ARL-TR-10443 ● SEP 2026

An AI/ML-Augmented Toolkit for Materials Intelligence through Knowledge Fusion and Reasoning

by Xiongjun Wu and B. Chad Hornbuckle

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-10443 ● SEP 2026

An AI/ML-Augmented Toolkit for Materials Intelligence through Knowledge Fusion and Reasoning

Xiongjun Wu and B. Chad Hornbuckle
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
September 2026	Technical Report	1 October 2025	31 July 2026

4. TITLE AND SUBTITLE
An AI/ML-Augmented Toolkit for Materials Intelligence through Knowledge Fusion and Reasoning

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER	5e. TASK NUMBER	5f. WORK UNIT NUMBER

6. AUTHOR(S)
Xiongjun Wu and B. Chad Hornbuckle

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-MC Aberdeen Proving Ground, MD 21005	ARL-TR-10443

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
Large-language models (Gemini 3.1 Pro) were used as editorial tools to review and polish the draft for grammar, clarity, and English-language fluency. The authors reviewed and approved the suggested revisions and remain responsible for the technical content and final text of the report.

14. ABSTRACT
High-throughput materials discovery requires the assimilation of structured and unstructured data distributed across public literature, proprietary records, external databases, and internal experimental observations. Although large-language models (LLMs) offer powerful capabilities for processing technical information, their unconstrained use risks hallucinations, weak provenance, and insufficient domain rigor. To address these challenges, this technical report describes the concept, architecture, and an example implementation of an AI/ML-Augmented Toolkit for materials intelligence that enables structured knowledge fusion and multistep reasoning.
This toolkit combines retrieval-augmented generation, an explicit materials ontology, a typed, provenance-aware knowledge graph (KG), and federated materials database search across external repositories (e.g., OPTIMADE, JARVIS) within an orchestrated tool-calling workflow. The ontology guides the extraction and normalization of materials entities and relationships, while the KG supports relational traversal across distributed evidence. Within the tool-calling loop, the LLM proposes structured tool requests, the Orchestrator executes the selected tools, and the resulting literature passages, graph relationships, database records, or conversation context are returned to the LLM for synthesis. This architecture constrains foundation models using explicit tools, domain boundaries, and semantic extraction to ensure auditable, cross-source validation.
The toolkit is demonstrated using refractory high-entropy alloys, whose complex design space requires integrated reasoning over composition, processing, phase stability, characterization, and properties. Accessed via an interactive widget, the workflow ingests literature, normalizes extracted entities, links relationships to source evidence, and synthesizes findings for evidence-based candidate prioritization and gap identification. By bridging otherwise isolated evidence sources through a unified, orchestrated interface, the toolkit provides a trustworthy, reproducible, and scalable materials-intelligence layer that supports hypothesis generation, candidate screening, and experiment planning for accelerated Army materials-discovery missions.

15. SUBJECT TERMS
high-throughput materials discovery; cross-platform data fusion; refractory high-entropy alloys; materials informatics; knowledge graph; ontology; retrieval-augmented generation; large-language models; OPTIMADE; JARVIS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	28

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Xiongjun Wu	(520) 691-5396

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

1.1 HTMDEC Context and Report Focus

The broader program context for this work is the High-Throughput Materials Discovery for Extreme Conditions (HTMDEC) Collaborative Research Alliance, which includes U.S. Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory (ARL) intramural research projects and extramural university-led activities that collaborate to accelerate materials discovery for extreme environments.[1]

This technical report focuses on the concept and reference implementation of an ontology-grounded, knowledge-graph-based AI/ML-Augmented Toolkit developed under the intramural effort Cross-Platform Data Fusion for High-Throughput Refractory HEA Discovery. It explains how the toolkit supports the broader collaborative materials-discovery mission. While extramural partners like the BIRDSHOT center provide complementary alloy design, synthesis, and evaluation context for refractory high-entropy alloys (RHEAs), the intramural toolkit provides a core knowledge-fusion capability for ARL's internal high-throughput materials development platform. The toolkit organizes, retrieves, relates, and supports reasoning over distributed and disparate data sources.

1.2 Overall Workflow of the Intramural Project

The intramural effort sits within an overall workflow designed to meet stringent operational requirements. As illustrated in Figure 1, this workflow links multiple domains—computational materials methods, including density functional theory, molecular dynamics, and CALPHAD (CALculation of PHAse Diagrams)—as well as human researchers, public-domain materials databases (AFLOW, JARVIS, NOMAD), open literature, the ARL materials data-management environment (RHEA database, ARL Materials Intelligence [MI] Data Management System [DMS]), and high-throughput experimental activities (Design of Experiments [DOE], Synthesis/Process, Characterization).

In this ecosystem, the AI/ML-Augmented Toolkit acts as the central information-fusion and reasoning layer. It connects upstream sources of knowledge to downstream research actions, supporting data collection, analysis, assimilation, candidate prioritization, and experiment planning. This integration is crucial in a high-throughput setting, where the sheer volume of candidate materials and associated measurements can easily overwhelm manual review and traditional isolated search methods.

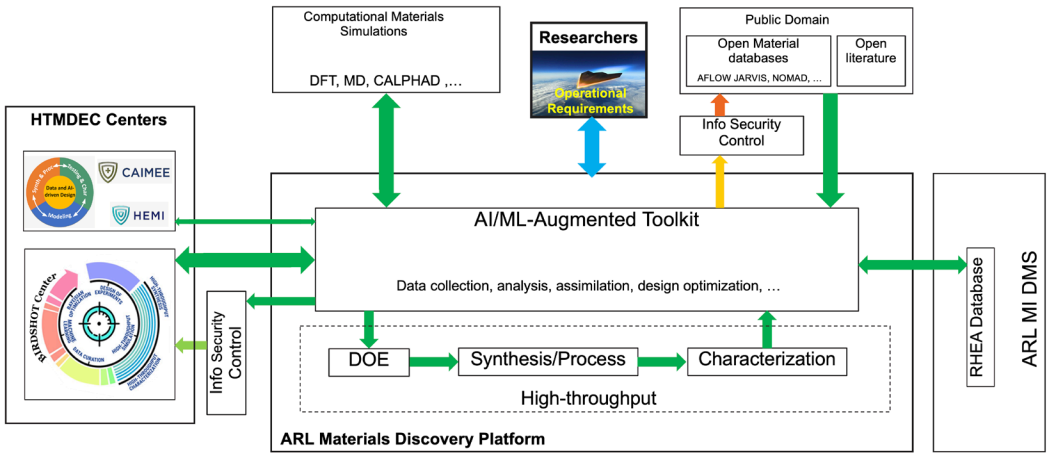


Figure 1. Overall workflow of the intramural project, showing the relationship among collaborating HTMDEC centers, computational inputs, public-domain sources, the AI/ML-Augmented Toolkit, high-throughput experimentation, and the ARL data-management environment.

1.3 Why a Large-Language Model–based Toolkit Matters Now

The rapid development of foundation models has made this kind of toolkit timely and technically important. Recent large-language models (LLMs) offer capabilities directly relevant to materials-discovery workflows, including structured information extraction, tool calling, summarization across long contexts, and the ability to interact with researchers using plain language.

Furthermore, the Department of War (DoW) push for generative AI has facilitated secure application program interface (API) access to high-capability cloud models via platforms like GenAI, AskSage, and Army Vantage. This allows the workflow to balance capability and security, bridging the gap between compute-limited local models and powerful cloud-based reasoning engines.

However, unconstrained LLM use can lead to hallucination, weak provenance, limited reproducibility, and ambiguity about what evidence underlies a generated answer.[2,3] In Army and ARL settings, these issues are not minor inconveniences; they directly affect trustworthiness, security posture, and operational usability. The technical problem addressed here is therefore not simply how to "use an LLM," but how to engineer a grounded workflow in which LLM capability is constrained by explicit data sources, tools, and domain structure. It is designed to provide researchers with a robust capability for knowledge discovery, hypothesis generation, and multistep reasoning for MI.

1.4 Retrieval-Augmented Generation, Knowledge Graphs, Ontologies, and Grounding Strategy

Several architectural strategies exist for adapting LLMs to a technical domain. Extensive model fine-tuning can improve domain familiarity but is expensive to maintain, provides limited provenance, and does not inherently ground responses in current evidence. Retrieval-augmented generation (RAG) improves grounding by tying answers to retrieved documents, but vector-only retrieval struggles with multi-hop reasoning across scattered sources.[2]

To overcome these limitations, the architecture reported here pairs RAG with a knowledge graph (KG) and an explicit materials ontology.[3] The ontology defines canonical classes, aliases, and relationship constraints. The graph stores typed entities extracted from documents. Together, these provide a highly inspectable reasoning substrate. The LLM handles extraction, entity resolution, and synthesis, while the ontology and graph enforce normalization, validation, and structural relationships. Table 1 summarizes and compares these grounding strategies across key dimensions such as updatability, provenance, relational reasoning, hallucination risk, and schema enforcement.

Table 1. Comparison of grounding strategies for MI, contrasting domain fine-tuning alone, vector-only RAG, and the present RAG + ontology + KG + tool-execution stack.

Property	Domain fine-tuning	Vector-only RAG	Toolkit Architecture (RAG + KG + Ontology)
Updatability	Low (requires expensive retraining)	High (add/remove documents from vector store)	High (dynamic updates to KG and document store)
Provenance	Poor (implicit knowledge in weights)	Good (links to specific text passages)	Excellent (explicit citations tied to graph nodes)
Relational reasoning	Limited (relies on learned weights)	Poor (struggles with multi-hop connections)	Excellent (explicit graph traversal and logic)
Hallucination risk	High (unguided generation)	Medium (context-bounded but unconstrained)	Low (ontology-validated extraction and tools)
Schema enforcement	None	None	Strict (enforced by materials ontology)

1.5 Objective and Scope

The objective of this technical report is to present and demonstrate a technical approach for grounded MI based on the integration of retrieval-augmented generation, an explicit materials ontology, a typed KG, and tool-mediated LLM reasoning. The significance of the work lies in this knowledge-fusion architecture: heterogeneous materials evidence is transformed into normalized, relational

knowledge that can be searched, traversed, compared, and synthesized in support of scientific decision-making.

The report documents the design, implementation, and demonstration of the workflow with an AI/ML-Augmented Toolkit for RHEAs; it explains why the toolkit is needed; how it is implemented; how it fits into the broader intramural project workflow; and what capabilities it currently provides to researchers to speed up materials discovery. RHEAs serve as an ideal proving ground to demonstrate the toolkit with a highly complex design space requiring integrated reasoning over composition, processing history, phase stability, and properties—information that is heavily distributed across literature, databases, and experimental observations.

The implementation described in Sections 2 and 3 provides traceability from the concept of an ontology-grounded knowledge fusion framework to an operational realization, while the RHEA demonstration in Section 4 illustrates how the framework can support evidence synthesis, hypothesis generation, and identification of knowledge gaps. Functionalities not covered in this report—such as autonomous first-principles compute execution, CALPHAD generation, phase-diagram generation, and batch Bayesian optimization (BBO)—are some of the functions to be developed and/or integrated in the future for the completion of the comprehensive materials discovery platform.

2. Toolkit Design and Architecture

2.1 Core Architecture Overview

Cross-platform data fusion requires the deliberate integration of multiple evidence and execution platforms into one operational workflow. The toolkit achieves this by fusing heterogeneous knowledge sources—ingested literature, typed KGs, and federated external materials databases—while maintaining explicit boundaries among them.

As illustrated in Figure 2, the architecture follows an Orchestrator and Tool-Calling loop paradigm. A researcher inputs a query into the interactive user interface (UI). The Orchestrator receives the query and either resolves it deterministically or routes it to an LLM planner that utilizes specialized internal tools to fetch and synthesize the required evidence. Details of each component are discussed in the following subsections.

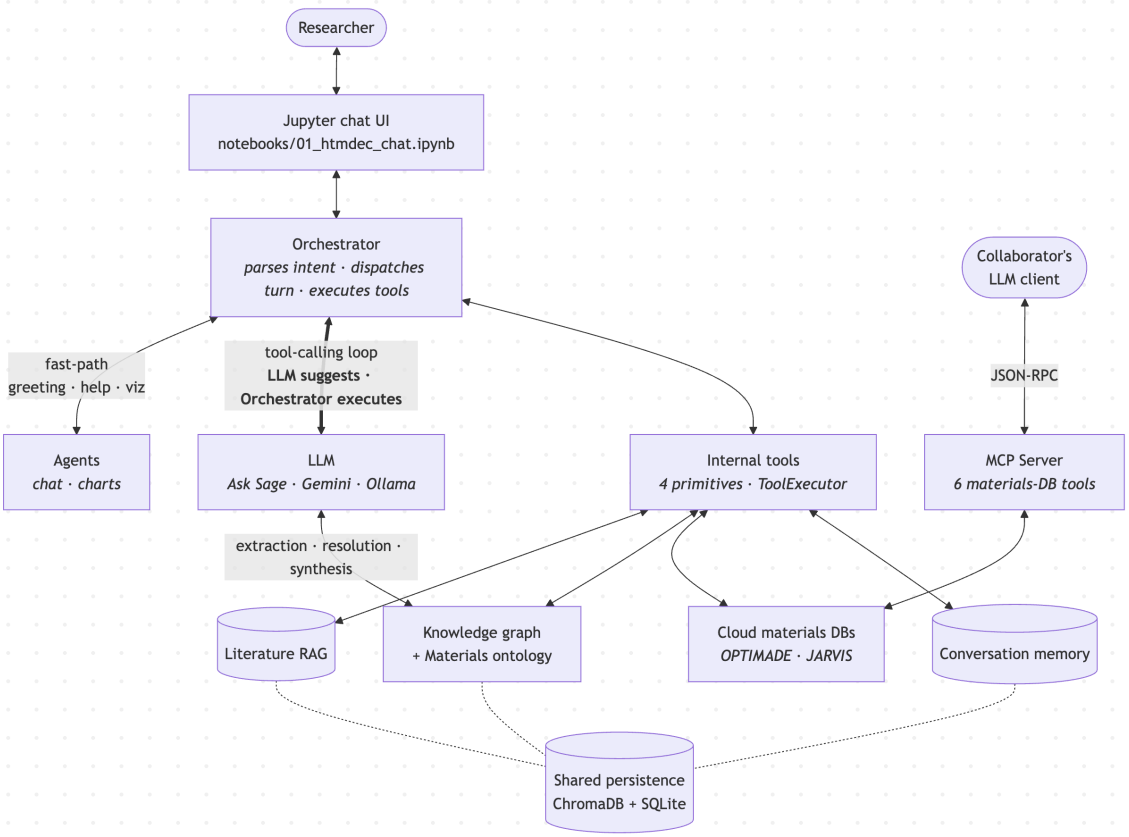


Figure 2. AI/ML-Augmented Toolkit Architecture showing the Orchestrator, LLM tool-calling loop, internal tools, and shared persistence layer.

2.2 Orchestrator and Intent Classification

The Orchestrator serves as the primary traffic controller for the system. When a query is received, the Orchestrator performs an initial evaluation to determine the most efficient execution path:

- **Fast-path routing:** If the user's request is simple and does not require complex reasoning (e.g., standard greetings, requests for basic data visualization, or system help), the Orchestrator routes the request directly to a deterministic agent. This provides an immediate, reliable response without unnecessary computational overhead and resource consumption.
- **LLM Tool-Calling Loop:** For complex materials science questions that bypass the fast-path, the Orchestrator hands the query over to the LLM.

2.3 Central Roles of the Foundation Models

Within the Tool-Calling Loop, foundation models (LLMs) act as the reasoning engine of the architecture. Instead of just answering questions blindly, the LLM takes on several distinct roles:

- **Intent classifier:** The LLM evaluates the complex query to determine the specific materials science objective (e.g., Is the user asking for candidate screening, causal explanation, or data gap analysis?).
- **Tool planner:** Based on the intent, the LLM decides which internal tools (e.g., RAG search vs. KG query) are necessary to answer the question, formatting the request as a structured tool call.
- **Entity resolver:** When querying the KG, the LLM helps resolve fuzzy user terms (e.g., "high-temp alloys") into canonical ontology entities.
- **Subgraph and Response synthesizer:** After the tools return their raw data (such as retrieved text chunks or JSON graph nodes), the LLM synthesizes this structured evidence into a coherent, prose-based explanation complete with citations.
- **Document ingest extractor:** During background ingestion, the LLM processes raw PDFs and extracts structured entities and relationships based on strict ontology prompts.

2.4 Internal Tools and Shared Persistence

When the LLM planner determines that additional information is needed to fulfill the user's request, it relies on four primary internal tool primitives:

1) **Literature RAG:** Executes semantic vector searches against the ingested document corpus to retrieve contextually relevant text passages.
2) **Typed KG + Materials Ontology:** Executes structured relationship queries and ontology-validated extractions to traverse logical links between materials, processes, and properties.
3) **Cloud Materials DBs:** Interfaces with standardized external APIs (e.g., OPTIMADE, JARVIS) to fetch crystallographic data, phase information, and computed properties directly from federated public repositories.[4,5]
4) **Conversation Memory:** Retrieves contextual history from the ongoing session, ensuring multi-turn conversations remain coherent.

The outputs from these tools are supported by a Shared Persistence layer. Semantic text chunks and embeddings are stored in ChromaDB, while conversational chat logs and the structured relational data for the KG are maintained via SQLite.

2.5 Model Context Protocol Server

To facilitate seamless collaboration across external platforms, the architecture includes an optional Model Context Protocol (MCP) server. This allows the toolkit's internal materials database primitives to be exposed via JSON-RPC, enabling external collaborator LLM clients to securely query the system's

structured data. Deploying the MCP server is an advanced integration step that typically requires dedicated IT support, meaning it is an optional capability that does not hinder the stand-alone functionality of the core toolkit.

2.6 Knowledge Beyond RAG: Dual-Pathway Ingestion

The defining feature of this architecture is its ability to push beyond "Vanilla RAG," which retrieves passages but cannot answer questions requiring structured, multi-hop reasoning (e.g., "Which refractory elements stabilize the BCC phase across all our ingested papers?").

To achieve this, the toolkit employs a dual-pathway ingestion system (Figure 3):

1) **Semantic Pathway (ChromaDB):** Documents are parsed into sentence-aware chunks and embedded for standard vector retrieval.
2) **Structured Pathway (KG):** The LLM extracts specific materials entities and their relationships as structured JSON. This JSON is then normalized and validated by the system: aliases are resolved (e.g., mapping HEA to HighEntropyAlloy), canonical IDs are constructed, and types are enforced against the ontology. Invalid relations are dropped.

This approach ensures idempotent inference (making it safe to re-ingest documents without duplication) and allows the system to simultaneously interrogate both semantic text and structured relationships during a single query turn.

2.7 The Knowledge Graph Schema and Discovery Capabilities

The backbone of the structured ingestion pathway is a highly curated materials ontology (Figure 4). The schema draws its lineage from established frameworks like MatOnto (Materials Ontology), EMMO, and PMDco, but is specifically adapted for the HTMDEC focus on high-entropy and refractory alloy research.

It features approximately 160 curated classes designed to cover the entire materials workflow: Composition → Phase → Property → Process → Characterization. This adaptive schema is not standards-strict; it is designed to accept new, validated additions, allowing the toolkit to dynamically map unprecedented discoveries and novel property correlations that do not yet exist in legacy ontologies.

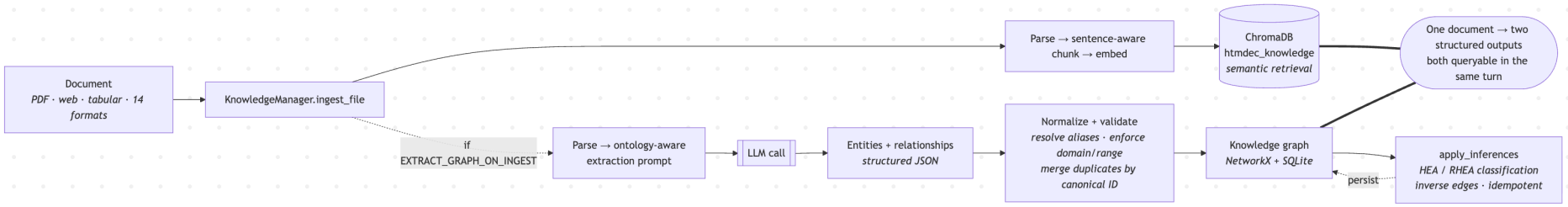


Figure 3. Knowledge beyond RAG: dual-pathway ingestion flowchart.

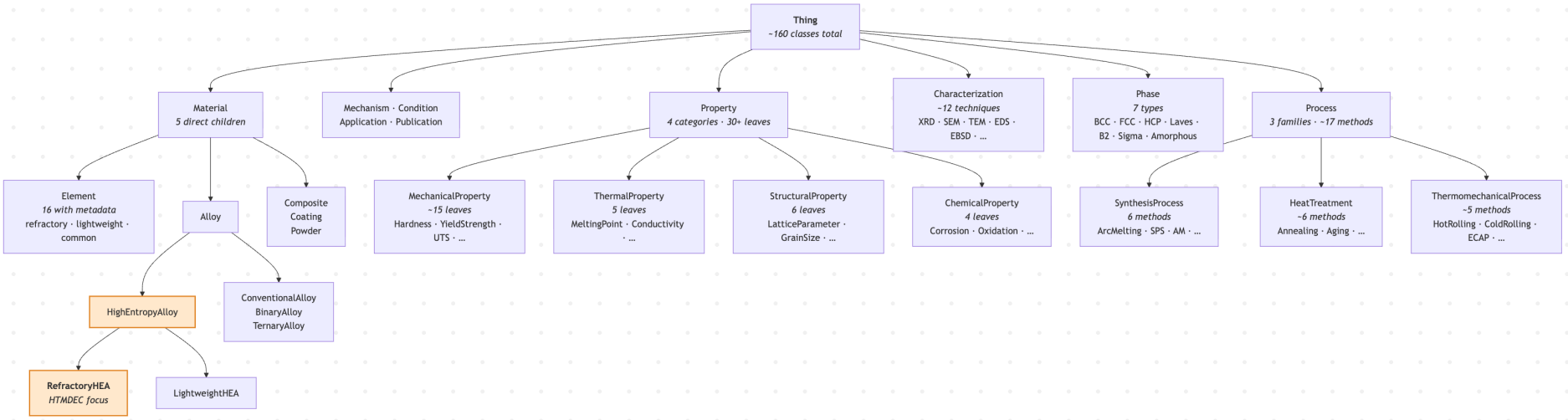


Figure 4. The knowledge graph schema showing curated classes mapped to the materials workflow.

Because the architecture fuses RAG with this ontology-grounded KG, the resulting system operates as a powerful discovery engine. It supports advanced queries natively through Orchestrator-mediated retrieval, graph traversal, and LLM synthesis. The toolkit supports the following researcher workflows that are composed from the implemented retrieval and knowledge-graph primitives rather than exposed as separate hard-coded API methods:

- **Candidate screening:** The engine systematically cross-references the ingested materials inventory against desired structural or mechanical property nodes to rank under-explored alloys, highlighting them for renewed experimental attention (e.g., Based on the ingested literature and KG, which refractory HEAs should be prioritized for further investigation, and why?).
- **Causal explanation:** By traversing interconnected process-to-property edges across multiple documents, the engine provides logical, multi-hop explanations for observed phenomena (e.g., detailing why cold rolling affects hardness in specific compositions).
- **Knowledge audit:** The system isolates and flags contradictory findings across disparate papers by mapping conflicting property values or measurements assigned to the exact same canonical alloy ID within the graph (e.g., show the property observations connected to TiZrNbHf and identify the source documents and passages that should be examined for potentially conflicting values).
- **Gap analysis:** The engine identifies chemical combinations or procedural spaces that exist theoretically within the ontology but lack supporting experimental or literature nodes in the database, directly informing future DOEs (e.g., which properties or processing steps are missing from the evidence chain for NbMoTaW).

The specific implementation details and code structures that bring these architectural layers and discovery capabilities to life are described in the following section.

3. Implementation of the Ontology-Grounded Toolkit for RHEAs

3.1 Core Software Implementation

The `htmdec` package provides an example implementation of the ontology-grounded knowledge-fusion architecture detailed in Section 2. It is realized through a combination of the orchestrator, the LLM gateway, the suite of internal tools, the

knowledge services, and the external materials-search layer. The specific implementation details and code structures for each of these layers are described in detail in the following subsections.

3.2 Orchestration Layer

The primary execution anchor for the user-facing workflow is `Orchestrator.route()`. This method handles the immediate intake and triage of user queries, dictating whether to bypass the LLM entirely or invoke the full tool-calling loop. As depicted in the snippet of Figure 5, a pivotal routing decision occurs after memory and intent logging. If the system flags the request as a `fast_path` scenario, it executes a deterministic agent directly. If complex reasoning is required, it triggers the `_query_with_tools()` method, unleashing the LLM to access the KG, literature RAG, and external databases.

```
    def route(
        self,
        message: str,
        model: str = None,
        sensitive: bool = False,
        **kwargs,
    ) -> dict:
        self.memory.add_user_message(message)
        intent = self._classify_intent(message, model=model)
        entities = self._extract_entities(message)
        system_prompt = self._get_system_prompt(intent)
        ...
        if fast_path or not has_tools:
            output = self.registry.execute(agent_name, agent_input)
        else:
            output = self._query_with_tools(
                message=message,
                system_prompt=system_prompt,
                conversation_history=self._get_recent_history(),
                model=model,
            )
```

Figure 5. Code snippet I.1: `Orchestrator.route()` as the primary researcher-to-system execution path.

3.3 Ingestion and Metadata Registration

The method `KnowledgeManager.ingest_file()` governs the dual-pathway ingestion architecture discussed in Section 2.6. It transforms raw unstructured PDFs and texts into highly indexed knowledge assets.

As shown in Figure 6, this function manages both the semantic chunking (via `self.ingestor`) and the structured graph extraction (via

`self._extract_graph_entities`). By tightly coupling standard database commits (`self._db.commit()`) with AI-driven metadata classification, the toolkit guarantees that every document is fully traceable within the RAG and KG components.

```
    def ingest_file(
        self,
        file_path: str,
        *,
        title: str = "",
        ...
        auto_extract_metadata: Optional[bool] = None,
        extract_graph_entities: Optional[bool] = None,
        gateway: Optional["LLMGateway"] = None,
        **kwargs,
    ) -> dict:
        ...
        result = self.ingestor.ingest_file(file_path, tags=tags,
**kwargs)
        ...
        if auto_extract_metadata and gateway and (need_bib or
need_domain):
            extracted = self._auto_extract_and_classify(file_path,
gateway)
            ...
        self._db.execute(...)
        self._db.commit()
        if extract_graph_entities and self.graph and gateway:
            self._extract_graph_entities(file_path, source, gateway)
            result["graph_entities_extracted"] = True
```

Figure 6. Code snippet I.2: `KnowledgeManager.ingest_file()` as the ingestion, metadata, and graph-linkage entry point.

3.4 Ontology-Guided Text-to-Graph Transformation

The `KnowledgeGraph.extract_from_text()` method embodies the toolkit's most advanced data-fusion capability. As shown in Figure 7, it prompts the LLM to return entities and relationships in a prescribed JSON structure. `MaterialsOntology.get_extraction_prompt()` supplies the allowed entity types, relationship constraints, naming conventions, few-shot examples, and required JSON schema. The returned text is then parsed and validated by the method: aliases are resolved, canonical identifiers are constructed, unknown entity and relationship types are rejected, self-loops are removed, and relationship constraints are checked before accepted records are stored. Even though the generated output remains probabilistic, the strict rules of the materials ontology are enforced during normalization and validation.

```
    def extract_from_text(
        self,
        text: str,
        gateway,
        source: str = "",
        chunk_id: str = "",
    ) -> dict:
        prompt = self.ontology.get_extraction_prompt(text)
        response = gateway.query(
            message=prompt,
            system_prompt="You are a materials science NER system.
Follow the ontology schema exactly.",
            temperature=0.0,
        )
        ...
        resolved_type = self.ontology.resolve_alias(raw_type)
        ...
        canonical_id = f"{canonical_type.lower()}:{canonical_name_slug}"
        self.add_entity(Entity(
            id=canonical_id,
            entity_type=canonical_type,
            name=raw_name,
            properties=e.get("properties", {}) or {},
            sources=[source] if source else [],
            chunk_ids=[chunk_id] if chunk_id else [],
            confidence=0.8,
        ))
```

Figure 7. Code snippet I.3: `KnowledgeGraph.extract_from_text()` as the ontology-guided text-to-graph transformation path.

By setting a low LLM sampling temperature (`temperature=0.0`), the system prioritizes low-variability output during entity and relationship extraction. This parameter does not by itself guarantee deterministic results; structural consistency is provided primarily by JSON parsing, ontology normalization, and post-generation validation. The code systematically intercepts the LLM's raw output, resolves aliases (`resolve_alias`), slugifies the nomenclature, and binds the entity to a `canonical_id`. This prevents the graph from fragmenting when analyzing different spellings of the same alloy across distinct papers.

3.5 Federated External Database Search

When local literature is insufficient, the system can invoke `MaterialsSearchTool.search()` to query data from configured external repositories.

As shown in Figure 8, this asynchronous method standardizes the query parameters (e.g., elements, formulas) so they can be dispatched simultaneously to multiple OPTIMADE-compliant endpoints[4] and the JARVIS platform.[5] The current

implementation queries enabled OPTIMADE endpoints through their public HTTPS interfaces and does not supply provider-specific API keys. Consequently, users do not need separate accounts for the database requests currently implemented, although availability remains subject to each provider's public-access policy, network status, and rate limits. The toolkit provides extensive coverage of integrated OPTIMADE databases—representing millions of materials structures from sources like NOMAD, OQMD, and the Materials Project. These databases are not manually selected by the user; instead, the underlying search tool automatically checks and routes queries to these federated repositories whenever additional information is required. It then merges the disparate JSON responses into a single `UnifiedSearchResult`, feeding formatted external evidence back to the LLM synthesizer.

```
    async def search(
        self,
        elements: Optional[list[str]] = None,
        formula: Optional[str] = None,
        ...
        providers: Optional[list[str]] = None,
        include_jarvis: Optional[bool] = None,
        page_limit: int = 10,
    ) -> UnifiedSearchResult:
        result = UnifiedSearchResult()
        ...
        optimade_result = await self._optimade.search_structures(
            providers=optimade_providers,
            elements=elements,
            formula=formula,
            ...
        )
        ...
        if use_jarvis and self._jarvis:
            jarvis_results = await self._jarvis.search(
                elements=elements,
                formula=formula,
                limit=page_limit,
            )
```

Figure 8. Code snippet I.4: `MaterialsSearchTool.search()` as the federated external database search interface.

3.6 Conversation Memory and Persistence Implementation

The state management of the toolkit ensures that chat histories and relational data are reliably preserved across sessions. The authoritative relational store is handled via `SQLite`.

The in-memory conversation is dynamically managed by the `MemoryManager` module. To persist this data, the system utilizes a dedicated `Database` class that

wraps the standard `sqlite3` library, routing data specifically to `data/htmdec.db`.

Every conversation turn is written to three distinct locations:

1) **SQLite Database (`data/htmdec.db`):** Maintains the `sessions` metadata and the `messages` table, creating a full temporal log containing roles, content, timestamps, intents, models, and execution routes.
2) **JSONL Flat Files:** Provides raw textual backup logs mapped directly to specific `session_id` structures.
3) **ChromaDB Vector Store:** Captures the semantic embedding of the turn, allowing the system to perform vector-based searches over past conversational context during prolonged research sessions.

Concurrently, the KG utilizes `NetworkX` to handle live, in-memory graph topologies, while utilizing `sqlite3` to commit the node and edge structures safely to disk.

4. Toolkit Demonstration

4.1 End-User Interactive Widget Interface

To ensure high usability for researchers, the toolkit provides a streamlined interactive widget interface that abstracts away the complexity of the underlying architecture. As depicted in Figure 9, researchers can execute complex discovery tasks through natural language inputs while retaining fine-grained control over execution parameters.

The interface includes a prominent model-selection drop-down menu, allowing users to dynamically switch between highly capable cloud models (e.g., Claude 4.7 Sonnet, Gemini 3.1 Pro) and secure local models (e.g., Ollama). To ensure data security, a "Sensitive Data (force local)" toggle allows researchers to mandate that proprietary or classified queries are processed entirely on-premise without ever transmitting data to cloud APIs.

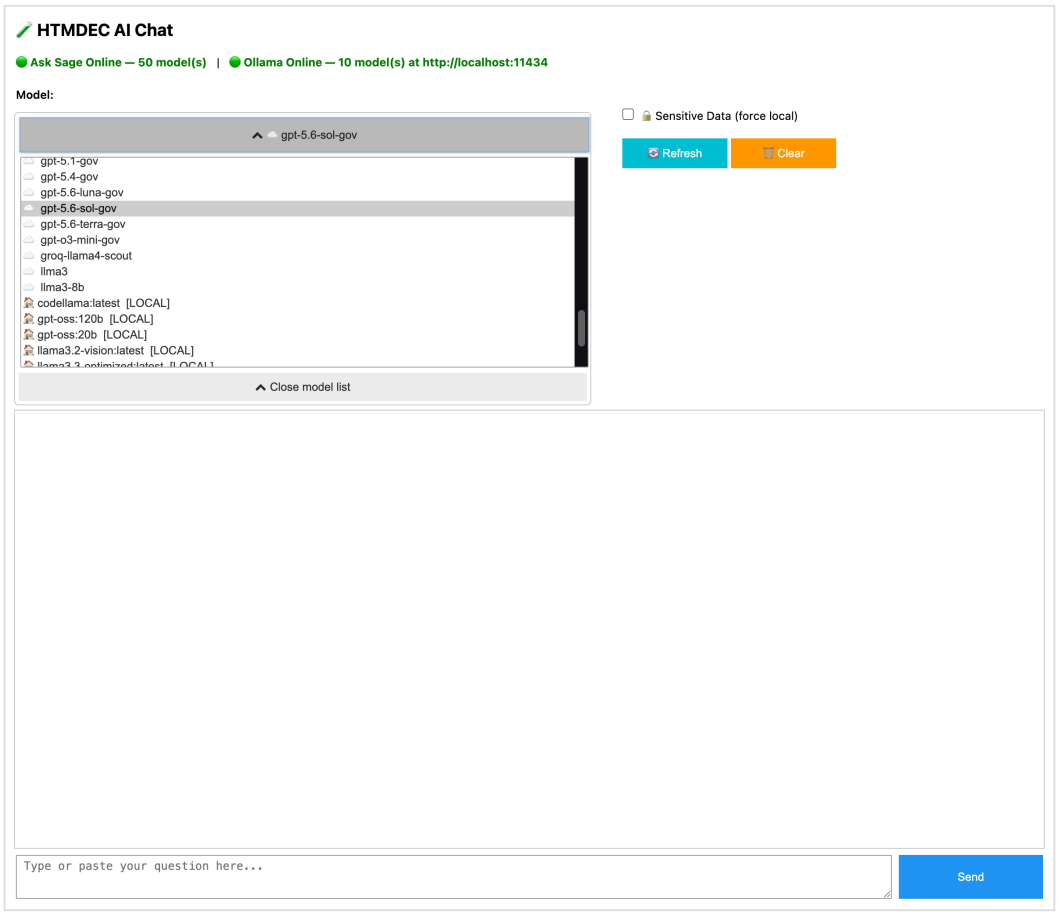


Figure 9. End-user interactive widget interface showcasing model selection.

4.2 Knowledge Graph Visualization

The toolkit can help researchers visualize the extracted structured data to inspect the ingested knowledge as the KG. Figure 10 highlights a subset of the knowledge graph generated during the project, detailing hundreds of distinct entities and over a thousand relationships. The visualization clearly clusters entities according to the ontology schema (Composition, Phase, Property, Process, and Characterization). Furthermore, the rendering is an interactive visualization; the user can interactively activate or deactivate the entities of interest to dynamically explore specific subgraphs and relationships. This macroscopic view is invaluable for researchers, as it visually exposes dense clusters of well-studied material phenomena alongside sparse regions that represent under-explored design spaces ripe for high-throughput experimentation.

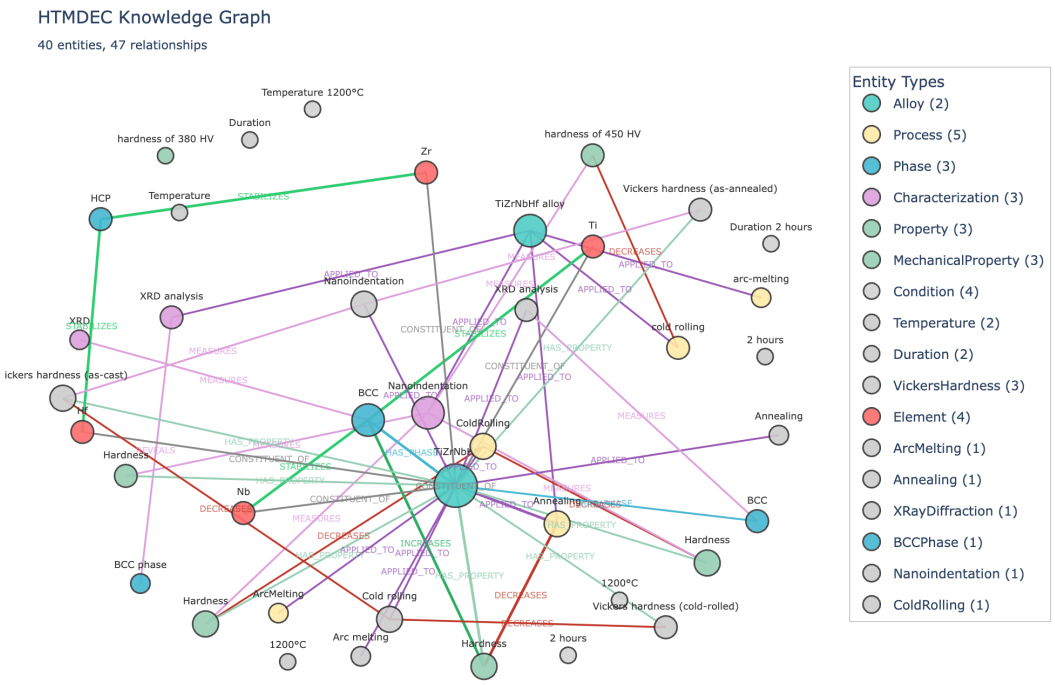


Figure 10. KG visualization showing entity clusters and structural relationships.

4.3 Candidate Screening and Evidence Synthesis

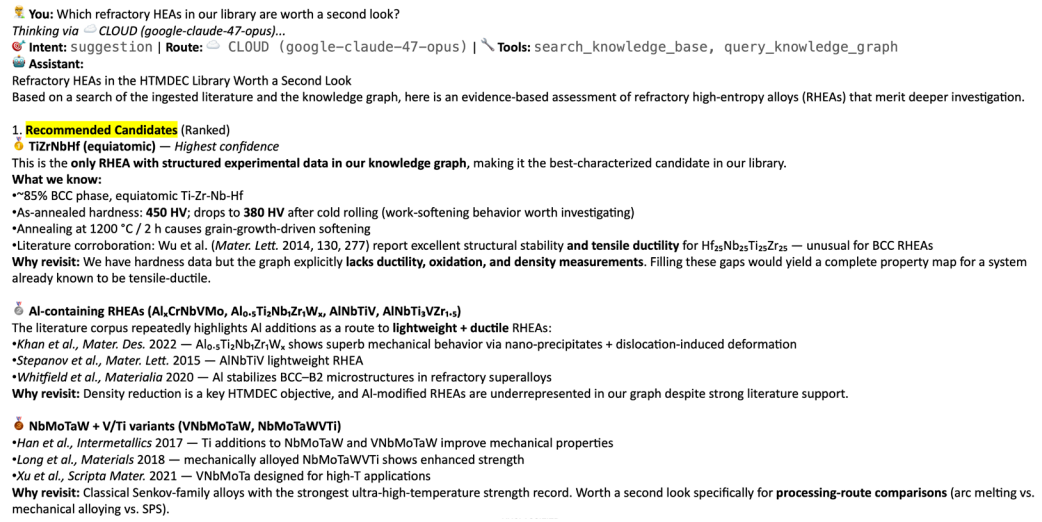
The true utility of the toolkit is demonstrated through a practical RHEA candidate screening query intended to prioritize systems for additional literature review, data extraction, or experimental consideration: "Based on the ingested literature and knowledge graph, which refractory HEAs in our library should be prioritized for further investigation, and what evidence gaps motivate that prioritization?"

Instead of merely retrieving a list of documents, the LLM planner deployed tools to search the 70-document knowledge base and traverse the 541 entities and 1497 relationships in the graph (Figures 11 and 12). The system synthesized a ranked, evidence-backed narrative highlighting top candidates:

1) **TiZrNbHf:** Identified as the best-instrumented system with a complete processing-to-property chain (arc-melting → annealing → cold rolling → X-ray diffraction). The system successfully highlighted a genuine scientific loose end (a counterintuitive decrease in hardness post-cold rolling) and noted untapped potential due to missing interstitial chemistry data.
2) **NbMoTaW Family:** Identified as the high-temperature workhorse. The system noted a major data gap: despite extensive review coverage, the graph lacked structured quantitative property nodes for these alloys, recommending immediate extraction work.

3) **Lightweight Al-bearing RHEAs:** Identified as ideal candidates for a Bayesian optimization campaign due to the well-parameterized effect of aluminum content on BCC-B2 ordering.

The demonstration example concluded the interaction by presenting actionable next steps, such as computing high-entropy alloy descriptors for TiZrNbHf or initiating a design space build for the Al-bearing RHEAs, seamlessly bridging the gap between historical literature and active experimental planning.



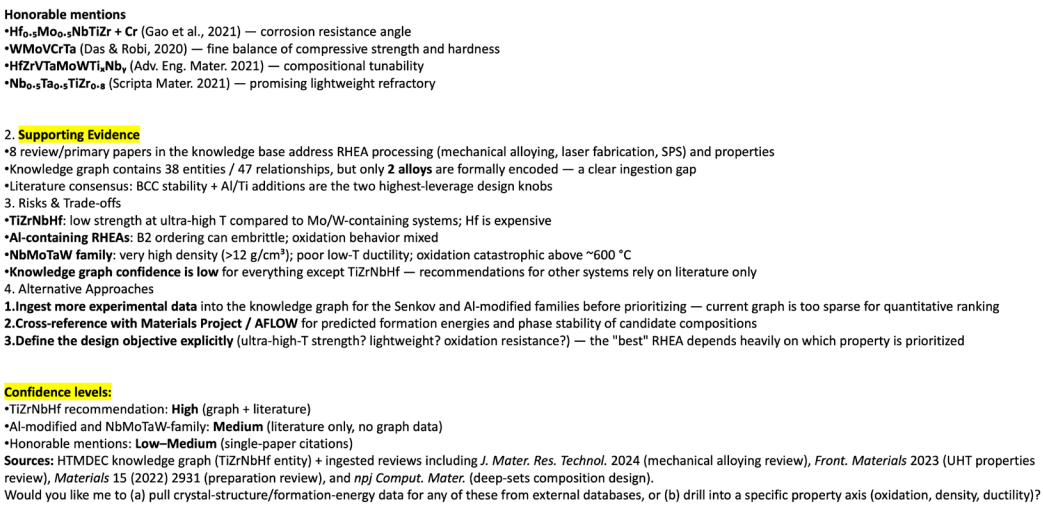
You: Which refractory HEAs in our library are worth a second look?
Thinking via ☁️ CLOUD (google-claude-47-opus)...
🎯 **Intent:** `suggestion` | **Route:** ☁️ `CLOUD (google-claude-47-opus)` | 🔧 **Tools:** `search_knowledge_base, query_knowledge_graph`
🤖 **Assistant:**
Refractory HEAs in the HTMDEC Library Worth a Second Look
Based on a search of the ingested literature and the knowledge graph, here is an evidence-based assessment of refractory high-entropy alloys (RHEAs) that merit deeper investigation.

1. **Recommended Candidates** (Ranked)
🥇 **TiZrNbHf (equiatomic)** — *Highest confidence*
This is the **only RHEA with structured experimental data in our knowledge graph**, making it the best-characterized candidate in our library.
What we know:
•~85% BCC phase, equiatomic Ti-Zr-Nb-Hf
•As-annealed hardness: **450 HV**; drops to **380 HV** after cold rolling (work-softening behavior worth investigating)
•Annealing at 1200 °C / 2 h causes grain-growth-driven softening
•Literature corroboration: Wu et al. (*Mater. Lett.* 2014, 130, 277) report excellent structural stability **and tensile ductility** for $Hf_{25}Nb_{25}Ti_{25}Zr_{25}$ — unusual for BCC RHEAs
Why revisit: We have hardness data but the graph explicitly **lacks ductility, oxidation, and density measurements**. Filling these gaps would yield a complete property map for a system already known to be tensile-ductile.

🥈 **Al-containing RHEAs ($Al_xCrNbVMo$, $Al_{0.5}Ti_2Nb_1Zr_1W_x$, AlNbTiV, $AlNbTi_3VZr_{1.5}$)**
The literature corpus repeatedly highlights Al additions as a route to **lightweight + ductile** RHEAs:
•*Khan et al., Mater. Des.* 2022 — $Al_{0.5}Ti_2Nb_1Zr_1W_x$ shows superb mechanical behavior via nano-precipitates + dislocation-induced deformation
•*Stepanov et al., Mater. Lett.* 2015 — AlNbTiV lightweight RHEA
•*Whitfield et al., Materialia* 2020 — Al stabilizes BCC–B2 microstructures in refractory superalloys
Why revisit: Density reduction is a key HTMDEC objective, and Al-modified RHEAs are underrepresented in our graph despite strong literature support.

🥉 **NbMoTaW + V/Ti variants (VNbMoTaW, NbMoTaWVTi)**
•*Han et al., Intermetallics* 2017 — Ti additions to NbMoTaW and VNbMoTaW improve mechanical properties
•*Long et al., Materials* 2018 — mechanically alloyed NbMoTaWVTi shows enhanced strength
•*Xu et al., Scripta Mater.* 2021 — VNbMoTa designed for high-T applications
Why revisit: Classical Senkov-family alloys with the strongest ultra-high-temperature strength record. Worth a second look specifically for **processing-route comparisons** (arc melting vs. mechanical alloying vs. SPS).

Figure 11. Sample query example (part 1) demonstrating candidate screening and LLM tool use.



Honorable mentions
•**$Hf_{0.5}Mo_{0.5}NbTiZr$ + Cr** (Gao et al., 2021) — corrosion resistance angle
•**WMoVCrTa** (Das & Robi, 2020) — fine balance of compressive strength and hardness
•**$HfZrVTaMoWTi_xNb_y$** (Adv. Eng. Mater. 2021) — compositional tunability
•**$Nb_{0.5}Ta_{0.5}TiZr_{0.8}$** (Scripta Mater. 2021) — promising lightweight refractory

2. **Supporting Evidence**
•8 review/primary papers in the knowledge base address RHEA processing (mechanical alloying, laser fabrication, SPS) and properties
•Knowledge graph contains 38 entities / 47 relationships, but only **2 alloys** are formally encoded — a clear ingestion gap
•Literature consensus: BCC stability + Al/Ti additions are the two highest-leverage design knobs
3. Risks & Trade-offs
•**TiZrNbHf**: low strength at ultra-high T compared to Mo/W-containing systems; Hf is expensive
•**Al-containing RHEAs**: B2 ordering can embrittle; oxidation behavior mixed
•**NbMoTaW family**: very high density (>12 g/cm³); poor low-T ductility; oxidation catastrophic above ~600 °C
•**Knowledge graph confidence is low** for everything except TiZrNbHf — recommendations for other systems rely on literature only
4. Alternative Approaches
1.Ingest more experimental data into the knowledge graph for the Senkov and Al-modified families before prioritizing — current graph is too sparse for quantitative ranking
2.Cross-reference with Materials Project / AFLOW for predicted formation energies and phase stability of candidate compositions
3.Define the design objective explicitly (ultra-high-T strength? lightweight? oxidation resistance?) — the "best" RHEA depends heavily on which property is prioritized

Confidence levels:
•TiZrNbHf recommendation: **High** (graph + literature)
•Al-modified and NbMoTaW-family: **Medium** (literature only, no graph data)
•Honorable mentions: **Low–Medium** (single-paper citations)
Sources: HTMDEC knowledge graph (TiZrNbHf entity) + ingested reviews including *J. Mater. Res. Technol.* 2024 (mechanical alloying review), *Front. Materials* 2023 (UHT properties review), *Materials* 15 (2022) 2931 (preparation review), and *npj Comput. Mater.* (deep-sets composition design).
Would you like me to (a) pull crystal-structure/formation-energy data for any of these from external databases, or (b) drill into a specific property axis (oxidation, density, ductility)?

Figure 12. Sample query example (part 2) demonstrating evidence synthesis, risks, and recommended actions.

5. Conclusion and Future Directions

As demonstrated with the AI/ML-Augmented Toolkit, the ontology-grounded framework that integrates semantic retrieval, typed knowledge representation, graph traversal, and tool-mediated LLM reasoning represents a significant advancement in cross-platform data fusion for MI. By bridging isolated evidence sources—proprietary documents, structured KGs, and federated external databases—into a unified, orchestrated interface, the system reduces data fragmentation and LLM hallucination in high-throughput materials discovery. Conceptually, the core architectural pillars of this framework—the Orchestrator, the LLM tool-calling loop, the dual-pathway ingestion, and memory management—are fundamentally domain-agnostic. This approach extends well beyond materials science, serving as a generalized blueprint applicable to other data-intensive scientific disciplines such as chemistry, biology, and pharmacology.

The architecture is modular, scalable, and inherently extensible to other materials classes by design. Within materials science, the toolkit can seamlessly adapt to new classes of materials beyond RHEAs. To extend the system to novel domains—such as ultra-high-temperature ceramics, lightweight alloys, or energy-storage materials—researchers only need to adapt the highly modular components. The materials ontology can be updated with new classes and property definitions, allowing the LLM-guided extraction prompts to begin populating the knowledge graph with new, domain-specific typed entities immediately. Similarly, because the external database layer utilizes standardized OPTIMADE protocols, adding new federated public repositories requires minimal adjustments to the core codebase. This structural modularity guarantees that the toolkit can evolve rapidly alongside emerging Army research priorities without requiring fundamental software re-architecture.

Moving forward, the platform is positioned to integrate more tightly with automated physical and computational workflows. Future development tasks include the seamless integration of BBO modules directly into the agentic workflow to drive active learning campaigns. Additionally, the system will be expanded to interface with internal ARL DMS for bidirectional data flow and will be actively demonstrated on non-RHEA material classes of interest. Ultimately, this scalable architecture provides a trustworthy, reproducible intelligence layer capable of accelerating hypothesis generation and experimental design for critical Army materials-discovery missions.

6. References

1. Mallick D et al. Year 2 activities of the High-Throughput Materials Discovery for Extreme Conditions (HTMDEC) Collaborative Research Alliance. DEVCOM Army Research Laboratory (US); 2025. Report No.: ARL-SR-0501.
2. Edge D et al. From local to global: a graph rag approach to query-focused summarization. arXiv; 2024. arXiv:2404.16130.
3. Zheng Y et al. Knowledge graphs in AI for science: a survey. arXiv. 2024; arXiv:2407.13524.
4. Andersen CW et al. OPTIMADE, an API for exchanging materials data. Sci Data. 2021;8(217).
5. Choudhary K et al. The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design. npj Comput Mater. 2020;6(173).

List of Symbols, Abbreviations, and Acronyms

AFLOW	Automatic FLOW for Materials Discovery
AI	Artificial Intelligence
API	Application Program Interface
ARL	Army Research Laboratory
BBO	Batch Bayesian Optimization
CALPHAD	CALculation of PHAse Diagrams
DEVCOM	U.S. Army Combat Capabilities Development Command
DMS	Data Management System
DOE	Design of Experiments
DoW	Department of the War
HEA	High-Entropy Alloy
HTMDEC	High-Throughput Materials Discovery for Extreme Conditions
JARVIS	Joint Automated Repository for Various Integrated Simulations
KG	Knowledge Graph
LLM	Large-Language Model
MCP	Model Context Protocol
MI	Materials Intelligence
ML	Machine Learning
NOMAD	Novel Materials Discovery
OPTIMADE	Open Databases Integration for Materials Design
OQMD	Open Quantum Materials Database
RAG	Retrieval-Augmented Generation
RHEA	Refractory High-Entropy Alloy
UI	User Interface

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLB CB TECH LIB