Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.084061

ARTICLE

STALAgent: A Multi-Agent System Based on Large Language Model (LLM) for Steel and Alloy Design

Jiayi Qiu[1], **Youle Wang**[1,*] **and Lei Zhang**[1,2,*]

[1]Department of Internet Engineering, School of Software, Nanjing University of Information Science & Technology, Nanjing, China
[2]Key Laboratory of Spectral Imaging Technology, Xi'an Institute of Optics and Precision Mechanics of the Chinese Academy of Sciences (CAS), Xi'an, China

*Corresponding Authors: Youle Wang. Email: youlew@foxmail.com; Lei Zhang. Email: 002699@nuist.edu.cn

Received: 15 April 2026; Accepted: 16 July 2026; Published: 13 August 2026

ABSTRACT: The design of steel and alloy materials is of critical importance across a wide range of industrial applications; however, effective intelligent agent-based assistants for this domain remain limited. To address this gap, we introduce STALAgent, a large language model (LLM)-based multi-agent system specifically tailored for intelligent and automated design of steel and alloy materials. STALAgent is centered on an LLM brain with several key agents (e.g., task assignment, semantic search, inverse design, and heat treatment simulation) that collectively form a closed-loop workflow from user query to material recommendation. This system leverages a CrewAI-based orchestrator to assign tasks and coordinate a suite of specialized agents, including tools for knowledge retrieval using a retrieval augmented generation (RAG), inverse materials design using variational encoder (VAE), and thermodynamic calculations using Pycalphad. Through case studies involving inverse alloy design tasks and knowledge-based steel design queries, we showcase the capacity of the LLM agent to offer effective and dependable guidance for steel and alloy material design. STALAgent is practical and scalable, serving as a supplementary tool for materials researchers and holding promise for extension to other materials science domains requiring scientific discovery and domain knowledge-intensive tasks.

KEYWORDS: Agent; LLM; large language model; materials informatics; materials genome

1 Introduction

In recent years, AI for Science has rapidly reshaped scientific research paradigms across physics, chemistry, and materials science, enabling data-intensive and knowledge-centered discovery at unprecedented speed [1–3]. Within this transformation, the frameworks of Materials Genome Engineering (MGE) [4] and Materials Informatics [5,6] have provided systematic pathways for integrating high-throughput computation, experimental data, and machine learning models to accelerate the design of new materials and exploration of their composition-processing-structure-property relationships [7–9]. These developments have established the foundation for intelligent, automated, and hypothesis-driven materials research [10].

The material science suffers from data scarcity and data sparsity [11]. In parallel with algorithmic progress, the expansion of materials databases has provided the essential data infrastructure for materials science [12]. Large-scale first-principles repositories such as the Materials Project (MP), the Automatic FLOW for Materials Discovery (AFLOW) database, the Open Quantum Materials Database (OQMD), and the Joint Automated Repository for Various Integrated Simulations (JARVIS) host millions of density-functional theory (DFT) calculations, including crystal structures, formation energies, phase stability, electronic and mechanical properties, and defect energetics [13,14]. Additional computational platforms, including 2DMatPedia, Atomly, and other domain-specific curated datasets such as matbench, enrich the landscape by providing information in various domains and multimodal metadata [15,16]. Complementing these computational resources, extensive experimental materials databases enhance data availability and reliability. Notable examples include the Cambridge Structural Database (CSD) for experimentally solved organic and metal-organic crystal structures, the Inorganic Crystal Structure Database (ICSD) for inorganic compounds, the Crystallography Open Database (COD), and a variety of specialized datasets capturing phase diagrams, thermophysical properties, alloy compositions, and heat-treatment records. Together, these computational and experimental repositories form the data backbone for modern materials research, enabling model training, benchmarking, generative design, and the discovery of new materials across energy, chemistry, metallurgy, and condensed-matter physics.

Concurrently, advances in machine learning [17] and deep learning have produced powerful predictive models capable of capturing complex nonlinear correlations in materials data [5,12]. These models range from classical kernel methods to graph neural networks and transformer-based architectures, demonstrating strong capability in learning structure-property relationships from limited, noisy, or imbalanced datasets. In particular, the emergence of large language models (LLMs) and general-purpose foundation models has further expanded this capability, enabling scientific reasoning, natural-language interaction, and multimodal knowledge integration across diverse materials domains [18,19]. These models provide an unprecedented opportunity to unify text-based knowledge, numerical data, simulation results, and domain expertise within a single computational framework. Recent advances in agentic AI systems and multimodal artificial intelligence have further demonstrated the potential of LLM-driven workflows to transform domain-specific scientific practices through structured agent coordination and in context tool integration. The power of LLMs lies in their emergent capabilities and autoregressive reasoning, which enable them to synthesize information, identify patterns, and generate coherent scientific hypotheses. In materials science, the rich textual knowledge embedded in research articles, patents, technical reports, and domain-specific handbooks provides a natural and abundant source for model construction, semantic understanding, and the formulation of data-driven scientific proposals. Building on these foundations, several LLM-based materials science agents have recently emerged, including agentic systems for materials synthesis planning, literature-guided reasoning, and database-driven property prediction [20–22]. In parallel, inverse materials design has become a central direction for generative AI in materials science, [23,24] employing VAE, GAN, and diffusion models to propose candidate structures or compositions directly from target properties. These generative-reasoning capabilities form the technological basis for autonomous and closed-loop exploration of new materials.

In steel and alloy materials science and engineering, alloy composition design, literature knowledge retrieval, and heat-treatment process simulation are critical steps for improving performance and accelerating research-engineering iteration [25]. With the rapid increase in experimental and computational studies, vast quantities of documents, process standards, and structured or unstructured data have accumulated. However, this knowledge is highly fragmented and distributed across technical reports, patents, handbooks, industrial standards, and academic publications. Moreover, prior metallurgical knowledge, especially

regarding synthesis feasibility and heat-treatment rationality, is essential to ensure that AI-generated design schemes are both scientifically sound and practically actionable. A limitation is the lack of a unified platform in the steel and alloy domain that integrates knowledge retrieval, composition design, generative recommendation, heat-treatment simulation, and interactive human-AI collaboration.

To address these limitations, we introduce STALAgent (Steel and Alloy Agent), an LLM-based multi-agent system that unifies literature retrieval, inverse design, composition recommendation, and heat-treatment simulation within a single interactive platform. STALAgent incorporates an intelligent task assignment agent, a hybrid semantic retrieval engine built on a curated metallurgical literature database, auxiliary generative modules for composition proposal based on inverse design, and a heat-treatment simulation component that links design recommendations to physical validation. A conversational interface with transparent agent-level decision tracing further enhances usability and interpretability. Through two representative case studies covering inverse design and retrieval-augmented generation, we demonstrate the ability of the system to inversely design, organize knowledge, generate preliminary hypotheses, and support data-driven material design workflows. Beyond qualitative case studies, we provide systematic quantitative evaluation on two benchmarks comprising 110 tasks and a retrieval evaluation set of 57 queries, with comprehensive baseline comparisons against four general-purpose LLMs evaluated under identical conditions. Rather than replacing laboratory experiments or materials simulations, STALAgent aims to provide an LLM-driven assistant for material science researchers, enabling efficient knowledge access, exploratory design, and analysis of heat-treatment behaviors of new materials within an integrated AI framework.

2 Methods

STALAgent adopts a cooperative multi-agent architecture with a large language model (LLM) at its core. The architecture follows a closed-loop process of user query → task analysis → agent team formation → team execution → result evaluation → output or iteration, ensuring system robustness while maintaining strong flexibility and scalability. Unlike traditional rule-based multi-module systems, the LLM is deeply embedded in the scheduling and decision-making links of the STALAgent system through the CrewAI framework. In this way, the LLM not only acts as a text generator but also assumes the central roles of task assignment, tool invocation, and agent cooperation via CrewAI. This design effectively improves the adaptability of the system in open task scenarios (Fig. 1a). Currently, the system integrates four core agents: (1) Task Assignment Agent, which is based on CrewAI and interprets user goals, generate subtasks, and assigns them to appropriate expert agents. (2) Semantic Search Agent, which is based on retrieval augmented generation (RAG) and performs keyword search and semantic vector fusion retrieval based on the literature database, and provides candidate results (Fig. 1b). (3) Inverse Design Agent, which is based on conditional variational autoencoder (cVAE) and generates steel material compositions and processes through inverse design using a generation model (Fig. 1c). (4) Heat Treatment Agent, which is based on Pycalphad and simulates the heat treatment process and returns phase information according to steel and alloy composition and conditions (Fig. 1d). These agents form a cooperative team through LLM unified scheduling, support multi-stage reasoning and interpretation of results, and finally provide users with reliable answers.

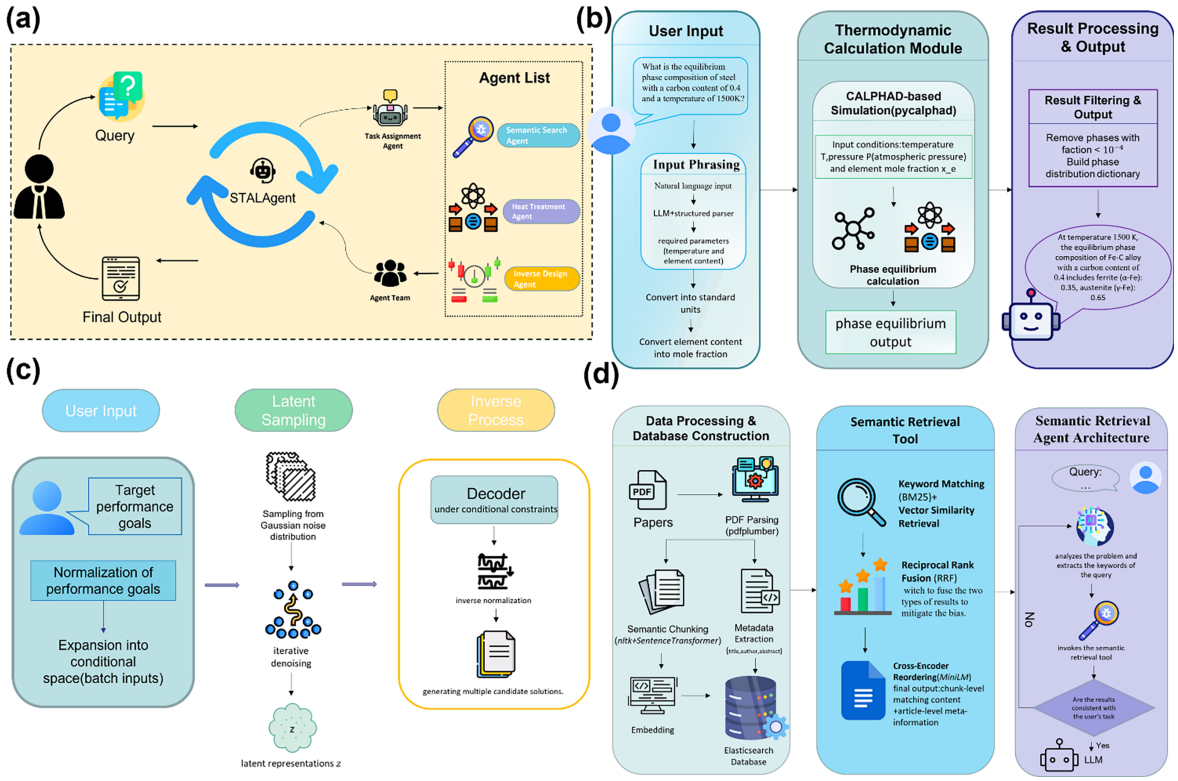

Figure 1: (**a**) Overview of the overall workflow. (**b**) Retrieval-augmented generation (RAG) process built upon ElasticSearch and a proprietary materials database. (**c**) Inverse design framework for alloy composition (e.g., steel and multicomponent alloys) using a combined conditional variational autoencoder (cVAE) and diffusion model (cVAE + Diffusion). (**d**) Heat-treatment simulation and phase-information extraction based on Pycalphad.

In detail, the operational workflow of STALAgent follows a seven-stage sequential and iterative execution pipeline (Fig. S1). The detailed implementation procedures are summarized as follows: (1) The workflow initiates with user interaction, where diverse material research requirements are submitted in natural language via the web-based interface. These user demands span customized steel performance design with targeted tensile strength and elongation metrics, alloy composition optimization, and domain-specific knowledge queries regarding steel and alloy systems, and all user inputs are captured by the Task Assignment Agent serving as the exclusive entry of the entire operational pipeline. (2) Leveraging the LLM and CrewAI framework, the Task Assignment Agent conducts fine-grained semantic parsing and in-depth analysis on raw user inputs, which distills core task objectives, explicit operational constraints and implicit functional requirements, and categorizes incoming tasks into inverse material design, literature retrieval and knowledge mining, heat treatment simulation analysis, and hybrid composite tasks, thereby adaptively configuring a collaborative team of four specialized functional agents to match current task demands. (3) The integrated scheduler of CrewAI decomposes the holistic task goal into decoupled atomic subtasks, quantifies the logical dependencies between subtask units, and distributes each subtask to the corresponding specialized agent; mutually independent subtasks including semantic literature retrieval and heat treatment numerical simulation are executed in parallel to enhance computational efficiency, while standardized output specifications and quantitative quality criteria are established to unify the deliverable format and evaluation benchmark of all distributed subtasks. (4) Then all configured functional agents execute assigned subtasks synchronously, where the Semantic Search Agent implements dual-path retrieval and hierarchical reranking

on a professionally curated steel literature database to acquire authoritative and targeted literature evidence, the Inverse Design Agent explores the latent space of the conditional VAE and conditional diffusion model under predefined performance constraints to generate viable steel alloy composition candidates, and the Heat Treatment Agent adopts the PyCALPHAD toolkit to perform thermodynamic equilibrium calculations over a temperature range of 673–2000 K, yielding continuous temperature-dependent phase fraction evolution data. (5) Upon the completion of parallel subtask execution, the central scheduler aggregates multiagent outputs and implements a comprehensive three dimensional evaluation system covering result completeness, cross module consistency and overall output quality, which validates the integrity and effectiveness of individual subtask deliverables, verifies the thermodynamic feasibility of generated alloy compositions and the literature basis of designed processing strategies, and examines whether all outputs satisfy domain-specific precision thresholds and professional constraint conditions, ultimately determining whether to finalize result output or trigger a closed loop iterative refinement procedure. (6) An iterative optimization loop is activated once inadequate, inconsistent or substandard results are identified in the evaluation phase, where the system adaptively revises task implementation parameters by refining literature search boundaries, tightening performance constraint conditions and supplementing discrete temperature calculation points before redistributing and re-executing targeted subtasks; supported by endogenous "reflection-critique-refinement" autoregressive mechanism of CrewAI, this cyclic optimization eliminates output defects and guarantees the reliability and scientific rigor of final results via dual validation of literature knowledge and computational thermodynamics. (7) After all subtask deliverables pass multi-dimensional consistency and quality validation, the scheduler integrates, polishes and systematizes valid outputs to form a holistic and interpretable response, which covers mass fraction-based alloy composition parameters, elaborate thermomechanical processing routes including thermo-mechanical controlled processing (TMCP) and quenching-tempering schemes, temperature-resolved phase analysis data, technical validation literature sources, and explanatory remarks on potential uncertainties and limitations, and all results are delivered through the web interface with complete intermediate decision-making and execution records retained for full workflow transparency and traceability.

2.1 Task Assignment Agent

The task assignment agent is mainly powered by an LLM and CrewAI. After the user inputs the query related to materials science, the LLM brain powered by DeepSeek-V3.2 will get the initial user goal and the information of each agent. The LLM then splits the user goal into subtasks and selects appropriate expert agents to complete these subtasks. Subsequently, the agent will sequentially assign these subtasks to the corresponding agents, evaluate the results returned by the agents, and determine whether rework is necessary or the assignment strategy needs to be adjusted, just like a project manager. Behind this coordination process, CrewAI provides a principled multi-agent orchestration paradigm grounded in LLM-based cognitive planning and iterative cooperative reasoning. At its core, CrewAI uses the LLM as a global planner to infer task dependencies, execution order, and information flows among agents, effectively constructing a dynamic task graph from natural language goals. Each agent is modeled as a functional unit with a defined role, capability description, and tool-handling policy to material science. During execution, CrewAI leverages an autoregressive "reflection-critique-refinement" loop, allowing agents to evaluate intermediate outputs, resolve conflicts, and refine each other's results through structured message passing. This theoretical foundation converts loosely defined user instructions into a coordinated sequence of agent actions, enabling emergent collaboration and reliable completion of complex scientific workflows.

The task allocation and coordination in STALAgent is built upon the following CrewAI primitives and design patterns (Fig. S2): (i) In CrewAI, each agent is defined by four structured attributes: a role, a goal, a

backstory, and a tools list. The backstory serves a critical function beyond narration: it functions as a soft constraint that guides the LLM interpretation of which tasks fall within the agent's competence. All specialist agents have delegation disabled so that they focus exclusively on substantive execution rather than meta-coordination, and each is capped at a maximum of three LLM reasoning iterations to prevent unbounded self-reflection loops. (ii) The coordination brain of STALAgent is a designated manager agent that holds the sole delegation authority. CrewAI provides two atomic delegation tools through which the manager coordinates specialists: DelegateWorkTool, which allows a manager to specify a task description, context, and coworker, and enables the designated coworker to independently execute the delegated subtask using its own LLM reasoning loop and domain tools before returning the result, and AskQuestionTool, which supports lighter-weight interaction for fact checking or quick consultation without initiating a full subtask execution. These tools are dynamically injected into the manager tool set at runtime, ensuring that delegation is traceable, the recipient is explicitly identified, and the subtask specification is well-formed. (iii) The manager agent backstory encodes an explicit set of task routing rules that map user intent categories to agent teams. These rules are not hard-coded in control flow but executed by the LLM as it parses the user goal and selects which coworkers to invoke, which combines the flexibility of natural-language intent interpretation with the reliability of explicit, auditable routing logic. (iv) STALAgent adopts the sequential execution model of CrewAI in which a single meta-task would be assigned to the manager, who then orchestrates the entire workflow through successive delegation calls. This single point design ensures that the manager maintains full visibility over the execution state. After each round of specialist execution, the manager evaluates the collected results against the original user goal. If gaps, inconsistencies, or quality deficiencies are detected, the manager will reinvoke the relevant specialists with refined subtask specifications. (v) STALAgent captures a structured execution trace for every interaction: each delegation event is logged with the delegating agent role, the subtask description, the assigned coworker, the returned result, and a timestamp.

2.2 Semantic Search Agent

This agent is a critical part of this multi-agent system, as it controls the steel knowledge database and acts as the primary knowledge source of the entire system. It effectively implements a retrieval-augmented generation (RAG) function. The agent comprises three main parts: Data Processing and Database Construction, Semantic Retrieval Tool, and Semantic Retrieval Agent Architecture. (1) Data Processing and Database Construction: We collect over 10,000 full-text published articles in the steel field. Documents are chunked, enriched with metadata, vectorized, and stored in an Elasticsearch database. At the document parsing stage, text is extracted page by page from PDF files using *pdfplumber*. The extracted text is then divided into semantically complete chunks of moderate length and by sentence level segmentation (*nltk*) and semantic similarity aggregation (*SentenceTransformer*). First, sentences are segmented with *nltk* to obtain the initial sentence sequence, and the semantic similarity of adjacent sentences is calculated using a pre-trained *SentenceTransformer*. When the similarity fell below a threshold, or the cumulative token count exceeds an upper limit, chunking was triggered to ensure that each chunk is concise yet semantically coherent. For metadata extraction, we use a lightweight, heuristic, rule-based approach to extract titles, authors and abstracts from the first pages of documents and store them separately as treatise-level metadata. At the same time, each text block is transformed into a dense vector representation by the same embedding model and stored in Elasticsearch database together with the corresponding document ID. The database design consists of two indexes: (i) *papers* index: stores meta-information at the paper level (title, author, abstract, etc.); (ii) *paper_chunks* index: stores text blocks, location indexes, and their embedding vectors (using 768-dimensional semantic vectors, cosine similarity as the retrieval metric). (2) Semantic Retrieval Tool: Based on the database, we construct a multi-stage semantic retrieval tool which combines keyword retrieval, vector

retrieval and reordering. The retrieval process is implemented as a sequential four stage pipeline (Fig. S3) designed to balance recall and precision: (i) Stage 1 is Best Matching 25 (BM25) Keyword Retrieval stage. During this stage, a query is issued against the text field of the chunk index using Elasticsearch's built in BM25 scoring, capturing exact terminology matches that dense embeddings may miss. (ii) Then the user query is encoded into a 768-dimensional vector using intfloat/e5-base-v2, and a kNN search (k = 100) is performed against the embedding field, capturing semantically related content that may use different terminology than the query. (iii) Results from Stages 1 and 2 are merged using Reciprocal Rank Fusion (RRF). To reflect the higher reliability of dense retrieval for scientific text, vector results are assigned a 2× weight relative to keyword results. Documents are deduplicated by paper id, retaining only the highest-scoring chunk per article to ensure diversity in the candidate set. (iv) The fused candidate set is re-ranked using a pre-trained Cross-Encoder model. Unlike the encoder used in Stage 2, the Cross-Encoder processes each pair jointly through full cross-attention, producing a fine-grained relevance score. Candidates are sorted by this score, and the top-5 results are returned to the agent. (3) Semantic Retrieval Agent Architecture. The semantic retrieval agent realizes the closed-loop feedback operation architecture through the mode of perception → analysis → retrieval → evaluation → reanalysis. When the agent receives a user task, it first analyzes the problem and extracts the keywords of the query. It then invokes the semantic retrieval tool to search for relevant literature segments. The system evaluates the retrieved content, and if the results are inconsistent with the user task, the agent refines the query and repeats the process. Through this iteration, suitable literature can be identified more accurately.

We constructed a retrieval evaluation set comprising 57 queries across 17 steel metallurgy categories, including martensite transformation, bainite kinetics, High-Strength Low-Alloy (HSLA) precipitation, Transformation-Induced Plasticity/Twinning-Induced Plasticity steel (TRIP/TWIP) steel, stainless steel corrosion, hydrogen embrittlement, and CALPHAD phase diagrams. Ground-truth relevant papers were established through Elasticsearch-based candidate generation with metallurgical keyword filtering, producing 3–4 graded relevance judgments per query. Standard information retrieval metrics, Precision@K, Recall@K, Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain at K (NDCG@K), were computed across the four-stage pipeline. The pipeline achieves a Precision@1 of 0.786, indicating that in nearly 79% of queries the top-ranked result is directly relevant. The Mean Reciprocal Rank of 0.819 demonstrates that the first relevant paper consistently appears near the top of the ranked list. The Recall@5 of 0.505 and Mean Average Precision of 0.405 reflect robust overall retrieval quality across diverse query types, with an average retrieval time of 0.55 s per query. Strongest per-category performance is observed for thermomechanical processing (MAP = 0.658), grain refinement (MAP = 0.583), HSLA precipitation (MAP = 0.579), and martensite transformation (MAP = 0.510, MRR = 1.000), where the specialized steel-domain vocabulary and well-defined metallurgical concepts facilitate precise matching. These results validate that the four-stage retrieval pipeline, built on a dedicated >10,000-article steel literature database, effectively serves as the knowledge foundation for downstream agent tasks. The detailed information is provided in Table S1.

2.3 Inverse Design Agent

To realize the reverse design of steel alloys under the target performance conditions, a generative agent combining CVAE and a conditional diffusion model is employed in this study. The agent is able to generate alloy compositions and process solutions that satisfy conditional constraints when the user provides target performance parameters.

The inverse design model was trained on a dataset of 842 steel alloy compositions sourced from the Citrination materials data platform (dataset ID: 153092), a community repository affiliated with the Materials

Genome Initiative. Each record contains the weight percentages (wt%) of 13 alloying elements (C, Mn, Si, Cr, Ni, Mo, V, N, Nb, Co, W, Al, and Ti) together with four target mechanical properties measured under specified thermal conditions: yield strength (YS, MPa), ultimate tensile strength (UTS, MPa), elongation (El, %), and testing temperature (T, °C). The dataset was split into 80% training (n = 673) and 20% test (n = 169) using stratified random sampling (random state = 42). All composition values were normalized to the range [0, 1] using a fitted on the training set, and the target properties were similarly standardized with a separate scaler fitted on the training set. The test set was reserved for evaluation purposes only and was not included in the training process.

The inverse design pipeline consists of a conditional variational autoencoder (cVAE) coupled with a conditional diffusion model operating in the latent space (Fig. S4). The cVAE encoder accepts a concatenated vector of composition (13-dim) and property conditions (4-dim), passes it through a two-layer MLP with 128 hidden units and ReLU activations, and outputs two 32 dimensional vectors representing the mean μ and log-variance $\log\sigma^2$ of the approximate posterior. A latent variable $z \in \mathbb{R}^{32}$ is sampled via the reparameterization trick: $z = \mu + \sigma \odot \varepsilon$, where $\varepsilon \sim N(0, I)$. The cVAE decoder takes the concatenated latent vector z (32-dim) and condition vector (4-dim), feeds it through a single hidden layer of 128 units with ReLU activation, and outputs a reconstructed 13 dimensional composition vector. The conditional diffusion model is a 4-layer MLP (hidden dimension 128, GELU activation, LayerNorm, Dropout = 0.1) that predicts the noise ε_t added to the latent variable z at diffusion timestep t, conditioned on the property vector. The sinusoidal time embedding t/1000 is concatenated with z_t and the condition vector as input to the denoising network.

First, the cVAE is used to model the nonlinear relationship between steel/alloy composition and performance conditions. The encoder maps the input composition-performance pairs to the latent space to obtain the mean and variance parameters, and generates the latent variable z_0 by reparameterization technique. The decoder then reconstructs the composition distribution under the given latent variables and conditional constraints, ensuring that the model can learn the conditional dependent latent representation. On this basis, a conditional diffusion model is introduced to enhance the diversity and robustness of the generation process. Diffusion models can generate latent representations z from random noise by injecting Gaussian noise into latent variables and training a denoising network to learn the inverse diffusion process. Combined with conditional information, the model can generate diverse candidate components while maintaining controllability.

During the training phase, the model is trained for 100 epochs with a batch size of 64 using the Adam optimizer (learning rate 1×10^{-3}). The composite loss function consists of three terms: (1) Reconstruction Loss (L_recon): mean squared error (MSE) between the decoder output and the true composition vector in the normalized space, ensuring that the generated compositions faithfully reproduce the input. (2) KL Divergence (L_KL): $L\_KL = -1/2 \sum (1 + \log \sigma^2 - \mu^2 - \sigma^2)$, which regularizes the approximate posterior toward the standard normal prior N(0, I), promoting a smooth and continuous latent space. (3) Diffusion Loss (L_diffusion): MSE between the predicted noise and the true noise added to z at randomly sampled timesteps $t \in \{1, \ldots, 1000\}$, enabling the denoising network to learn the reverse diffusion process. The overall loss is:

$$\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_{diffusion} + \lambda \mathcal{L}_{KL}$$

where $\lambda = 0.1$ controls the strength of *KL* regularization.

In the inference phase, the system first normalizes the performance goals provided by the user and repeatedly expands them in the condition space to form batch inputs. At inference time, the user specified target properties are first extracted from natural language using an LLM-based structured parser. The property values are standardized using the same scaler_cond fitted on the training set. A batch of 32

dimensional latent vectors is sampled from N(0, I) and iteratively denoised over T = 1000 steps using the DDPM reverse process. The denoised latent representation z_0 is then passed through the trained cVAE decoder to generate the alloy composition vector. Output values are clipped to the [0, 100] range and inverse transformed via scaler_gen to recover the actual weight percentages. Multiple candidate compositions (typically 100) can be generated in a single batch by repeating the condition vector, with the top-k most chemically plausible candidates selected by filtering on compositional constraints (sum ≈ 100 wt%, individual element feasibility ranges). The agent can satisfy target performance constraints and provide diversified steel/alloy composition generation results, thus offering feasible candidate solution space for reverse design of materials.

To quantitatively validate the reconstruction fidelity of the model, we evaluate the CVAE and conditional diffusion model on the held-out test set (n = 169). The model achieves an overall reconstruction R^2 of 0.971 and a mean absolute error (MAE) of 0.311 wt% across all 13 alloying elements (Table 1, Fig. 2). Per-element R^2 values range from 0.865 to 0.992, with major alloying elements such as Cr (R^2 = 0.925) and Ni (R^2 = 0.939) reconstructed with high fidelity. The composition sum (the total weight percentage of the 13 alloying additions, Fe balance excluded) is reconstructed with R^2 = 0.893 and a mean absolute error of 1.23 wt%, with 52.7% of test samples within ±1 wt% of the true sum. These results confirm that the model accurately captures the conditional distribution of alloy compositions from mechanical property targets.

Table 1: Reconstruction accuracy of the cVAE-diffusion inverse design model evaluated on an 80/20 train/test split (n_{train} = 673, n_{test} = 169). The 13 elements represent alloying additions only (Fe is not included); composition sums are compared against ground truth rather than 100%. Generated Range denotes the span of values (maximum minus minimum) generated by the model for each element across the test set. By comparing the generated range against the ground truth distribution, we assess the model capacity to capture the diversity of the design space.

Element	R^2 (Recon.)	MAE (wt%)	Generated Range (wt%)
C	0.944	0.015	0.0–0.1
Mn	0.949	0.056	0.0–0.1
Si	0.966	0.059	0.0–0.1
Cr	0.925	1.035	2.9–12.5
Ni	0.939	0.905	5.0–14.0
Mo	0.939	0.305	2.0–3.2
V	0.966	0.042	0.0–0.2
N	0.992	0.001	0.0–0.0
Nb	0.865	0.017	0.0–0.0
Co	0.935	1.103	0.0–13.6
W	0.949	0.101	0.0–0.2
Al	0.964	0.070	0.1–1.0
Ti	0.963	0.076	0.0–1.1

Note: Overall: R^2 = 0.971, MAE = 0.311 wt%. Composition sum R^2 = 0.893 (true vs. recon.).

2.4 Heat Treatment Agent

To support the phase composition prediction of steel materials under different composition and temperature conditions, a heat treatment agent based on thermodynamic calculation is designed. The agent is able to automatically simulate the phase evolution of Fe-based alloys under arbitrary multi-element compositions and temperature ranges, and returning quantitative phase fractions for each phase.

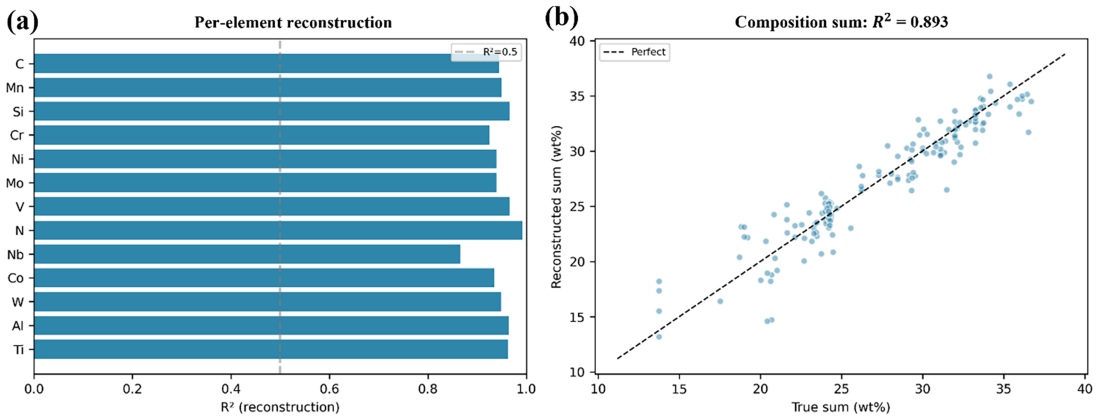

Figure 2: (**a**) Per-element reconstruction R^2. Per-element R^2 ranges from 0.865 (Nb, a sparse microalloying element) to 0.992 (N), with major elements such as Cr (R^2 = 0.925) and Ni (R^2 = 0.939) reconstructed with high fidelity. (**b**) The total alloying element sum is reconstructed with R^2 = 0.893.

During the input parsing stage, the system employs a large language model (LLM) coupled with a structured parser to extract elemental compositions from the user's natural language input. Each alloying element (such as C, Mn, Si, Cr, Ni, Mo, V, N, Nb, Co, W, Al, and Ti) is recognized and converted into standardized units (mass fraction, wt%). Missing values are automatically filled with zeros to ensure a complete composition vector. The extracted data are then formatted into a structured request object and passed to the computational module.

In the calculation phase, the agent invokes the thermodynamic calculation engine built upon the *pycalphad* library, which implements CALPHAD (CALculation of PHAse Diagrams) methodology. The system converts the input mass fractions into molar fractions based on atomic weights, supplements Fe to maintain a total of 100 wt%, and defines equilibrium conditions characterized by temperature T, pressure P (fixed at 101,325 Pa), and elemental mole fractions $\{x_i\}$. Equilibrium calculations are performed iteratively across a temperature range of 673–2000 K to obtain the temperature-dependent phase fraction data for all active phases.

During result processing, the agent filters and formats the phase equilibrium output, retains only the stable phases with fractions greater than the threshold ($>10^{-4}$), and constructs a structured dictionary describing the evolution of each phase as a function of temperature. The final output includes the active elements, temperature range, normalized molar composition, and phase fraction evolution. The final result is returned to the user in natural language, for example: "At temperature T K, the equilibrium microstructure consists of 0.52 austenite (γ-Fe) and 0.48 ferrite (α-Fe)". Through the above process, the heat treatment agent can not only realize the automatic mapping from natural language input to thermodynamic calculation, but also ensure the scientific rationality and interpretability of the results. Within the multi-agent systems, this module can be used in conjunction with the component generation agent to achieve a complete closed-loop design from target performance specification to alloy component recommendation and heat treatment simulation. The implementation is publicly available via a GitHub repository (https://github.com/siqiany/steelbackend/tree/master).

The role of the Heat Treatment Agent within the STALAgent system is defined by explicit interactions with three other agents through the CrewAI orchestration layer: (i) When the Task Assignment Agent routes an alloy design task to the Inverse Design Agent, the generated composition vector is forwarded as structured context to the Heat Treatment Agent. An LLM-based structured parser automatically extracts the composition values from the natural-language task description, with missing elements defaulting to

zero. This composition vector is then passed to the PyCALPHAD equilibrium solver, establishing a direct inverse design → thermodynamic validation pipeline. (ii) For tasks requiring both literature evidence and thermodynamic verification, the Task Assignment Agent may invoke the Semantic Search Agent first to retrieve relevant prior compositions, heat treatment schedules, and phase transformation guidelines from the literature database. This retrieved knowledge provides the Heat Treatment Agent with experimentally validated parameter ranges against which the PyCALPHAD simulation results can be interpreted and critiqued. (iii) Output to the Task Assignment Agent. The Heat Treatment Agent returns a structured result containing: the temperature range (673–2000 K, 100 points), active elements, six target phases monitored (LIQUID, BCC_A2, FCC_A1, $M_{23}C_6$, σ-phase, Laves phase), and a temperature dependent phase fraction dictionary for each phase. The Task Assignment Agent receives this output and synthesizes it with results from other agents to produce the final report. The thermodynamic database used (steel_database_fix.tdb) is a CALPHAD-format steel-specific database containing Gibbs energy descriptions for Fe based multicomponent systems. (iv) Current Limitation: One Directional Validation. It is important to note that the current interaction is one-directional: the Heat Treatment Agent validates a composition proposed by the Inverse Design Agent, but if the thermodynamic simulation reveals undesirable phases, there is no automatic feedback loop to trigger the Inverse Design Agent to regenerate an improved composition. The manager may, in principle, re-assign the task based on the Heat Treatment Agent's critique, but this meta-cognitive re-planning capability depends on the reasoning quality of LLM. A closed loop refinement mechanism is planned as a future extension.

3 Results and Discussion

The main contributions of this software include the following aspects. (1) Task Assignment. The task assignment mechanism is based on the CrewAI framework, which dynamically allocates tasks according to the semantics of user queries. (2) Retrieval-Augmented Generation (RAG) Pipeline. The system features an end-to-end RAG pipeline centered on a specialized literature knowledge base. After document segmentation and metadata extraction, a dual-path retrieval (keyword and dense vector) followed by re-ranking ensures precise, context-aware fetching of relevant document fragments and prior knowledge in response to user queries. (3) Generative Design Assistants. The software integrates preliminary modules dedicated to the following key generative task: inverse materials design with alloy composition recommendation. (4) Heat Treatment Simulation. This module provides *in-silico* heat treatment simulations to facilitate the evaluation of physical rationale and practical potential, thereby supporting the critical link from computational recommendation to experimental validation. (5) Interactive Interface and Interpretability. A conversational web front end is constructed that supports user queries and PDF uploads, while the system displays decision traces between query processes and agents to ensure transparency and understanding. To validate the STALAgent framework, we present case studies that exemplify its application in materials design, leveraging the synergy of its LLM-driven task planning, semantic RAG, generative design modules, and *in-silico* heat treatment validation.

3.1 Case 1: High-Strength and Tough Steel Design

This case demonstrates the capability of STALAgent for designing a high-strength and tough steel with a target tensile strength of approximately 650 MPa and an elongation rate of ≥15%. This workflow comprises five sequential stages: Stage 1, task analysis and assignment; Stage 2, semantic search via an RAG agent; Stage 3, inverse design executed by an inverse design agent; Stage 4, heat-treatment optimization conducted by a specialized heat-treatment agent; and Stage 5, generation of the final output (Fig. 3).

(1) Task Analysis and Assignment. When the user proposed the task *"Design a high-strength and tough steel with a tensile strength of about 650 MPa and an elongation rate of at least 15%,"* the Task Assignment

Agent first performed semantic parsing on the input. Based on the intent and constraints, the system automatically determine that Semantic Search Agent, Inverse Design Agent and Heat Treatment Agent are required.

(2) Semantic Search Agent. The Task Assignment Agent delegate the following subtask to the Semantic Search Agent: *"Research and provide information on existing steel grades and compositions that achieve tensile strength around 650 MPa and elongation rate of 15% or higher. Focus on high-strength low-alloy (HSLA) steels, quenched and tempered steels, or other relevant categories."* The core objective is to retrieve peer-reviewed literature and technical data for reference in subsequent alloy design. The Semantic Search Agent invokes specialized database retrieval tools to conduct targeted searches. Key retrieval criteria include tensile strength (TS) ≈ 650 MPa, elongation (El) ≥ 15%, and alloy categories limited to HSLA steels, quenched and tempered (Q&T) steels, and medium-Mn steels. The following representative examples are identified: (i) AISI 1020 carburized and quenched steel: Tensile strength (TS) = 365 MPa, elongation (El) = 18%. This grade exhibits high ductility but fails to meet the target strength requirement. (ii) HSLA steel with gradient dislocation-cell structures: Yield strength (YS) = 522 MPa, El = 25.5%. Its excellent ductility and near-target strength indicate significant optimization potential for reaching the 650 MPa class. (iii) Q&T structural steels (e.g., S690QL, F500W): Microalloyed with Nb, Ti, and V, these grades achieve YS ≈ 690 MPa and demonstrate superior toughness, confirming the feasibility of microalloying for high-strength requirements. (iv) Medium-Mn dual-phase steels (annealed at 680°C–760°C): TS > 650 MPa and El > 20%, realizing a superior strength-ductility synergy through microstructure regulation. Through comparative analysis of mechanical properties and alloying systems, the Semantic Search Agent infers two viable technical pathways: 1. HSLA steels microalloyed with Nb + Ti + V, optimized via quenching and tempering; 2. medium-Mn steels with dual-phase microstructure control. Both pathways are deemed capable of simultaneously meeting the target strength (≈650 MPa) and ductility (≥15%) requirements.

(3) Inverse Design Agent. Building on the literature insights from the Semantic Search Agent, the Task Assignment Agent assigns the following subtask to the Inverse Design Agent: *"Based on the reference steel compositions and properties provided, design a new steel alloy composition that can achieve tensile strength of 650 MPa and elongation rate of ≥15%. Consider microalloying elements and processing routes."* First, the Inverse Design Agent extracts key performance indicators from the task: ultimate tensile strength (UTS) = 650 MPa, El ≥ 15%. It then invokes a machine learning-based alloy design tool to generate a preliminary composition, which is returned as: [{'C': 0.00329, 'Mn': 0.589, 'Si': 0.040, 'Cr': 14.936, 'Ni': 8.765, 'Mo': 2.580, 'V': 0.000, 'N': 0.020, 'Nb': 0.000, 'Co': 0.000, 'W': 0.107, 'Al': 0.750, 'Ti': 0.131}] (all in wt%). To balance performance and industrial applicability, the agent refines the preliminary composition by integrating HSLA steel design principles and reference data: (i) Reducing noble metal content (Cr, Ni, Mo) to lower production costs while retaining core strengthening elements. (ii) Adjusting C content to optimize the strength-ductility balance, aligned with HSLA steel's low-carbon design philosophy. (iii) Adding microalloying elements (Nb, V, Ti) to enhance grain refinement and precipitate strengthening.

The final proposed alloy composition and processing route are defined as follows. The carbon content is set at 0.10 wt% (≤0.15 wt%) to achieve an optimal balance between strength and ductility while maintaining good weldability. Manganese is added at 1.50 wt% (0.65–1.69 wt%) to enhance strength and hardenability through solid-solution strengthening. Silicon is fixed at 0.25 wt%, serving primarily as a deoxidizing agent during smelting while providing additional solid-solution strengthening. Microalloying with niobium (0.03 wt%), vanadium (0.05 wt%), and titanium (0.01 wt%) is employed to refine prior-austenite grain size and to promote the formation of nanoscale carbide precipitates (e.g., NbC and VC), thereby contributing to

precipitation strengthening. Aluminum is controlled at 0.02 wt% for effective deoxidation, while impurity elements such as phosphorus and sulfur are strictly limited to below 0.015 wt% each to prevent degradation of ductility and toughness.

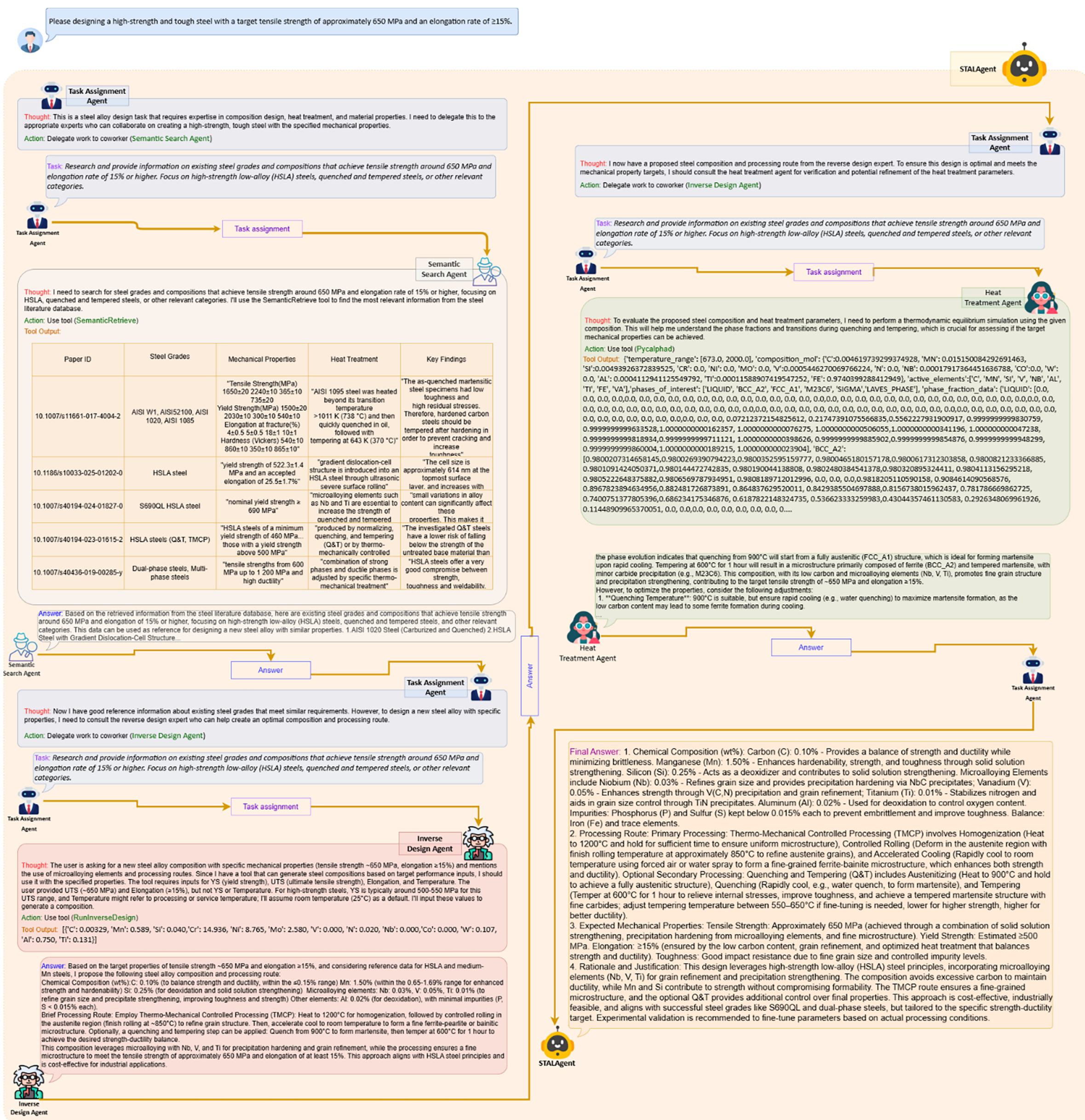

Figure 3: A case study on high-strength and tough steel design (Case 1). The target is to develop a steel with a tensile strength of approximately 650 MPa and an elongation of ≥15%.

The proposed processing route is based on TMCP. The alloy is first homogenized at 1200°C, followed by controlled rolling in the austenite region with a finish rolling temperature of approximately 850°C to achieve effective grain refinement. Subsequently, accelerated cooling to room temperature is applied to promote the formation of a fine ferrite–pearlite or bainitic microstructure. In addition, an optional quenching and tempering (Q&T) treatment may be employed, in which the material is quenched from 900°C to obtain a martensitic structure and then tempered at 600°C for 1 h to relieve residual stresses and optimize the balance

between strength and ductility. This design leverages microalloying and controlled processing to achieve the target properties, with cost-effectiveness and industrial scalability as considerations.

(4) Heat Treatment Agent. To validate the feasibility of the proposed composition and processing route, the Task Assignment Agent assigns the following subtask to the Heat Treatment Agent: *"Please review and verify the proposed steel composition and processing route for achieving tensile strength ~650 MPa and elongation ≥15%. Specifically, evaluate the heat treatment parameters (quenching and tempering) and suggest any adjustments to optimize the microstructure and mechanical properties."* The Heat Treatment Agent extracts the weight percentages of effective elements (C, Mn, Si, V, Nb, Al, Ti, Fe) from the response from the inverse design agent. It then invokes Pycalphad for thermodynamic equilibrium simulation, calculating the behavior of the following phases across 100 temperature points in the range of 673 to 2000 K: LIQUID, BCC_A2, FCC_A1, M23C6, SIGMA, and LAVES_PHASE. The Heat Treatment Agent provides its verification response by integrating the thermodynamic simulation results from Pycalphad.

(5) Final Output. After synthesizing technical insights from the Semantic Search Agent, Inverse Design Agent, and Heat Treatment Agent, the Task Assignment Agent generates a comprehensive final report. The report includes the optimized alloy composition (wt%: C 0.10, Mn 1.50, Si 0.25, Nb 0.03, V 0.05, Ti 0.01, Al 0.02, P/S ≤ 0.015), detailed processing route (TMCP + optional Q&T), and performance validation results. We note that the proposed composition represents a preliminary hypothesis generated by the system, not a validated steel alloy. Experimental confirmation of the predicted mechanical properties through synthesis and testing would be required to verify the design. The system provides a computationally grounded starting point that narrows the experimental search space, rather than a substitute for empirical validation.

3.2 Case 2: Steel Knowledge Q&A Task

This example complements Case 1 by demonstrating the capabilities of the semantic retrieval agent (Stage 1: Task Analysis and Assignment; Stage 2: Semantic Search Agent) (Fig. 4). (1) Task Analysis and Assignment. When the user posed the question "*What are the characteristics of concrete materials reinforced with steel fibers in terms of structure and performance?*", The Task Assignment Agent first conducts semantic analysis of the query, identifying its core focus on the structural and performance-related characteristics of steel fiber reinforced concrete (SFRC). It then assigns the task to the Semantic Search Agents. (2) Semantic Search Agent. Upon receiving the task, the Semantic Search Agent extracts initial keywords from the query, forming the first search prompt: "steel fiber reinforced concrete characteristics microstructure mechanical properties strength toughness durability structural performance". This prompt is used to call the Elasticsearch-based semantic retrieval tool, which returned relevant academic literature fragments, including studies on SFRC's blast resistance, flexural behavior, and fiber-matrix bond performance. After evaluating the retrieved results, the agent identifies gaps in the information, such as insufficient details on specific mechanical properties (e.g., tensile strength, crack resistance) and quantitative performance data. To address this, the Sematic Search Agent optimizes the search prompt via the inclusion the following keywords related to the user query: *steel fiber reinforced concrete, microstructure, mechanical properties, compressive strength, flexural strength, toughness, durability, crack resistance, shrinkage, impact resistance, structural performance*. Subsequently, a second round of retrieval is executed, supplementing key data including SFRC's 49.0% increase in compressive strength for ceramsite concrete, 69% improvement in flexural strength (1.2% fiber volume in 3D-printed geopolymer concrete at 7 days), and significant enhancements in toughness indices ($W_{0.85}$ +85.1%, $W_{0.5}$ +257.6%). When the retrieved information fully covers the structural information (e.g., crack-bridging microstructure, fiber-matrix bonding) and performance (mechanical properties, durability,

structural behavior) dimensions of the user query, the agent encapsulates the validated literature fragments into a structured dataset, marking the completion of evidence collection.

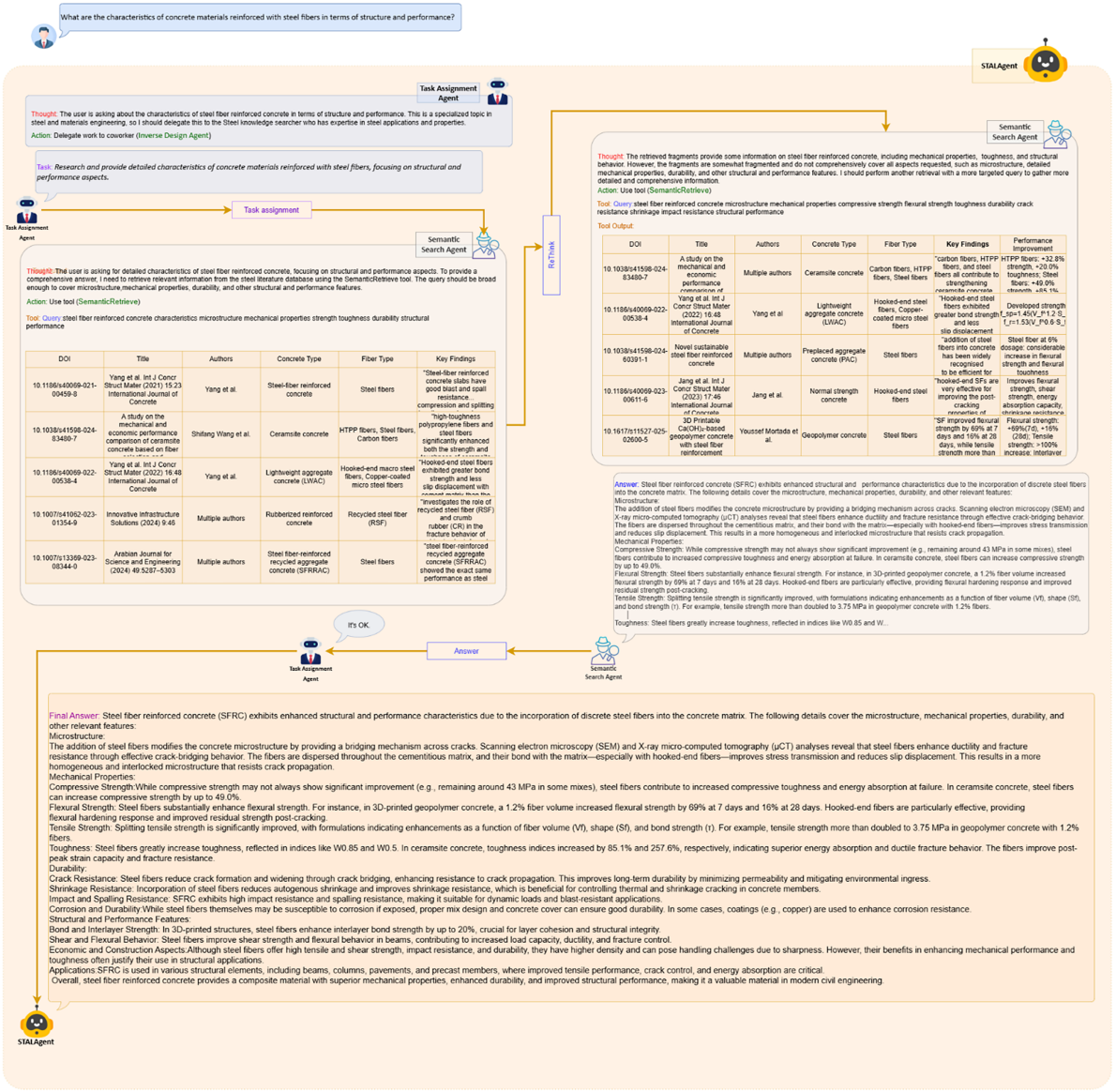

Figure 4: A case study on retrieval-augmented generation (RAG) (Case 2) illustrating the user query on the structural characteristics and performance properties of steel fiber-reinforced concrete.

3.3 Case 3: 30-Task Alloy Design Benchmark and 80-Task General Steel QA Benchmark

We employ two benchmarks to first assess the end-to-end answer quality of STALAgent: (1) Alloy Design Benchmark (30 tasks): Comprising 20 B_phase tasks (equilibrium phase calculation for industrial steel grades including AISI 1045, 4140, 4340, H13, D2, 316L, 2205, M2) and 10 C_integrated tasks (end-to-end alloy design with composition proposal, phase validation, and heat treatment specification). Each answer was independently scored by an LLM judge (qwen3-max) on 5–6 domain-specific criteria at the 1–5 scale. (2) General Steel QA Benchmark (80 tasks): Covering broad steel metallurgy topics from phase

diagrams to processing-microstructure-property relationships. Each answer was scored on four criteria: explanation_quality, feasibility, target_match, and novelty.

To isolate the contribution of the multi-agent architecture from the underlying LLM capability, we compared STALAgent (DeepSeek Flash backend) against the same DeepSeek Flash model as a standalone LLM, evaluated under identical conditions on the 30-task benchmark. We further benchmarked three additional general purpose LLMs (DeepSeek Pro, GPT-5.4, Qwen3-max) on the 80-task benchmark using the same LLM judge and scoring rubric.

On the 30-task benchmark, STALAgent achieves an overall score of 4.57, compared to 4.30 for the single LLM baseline (+6.3%) (Fig. 5). The advantage is more pronounced on C_integrated design tasks (score: 4.90 vs. 4.70), where the coordinated Search → Inverse → Heat pipeline provides compounding benefits, than on B_phase calculation tasks (score: 4.40 vs. 4.10). STALAgent wins 11 of 15 non-tie tasks in head-to-head comparison (73% win rate). On C_integrated design tasks, STALAgent achieves perfect Constraint Satisfaction Rate (5.00), indicating that all generated alloy compositions fall within feasible metallurgical ranges. High scores on Design Completeness (4.90) and Thermodynamic Soundness (4.80) further confirm that the agent pipeline successfully incorporates CALPHAD predictions and materials-domain constraints into the design process. What's more, STALAgent averages 113 s per task and 239,527 tokens. The additional ~40 s compared to the single LLM (73 s) is attributed to PyCALPHAD thermodynamic computation (~30 s per task) and Elasticsearch-based literature retrieval (~10 s per task), both components that generate the quantitative phase data and expert references underpinning the higher evaluation scores of STALAgent.

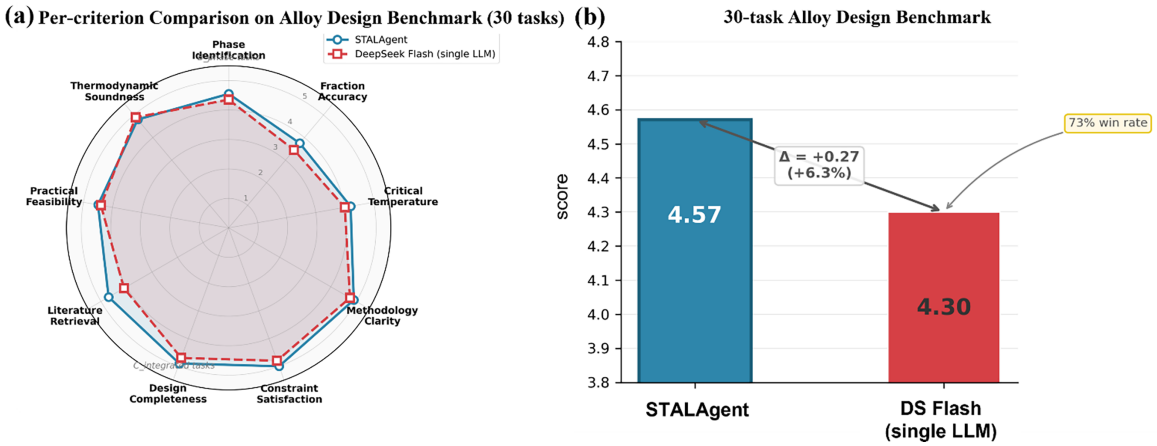

Figure 5: (**a**) Radar chart for STALAgent (driven by DeepSeek-Flash) against the single LLM (DeepSeek-Flash) on the 30-task benchmark. (**b**) Bar chart that compares the overall score of STALAgent and DeepSeek-Flash on the 30-task benchmark.

On the 80-task benchmark, STALAgent achieves an overall score of 4.54, outperforming all single-LLM baselines (score: Qwen: 4.37, GPT-5.4: 4.32, DeepSeek Flash: 4.31, DeepSeek Pro: 4.30) (Fig. 6). In head-to-head paired comparison against the identical base model (DeepSeek Flash), STALAgent wins 51 of 55 non-tie tasks (93% win rate), providing a controlled demonstration that the multi-agent architecture, not the base LLM, drives the performance gain. On this benchmark, STALAgent averages 61 s and 727,882 tokens per task. More detailed information is summarized in Tables S2–S6 and Figs. S5 and S6.

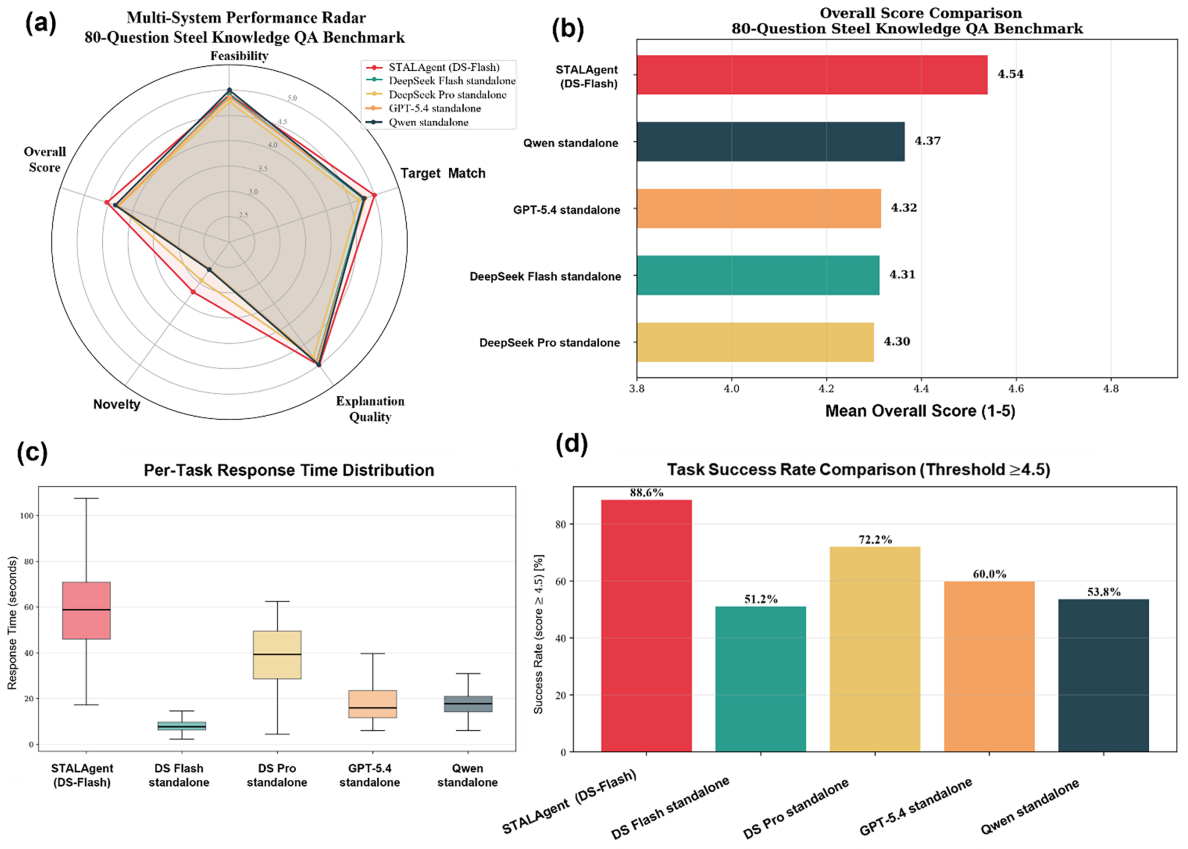

Figure 6: (**a**) Multi-system performance rander on the 80-question steel knowledge QA benchmark. (**b**) Overall score comparison of STALAgent (driven by DeepSeek-Flash) and single LLMs (DeepSeek-Flash, DeepSeek-Pro, GPT-5.4 and Qwen3-max) on the 80-question steel knowledge QA benchmark. (**c**) Per-task response time distribution of STALAgent (driven by DeepSeek-Flash) and single LLMs (DeepSeek-Flash, DeepSeek-Pro, GPT-5.4 and Qwen3-max). (**d**) Task success rate comparison of STALAgent (driven by DeepSeek-Flash) and single LLMs (DeepSeek-Flash, DeepSeek-Pro, GPT-5.4 and Qwen3-max). We define a task whose score is above 4.5 as success.

Through the above case studies, it is evident that the task processing flow of STALAgent is composed of sub-agents working sequentially. The quantitative results across 110 benchmark tasks confirm that the multi-agent architecture provides measurable domain-specific advantages over general purpose LLMs, particularly on tasks requiring integrated knowledge retrieval, thermodynamic computation, and constraint driven inverse design. In addition to the task assignment agent and the summarization agent, the intermediate sub-agent can be flexibly adjusted according to the task. The architecture makes STALAgent extensible: in the future, integrating new functional modules will be straightforward, one only needs to define a new agent through the CrewAI framework, specify its functions, connect the relevant tool API, and update the task assignment mechanism.

3.4 Robustness, Limitations, Generalizability, and Deployment

System Robustness and Interpretability of STALAgent. STALAgent handles ambiguous and incomplete queries through a five-layer defense: (1) LLM-based semantic completion infers missing numerical targets from domain context; (2) statistical fallback to training-set defaults ensures the pipeline always produces output; (3) a composition validation layer checks all inputs against the documented limits of the

MatCalc thermodynamic database (e.g., Cr < 25 wt%, Ni < 26 wt%) and returns descriptive error messages for out-of-range inputs; (4) the final synthesis agent is instructed to report upstream errors transparently rather than fabricating answers; (5) all tool-execution paths are wrapped in try/except blocks. For interpretability, STALAgent provides five complementary mechanisms: natural-language explanation synthesis (scored 5.00/5.00 on our QA benchmark), input-output transparency where target properties are paired with generated compositions in a side-by-side format, thermodynamic grounding via PyCALPHAD against a validated database, a structured agent execution trace that records every delegation event with timestamps for full auditability, and source-attributed knowledge retrieval with paper-level metadata. Current limitations include the black box nature of the diffusion model and the absence of explicit RAG citations in user-facing answers, improvements planned for future versions.

Limitations: Hallucination, Data Quality, and Uncertainty of STALAgent. STALAgent mitigates LLM hallucination through four architectural layers: (i) retrieval grounded text generation anchors answers in peer-reviewed literature rather than parametric memory; (ii) the VAE/Diffusion model generates compositions on a learned manifold of physically realizable alloys; (iii) all compositions are verified through PyCALPHAD equilibrium calculations with documented database constraints; (iv) multi-agent cross-checking captures physically anomalous results. The LLM serves as orchestrator and synthesizer, not as a direct source of numerical predictions. Regarding uncertainty, CALPHAD simulations are inherently deterministic, solving for the global minimum of Gibbs energy, and uncertainty is managed through database boundary enforcement rather than probabilistic quantification. For inverse design predictions, the VAE encoder computes per-element log-variance during training but this information is currently discarded at inference; the system generates a single candidate without confidence intervals. Planned improvements include exposing VAE variance, multi-sample generation with clustering, and ensemble-based uncertainty estimation. The literature corpus draws from peer-reviewed publications, and the thermodynamic database is a validated community resource; extending coverage to emerging alloy systems remains ongoing work.

Complementary Methods and Generalizability of STALAgent. Graph neural networks (GNNs) and reinforcement learning (RL) provide complementary capabilities to STALAgent's current VAE/Diffusion approach. GNNs excel at structure-aware predictions when crystallographic data is available, and could be integrated as an alternative inverse design tool. RL formulates materials discovery as a sequential decision process and can leverage the existing PyCALPHAD validation layer as an environment simulator, with the CrewAI sequential execution model supporting iterative optimization. All three paradigms, VAE/Diffusion for rapid screening, GNNs for structure-aware prediction, and RL for iterative optimization, can coexist as tools within the agent framework, dynamically selected by the orchestrator LLM based on task context. The STALAgent architecture is material-agnostic. Adapting to a new alloy system requires only three configuration changes: (1) substituting the CALPHAD thermodynamic database (mature databases exist for Ni-based superalloys, Al alloys, Ti alloys, and high-entropy alloys); (2) retraining or fine-tuning the VAE/Diffusion model on the new system's composition-property data; and (3) re-indexing the literature corpus. The agent orchestration, retrieval pipeline, and model architecture remain unchanged. Ni-based superalloys and Al alloys are planned as the next validation targets.

Toward Practical Deployment: Experiment Integration, Sensitivity Analysis, and Future Extensions. STALAgent is designed to accelerate, not replace, experimental alloy development. We envision three integration modes of increasing automation: (1) offline advisory (current), where domain experts evaluate computational predictions before committing experimental resources; (2) active learning, where experimentally validated compositions are fed back as training data to progressively refine the VAE/Diffusion model; (3) autonomous closed-loop discovery (long-term), integrating STALAgent with high-throughput experimentation platforms for end-to-end automated alloy development. Quantitative sensitivity analysis,

perturbing input conditions, processing parameters, and compositions near database boundaries, would further strengthen design robustness by identifying which variables most strongly influence predictions and where confidence is highest. This can be implemented as an optional analytical mode invoked on demand. For human-in-the-loop integration, we propose three levels: mid-pipeline expert intervention (pausing the workflow for metallurgist review after inverse design), structured feedback collection for model fine-tuning, and interactive collaborative design with iterative refinement. Several technical extensions are planned. Real-time optimization can reduce the current 61.3 s mean response time toward the sub-10-s range via PyCALPHAD caching, lightweight model serving, and streaming token generation. Physics-informed neural networks (PINNs) offer two integration pathways: augmenting the VAE/Diffusion loss with thermodynamic constraints (mass conservation, Gibbs phase rule), and training a PINN surrogate for rapid PyCALPHAD approximation, an approach recently demonstrated for multi-scale thermal conductivity prediction using PINNs. Beyond equilibrium thermodynamics, phase-field models and finite element solvers can extend the system to non-equilibrium microstructural evolution and thermo-mechanical analysis, forming a tiered computational hierarchy: fast PINN surrogate → PyCALPHAD validation → phase-field/FEM detailed analysis.

4 Conclusions

In this study, an integrated LLM agent for steel and alloy design is developed. This system leverages an LLM based on the CrewAI orchestrator as its central brain to intelligently assign tasks and coordinate a suite of specialized agents for materials design, including tools for thermodynamic calculations (Pycalphad), knowledge retrieval (RAG), and inverse design (VAE). The system provides materials science domain knowledge support and guidance to researchers by integrating semantic literature search, reverse alloy design, and thermodynamic simulation into an interactive workflow. Case studies show that STALAgent can generate candidate alloy compositions following user query, inverse design with composition recommendation, and perform heat treatment analysis toward high-value materials recommendations. In addition to its direct application in steel and alloy materials research, the framework can be extended to other fields where knowledge retrieval, generative modeling, and simulation need to be integrated. Overall, STALAgent provides an efficient way to accelerate materials design through intelligent, interpretable, and scalable multi-agent LLM systems.

Acknowledgement: This work was supported by the NUIST Students' Platform for Innovation Training Program (No. XJDC202510300646).

Funding Statement: This work was supported by the Chinese Academy of Sciences Pioneer Initiative (CAS Leading Action Plan) and the NUIST Students' Platform for Innovation Training Program (No. XJDC202510300646).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Jiayi Qiu, Lei Zhang; data collection: Jiayi Qiu, Youle Wang, Lei Zhang; draft manuscript preparation: Jiayi Qiu, Lei Zhang. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that supports the findings of this study are available within the article.

Ethics Approval: Not applicable. This study did not involve human participants, animals, or any research requiring ethical approval.

Conflicts of Interest: Given his role as Editorial Board Member of this journal, Lei Zhang had no involvement in the peer review of this article and had no access to information regarding its peer review. Full responsibility for the editorial process for this article was delegated to another journal editor. The authors declare no other conflicts of interest.

Supplementary Materials: The supplementary material is available online at https://www.techscience.com/doi/10.32604/cmc.2026.084061/s1.

References

1. Ge H, Liu J, Sebek M, Li Z, Fu W, Wang Z, et al. AI-assisted wafer-scale exfoliation and transfer of 2D materials: status, challenges and perspectives. AI Sci. 2025;1(1):013002. doi:10.1088/3050-287X/ae0353.
2. He J, Guan H, Feng W, Min Y, Yi J, Tang K, et al. Controlling risks of AI in chemical science with agents. AI Sci. 2025;1(1):015002. doi:10.1088/3050-287x/adfee5.
3. Yang Y, Cao L, Ren J, Gao W, Ling S. Artificial intelligence for fibrous network design and mechanics. AI Sci. 2025;1(1):012001. doi:10.1088/3050-287X/adfaf7.
4. Wei J, Yang Y, Zhang X, Chen Y, Zhuang X, Gao Z, et al. From AI for science to agentic science: a survey on autonomous scientific discovery. arXiv:2508.14111. 2025.
5. Li L, Xu K, Su R, Gu H, Ma P. Data speaks: LLM-enabled evidence auditing resets the "migration barrier" playbook for solid-state electrolytes. AI Mater. 2025;1(1):2. doi:10.53941/aimat.2026.100002.
6. Yuan Y, Sui Y, Li P, Quan M, Zhou H, Jiang A. Multi-model integration accelerates Al-Zn-Mg-Cu alloy screening. J Mater Inf. 2024;4(4):23. doi:10.20517/jmi.2024.34.
7. Mannan S, Bihani V, Anoop Krishnan NM, Mauro JC. Navigating energy landscapes for materials discovery: integrating modeling, simulation, and machine learning. Mater Genome Eng Adv. 2024;2(1):e25. doi:10.1002/mgea.25.
8. Wang WY, Zhang S, Li G, Lu J, Ren Y, Wang X, et al. Artificial intelligence enabled smart design and manufacturing of advanced materials: the endless Frontier in AI^{+} era. Mater Genome Eng Adv. 2024;2(3):e56. doi:10.1002/mgea.56.
9. Lv S, Peng L, Wu W, Yao Y, Jiao S, Hu W. Bridging language models and computational materials science: a prompt-driven framework for material property prediction. Mater Genome Eng Adv. 2025;3(2):e70013. doi:10.1002/mgea.70013.
10. Ding K, Yu J, Huang J, Yang Y, Zhang Q, Chen H. SciToolAgent: a knowledge-graph-driven scientific agent for multitool integration. Nat Comput Sci. 2025;5(10):962–72. doi:10.1038/s43588-025-00849-y.
11. Dou B, Zhu Z, Merkurjev E, Ke L, Chen L, Jiang J, et al. Machine learning methods for small data challenges in molecular science. Chem Rev. 2023;123(13):8736–80. doi:10.1021/acs.chemrev.3c00189.
12. Yao T, Yang Y, Cai J, Liu R, Dong Z, Tang X, et al. From LLM to agent: a large-language-model-driven machine learning framework for catalyst design of MgH_2 dehydrogenation. J Magnes Alloys. 2026;14(24):101858. doi:10.1016/j.jma.2025.08.021.
13. Jain A, Ong SP, Hautier G, Chen W, Richards WD, Dacek S, et al. Commentary: the materials project a materials genome approach to accelerating materials innovation. APL Mater. 2013;1(1):011002. doi:10.1063/1.4812323.
14. Kirklin S, Saal JE, Meredig B, Thompson A, Doak JW, Aykol M, et al. The open quantum materials database (OQMD): assessing the accuracy of DFT formation energies. npj Comput Mater. 2015;1(1):15010. doi:10.1038/npjcompumats.2015.10.
15. Zhang L, Huang Y, Yan L, Ge J, Ma X, Liu Z, et al. Fast exploring literature by language machine learning for perovskite solar cell materials design. Adv Intell Syst. 2024;6(6):2300678. doi:10.1002/aisy.202300678.
16. Choudhary K. The JARVIS infrastructure is all you need for materials design. Comput Mater Sci. 2025;259(5330):114063. doi:10.1016/j.commatsci.2025.114063.
17. Ezugwu AE, Ho YS, Egwuche OS, Ekundayo OS, Van Der Merwe A, Saha AK, et al. Classical machine learning: seventy years of algorithmic learning evolution. Data Intell. 2025;7(4):947–96. doi:10.3724/2096-7004.di.2024.0051.
18. Zhang L, Liu Z, Ni B, Wang Q. Large language models (LLMs) for materials design. Adv Funct Mater. 2026;36(30):e25897. doi:10.1002/adfm.202525897.
19. Guo D, Yang D, Zhang H, Song J, Wang P, Zhu Q, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. Nature. 2025;645(8081):633–8. doi:10.1038/s41586-025-09422-z.

20. Ansari M, Moosavi SM. Agent-based learning of materials datasets from the scientific literature. Digit Discov. 2024;3(12):2607–17. doi:10.1039/d4dd00252k.
21. Feng R, Liang Y, Yin T, Gao P, Wang W. Agentic assistant for materials scientists. Electrochem Soc Interface. 2025;34(2):45–8. doi:10.1149/2.f09252if.
22. Zimmermann Y, Bazgir A, Al-Feghali A, Ansari M, Bocarsly J, Brinson LC, et al. 32 examples of LLM applications in materials science and chemistry: towards automation, assistants, agents, and accelerated scientific discovery. Mach Learn Sci Technol. 2025;6(3):030701. doi:10.1088/2632-2153/ae011a.
23. Cheng G, Zhou K, Gong XG, Yin WJ. D^3REAM: a framework for full-space inverse materials design with target properties via machine learning models and global optimization algorithms. AI Sci. 2026;2(2):025005. doi:10.1088/3050-287X/ae70c4.
24. Liu ZK, Li B. Perspectives on thermodynamics for AI and AI safety. AI Sci. 2026;2(2):022001. doi:10.1088/3050-287X/ae7e9b.
25. Ma J, Cao B, Dong S, Tian Y, Wang M, Xiong J, et al. MLMD: a programming-free AI platform to predict and design materials. npj Comput Mater. 2024;10(1):59. doi:10.1038/s41524-024-01243-4.