing a single numerical agreement coefficient, robustness was assessed by comparing the distributions of support labels and the average evaluator scores produced by different extractor–evaluator combinations. Because all configurations evaluate the same set of extracted facts, differences in these distributions provide insight into the stability and calibration of model judgments.

Consistent support distributions and score patterns across configurations indicate stable evaluator behavior, whereas systematic shifts in these statistics reveal differences in how models interpret evidence and assign support levels.

Together, these comparisons provide insight into the internal consistency of the LLM-as-judge evaluation framework and help identify systematic tendencies among different model architectures.

3.6.3 Quality Control and External Validation

To further assess evaluator reliability, an independent human-annotated ZnO dataset [32] was used as a gold-standard reference. Unlike the primary ALD corpus, which relies on LLM-generated factual claims, the ZnO dataset contains manually curated fact–evidence pairs in which the supporting sentence explicitly validates the corresponding fact, enabling external benchmarking of evaluator judgments.

Rather than serving merely as a transferability test, the ZnO dataset was employed to quantify alignment between LLM-based evaluations and human-annotated ground truth labels. Evaluator reliability was assessed by measuring how frequently models correctly identified the evidence as directly supporting the fact, as well as how often partial support was assigned instead of direct support. This gold-standard comparison provides an external benchmark for evaluating the reliability of the LLM-as-judge component.

All metric computations and dataset aggregations were implemented as dedicated post-processing utilities

While cross-family extractor–evaluator combinations (e.g., GPT → Llama) were not executed in the present study, the selected intra-family role-swapping design provides a controlled first-order robustness analysis. Future work may extend this framework to cross-family configurations to further evaluate architectural generalization effects.

4

Results and Discussion

This chapter presents and interprets the results obtained from the Large Language Model (LLM)-based framework for evidence-grounded information extraction and validation in ALD literature. The experiments evaluate how effectively different extractor–evaluator model combinations identify, verify, and quantify factual statements within scientific papers. The analysis focuses on three main aspects: the volume and quality of extracted facts, the distribution of factual support levels, and the consistency of relevance and accuracy judgments.

Importantly, extraction performance and evaluator reliability are analyzed separately. The AWASES-ALD corpus is used to evaluate full pipeline behavior (extraction + evaluation), while a separate human-annotated ZnO dataset is employed exclusively to benchmark evaluator reliability against expert-labeled ground truth. Figure 4.1 summarizes the datasets used for pipeline evaluation and independent benchmarking of evaluator reliability.

4.1 Extraction and Evaluation Outcomes

Four extractor–evaluator configurations were evaluated, grouped into proprietary (closed-source) and open-source model families, as illustrated in Figure 4.2. Within the proprietary family, two role-swapping configurations were tested: GPT-4 as extractor with GPT-3.5 as evaluator, and the reverse configuration where GPT-3.5 acts as the extractor and GPT-4 serves as the evaluator. Similarly, two configurations were evaluated within the open-source family: Llama-3.1-8B as extractor with Qwen2.5-7B as evaluator, and the reciprocal configuration where Qwen2.5-7B

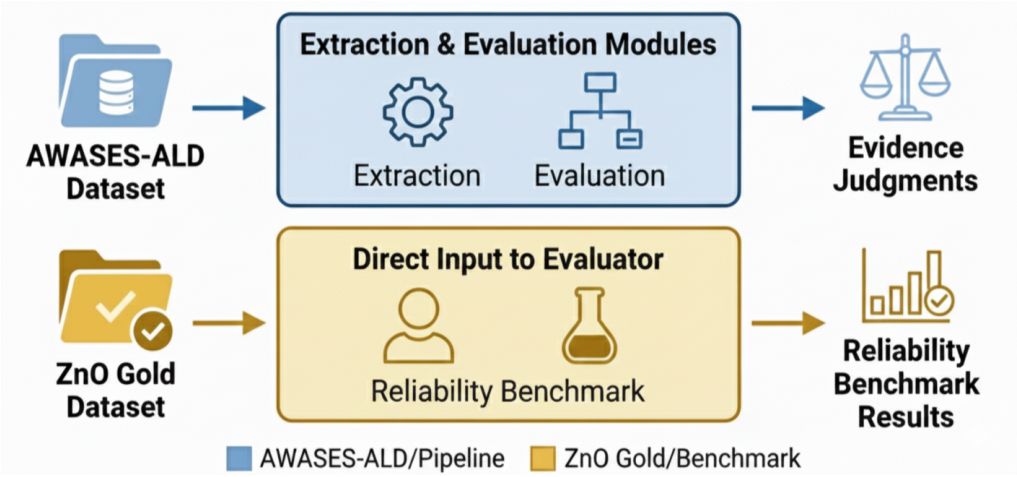


Figure 4.1: Datasets used in the experiments: AWASES-ALD for pipeline evaluation and the ZnO dataset for evaluator benchmarking.

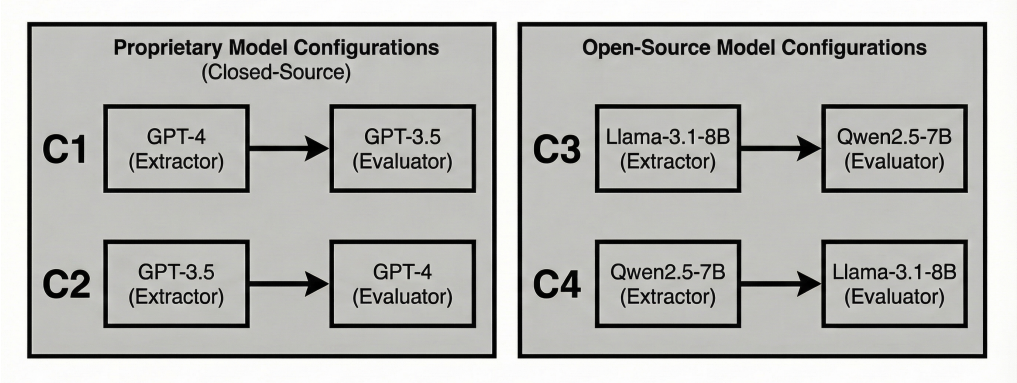


Figure 4.2: Extractor–evaluator model combinations used in the experimental evaluation.

performs extraction while Llama-3.1-8B acts as the evaluator. This experimental design enables controlled intra-family robustness analysis through systematic swapping of extractor and evaluator roles, while the corresponding configurations are visually summarized in Figure 4.2.

All four configurations were evaluated on a subset of the AWASES-ALD corpus comprising 141 research papers. Across these documents, a total of 2,195 structured factual statements were extracted and used as the evaluation set. Because the same fact set is assessed under each configuration, the extraction volume remains constant, allowing differences in performance to be attributed to variations in evidence retrieval and evaluator behavior rather than differences in the underlying data.

The total number of evaluated facts therefore remains fixed across configurations, while the

evidence-found rate reflects the effectiveness of each model pairing in locating supporting passages within the source documents. Table 4.1 and Figure 4.3 summarize the proportion of extracted facts for which supporting evidence was successfully identified in the corresponding paper.

Combination 3 achieves complete evidence retrieval (100%), indicating that the Llama–Qwen configuration identifies at least one relevant evidence segment for every extracted fact. Combination 2 also demonstrates strong coverage with an evidence-found rate of 92.20%, followed by Combination 1 with 84.23%. Combination 4 shows the lowest retrieval coverage (78.67%), suggesting that the Qwen–Llama configuration is somewhat less effective at locating supporting passages within the corpus.

This result indicates that the extractor identifies at least one candidate evidence span for every fact, although the subsequent evaluation stage determines whether the relationship is direct, partial, or unsupported. The detailed distribution of direct, partial, and unsupported

Table 4.1: Extraction coverage and evidence identification results for the four extractor–evaluator combinations on the AWASES-ALD corpus (141 papers, 2,195 extracted facts).

Metric	C1	C2	C3	C4
Facts Extracted	2195	2195	2195	2195
Evidence Found (%)	84.23	92.20	100.00	78.67

judgments is presented in Table 4.2. Combination 1 yields the highest proportion of `direct` supports (75.63%), indicating strong alignment between extracted facts and their textual evidence. Combination 2 shows a slightly lower direct-support rate (72.30%) but a higher proportion of `partial` supports (11.80%), suggesting a somewhat more conservative evaluator interpretation.

The open-source configuration C3 exhibits the lowest proportion of direct supports (59.64%) and the highest proportion of partial supports (29.52%). This pattern indicates that the evaluator frequently recognizes semantic relevance but refrains from assigning full direct support when the relationship between fact and evidence is less explicit.

Combination 4 presents a more balanced distribution, with 66.43% direct support, 12.66% partial support, and the highest unsupported rate (20.91%). This suggests that the Qwen–Llama configuration applies stricter criteria when determining whether the evidence fully substantiates the extracted claim.

Figure 4.4 compares the support-level distributions across proprietary and open-source model

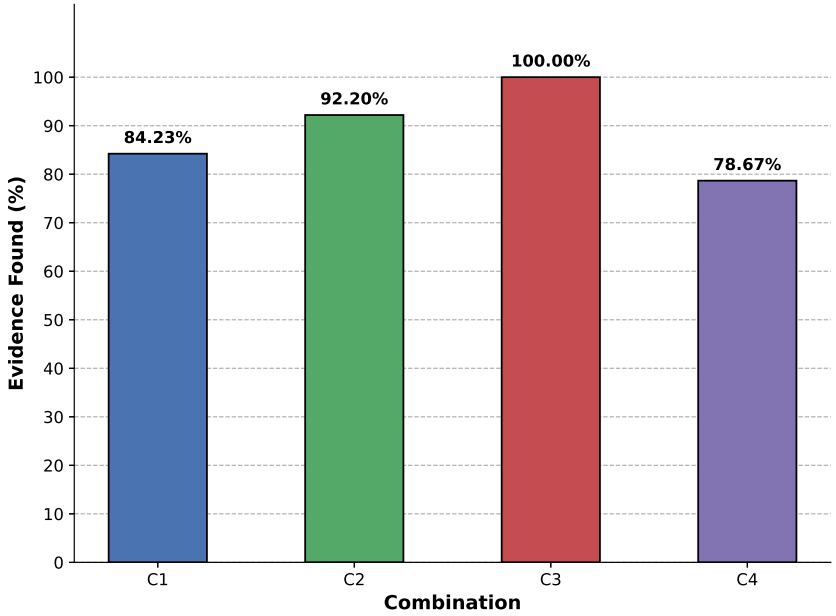


Figure 4.3: Evidence retrieval rates across the four extractor–evaluator configurations on the AWASES-ALD corpus.

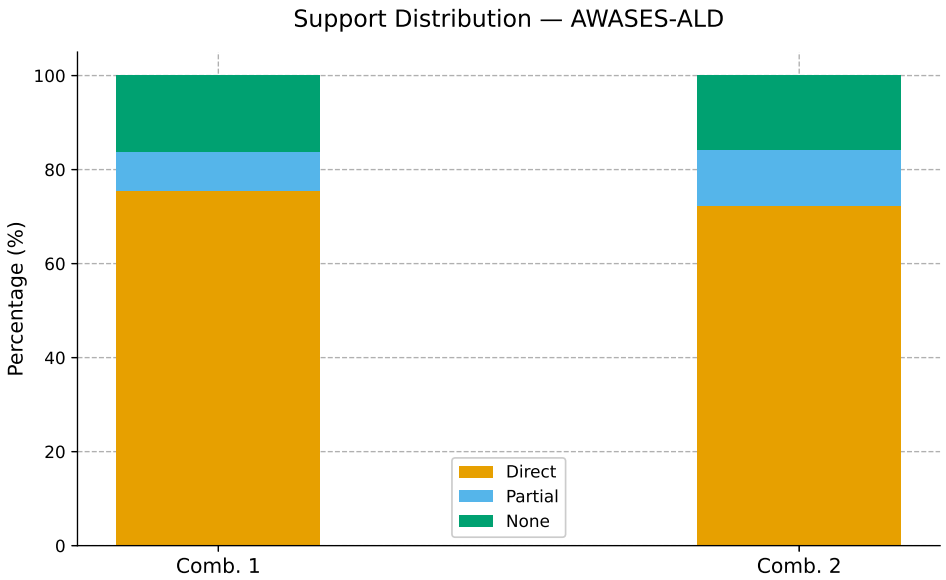
Table 4.2: Distribution of support levels for all four combinations on the AWASES-ALD corpus.

Support Level (%)	C1	C2	C3	C4
Direct	75.63	72.30	59.64	66.43
Partial	8.11	11.80	29.52	12.66
None	16.26	15.90	10.84	20.91

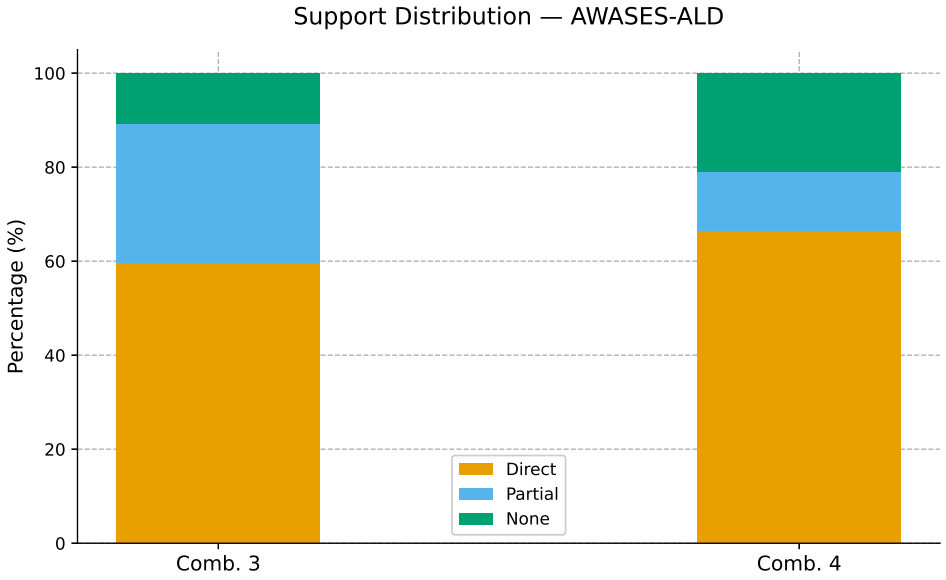
families.

Evaluator scoring trends generally align with the categorical support distributions shown above (Table 4.3). Combination 4 achieves the highest average relevance and accuracy scores (both 4.83), indicating that the Llama evaluator produces particularly confident judgments when paired with the Qwen extractor. Combination 1 also demonstrates strong evaluator performance with an average relevance score of 4.61 and accuracy score of 4.36.

Combination 2 yields slightly lower evaluator scores despite its high evidence retrieval rate, suggesting that the GPT-4 evaluator applies somewhat stricter scoring criteria. Combination 3 records the lowest average relevance score (4.12) but maintains competitive accuracy (4.29), reinforcing the observation that the Qwen evaluator tends to assign partial support rather than direct support in ambiguous cases.



(a) Proprietary model configurations (C1, C2).



(b) Open-source model configurations (C3, C4).

Figure 4.4: Distribution of support levels (**`direct`**, **`partial`**, and **none**) for the AWASES-ALD dataset across extractor–evaluator configurations. (a) Proprietary model combinations. (b) Open-source model combinations.

4.2 Comparative Analysis of Model Combinations

Within the proprietary family (C1 and C2), both configurations extract identical numbers of facts but exhibit noticeable differences in evidence retrieval and evaluator behavior. Combi-

Table 4.3: Average evaluator scores (0–5 scale) across combinations on the AWASES-ALD corpus.

Metric	C1	C2	C3	C4
Avg. Relevance	4.61	4.42	4.12	4.83
Avg. Accuracy	4.36	4.33	4.29	4.83

nation 2 achieves the highest evidence-found rate among proprietary configurations (92.20%), whereas Combination 1 yields the highest proportion of direct supports (75.63%). These results suggest that the GPT-4 extractor paired with GPT-3.5 evaluator produces slightly more decisive support classification, while the reverse pairing favors broader evidence retrieval but results in a higher proportion of partial judgments.

Among the open-source configurations, Combination 3 (Llama extractor, Qwen evaluator) achieves perfect evidence retrieval (100%), indicating that the extractor successfully identifies at least one candidate evidence segment for every extracted fact. However, this configuration also shows the highest proportion of partial supports (29.52%) and the lowest direct-support rate (59.64%). This pattern suggests that the Qwen evaluator frequently detects semantic relevance but applies conservative criteria when determining whether the evidence fully substantiates the extracted fact.

Combination 4 (Qwen extractor, Llama evaluator) presents a different behavior profile. Although it exhibits the lowest evidence-found rate among all configurations (78.67%), it achieves the highest evaluator confidence scores, with both average relevance and accuracy reaching 4.83. At the same time, this configuration records the highest unsupported rate (20.91%), indicating that the Llama evaluator applies stricter criteria when determining whether the retrieved evidence sufficiently supports the claim.

Taken together, these observations indicate that differences between model combinations arise primarily from evaluator calibration rather than extraction volume. While the Llama extractor demonstrates strong retrieval coverage in C3, the corresponding Qwen evaluator assigns partial support more frequently. Conversely, the Llama evaluator in C4 produces more decisive relevance and accuracy scores but rejects unsupported claims more aggressively. These findings highlight the importance of separating extraction and evaluation roles when analyzing LLM-based verification pipelines.

These results demonstrate that the behavior of the pipeline is strongly influenced by evaluator calibration, with different models exhibiting varying thresholds for assigning direct support, partial support, or rejection of claims.

4.3 Evaluator Reliability on the ZnO Gold Dataset

To independently assess evaluator reliability, a separate human-annotated ZnO dataset was used as a gold-standard benchmark. Unlike the AWASES-ALD corpus, this dataset does not involve fact extraction and does not represent a multi-class support classification task. Instead, it contains manually curated fact–evidence pairs in which the evidence sentence explicitly supports the corresponding fact.

Each instance therefore represents a confirmed positive support case. Consequently, the purpose of this evaluation is not to determine which support category applies, but rather to measure whether evaluator models correctly recognize that a given evidence sentence directly supports the associated fact.

In this setting, evaluator reliability is measured by the proportion of instances for which the evaluator correctly assigns the label **`direct`**. Cases where the evaluator assigns **`partial`** or **`none`** represent missed support detections (false negatives).

Table 4.4: Evaluator recognition of fact–evidence support in the ZnO gold dataset containing 327 human-annotated fact–evidence pairs.

Metric	GPT-4	GPT-3.5	Llama-3.1-8B	Qwen2.5-7B
Direct Agreement (%)	76.15	86.24	81.04	67.58
Direct + Partial (%)	88.38	98.17	96.02	89.91
Avg. Relevance Score	4.29	4.69	4.68	4.28
Avg. Accuracy Score	4.20	4.48	4.69	4.43

Direct Agreement measures the proportion of instances in which the evaluator correctly identifies the fact–evidence pair as directly supported. The metric Direct + Partial reflects cases where the evaluator recognizes a meaningful relationship between the fact and evidence, even if the support is judged conservatively as partial.

The ZnO dataset contains a total of 327 curated fact–evidence pairs, each representing a confirmed positive support relationship. Table 4.4 summarizes evaluator performance on this benchmark.

Among the evaluated models, GPT-3.5 achieves the highest Direct Agreement rate (86.24%), indicating the strongest ability to recognize explicit fact–evidence grounding. The Llama-3.1-8B evaluator follows with a Direct Agreement of 81.04%, while GPT-4 achieves 76.15%. Qwen2.5-7B shows the lowest Direct Agreement rate at 67.58%, suggesting that it more frequently assigns the **`partial`** label even when the support relationship is explicit.

When partial support assignments are included, recognition rates increase substantially for all evaluators. GPT-3.5 again demonstrates the highest support detection capability with a Direct+Partial rate of 98.17%, followed by Llama-3.1-8B (96.02%), Qwen2.5-7B (89.91%), and GPT-4 (88.38%). These results indicate that most evaluators correctly detect the existence of a semantic relationship between the fact and evidence even when the support classification is conservative.

The evaluator scoring metrics show similar trends. GPT-3.5 achieves the highest average relevance score (4.69), closely followed by Llama-3.1-8B (4.68), indicating strong semantic alignment between predicted judgments and the supporting text. Llama-3.1-8B attains the highest average accuracy score (4.69), while GPT-3.5 (4.48), Qwen2.5-7B (4.43), and GPT-4 (4.20) follow with slightly lower values.

Because every instance in the ZnO dataset represents a true support relationship, instances labeled as `partial` or `none` correspond to cases where the evaluator does not fully recognize the directness of the supporting evidence. Such cases therefore represent conservative evaluator judgments rather than incorrect fact–evidence pairings.

Importantly, the ranking of evaluators observed in the ZnO benchmark is broadly consistent with the behavioral patterns observed in the AWASES pipeline experiments. Models that exhibit higher rates of partial-support assignments in the AWASES evaluation similarly show lower Direct Agreement in the ZnO benchmark. This consistency suggests that the variations observed in the pipeline results are primarily attributable to differences in evaluator calibration rather than extraction errors.

4.4 Discussion, Design Implications, and Limitations

The combined evidence from internal robustness testing on the AWASES-ALD corpus and external benchmarking using the ZnO gold dataset strengthens confidence in the proposed evidence-grounded verification framework.

The ZnO evaluation confirms that all evaluators are capable of reliably identifying true fact–evidence relationships, with proprietary models demonstrating slightly higher recognition rates and more decisive support classification. Open-source evaluators remain competitive while showing a modest tendency toward conservative classification behavior.

From a methodological perspective, the separation between pipeline evaluation (AWASES dataset) and evaluator benchmarking (ZnO dataset) improves interpretability of the results. Pipeline metrics reflect the interaction between extraction and evaluation stages, whereas ZnO

benchmarking isolates the evaluator component and quantifies its alignment with human-curated ground truth.

Computational considerations also differ across model families. Proprietary extended-context models provide highly consistent support recognition but incur higher computational cost, while open-source models offer scalable alternatives with only moderate reductions in evaluator decisiveness. This trade-off highlights the potential for hybrid systems in which open-source models perform large-scale extraction while proprietary models are reserved for high-confidence evaluation tasks.

Overall, the results demonstrate that the proposed evidence-grounded pipeline can reliably extract and verify scientific facts at scale while maintaining strong alignment with expert-validated fact–evidence relationships. The integration of independent evaluator benchmarking further enhances the transparency and credibility of the framework.

Despite these promising results, several limitations remain. First, the current pipeline relies primarily on LLM-based semantic retrieval rather than structured information retrieval techniques, which may introduce variability when processing highly complex scientific descriptions. Second, evaluator judgments are generated through prompt-based reasoning, meaning that classification behavior may be sensitive to prompt design and model calibration. Finally, although the ZnO dataset provides a useful gold-standard benchmark, its domain coverage remains limited to a specific class of materials science experiments.

Future work may address these limitations by integrating hybrid retrieval approaches that combine semantic search with structured knowledge extraction. Further improvements could also be achieved through domain-adapted language models trained on materials science literature and through the development of larger human-annotated benchmarks for scientific fact verification.

5 Conclusion and Future Work

5.1 Conclusion

This thesis presented the design, implementation, and evaluation of an evidence-grounded framework for Large Language Model (LLM)–based factual extraction and validation in Atomic Layer Deposition (ALD) literature. The central objective was to develop a transparent and reproducible methodology capable of verifying structured scientific claims against their original textual sources while minimizing model self-confirmation bias.

The proposed pipeline separates fact extraction and evidence evaluation into distinct stages and assigns these roles to different language models. This architectural separation mitigates circular validation effects and enables systematic robustness analysis of model behavior. Four extractor–evaluator configurations were implemented across proprietary and open-source model families, with controlled intra-family role swapping to isolate evaluator calibration effects from extraction variability.

Experimental results on the AWASES-ALD corpus demonstrate that the framework can reliably process large volumes of structured scientific information. Across a subset of 141 research papers, a total of 2,195 factual statements were extracted and evaluated against their corresponding source texts. Because the same fact set was used across all configurations, differences in performance reflect variations in evidence retrieval effectiveness and evaluator calibration rather than differences in extraction volume.

The results show that proprietary model configurations exhibit tightly clustered support distributions and consistent evaluator scoring behavior, indicating stable and decisive support classification. In contrast, open-source configurations demonstrate greater dispersion in support categorization, particularly through increased assignment of partial-support judgments. These patterns suggest that differences between model combinations arise primarily from evaluator calibration rather than from the ability to identify candidate evidence passages.

To independently assess evaluator reliability, a human-annotated ZnO dataset was used as a gold-standard benchmark. This dataset contains 327 curated fact–evidence pairs derived from materials science literature, where the evidence sentence explicitly supports the associated fact. Unlike the AWASES corpus, which evaluates the full pipeline, the ZnO dataset isolates evaluator behavior by presenting predefined fact–evidence pairs without requiring fact extraction.

Evaluation on this dataset measures the ability of each model to correctly recognize explicit fact–evidence grounding. The results show strong support recognition across all evaluators, with proprietary and open-source models both demonstrating high rates of semantic alignment between facts and evidence. Differences between models primarily reflect varying thresholds for assigning direct versus partial support rather than an inability to detect supporting relationships.

Taken together, the findings validate three central contributions of this thesis. First, the work establishes a modular evidence-grounded verification pipeline capable of extracting and validating structured scientific knowledge from ALD literature at scale. Second, it introduces a controlled intra-family role-swapping experimental design that isolates evaluator calibration effects and enables systematic comparison of model behavior without conflating extraction variability. Third, it incorporates an external human-aligned benchmarking strategy using a gold-standard ZnO dataset, allowing evaluator reliability to be assessed independently of the extraction component.

The results demonstrate that LLM-based scientific fact verification can achieve reliable grounding when extraction and evaluation roles are explicitly decoupled and when evaluation performance is validated against expert-curated evidence pairs. The framework therefore provides a scalable and reproducible foundation for automated knowledge grounding in materials science literature and potentially other scientific domains.

Despite the promising results obtained in this study, several limitations should be acknowledged.

First, although intra-family role swapping provides controlled robustness analysis, cross-family extractor–evaluator combinations were not executed. Such experiments could further clarify

architectural interaction effects between proprietary and open-source models and provide a more comprehensive evaluation of pipeline robustness.

Second, while evaluator reliability was benchmarked using a human-annotated dataset, the ZnO corpus represents a relatively narrow materials-science subdomain. Broader domain coverage and additional gold-standard datasets would strengthen claims of generalizability across different materials systems and experimental contexts.

Third, the categorical support schema used in the evaluation process (direct, partial, none) simplifies the complexity of scientific reasoning. Experimental results in materials science often involve conditional dependencies or contextual qualifiers that may not be fully captured within a three-category support framework.

Finally, although prompt standardization and deterministic inference settings were applied, LLM-based evaluation remains sensitive to prompt formulation and model calibration. Minor variations in judgment may therefore occur due to the probabilistic nature of large language model inference.

These limitations do not undermine the core findings of the study but highlight opportunities for methodological refinement and further empirical validation.

5.2 Future Work

Several directions can extend and strengthen the framework developed in this thesis.

Future research could incorporate cross-family extractor–evaluator configurations, such as pairing proprietary extractors with open-source evaluators and vice versa. Such experiments would help disentangle architectural biases from model-family calibration effects and provide a more comprehensive robustness analysis of LLM-based verification pipelines.

Another important direction involves expanding the availability of human-annotated benchmark datasets. Constructing additional gold-standard corpora spanning multiple materials systems and experimental conditions would enable more rigorous evaluation of evaluator generalization and support the use of formal agreement metrics such as Cohen's κ or macro-averaged F1 scores.

Further improvements may also arise from incorporating uncertainty-aware evaluation mechanisms. Techniques such as confidence estimation, probabilistic scoring, or ensemble-based judgment aggregation could help distinguish between ambiguous partial-support cases and genuine evaluator disagreement.

In addition, the current pipeline verifies extracted facts but does not yet integrate them into a dynamic knowledge representation system. Extending the framework to support automated knowledge graph construction, contradiction detection, and cross-paper consistency analysis would significantly enhance its utility for large-scale scientific knowledge management.

Human-in-the-loop hybrid systems represent another promising direction. While automated evaluation enables scalability, integrating targeted expert review for borderline or partial-support cases could improve final knowledge quality and provide valuable feedback for model calibration.

Finally, prompt optimization and domain adaptation strategies may improve the performance of open-source models. Fine-tuning or instruction alignment using domain-specific scientific corpora could reduce partial-support inflation while preserving the cost-efficiency and scalability advantages of open-source language models.

The rapid development of large language models presents both opportunities and methodological challenges for scientific knowledge extraction and verification. This thesis demonstrates that reliable scientific fact grounding requires careful architectural design, explicit separation between extraction and evaluation roles, and independent validation against human-curated evidence.

By combining modular pipeline design, controlled model comparisons, and external gold-standard benchmarking, the proposed framework advances the reliability and interpretability of automated fact verification in materials science literature. Although further research is needed to extend its generalizability and integrate it with broader knowledge management systems, the methodological principles developed in this work provide a strong foundation for trustworthy LLM-assisted scientific knowledge extraction.

References

[1] V. Venugopal and E. Olivetti, "Matkg: An autonomously generated knowledge graph in material science," *Scientific Data*, vol. 11, no. 1, p. 217, 2024.

[2] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg *et al.*, "The fair guiding principles for scientific data management and stewardship," *Scientific Data*, vol. 3, no. 1, p. 160018, 2016.

[3] M. C. Swain and J. M. Cole, "Chemdataextractor: A toolkit for automated extraction of chemical information from the scientific literature," *Journal of Chemical Information and Modeling*, vol. 56, no. 10, pp. 1894–1904, 2016.

[4] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "Fever: A large-scale dataset for fact extraction and verification," in *Proceedings of NAACL-HLT*. Association for Computational Linguistics, 2018, pp. 809–819.

[5] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, and H. Hajishirzi, "Fact or fiction: Verifying scientific claims," in *Proceedings of EMNLP*. Association for Computational Linguistics, 2020, pp. 7534–7550.

[6] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, D. Chen, W. Dai *et al.*, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 14, p. 248, 2023.

[7] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," *Advances in neural information processing systems*, vol. 36, pp. 46 595–46 623, 2023.

[8] S. M. George, "Atomic layer deposition: an overview," *Chemical reviews*, vol. 110, no. 1, pp. 111–131, 2010.

[9] R. W. Johnson, A. Hultqvist, and S. F. Bent, "A brief review of atomic layer deposition: from fundamentals to applications," *Materials today*, vol. 17, no. 5, pp. 236–246, 2014.

[10] M. Leskelä and M. Ritala, "Atomic layer deposition chemistry: recent developments and future challenges," *Angewandte Chemie International Edition*, vol. 42, no. 45, pp. 5548–5554, 2003.

[11] M. Ritala and M. Leskelä, "Atomic layer deposition," in *Handbook of Thin Films*. Elsevier, 2002, pp. 103–159.

[12] H. C. Knoops, S. E. Potts, A. A. Bol, and W. Kessels, "Atomic layer deposition," in *Handbook of Crystal Growth*. Elsevier, 2015, pp. 1101–1134.

[13] F. Rahman and J. C. Runyon, "Atomic layer processes for material growth and etching—a review," *IEEE Transactions on Semiconductor Manufacturing*, vol. 34, no. 4, pp. 500–512, 2021.

[14] D. Mrdjenovich, M. K. Horton, J. H. Montoya, C. M. Legaspi, S. Dwaraknath, V. Tshitoyan, A. Jain, and K. A. Persson, "Propnet: a knowledge graph for materials science," *Matter*, vol. 2, no. 2, pp. 464–480, 2020.

[15] V. Venugopal, S. Pai, and E. Olivetti, "Matkg: The largest knowledge graph in materials science–entities, relations, and link prediction through graph representation learning," *arXiv preprint arXiv:2210.17340*, 2022.

[16] V. Nechakhin, J. D'Souza, and S. Eger, "Evaluating large language models for structured science summarization in the open research knowledge graph," *Information*, vol. 15, no. 6, p. 328, 2024.

[17] Z. Nasar, S. W. Jaffry, and M. K. Malik, "Information extraction from scientific articles: a survey," *Scientometrics*, vol. 117, no. 3, pp. 1931–1990, 2018.

[18] G. Bekoulis, C. Papagiannopoulou, and N. Deligiannis, "A review on fact extraction and verification," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–35, 2021.

[19] X. Deng, X. Wang, and M. Stevenson, "The next phase of scientific fact-checking: advanced evidence retrieval from complex structured academic papers," in *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, 2025, pp. 436–448.

[20] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu *et al.*, "A survey on llm-as-a-judge," *arXiv preprint arXiv:2411.15594*, 2024.

[21] H. Li, Q. Dong, J. Chen, H. Su, Y. Zhou, Q. Ai, Z. Ye, and Y. Liu, "Llms-as-judges: a comprehensive survey on llm-based evaluation methods," *arXiv preprint arXiv:2412.05579*, 2024.

[22] H. Wei, S. He, T. Xia, F. Liu, A. Wong, J. Lin, and M. Han, "Systematic evaluation of llm-as-a-judge in llm alignment tasks: Explainable metrics and diverse prompt templates," *arXiv preprint arXiv:2408.13006*, 2024.

[23] B. Murugadoss, C. Poelitz, I. Drosos, V. Le, N. McKenna, C. S. Negreanu, C. Parnin, and A. Sarkar, "Evaluating the evaluator: Measuring llms' adherence to task evaluation instructions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 18, 2025, pp. 19 589–19 597.

[24] S. Badshah and H. Sajjad, "Reference-guided verdict: Llms-as-judges in automatic evaluation of free-form text," *arXiv preprint arXiv:2408.09235*, 2024.

[25] M. Boyapati, L. Meesala, R. Aygun, B. Franks, H. Choi, S. Riordan, and G. Modgil, "Leveleval: Adaptive pipeline for evaluating llm as a judge-analysis on open llms as judges," in *2024 International Conference on AI x Data and Knowledge Engineering (AIxDKE)*. IEEE, 2024, pp. 74–77.

[26] L. Bergeron, I. Buhnila, J. François, and R. State, "Halluguard: Evidence-grounded small reasoning models to mitigate hallucinations in retrieval-augmented generation," *arXiv preprint arXiv:2510.00880*, 2025.

[27] A. Mackus, B. Macco, B. Karasulu, J. D'Souza, S. Auer, and E. Kessels, "Turning online ald and ale databases into ai-ready tools for development of new sustainable materials and fabrication processes," in *Proceedings of the AVS 24th International Conference on Atomic Layer Deposition (ALD 2024)*. AVS, 2024.

[28] OpenAI, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[29] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[30] Q. Team, "Qwen2.5 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2412.15115

[31] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[32] J. Malm, E. Sahramo, J. Perälä, T. Sajavaara, and M. Karppinen, "Low-temperature atomic layer deposition of zno thin films: Control of crystallinity and orientation," *Thin Solid Films*, vol. 519, no. 16, pp. 5319–5322, 2011.

Acknowledgments

I would like to express my sincere gratitude to the University of Padova for providing the academic foundation and support for this thesis.

I am especially grateful to Prof. Gianmaria Silvello for his guidance and supervision throughout this work. His feedback and support were invaluable during the development of this research.

I would also like to thank TIB – Leibniz Information Centre for Science and Technology University Library for providing the research context and resources that supported this study. In particular, I am grateful to Prof. Sören Auer for his leadership and for enabling the research environment in which this work was carried out.