



Cite this: DOI: 10.1039/d5dd00435g

Multi-agentic AI framework for end-to-end atomistic simulations

Aikaterini Vriza, ^{†*}^a Uma Kornu, ^{†*}^b Aditya Koneru, ^c Henry Chan ^{ab} and Subramanian K. R. S. Sankaranarayanan ^{ab}

One of the main bottlenecks for the wide adoption of atomistic simulation pipelines for computational materials design is the high complexity of the workflows which many times requires the use of a diverse set of specialized toolkits and libraries. Here, we introduce a multi-agent artificial intelligence (AI) framework that autonomously performs end-to-end atomistic simulations, *i.e.* molecular dynamics (MD), with automated input and associated full suite of analyses, using large language models (LLMs) and multiple specialized AI agents. Our system orchestrates the entire simulation pipeline, from structure generation *via* Atomsk and interatomic potential discovery through automated web mining, to simulation setup and execution using LAMMPS on high-performance computing (HPC) platforms. Post-simulation, our agentic framework performs automated data analysis and visualization with popular analysis tools like OVITO and Phonopy. Each expert agent operates within a defined role, equipped with domain-specific functions and a shared memory context for coordination. Using a diverse set of representative elemental and alloy systems, we demonstrate the capability of our framework to execute a range of static and dynamic materials modeling tasks, including lattice parameter and cohesive energy estimation, elastic constants computation, phonon dispersion analysis, as well as perform MD simulations to determine dynamical properties that aid estimation of melting point. The results produced by the agents show strong agreement with those obtained by a human expert, highlighting the reliability of the agentic approach. By combining automation, reproducibility, and human-in-the-loop control, our framework lowers the barrier to the widespread adoption of scalable, AI-driven discovery tools in materials science.

Received 28th September 2025

Accepted 8th December 2025

DOI: 10.1039/d5dd00435g

rsc.li/digitaldiscovery

1. Introduction

Atomistic simulations represent a popular and powerful method to understand structure and dynamics of complex systems in materials science spanning a broad range of applications from energy to environment.^{1–3} In particular, techniques such as molecular dynamics (MD) provide detailed insights into atomic-scale behavior, enabling the study of structural relaxation, phase transitions,⁴ transport phenomena,⁵ and thermomechanical properties of solids and fluids. The level of accuracy is determined by an interatomic potential (force field) which defines the potential energy of the system as a function of atom positions.⁶ However, despite their wide adoption and availability, these simulation workflows remain fragmented, manual, and highly dependent on domain expertise. Researchers must typically integrate multiple tools such as, for example, Atomsk for

structure generation,⁷ LAMMPS for dynamics simulation,⁸ Phonopy for simulation analysis,⁹ and OVITO for visualization,¹⁰ alongside literature or web search to retrieve force fields from relevant databases like the NIST Interatomic Potentials Repository.¹¹ This fragmented collection of tools and processes introduces challenges in interoperability, reproducibility and scalability, especially for high-throughput or exploratory simulations. It also limits accessibility for non-specialists and poses a significant challenge for integrating simulations into autonomous or data-driven materials design pipelines.

Recent advances in large language models (LLMs) and agent-based AI systems offer promising routes toward automating complex scientific workflows that lower the barrier for scientific usage by non-expert users.^{12–16} There has been a significant surge in the development of agentic pipelines designed to automate experimental workflows, simulation tools and the subsequent data analysis processes.^{17–20} These workflows aim to leverage intelligent agents, often powered by machine learning or AI, to orchestrate complex computational tasks, streamline parameter exploration, and autonomously extract insights, thereby accelerating scientific discovery and reducing human intervention in traditionally labor-intensive simulation pipelines.^{21,22} For example MDCrowd introduced a single agent system built on LangChain,²³

^aCenter for Nanoscale Materials, Argonne National Laboratory, Lemont, IL, 60439, USA. E-mail: kvriza@gmail.com; hchan@anl.gov; skrssank@anl.gov

^bDepartment of Mechanical and Industrial Engineering, University of Illinois, Chicago, IL, 60607, USA

^cArgonne Leadership Computing Facility, Argonne National Laboratory, Lemont, IL, 60439, USA

[†] Equal contribution.

leveraging GPT-4o and Llama3-405B to automate biomolecular MD workflows using tools like OpenMM, MDTraj, and PDBFixer.²⁴ Similarly, DynaMate offers modular multi-agent pipeline for scientific workflows built on a flexible Python template that leverages LLM agents to automate molecular simulations across diverse domains mainly focused on solvents and metal-organic frameworks.²⁵ El Agente demonstrated a hierarchical framework for complex computational and quantum chemistry workflows by integrating with GPT-4 to autonomously plan, execute, and debug DFT-based computations using tools like ORCA and xTB.²⁶ AtomAgents, on the other hand applied multi-modal GPT-4 based

multi-agent system, where each agent is designed to act like planner, scientist and engineer to automate alloy design coordinating MD workflows.²⁶ More recently, DREAMS framework was used for DFT calculations on a materials benchmark, demonstrating comparable performance to human DFT experts.²⁷ While each of these agentic systems contribute valuable ideas, they often rely on static templates, lack flexibility in input generation, and do not fully support classical MD workflows involving multiple diverse tasks ranging from structure generation, interatomic potential selection by searching the web, visual inspection of the structures, or trajectory analysis.

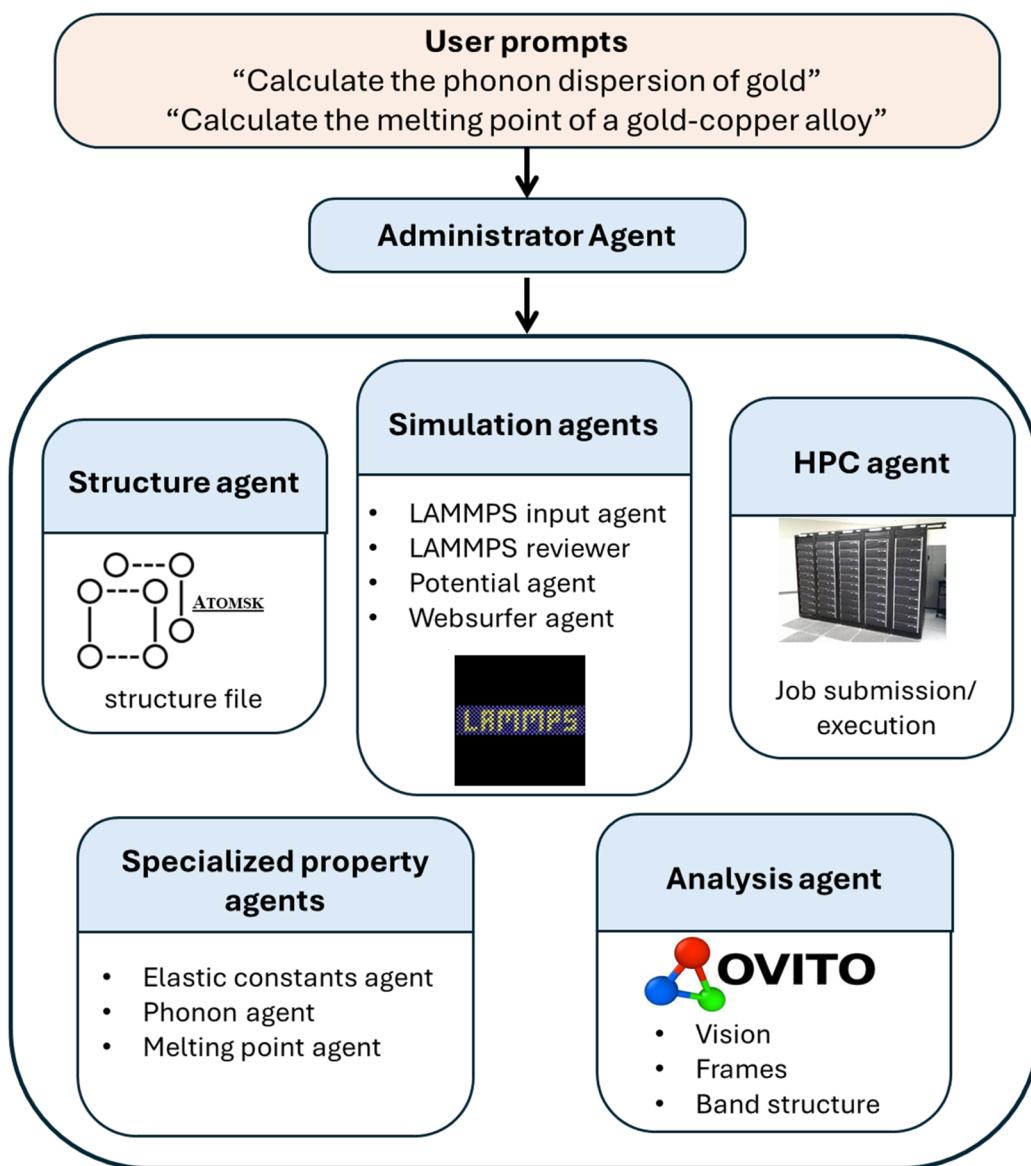


Fig. 1 Overview of the multi-agent system for molecular dynamics (MD) simulations. The system enables end-to-end automation of atomistic simulations and analysis through coordinated agents. The administrator agent receives high-level natural language instruction from the user and assigns tasks to specialized agents. These include: (i) the structure agent, which generates initial candidate structures that satisfy user requirements, (ii) the simulation agents, which prepare input files for execution, (iii) the HPC agent, which submits jobs to an external high-performance computing cluster (HPC) (https://wiki.anl.gov/cnm/HPC/Network_Access), (iv) specialized property agents, which are integrated with functions tailored to specific materials properties and (v) the analysis agent, which performs multimodal reasoning over simulation outputs and generates property specific plots. Each agent operates within a focused scope defined by its system message and interacts with registered functions via the administrator agent.

In this work, we introduce a multi-agent AI framework for automating the full pipeline of atomistic simulations, from structure generation to post-processing and visualization. Each agent is provided with a specialized set of domain-specific tools and a clearly defined role. This modular, role-specialized architecture is motivated by recent studies showing that distributing reasoning and tool control across multiple collaborating agents improves robustness and reasoning efficiency compared to single-agent systems.^{28–30} Such architectures reduce the contextual load on any single agent and enable complex workflows to be decomposed into manageable subtasks, mitigating context-window limitations observed in single-agent designs.²⁶ Unlike prior works which are largely focused on single-agent systems, our framework distributes tasks across agents that can reason, self-correct, and collaborate dynamically *via* agent-to-agent communication through a shared memory and execution interface. This enables our system to respond autonomously to simulation errors, adapt tool selection based on material context, and incorporate human guidance for learning and explainability. To demonstrate the capabilities of this agentic platform, we apply it to several distinct simulation tasks of increasing complexity, from static properties such as lattice parameter and cohesive energy calculation, elastic constant computation and extraction, and phonon dispersion calculation, to dynamical properties such as melting point estimation for representative elemental and binary alloy structures that span across different crystal systems (Fig. 1).

2. Methods

2.1 Agentic architecture

The three core components of the system are: (i) the agent factory, where all agents are instantiated and assigned an LLM

configuration along with a system prompt that defines their roles and available tools, (ii) a suite of specialized tools, which includes functions for materials property calculations, simulation input generation and file handling and (iii) a function registry, which maps specific functions to the appropriate agents (see SI Section S1 and Fig. S1). The agentic infrastructure builds upon the AutoGen framework (AG2), which enables dynamic agent selection, supports human-in-the-loop feedback, and allows agents to collaborate adaptively to complete complex tasks.²⁹ Each agent is designed with a narrow, expert-level task definition, enabling more focused reasoning and minimizing prompt complexity. The full list of agents used in this work and their capabilities is provided in Table 1.

The agents were assigned with several domain-specific software libraries through wrapper functions exposed in the function registry. Table 2 lists the main software tools and the agents they were assigned to.

2.2 LAMMPS simulation workflows

We utilize LAMMPS, a widely adopted open-source molecular dynamics (MD) simulation package, for modeling and analyzing materials at the atomic scale.⁸ Its flexibility and extensibility make it well-suited for simulating a broad range of materials phenomena, from lattice dynamics and defect behavior to phase transitions and interfacial properties. LAMMPS is typically used to compute a range of material properties and dynamics. Lattice constants and cohesive energies are used to predict phase stability and guide alloy or semiconductor design. Elastic constants inform mechanical reliability of materials used in structural, aerospace, and microelectronic applications. Phonon dispersion of materials underpins predictions of thermal conductivity, thermoelectric performance, and superconductivity. Together, calculation of these properties by users is often aimed at bridging atomistic

Table 1 Overview of the agents designed for the multi-agent MD system. Each agent has a focused, expert-level task and equipped with access to specialized functions. The table summarizes the main responsibilities of each agent within the system

Agent	Description
Structure creation agent	Responsible for creating the appropriate structure file
LAMMPS input agent	Responsible for the generation of the LAMMPS input file tailored to the property prediction requested from the user
LAMMPS input reviewer agent	Responsible to review the LAMMPS input files when an execution error appears and provide feedback and corrections
Potential agent	Responsible for identifying the correct potential files and checking their validity. The agent can download files from existing known resources. Collaborates with the WebScraper agent in case Google search is required to find the potential file
Web scraper agent	Data extraction (e.g., CIFs, potential files, thermophysical constants). Web scraper agent enhanced to: fetch known lattice parameters, elastic constants, and empirical potentials. Validate against computed values
Melting point calculation agent	Responsible for handling calculations for melting point estimation
Elastic constants agent	Responsible for handling calculations for elastic constants calculation
Phonopy agent	Responsible for handling calculations for phonon dispersion calculation
Results analysis agent	With integrated vision: responsible for analysis the results of the simulations by checking the output files
HPC agent	Responsible for working on an HPC cluster (carbon cluster)
Administrator with integrated executor	Coordinates the interactions between the human user and the agents. Executes the functions that are associated with the agents



Table 2 The main packages used in this work with their description and the agent they are assigned to

Tool/package	Description	Agent
Atomsk	Generates crystal structures with defined lattices, defects or supercells	Structure creation agent
LAMMPS	Molecular dynamics code used for all simulations	LAMMPS input agent
Phonopy	Computes phonon band structures using finite displacements	Results analyzer agent
OVITO	Parses and visualizes simulation trajectories and outputs	Results analyzer agent
Playwright	Webscraping and file downloading from online sources with potentials, like NIST	Web scraper agent
ASE	Atomistic structure manipulation and file conversion	Phonopy agent
Custom tools	Includes HPC management scripts and file converters	HPC agent, admin agent

simulations with practical materials design and engineering. In our agentic pipeline, each property calculation using LAMMPS requires a different set of agents and workflow routines. At the core of each property calculation is the automated generation and execution of a LAMMPS input script tailored to the task and the selection of the appropriate force field file. Depending on the task, the LAMMPS input agent can use pre-defined templates and dynamically inserted parameters such as time-step, ensemble type, temperature, pressure, and dump frequency or create and save a new input file. Note that we initiate the pipeline *via* generation of a crystal structure, the selection of the appropriate potential file and the creation of a LAMMPS input script. The associated main routines for each of the simulations performed are described below.

2.2.1 Lattice parameters and cohesive energy calculations.

To calculate lattice parameters and cohesive energy using LAMMPS, a user would perform a series of energy minimizations across different lattice constants and identify the value that yields the lowest potential energy, which gives the equilibrium lattice parameter. Cohesive energy is then determined by subtracting the energy per atom in the bulk solid from the energy of an isolated atom. The bulk energy is obtained from a relaxed crystal structure, while the isolated atom energy is calculated by placing a single atom in a large vacuum box to eliminate interactions. For many interatomic potentials like EAM, the isolated atom energy is effectively zero, so cohesive energy equals the negative of the bulk energy per atom. Calculating lattice parameters and cohesive energy involves an input file that defines the simulation cell and atomic structure (generated *via* Atomsk in this work). The pair_style and pair_coeff commands specify the interatomic potential (e.g., EAM for metals), while minimize performs energy minimization to relax the structure. For cohesive energy, one script is used to compute the energy per atom in the bulk, and a separate script places a single atom in a large box (vacuum) to compute its isolated energy. Output quantities like potential energy are accessed using the appropriate keywords in the thermo_style parameter for appending the LAMMPS output file with the dimensions of the cell of the relaxed structure. By looping over lattice constants and analyzing the energy trends, users can extract both equilibrium lattice constants and cohesive energy.

2.2.2 Elastic constants computation and extraction.

This includes the creation of the starting crystal structure file, selection of the appropriate potential, relaxation of the structure using the appropriate LAMMPS input, and the use of

predefined LAMMPS template files for the elastic constants simulation. The template files are 'in.elastic', 'potential.mod', 'displace.mod', 'init.mod'. After the template files are copied to the working directory, the 'init.mod' requires modifications, such that the correct name of the structure file will be added and also several values for the 'variable up equal' parameter should be selected with possible values of 0.0001, 0.001, 0.01. The 'potential.mod' file should also be updated with the correct potential. The expected output of this simulation is a 6×6 matrix *i.e.* C_{ij} with the elastic constants, where C_{11} , C_{22} , C_{33} represent the longitudinal elastic moduli and the C_{44} , C_{55} , C_{66} the shear elastic moduli. The off-diagonal terms represent the coupling between stress and strain components. To ensure a reliable simulation, at least two different values of the 'variable up equal' should be tested and the resulting tables should have comparable values.

2.2.3 Phonon dispersion calculation.

Phonon dispersion calculations in LAMMPS involve analyzing the vibrational properties of a crystal by constructing and diagonalizing the dynamical matrix, typically using the finite displacement method. This is often done in conjunction with external packages like Phonopy, which interfaces with LAMMPS to generate displaced structures, compute forces, and extract phonon frequencies and dispersion relations across the Brillouin zone. Phonon dispersion calculation includes the initial creation of the structure with a unit cell above $3 \times 3 \times 3$, downloading the correct potential file, run the initial relaxation and then prepare for the Phonopy workflow. Phonopy calculation steps involve converting the structure file to POSCAR format, generating supercells and displacement files after applying modifications to the starting structure. A force calculation LAMMPS simulation is performed for each of the displacement structures and then all the forces are compiled, and a FORCE_SETS file is generated for band structure analysis.

2.2.4 Melting point estimation.

Unlike static material properties such as elastic or lattice constants, the melting point is significantly more challenging to compute, both numerically and computationally.³¹ The method to determine the melting point at constant pressure is by finding the temperature at which the Gibbs energy of the solid and liquid phases are equal. In LAMMPS, melting point estimation is typically performed by gradually heating a solid system under canonical ensemble (constant number of particles N , volume V , temperature T) or isothermal-isobaric ensemble (constant number of particles N , pressure P , temperature T), while monitoring structural and



thermodynamic indicators such as abrupt changes in volume, potential energy, or radial distribution functions.³² This approach provides insight into the thermal stability of materials and serves as a critical validation metric for interatomic potentials, ensuring accurate reproduction of phase transitions and high-temperature behavior.

We employed the interface method (coexistence approach) for melting point estimation,^{33,34} which involves simulating a system containing both crystalline and disordered phases within the same periodic simulation box, separated by a well-defined interface. The high-level procedure for the melting point estimation is a multi-step process that involves three main steps: (i) generation and relaxation of the crystalline structure generated by Atomsk, (ii) creation of the solid–liquid interface and (iii) melting point simulation *via* heating of the system created in step (ii). The first (i) part is the standardized protocol followed previously for the structure relaxation. The second (ii) part includes the identification of the top half of the structure using box-bound variables (*e.g.*, `bound(all, zmax)` and `bound(all, zmin)`), so the region definition is independent of the system size. Then this top half structure should be frozen using `fix setforce 0.0 0.0 0.0` to prevent atomic motion in that region. Then the initial velocities to the bottom half using should be assigned and apply a high temperature thermostat (*e.g.*, 3000 K) using canonical ensemble (*NVT*) to melt that part. The simulation should run for sufficient time (*e.g.*, 50 ps or longer) to allow a solid–liquid interface to form naturally.³⁵ The resulting structure of the solid–liquid interface is saved for use in the melting point simulation. An OVITO visualization is then generated by saving the front view of the last frame of the melting point simulation file and visually verify the two phases (solid/liquid). If the structure is half melted, we can proceed to the melting point estimation. Otherwise, a new LAMMPS input file should be generated for rerunning the interface simulation by applying higher temperature or for longer time. Once a 50 : 50 solid–liquid ratio is visually confirmed, the final step (iii) involves a heating simulation starting from room temperature and ramping up beyond and expected melting point (*e.g.* by 1000 K). The system is monitored until full melting is observed. This marks the upper bound of the melting point for the given interatomic potential.

2.3 Error detection and autocorrection routines

A core ability of our agentic system is the ability to detect, explain, and correct execution failures. The LAMMPS input creator first calls an integrated function to check the workflow status for the existence of a valid potential and a valid structure file at the working directory. Only if the workflow status result is true, the agent will proceed to create the LAMMPS input file. Furthermore, the LAMMPS input reviewer agent monitors the output logs from each simulation and parses for known failure signatures, including invalid commands (*e.g.*, misnamed keywords or deprecated syntax), numerical instabilities (*e.g.*, atom loss, box deformation) and missing potential files or incorrect atom types. Upon detection, the reviewer agent collaborates with the relevant task agent (*e.g.*, potential agent,

LAMMPS input agent) to regenerate or fix the input scripts. This autocorrection loop is repeated until the simulation is executed without any errors. In cases of failure, the user has also the option to interact with the administrator agent and suggest improvements or provide guidance.

2.4 High-performance computing execution

The simulations were executed on the carbon HPC cluster located within the Center for Nanoscale Materials at Argonne (https://wiki.anl.gov/cnm/HPC/Network_Access) or using a local installation of LAMMPS software (version 29 Aug 2024 – update 3). Carbon HPC uses a torque-based job scheduler. The HPC agent is responsible for packaging simulation files (structure, potential, input scripts), uploading *via* secure copy protocol (SCP), submitting jobs, monitoring job completion, and downloading results and notifying downstream agents for analysis.

2.5 Human expert evaluation

In all case study simulations, we validated the capabilities of the agentic pipeline by comparing all the results with human experts. For consistent comparison and due to the deterministic nature of the LAMMPS simulations, the human expert evaluation used the same initial structure and same potential files as those generated by the structure agent and the potential agent respectively. Note that the choice of interatomic potential exerts the largest influence on all computed physical properties, often leading to significant variability in lattice constants, cohesive energies, and elastic moduli across different parameterizations. Such variability stems from the intrinsic limitations and transferability of the potential itself, rather than from the agent's operation. Consequently, the agent should not be held responsible for errors originating from the underlying potential choice. These decisions inherently require human scientific oversight, as the selection of a physically appropriate potential remains a knowledge-driven step that depends on system chemistry, training data quality, and intended property predictions. The % error was then measured by the following equation:

$$\% \text{Error} = \text{abs} \left(\frac{A - H}{H} \right) \times 100\%$$

where *A* is agent and *H* is human. The human output is treated as the correct value (general truth).

3. Results and discussion

The agentic system was extensively tested for static and dynamic analysis pipelines using a representative set of structures and crystal systems, *i.e.*, gold (Au),³⁶ iron (Fe),³⁷ titanium (Ti),³⁸ silicon (Si),³⁹ nickel–copper alloy (Ni–Cu), gold–copper alloy (Au–Cu).⁴⁰ Static simulations typically aim to find the minimum energy configuration of a system or properties of a system at 0 kelvin (or a specific, non-evolving atomic configuration). LAMMPS commands usually involve functions to relax structures or to compute properties of a given state without



advancing time, whereas the dynamic properties are obtained by integrating the equations of motion over finite time steps. In this case, simulations capture the temporal evolution of atomic trajectories, enabling the calculation of temperature-dependent quantities such as thermal conductivity, or structural phase transitions. Together, these complementary modes of operation allowed us to benchmark the system across both equilibrium property calculations and time-dependent processes, ensuring robustness and generality of the pipelines.

3.1. Static property simulations

3.1.1 Lattice constants calculation. The initial task includes calculation of the lattice parameters and cohesive energy for the selected systems (Fig. 2). The expected output is a list of 6 numbers that represent the length along different unit cell axes and the angles between them. A detailed system message was provided to the LAMMPS input creator for enabling the creation of the correct input files to perform these calculations (Fig. S3). The agentic system initially created the structure using the structure agent, then the potential agent identified and downloaded the correct potential file from the web, and LAMMPS input agent created the input file with the appropriate thermo keywords. The files were uploaded to HPC through the HPC agent. Once the simulation was successfully

completed, all the files were downloaded and the log.lammps file was read by the results analysis agent to extract the required quantities and provide a response to the user (Fig. S4).

To ensure transferability to several crystal systems, we further tested the framework using an extensive selection of crystal systems, *i.e.*, face centered cubic (FCC), BCC, HPC. The system achieved average errors below $x\%$ compared to the results of human LAMMPS simulation experts. Comparison of all agents *vs.* human results are demonstrated in Tables 3, 4 and S3.

3.1.2 Elastic constants calculation. The high-level pipeline for performing an elastic constants calculation with the agentic system is described in Fig. 3a. The LAMMPS elastic agent plays a key role in this process and is provided with detailed prompting for retrieving/modifying the appropriate files (Fig. S5). In the case of an FCC gold crystal, the crystal structure was generated with a lattice parameter of 4.078 Å in a $5 \times 5 \times 5$ supercell configuration containing 500 atoms. The interatomic interactions were described using the embedded atom method (EAM) potential (Au_u3.eam) obtained from the LAMMPS repository. The elastic constants calculation utilized a finite deformation approach, where small strain perturbations ($\delta = 0.001$) were systematically applied to the simulation cell in different directions, followed by energy minimization using the conjugate gradient method with convergence criteria of

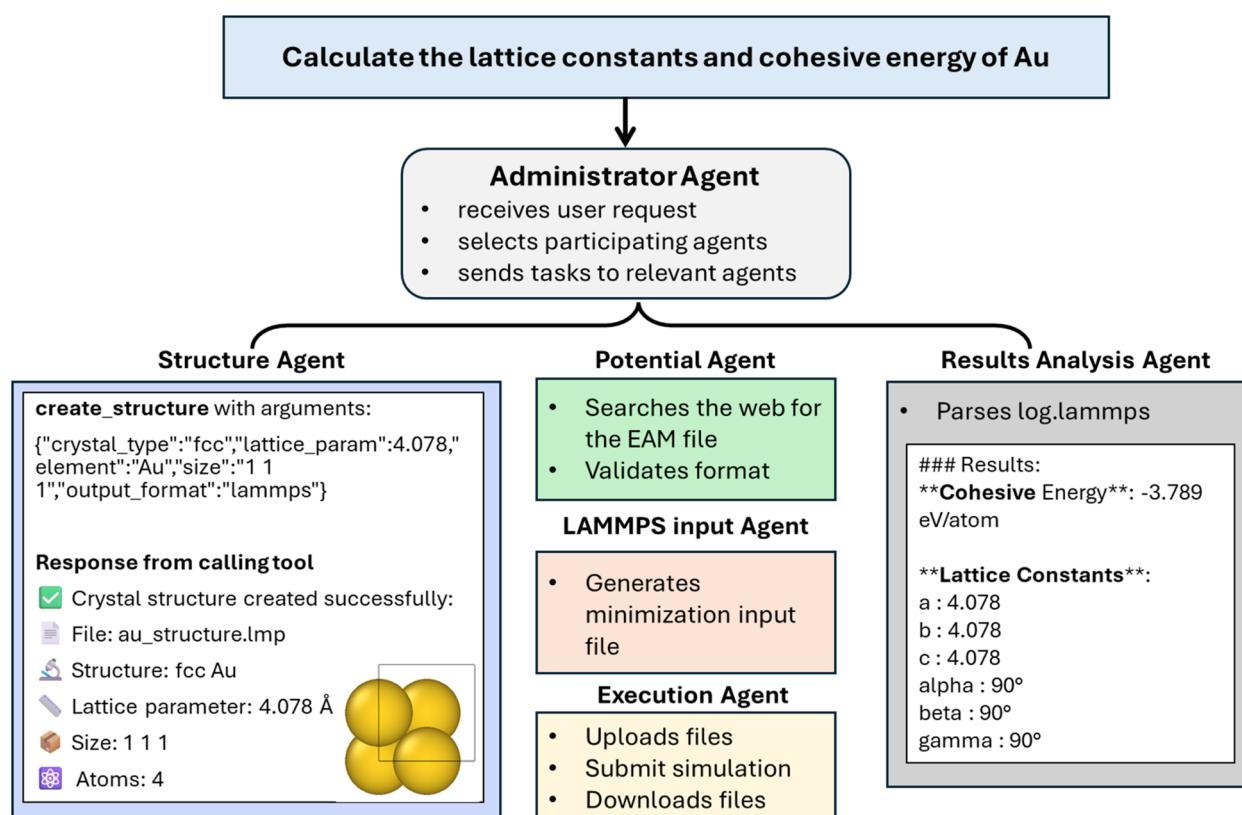


Fig. 2 An example of a typical agentic workflow for the calculation of the lattice constants and cohesive energy for gold (Au) using MD simulations. The administrator agent coordinates the task, while specialized agents generate the crystal structure, locate and validate the EAM potential, prepare and submit the LAMMPS input, and parse the simulation output from HPC. The results yield a stable FCC Au structure with lattice constant 4.078 Å and cohesive energy -3.789 eV per atom which is consistent with predictions by expert human users.

Table 3 Comparison between the results obtained from a human vs. an agentic pipeline on the calculated values of lattice constants for various elemental and alloy systems

System	Lattice parameters – agent	Lattice parameters – human
Fe (BCC)	$a = b = c = 2.866 \text{ \AA}$ $\alpha = \beta = \gamma = 90^\circ$	$a = b = c = 2.866 \text{ \AA}$ $\alpha = \beta = \gamma = 90^\circ$
Au (FCC)	$a = b = c = 4.078 \text{ \AA}$ $\alpha = \beta = \gamma = 90^\circ$	$a = b = c = 4.078 \text{ \AA}$ $\alpha = \beta = \gamma = 90^\circ$
Ni–Cu (FCC)	$a = b = c = 3.58177 \text{ \AA}$ $\alpha = \beta = \gamma = 90^\circ$	$a = b = c = 3.58177 \text{ \AA}$ $\alpha = \beta = \gamma = 90^\circ$
Ti (HPC)	$a = 2.96357 \text{ \AA}$ $b = 2.5665 \text{ \AA}$ $c = 4.7034 \text{ \AA}$ $\alpha = \beta = 90^\circ$ $\gamma = 120^\circ$	$a = 2.96357 \text{ \AA}$ $b = 2.5665 \text{ \AA}$ $c = 4.7034 \text{ \AA}$ $\alpha = \beta = 90^\circ$ $\gamma = 120^\circ$
Si–diamond	$a = b = c = 5.43094 \text{ \AA}$ $\alpha = \beta = \gamma = 90^\circ$	$a = b = c = 5.43094 \text{ \AA}$ $\alpha = \beta = \gamma = 90^\circ$
Au–Cu (FCC)	$a = b = c = 3.8465 \text{ \AA}$ $\alpha = \beta = \gamma = 90^\circ$	$a = b = c = 3.8388 \text{ \AA}$ $\alpha = \beta = \gamma = 90^\circ$

Table 4 Comparison between the results obtained from a human vs. an agentic pipeline on the calculated values for the cohesive energy for various elemental and alloy systems. The percentage of error (% error) measures the deviation of the agent relative to the human expert which we treat as the ground truth

System	Cohesive energy (eV per atom) – agent	Cohesive energy (eV per atom) – human	% Error
Fe (BCC)	–4.3159	–4.3159	0%
Au (FCC)	–3.789	–3.789	0%
Ni–Cu (FCC)	–3.9682	–3.9682	0%
Ti (HPC)	–4.8525	–4.8525	0%
Si–diamond	–4.3366	–4.3366	0%
Au–Cu (FCC)	–3.8239	–3.8246	0.018%

$10^{-10} \text{ eV \AA}^{-1}$ for forces. The full 6×6 elastic constant tensor was extracted from the stress–strain relationships, yielding the key elastic properties: bulk modulus of 166.89 GPa, primary shear modulus of 44.73 GPa, and Poisson ratio of 0.46. The computed elastic matrix was validated for physical consistency through symmetry checks and Born stability criteria, and the results were visualized through comprehensive plots showing both the elastic constant tensor and convergence behavior during the calculations. The elastic constants simulations part involves the modification of the variable to generate a 6×6 matrix. The generated elastic constant matrices are shown in Fig. 3b and S6 confirms that the agentic system has correctly calculated the stiffness tensor for gold in GPa. A table summarizing the elastic constants calculation of both agent and human expert for all six atomic systems (Fe, Au, Au–Cu Ni–Cu, Ti, Si) is provided in Table S4. The low % error for each of the systems shows that the agentic pipeline can perform these simulations at a human level.

3.1.3 Phonon dispersion. In this workflow, the user initiated a request to calculate the phonon dispersion of an element or alloy using an automated LAMMPS-Phonopy pipeline (Fig. 4a

and S7). For the gold example, the agent began by generating an FCC crystal structure of gold with a $2 \times 2 \times 2$ supercell and a lattice parameter of 4.078 \AA . An appropriate EAM potential (Au_u3.eam) was fetched from an online repository after initial download attempts failed. A LAMMPS input script was created to relax the gold structure, but the simulation initially failed due to unrecognized or invalid commands (after read pause 0, then after minimize). These errors were progressively corrected, leading to successful relaxation and generation of a relaxed structure. The relaxed structure was then used to generate a POSCAR and displacement files required for phonon calculations. After uploading all 192 displacement directories to the HPC system and running them in batch, the forces were collected, and phonon band structure data was generated. Finally, the phonon dispersion was successfully computed and visualized in a band.pdf file along the $\Gamma \rightarrow X \rightarrow W \rightarrow K \rightarrow \Gamma \rightarrow L \rightarrow U$ path for the cubic gold system, completing the workflow (Fig. 4b). In Fig. 4c, the same process was performed for the nickel–copper alloy. The agenting approach was tested for the exemplary systems with all the results and comparison with the human derived plots shown in the Fig. S8–S13.

3.2. Dynamical property simulations

The scope of our study was extended beyond static property simulations by exploring the thermodynamic properties for the exemplary materials systems. Melting properties play a key role for the in-depth understanding of a material as they affect the synthesis method, processing and performance in various areas of application.⁴¹ We evaluated the agentic system's ability to perform time-dependent molecular dynamics (MD) simulations to explore melting behavior using automated orchestration of the process. A critical part for the understanding of the process is the visual inspection of the structures during the melting, which is performed by human experts that have an understanding on how a melted structure should look like. In this work we tried to map the human perception of an image with the use of advanced reasoning multimodal models that can not only look at the generated image, but also take critical decisions during the simulation and propose ways forward based in the observations (SI Section S1.4 and Fig. S2). These workflows require agents to operate across long MD trajectories, manage temperature control, and extract properties like melting point and heat capacity from post-processed data.

The main corpus of the agentic simulation consists of three sequential MD simulations, (i) generation of an energy minimum configuration, (ii) solid–liquid interfacial pre-melting, (iii) full melting (Fig. S13). The term pre-melting refers to the formation of thermodynamically stable liquid films at solid interfaces subjected to temperatures below but near the bulk melting temperature (T_m).⁴² The role of the visual inspection is particularly important for parts ii and iii, as the vision model should verify the successful completion of these tasks or propose changes in the MD workflow otherwise until the final goal of each stage is reached (Fig. 5a). After each of the main steps, an image of the final frame of the completed simulation is given to the vision agent for inspection. An



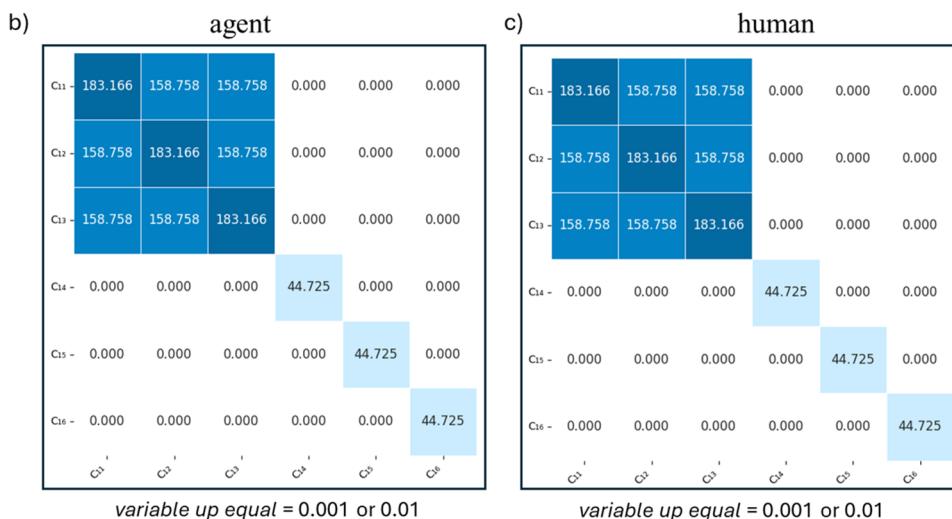
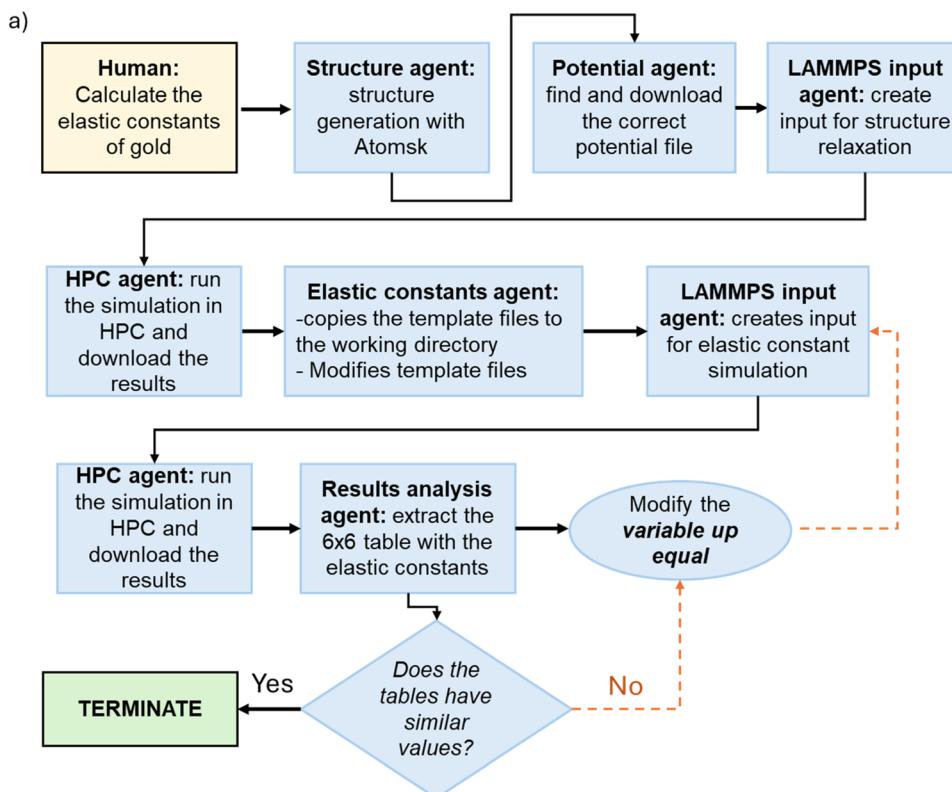


Fig. 3 Agent vs. human calculated elastic constants simulation for FCC gold. (a) Schematic of the agentic workflow for generating the elastic constants matrix for gold. The structure agent generated a gold (Au) crystal in a $5 \times 5 \times 5$ supercell configuration, the potential agent selected an EAM file, and the elastic constants agent applied six independent strain deformations. All results are in GPa. Since both matrices are exactly the same, the agent concludes that the simulation is successful and can be terminated without the need to apply different variable up equal values. (c) Elastic constants matrix computed by a human expert.

exemplary structural evolution of an AuCu alloy during the melting point simulation performed by the agent is shown in Fig. 5b. Once the simulation is completed successfully, the melting characteristics can be concluded following the changes in potential energy and heat capacity as functions of temperature (Fig. 5c). Similar analysis and results for all the explored systems is provided in the SI Fig. S15 and S18–S22.

The key component of the process is the integration of the results analyzer agent with two vision models which are assigned with different observatory tasks and are based on o3 vision model. The vision models are called to operate every time a new OVITO frame is saved in the working directory during the melting point calculation (Fig. S14). The first vision model is assigned the task of observing the solid–liquid interface frame



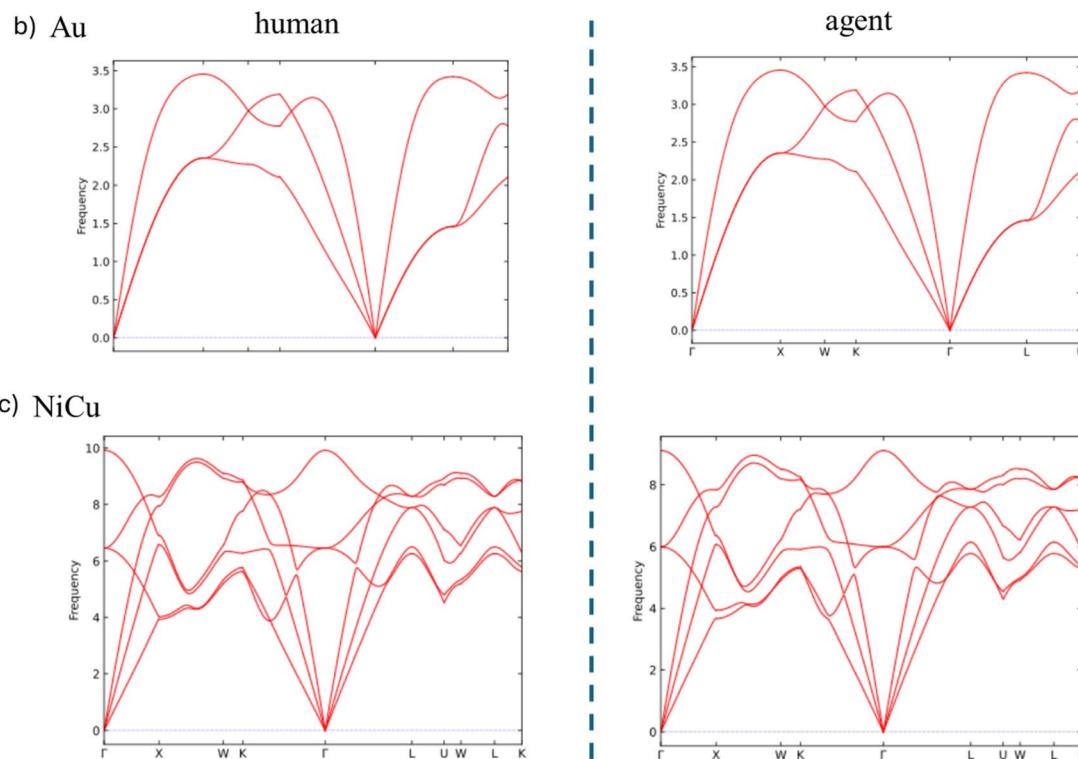
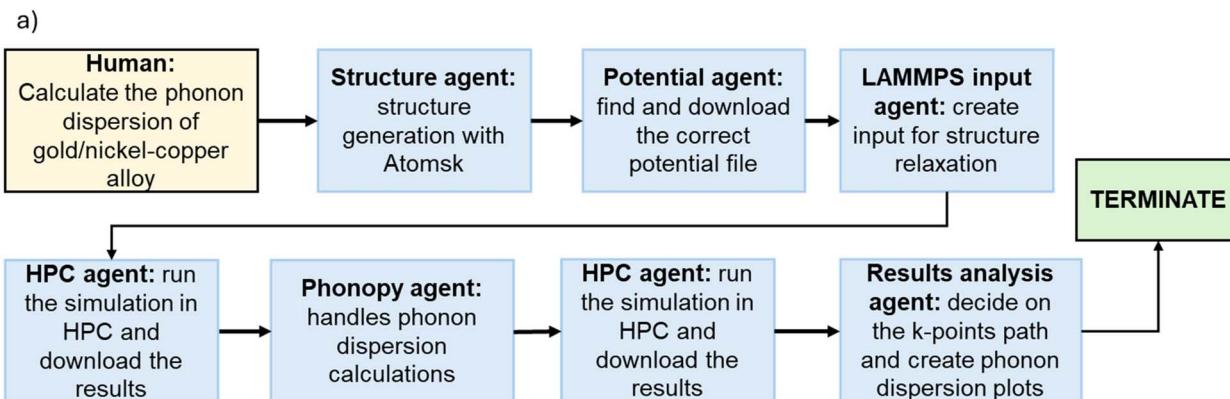


Fig. 4 Agent vs. human calculated phonon dispersion for Au and NiCu alloy. (a) Schematic of the steps performed from the agents to execute a phonon dispersion simulation following a user prompt. (b) Phonon dispersion plots generated by human vs. agent for Au. (c) Phonon dispersion plots generated by human vs. agent for NiCu. The human and agents generated results were identical.

and critically decide whether the composition of the solid liquid phase is around the intended 50 : 50 ratio. If that is not the case, then the LAMMPS simulation is resubmitted after fixing or unfixing the atoms in the structure accordingly. The second vision agent is responsible for understanding if a structure has fully melted or not. Conclusions are made by checking whether the distribution of the atomic positions appear to be random, the absence of lattice structures, non-aligned packing and disordered fluid-like appearance. If the structure is not fully melted the agent recommends parameter adjustments, such as increasing the temperature range, extending the simulation time by increasing the timesteps or adjusting the heating rate for slower heating. The detailed prompts and benchmarking of the vision agents is provided in the SI Section S1.4, Fig. S2 and

S3. The results generated by the human expert are also described in the SI. The comparison between the human experts and agent is provided in the Table S5.

Briefly, the simulations are executed by our workflow on a high-performance computing (HPC) resource, where large-scale molecular dynamics runs provide statistically reliable sampling across a broad temperature range as shown in Fig. 5. As the NiCu system is progressively heated, the potential energy *vs.* temperature trajectory reveals the gradual destabilization of the ordered lattice, with a pronounced change in slope marking the onset of melting. To pinpoint the transition more rigorously, the heat capacity curve is computed from fluctuations in enthalpy, and it exhibits a sharp peak at approximately 1604 K, which corresponds to the alloy's melting point. Our agentic



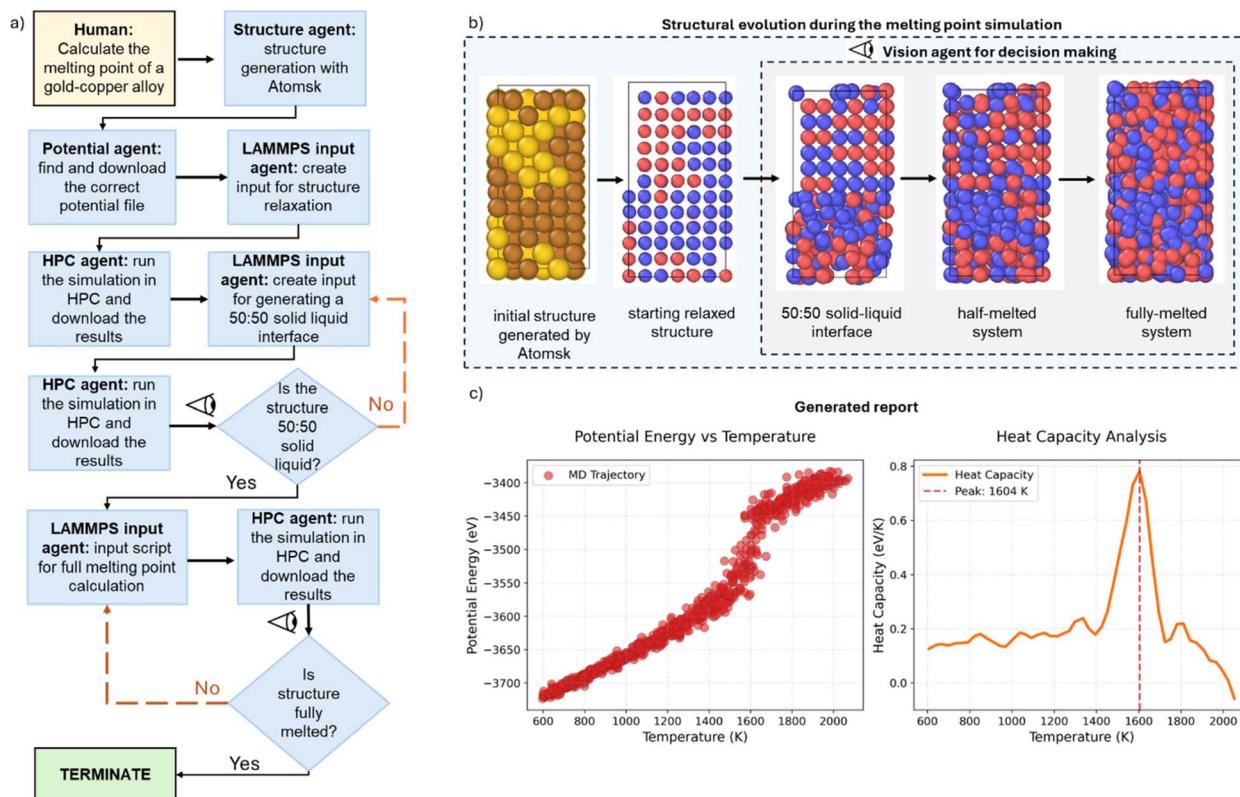


Fig. 5 Melting point predictions made by agentic workflow. (a) Agentic pipeline demonstrating the sequential steps for performing an end-to-end melting point simulation starting from a simple user prompt. When the process is terminated, a report is generated as shown in (c). (b) Progression of the melting simulation observed by the OVITO frames acquired after each simulation. The vision agent encodes these images to base 64 string and decides whether a simulation is complete or not. The main decisions the vision agent has to make is whether a 50 : 50 solid-liquid interface has been created and whether the structure is fully melted. (c) Melting characteristics were studied by following the changes in potential energy and heat capacity as functions of temperature. Report generated by the results analysis agent at the end of the whole process. The analysis plot displays a scatterplot with the potential energy versus temperature (left) and a plot with the heat capacity versus temperature (right).

workflow not only reproduces the expected thermodynamic signature of melting but also highlights how multi-agent orchestration—spanning structure preparation, simulation setup, execution, and analysis—can seamlessly integrate with physics-based MD to deliver accurate, reproducible predictions of key material properties. All the potential files identified by the potential agent and web scraper agent for each system and task are summarized in the Table S6.

3.3 Limitations and future work

While our multi-agent framework demonstrates substantial improvements in automation, modularity, and reproducibility for atomistic simulations, several limitations and opportunities for enhancement remain. First, dependency management and environment reproducibility, especially for tools like LAMMPS, Atoms, and Phonopy, can become brittle across platforms or HPC systems. We currently address this with curated environments, but robust containerization and environment self-checks will be important in future iterations.

Second, although the system is capable of autonomous decision-making and error recovery, the trustworthiness and explainability of some AI-driven actions, particularly LLM-based

reasoning, remain open challenges. Future work should incorporate transparent logging, agent-level confidence scores, and potentially symbolic reasoning layers to increase interpretability. Third, while our framework already scales to a broad class of static and dynamic materials properties, accuracy *vs.* automation trade-offs become critical for complex tasks such as phonon dispersion, thermal conductivity, or defect energetics. These simulations may benefit from adaptive sampling, uncertainty quantification, and self-correction loops that trigger convergence checks or higher-fidelity reruns when needed.

In addition, we note that there are already several agentic frameworks available in the community (*e.g.*, LangChain, AutoGen, CrewAI), and our approach is intentionally designed to be framework-agnostic. The predefined functions, system messages, and agent roles that we developed can be readily transferred or reconfigured within other pipelines, making the framework portable and easy to integrate with emerging ecosystems. Beyond this, opportunities exist to build standardized APIs for agent-simulation interaction, incorporate benchmarking datasets for cross-framework comparisons, and establish community-driven best practices for reproducibility, logging, and evaluation. Such steps will help ensure that agentic

AI for materials science evolves not only as a powerful tool for automation but also as a sustainable, extensible infrastructure that supports long-term collaboration between computational scientists, AI developers, and experimentalists.

Looking forward, we envision integrating the framework with HPC job schedulers for intelligent resource management and embedding it within closed-loop inverse design pipelines that use reinforcement learning or Bayesian optimization. The multi-agent design, with its modular and extensible structure, offers a natural pathway to scale toward increasingly complex materials discovery tasks and tighter integration with experimental or autonomous laboratory workflows.

4. Conclusion

We investigate the application of generative approaches to materials science, and for automating MD simulations. In particular, we introduced a multi-agent AI framework that automates the end-to-end pipeline of atomistic simulations, from structure generation to property analysis. By coordinating specialized agents equipped with domain-specific tools, our system successfully reproduced both static properties (lattice constants, cohesive energies, elastic constants, phonon dispersion) and dynamic properties (melting points) across a diverse set of elemental and alloy systems with accuracy comparable to human experts. Our results highlight that multi-agent orchestration is not only capable of reproducing core methodologies in LAMMPS but also ensures reproducibility, modularity, and scalability across different workflows.

Our agentic AI pipeline, which is a coordination of multiple agents with diverse tasks, presents our multi-agent system for automating the atomistic simulations and the running of scientific workflows with LLMs. If we want computational tools to be widely used and help us accelerate the discovery, they should be easy to use from a broader audience – for example, even human experimentalists with little computational background should be able to describe their problem and the agents help them develop solutions and understand their materials. The agentic pipeline could be also combined with autonomous laboratories and enable physics informed pipeline design. We used the GPT-03 model as the primary vision agent, as it consistently yielded the most accurate image-based reasoning results. However, we also benchmarked GPT-4o and the Qwen3-VL-8B-Thinking model under identical settings. While both GPT-4o and Qwen showed slightly lower accuracy for complex atomic structures, they remain reliable and consistent, making them a promising and accessible alternative for future multi-modal agentic workflows. We demonstrated the utility of our agentic workflow by running both static and dynamic simulations to capture the core methodology used in LAMMPS. Our multi-agentic system showed that it can effectively work when looking for a time-averaged property or a time-evolving process (dynamic workflow) or when looking for the lowest energy state or the properties of a fixed atomic arrangement (static). At the end of the day, we want to design systems that learn and are adaptable. Although we have demonstrated our workflow for inorganic alloys and materials, our framework is easily

customizable and can potentially serve a wide range of simulation tasks for soft materials as well. Our future work includes adding more capabilities to the agentic system such as connecting Bayesian optimization interfaces for parameter tuning or adaptive sampling. Finally, tighter integration with HPC job schedulers and autonomous laboratories will transform the framework into a powerful “digital twin” engine for closed-loop discovery, bridging AI-driven simulations with experimental validation in a reproducible and scalable manner.

Conflicts of interest

There are no conflicts of interest to declare.

Data availability

Code availability: all source data and code associated with this work, including representative examples of the outputs from the agentic simulations, are publicly available on GitHub at <https://github.com/ANL-NST/LAMMPS-Agents> and also archived in Zenodo under the DOI: <https://doi.org/10.5281/zenodo.17849607>.

The complete set of simulation data generated by the agent, together with the chat logs of the full workflow and the outputs of the human expert simulations, are available *via* Zenodo <https://doi.org/10.5281/zenodo.17849459>.

Supplementary information (SI) is available. See DOI: <https://doi.org/10.1039/d5dd00435g>.

Acknowledgements

This is based upon work supported by the US Department of Energy, Office of Science, Office of Basic Energy Sciences Data, Artificial Intelligence, and Machine Learning at DOE Scientific User Facilities program under Award Number 34532 (Digital Twins). Work was performed at the Center for Nanoscale Materials and Advanced Photon Source, both US Department of Energy Office of Science User Facilities, supported by the US DOE, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a US Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231. This research used resources from the Argonne Leadership Computing Facility, a US DOE Office of Science user facility at Argonne National Laboratory, which is supported by the Office of Science of the US DOE under Contract No. DE-AC02-06CH11357.

References

- 1 J. D. Evans, in *Computer Simulation of Porous Materials: Current Approaches and Future Opportunities*, ed. K. Jelfs, The Royal Society of Chemistry, 2021.
- 2 A. Koneru, P. S. Dutta, A. Muhammed, H. Chan, K. Balasubramanian, S. Manna, K. Sasikumar, P. Darancet



and S. K. R. S. Sankaranarayanan, *Mater. Today Adv.*, 2025, **26**, 100583.

3 C. Chu-Jon, E. Martinez, A. A. Bertolazzo, A. Koneru, S. K. R. S. Sankaranarayanan, J. D. Rimer and V. Molinero, *J. Am. Chem. Soc.*, 2025, **147**, 20456–20465.

4 A. Koneru, R. Batra, S. Manna, T. D. Loeffler, H. Chan, M. Sternberg, A. Avarca, H. Singh, M. J. Cherukara and S. K. R. S. Sankaranarayanan, *J. Phys. Chem. Lett.*, 2022, **13**, 1886–1893.

5 A. Gogoi, A. Koneru and K. A. Reddy, *Nanoscale Adv.*, 2019, **1**, 3023–3035.

6 M. Griebel, S. Knapek and G. Zumbusch, *Numerical Simulation in Molecular Dynamics: Numerics, Algorithms, Parallelization, Applications*, Springer Publishing Company Incorporated, 1st edn, 2007.

7 P. Hirel, *Comput. Phys. Commun.*, 2015, **197**, 212–219.

8 A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott and S. J. Plimpton, *Comput. Phys. Commun.*, 2022, **271**, 108171.

9 A. Togo, *J. Phys. Soc. Jpn.*, 2023, **92**(1), 012001.

10 A. Stukowski, *Modell. Simul. Mater. Sci. Eng.*, 2009, **18**, 015012.

11 Interatomic Potentials Repository.

12 S. Yang, S. Batzner, R. Gao, M. Aykol, A. Gaunt, B. Mc Morrow, D. Rezende, D. Schuurmans, I. Mordatch, E. Dogus and C. G. Deepmind, *Generative Hierarchical Materials Search*.

13 J. Kaiser, A. Lauscher and A. Eichler, *Sci. Adv.*, 2025, **11**, eadrl4173.

14 T. Song, M. Luo, X. Zhang, L. Chen, Y. Huang, J. Cao, Q. Zhu, D. Liu, B. Zhang, G. Zou, G. Zhang, F. Zhang, W. Shang, Y. Fu, J. Jiang and Y. Luo, *J. Am. Chem. Soc.*, 2025, **147**, 12534–12545.

15 S. Mathur, N. van der Vleuten, K. G. Yager and E. H. R. Tsai, *Mach. Learn. Sci. Technol.*, 2025, **6**, 2632–2153.

16 J. Gottweis, W.-H. Weng, A. Daryin, T. Tu, A. Palepu, P. Sirkovic, A. Myaskovsky, F. Weissenberger, K. Rong, R. Tanno, K. Saab, D. Popovici, J. Blum, F. Zhang, K. Chou, A. Hassidim, B. Gokturk, A. Vahdat, P. Kohli, Y. Matias, A. Carroll, K. Kulkarni, N. Tomasev, Y. Guan, V. Dhillon, E. D. Vaishnav, B. Lee, T. R. D. Costa, J. R. Penadés, G. Peltz, Y. Xu, A. Pawlosky, A. Karthikesalingam and V. Natarajan, *Towards an AI Co-Scientist*, 2025.

17 T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda and T. Scialom, Toolformer: Language Models Can Teach Themselves to Use Tools, *NIPS '23: Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023, pp. 68539–68551.

18 D. A. Boiko, R. MacKnight, B. Kline and G. Gomes, *Nature*, 2023, **624**, 570–578.

19 M. Muhoberac, A. Parikh, N. Vakharia, S. Virani, A. Radujevic, S. Wood, M. Verma, D. Metaxotos, J. Soundararajan, T. Masquelin, A. G. Godfre, S. Gardner, D. Rudnicki, S. Michael and G. Chopra, *State and Memory is All You Need for Robust and Reliable AI Agents*, 2024.

20 A. Vriza, M. H. Prince, T. Zhou, H. Chan and M. J. Cherukara, *Operating advanced scientific instruments with AI agents that learn on the job*, 2025.

21 A. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White and P. Schwaller, *Nat. Mach. Intell.*, 2024, **6**, 525–535.

22 M. H. Prince, H. Chan, A. Vriza, *et al.*, *NPJ Comput. Mater.*, 2024, **10**, 251.

23 *langchain-ai, LangGraph*.

24 Q. Campbell, S. Cox, J. Medina, B. Watterson and A. D. White, *MDCrow: Automating Molecular Dynamics Workflows with Large Language Models*, 2025.

25 O. A. Mendible-Barreto, M. Díaz-Maldonado, F. J. Carmona Esteva, J. E. Torres, U. M. Córdova-Figueroa and Y. J. Colón, *Mol. Syst. Des. Eng.*, 2025, **10**, 585–598.

26 Y. Zou, A. H. Cheng, A. Aldossary, J. Bai, S. X. Leong, J. A. Campos-Gonzalez-Angulo, C. Choi, C. T. Ser, G. Tom, A. Wang, Z. Zhang, I. Yakavets, H. Hao, C. Crebolder, V. Bernales and A. Aspuru-Guzik, *El Agente: An Autonomous Agent for Quantum Chemistry*, 2025.

27 Z. Wang, H. Huang, H. Zhao, C. Xu, S. Zhu, J. Janssen and V. Viswanathan, DREAMS: Density Functional Theory Based Research Engine for Agentic Materials Simulation, *arXiv*, 2025, preprint, arXiv:2507.14267, DOI: [10.48550/arXiv.2507.14267](https://doi.org/10.48550/arXiv.2507.14267).

28 B. Liu, X. Li, J. Zhang, J. Wang, T. He, S. Hong, H. Liu, S. Zhang, K. Song, K. Zhu, Y. Cheng, S. Wang, X. Wang, Y. Luo, H. Jin, P. Zhang, O. Liu, J. Chen, H. Zhang, Z. Yu, H. Shi, B. Li, D. Wu, F. Teng, X. Jia, J. Xu, J. Xiang, Y. Lin, T. Liu, T. Liu, Y. Su, H. Sun, G. Berseth, J. Nie, I. Foster, L. Ward, Q. Wu, Y. Gu, M. Zhuge, X. Liang, X. Tang, H. Wang, J. You, C. Wang, J. Pei, Q. Yang, X. Qi and C. Wu, *Advances and Challenges in Foundation Agents: From Brain-Inspired Intelligence to Evolutionary, Collaborative, and Safe Systems*, 2025.

29 Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, A. Awadallah, R. W. White, D. Burger, C. Wang, AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation, in, *COLM 2024*, 2024.

30 W. Chen, Y. Su, J. Zuo, C. Yang, C. Yuan, C.-M. Chan, H. Yu, Y. Lu, Y.-H. Hung, C. Qian, Y. Qin, X. Cong, R. Xie, Z. Liu, M. Sun and J. Zhou, AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors, in, *ICLR*, 2024.

31 L.-F. Zhu, J. Janssen, S. Ishibashi, F. Körmann, B. Grabowski and J. Neugebauer, *Comput. Mater. Sci.*, 2021, **187**, 110065.

32 M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, 2017.

33 J. R. Morris, C. Z. Wang, K. M. Ho and C. T. Chan, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1994, **49**, 3109–3115.

34 Y. Zhang and E. J. Maginn, *J. Chem. Phys.*, 2012, **136**, 144116.

35 A. Koneru, H. Chan, S. Manna, T. D. Loeffler, D. Dhabal, A. A. Bertolazzo, V. Molinero and S. K. R. S. Sankaranarayanan, *NPJ Comput. Mater.*, 2023, **9**, 125.

36 J. W. Arblaster, *J. Phase Equilib. Diffus.*, 2016, **37**, 229–245.



37 P. H. T. Philipsen and E. J. Baerends, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1996, **54**, 5326–5333.

38 W.-S. Ko, B. Grabowski and J. Neugebauer, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 2015, **92**, 134107.

39 H. Balamane, T. Halicioglu and W. A. Tiller, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1992, **46**, 2250–2279.

40 M. A. Turchanin and P. G. Agraval, *Powder Metall. Met. Ceram.*, 2008, **47**, 26–39.

41 S. K. R. S. Sankaranarayanan, V. R. Bhethanabotla and B. Joseph, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 2005, **71**(19), 195415.

42 Y. Yang, M. Asta and B. B. Laird, *Phys. Rev. Lett.*, 2013, **110**(9), 096102.