Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal

Cite this: DOI: 10.1039/d6dd00043f

Masgent: an AI-assisted materials simulation agent

Guangchen Liu, Songge Yang and Yu Zhong*

Density functional theory (DFT) and machine learning potentials (MLPs) are essential for predicting and understanding materials properties, yet preparing, executing, and analyzing these simulations typically requires extensive scripting, multi-step procedures, and significant high-performance computing (HPC) expertise. These challenges hinder reproducibility and slow down discovery. Here, we introduce *Masgent*, an AI-assisted materials simulation agent that unifies structure manipulation, automated VASP input generation, DFT workflow construction and analysis, fast MLP-based simulations, and lightweight machine learning (ML) utilities within a single platform. Powered by large language models (LLMs), Masgent enables researchers to perform complex simulation tasks through natural-language interaction, eliminating most manual scripting and reducing setup time from hours to seconds. By standardizing protocols and integrating advanced simulation and data-driven tools, Masgent lowers the barrier to performing state-of-the-art computational methodologies, enabling faster hypothesis testing, pre-screening, and exploratory research for both new and experienced practitioners.

Received 26th January 2026
Accepted 8th April 2026

DOI: 10.1039/d6dd00043f

rsc.li/digitaldiscovery

1. Introduction

Density functional theory (DFT) simulations and machine learning potentials (MLPs) have become indispensable tools in materials science for predicting and understanding materials properties, screening candidate compounds, and guiding experimental design.[1–9] Despite their widespread adoption, the practical implementation of these simulations remains challenging. Even standard workflows, such as structural relaxations, equation of state (EOS) evaluations, or elastic property calculations, require extensive scripting, careful input preparation, detailed knowledge of simulation software (*e.g.*, VASP[10,11]), and expertise in high-performance computing (HPC) environments. These complexities create barriers for new users, complicate reproducibility, and hinder the rapid prototyping and iterative exploration essential for modern materials research.

A broad ecosystem of tools has emerged to simplify portions of these workflows. Libraries like Pymatgen[12] and ASE[13] provide powerful APIs for structure manipulation and input generation, while tools like VASPkit,[14] qvasp,[15] and Atomate[16,17] support pre-processing, workflow construction, and post-processing of VASP calculations. Machine learning frameworks, including Scikit-learn,[18] TensorFlow,[19] and PyTorch,[20] enable data-driven modeling and property prediction, and large-scale materials databases such as the Materials Project[21] and Open Quantum Materials Database (OQMD)[22,23] provide curated structural and thermodynamic data. Recent advances in MLPs, such as SevenNet,[24] CHGNet,[25] Orb-v3,[26,27] MatterSim,[28] M3GNet,[29] MACE,[30] and Matlantis,[31] have further accelerated materials simulations by providing near-DFT accuracy at orders-of-magnitude lower cost. However, these tools remain largely fragmented, require significant manual integration, and often demand significant technical expertise to deploy effectively.

The rapid development of large language models (LLMs) and agentic AI systems[32–38] has enabled a new class of autonomous or semi-autonomous agents capable of planning and executing scientific workflows, including general-purpose infrastructures for AI-enabled scientific tool usage such as ToolUniverse[39] and Claude Scientific Skills.[40] In addition, several LLM-based agentic systems designed for computational chemistry and materials science have recently been reported, including DREAMS,[41] VASPilot,[42] GENIUS,[43] CatMaster,[44] QUASAR,[45] El Agente,[46] ChemGraph,[47] CACTUS,[48] CASCADE,[49] and Aitomia.[50] The field is evolving rapidly, with recent preprints such as El Agente Gráfico[51] and studies leveraging OpenClaw automating computational chemistry workflows *via* domain-specific skills[52] further expanding the design space of structured and skill-based agent architectures. As summarized in Table 1, these frameworks differ in several key capabilities. Most existing systems are designed for either periodic or molecular modeling tasks and primarily emphasize the automated execution on HPC resources. However, capabilities for automated structure generation or modification and reusable workflow templating remain limited in many frameworks. In addition, support for MLP-based simulations varies across systems, and the availability of integrated lightweight ML utilities for data analysis and model-assisted simulation workflows remains relatively limited.

To address these challenges, we introduce *Masgent*, an AI-assisted materials simulation agent designed to lower the

Mechanical and Materials Engineering Department, Worcester Polytechnic Institute, 100 Institute Rd, Worcester, MA, 01609, USA. E-mail: yzhong@wpi.edu

Table 1 Comparison of Masgent with existing LLM-based agentic systems. SE denotes semi-empirical method[a]

Framework	Primary domain	Structure manipulation	Workflow template	HPC execution	MLP simulations	ML utilities
DREAMS	DFT	No	Limited	Yes	No	No
VASPilot	DFT	Yes	Limited	Yes	No	No
GENIUS	DFT	Limited	No	Yes	No	No
CatMaster	DFT, MLP	Yes	Yes	Yes	Limited	No
QUASAR	DFT, MD, MLP	No	Limited	Yes	Limited	No
EI Agente	DFT, SE	Limited	No	Yes	No	No
ChemGraph	MD, MLP	Yes	No	No	Yes	No
Aitomia	DFT, SE	Limited	Limited	Yes	Yes	No
CASCADE	General	Limited	No	No	No	No
CACTUS	General	No	No	No	No	No
Masgent	DFT, MD, MLP, ML	Yes	Yes	No	Yes	Yes

[a] "Yes" indicates native or integrated support within the framework; "Limited" indicates partial or indirect support; "No" indicates that the capability is not explicitly supported based on the reported implementation.

barrier to performing advanced computational techniques. Masgent unifies DFT automation, fast MLP simulations, and lightweight ML utilities into a single, extensible framework. At its core is an interactive AI agent that enables natural-language interaction, allowing users to prepare simulations, generate inputs, and construct workflows without extensive scripting. The framework automates key tasks such as structure manipulation, defect generation, supercell construction, surface and interface creation, and complete VASP input preparation for different scenarios such as relaxation and static calculations. It further provides standardized workflows for common simulations, such as convergence tests, EOS fitting, elasticity evaluations, *ab initio* molecular dynamics (AIMD) simulations, and nudged elastic band (NEB) calculations, along with built-in analysis tools for extracting relevant properties. Masgent additionally incorporates high-performance MLP engines, including SevenNet, CHGNet, Orb-v3, and MatterSim, enabling rapid estimation of material properties at a fraction of the computational cost of traditional DFT. These capabilities enable near-instantaneous testing of hypotheses, pre-screening of candidate structures, and exploratory simulations that would otherwise be prohibitively expensive. Beyond physical-based simulations, Masgent includes lightweight ML utilities for data preparation, feature analysis, dimensionality reduction, data augmentation, hyperparameter tuning, model training and evaluation, and pre-trained model applications, bridging traditional simulations and emerging data-driven materials research.

By integrating these diverse functionalities into a coherent platform, Masgent significantly lowers the barrier to performing advanced materials simulations. Users can prepare, modify, and analyze computational workflows using conversational commands, eliminating much of the boilerplate scripting and manual file manipulation traditionally required. This design improves accessibility for students and newcomers while accelerating productivity for expert practitioners. The framework additionally promotes reproducibility through standardized input templates, validated tool calls, and structured workflow organization. In the following sections, the architecture and capabilities of Masgent are described in detail, demonstrating its utility through representative case studies, and discussing its broader impact on materials simulation and autonomous scientific computing.

2. Architecture and design

Masgent is designed as a modular, extensible, and AI-native framework that unifies DFT automation, MLP simulations, ML utilities, and natural-language interactions into a cohesive software ecosystem. The architecture follows a layered design that cleanly separates user interaction from computational backends, enabling flexible use across conversational sessions, scripted workflows, and high-throughput pipelines. The overall structure is illustrated in Fig. 1.

At the highest level, Masgent consists of three interconnected components:

(1) An AI-driven natural-language interface.

(2) A deterministic, menu-based command–line interface (CLI).

(3) A core utilities layer implementing all computational functionality.

This separation ensures both usability and reproducibility: users may interact with the system through natural language, while all underlying operations remain structured, validated, and fully deterministic.

2.1 AI mode (src/ai_mode)

The AI mode provides Masgent's natural-language interface, enabling users to request simulation tasks conversationally. The LLM interprets user instructions and maps them to specific tools through a structured tool-calling system. The LLM is guided by robust system prompts that regulate its behavior, ensuring reliability and preventing unsafe or ambiguous actions.

Each tool is associated with a Pydantic schema, which precisely defines required inputs, permissible values, and strict type constraints. This guarantees that all operations invoked through AI mode undergo rigorous validation before execution.

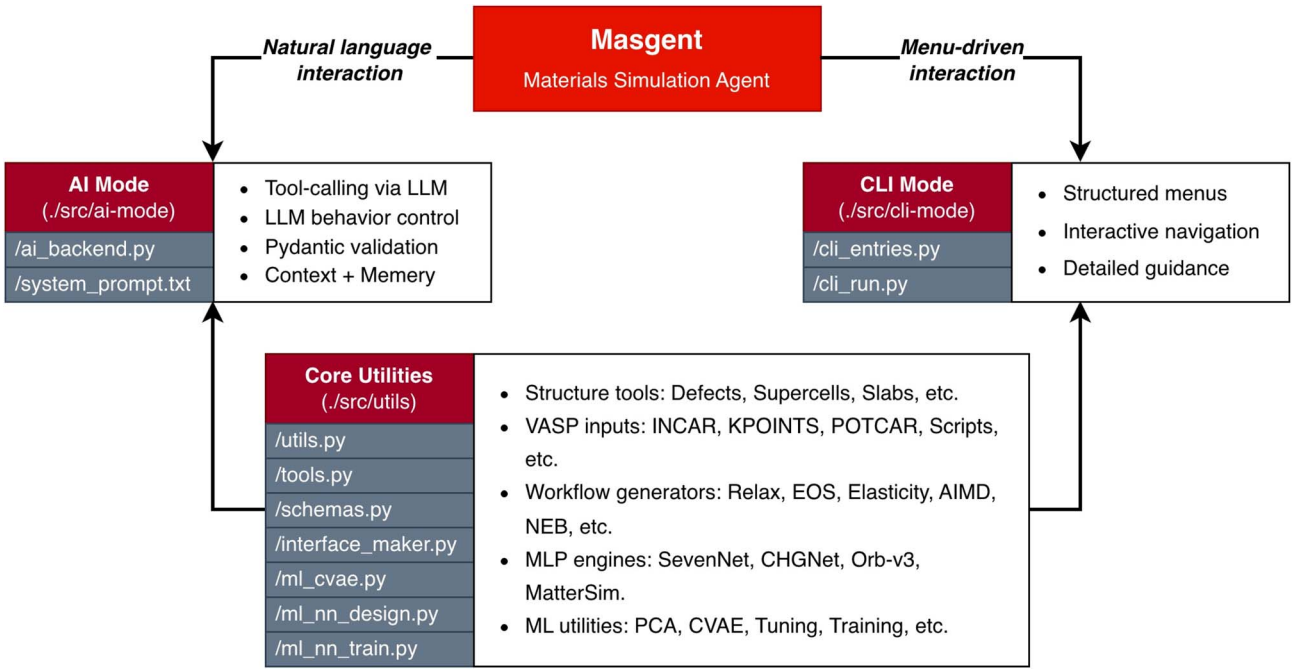

Fig. 1 Schematic overview of Masgent's modular architecture. The AI Mode and CLI Mode provide two user interfaces that both access the core utilities layer, which contains all computational tools for structure operations, VASP input generation, workflow automation, MLP simulations, and ML utilities.

The AI agent also maintains conversational context, allowing users to refine requests over multiple turns, ask follow-up questions, and receive intelligent feedback on planned operations or missing information.

2.2 CLI mode (src/cli_mode)

The CLI mode provides an alternative deterministic interface built around a menu-driven workflow. Using the Bullet library, users can navigate a structured set of commands to perform common tasks without writing scripts or interacting with the AI agent. This mode is particularly valuable for users who prefer a guided experience or when operating in environments where natural-language processing may be impractical.

The CLI uses the same core utilities as AI mode, ensuring that all interaction pathways produce consistent and reproducible outcomes. When Masgent is launched, the CLI serves as the default entry point, and the AI mode may be activated at any point during a session whenever more flexible or conversational interaction is desired.

2.3 Core utilities (src/utils)

The core utilities layer implements all substantive computations within Masgent and is designed to be modular, framework-agnostic, and callable from both the AI agent and the CLI. This layer includes:

- Structure manipulation tools, including vacancy and substitution defects, interstitial generation, supercell construction, Special Quasirandom Structure (SQS) generation, slab creation, and interface construction *via* lattice matching.
- Automated VASP input preparation, including INCAR, KPOINTS, POTCAR, and HPC job scripts based on standardized, best-practice templates.
- Workflow generators for structure relaxation, static calculations, EOS fitting, elasticity evaluation, AIMD simulations, and NEB calculations.
- Fast-simulation capabilities *via* unified interfaces to MLP engines such as SevenNet, CHGNet, Orb-v3, and MatterSim, enabling rapid energy evaluations, EOS, elasticity, and molecular dynamics (MD) simulations.
- Lightweight ML utilities, including feature analysis *via* principal component analysis (PCA),[53] data augmentation *via* conditional variational autoencoders (CVAEs),[54–56] hyperparameter optimization *via* Optuna,[57] model training and evaluation, and pre-trained model applications.

Together, these core utilities encapsulate all algorithmic logic, allowing the higher-level interfaces (AI mode and CLI mode) to remain thin, maintainable, and focused on user interaction rather than computation.

3. Features and capabilities

3.1 AI agent

Masgent incorporates an AI-native interface that allows users to perform complex material simulation tasks through natural-language interactions. The AI agent acts as a high-level orchestrator: it interprets user intents, translates natural-language instructions into well-defined computational operations, and invokes the appropriate tools within the Masgent framework to execute these tasks. This design removes the need

for extensive scripting or deep HPC expertise, enabling researchers to focus on conceptual and scientific questions rather than technical implementation details.

As shown in Fig. 2, the agent is built on a structured tool-calling architecture implemented with Pydantic AI. Incoming user messages are processed by the selected LLM backend (Table 2), which may be powered by OpenAI, Anthropic, Google, DeepSeek, or other supported providers. Masgent also provides its own managed LLM backend service that operates through a proxy layer, which communicates with external model providers (*e.g.*, OpenAI) while managing API credentials

Table 2 Overview of large language models (LLMs) supported by Masgent

Provider	Model	API key required
Masgent	Masgent AI	No
OpenAI	GPT-5 Nano	Yes
Anthropic	Claude Sonnet 4.5	Yes
Google	Gemini 2.5 Flash	Yes
xAI	Grok 4.1 Fast	Yes
DeepSeek	DeepSeek Chat	Yes
Alibaba	Qwen Flash	Yes

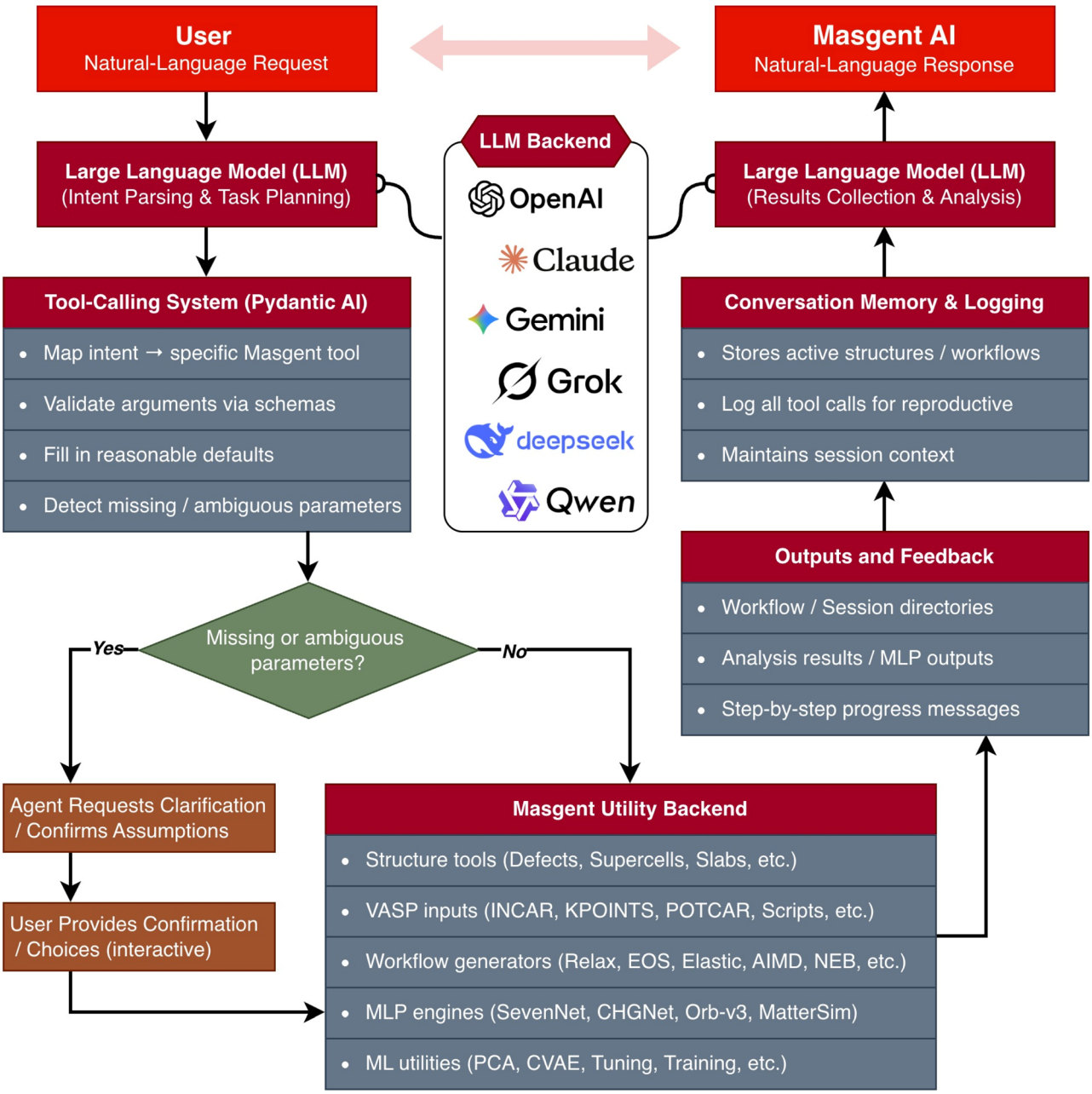

Fig. 2 AI agent workflow of Masgent. User requests are parsed by an LLM and routed through a schema-validated tool-calling system. The agent confirms assumptions, requests clarification when needed, and generates step-by-step feedback during execution. Validated operations are executed by Masgent's computational backend, and all actions are stored in conversation memory for context-aware, multi-turn interactions.

internally. This design allows users to interact with the system without configuring their own API keys. The LLM backend identifies the intended operation, such as structure generation, defect creation, VASP input preparation, workflow construction, MLP evaluation, or results analysis, and maps it to a specific tool. The agent supports multi-step workflows by breaking down complex user requests into sequential tasks, managing dependencies, and ensuring that intermediate results are appropriately passed between steps.

Each tool in Masgent is defined by a dedicated Pydantic schema that precisely specifies valid argument types, constraints, and default behavior. These schemas enforce robust input validation, detect missing or ambiguous parameters, and ensure that all operations are syntactically and physically consistent (*e.g.*, valid chemical formulas, sensible *k*-point densities, admissible supercell sizes). When ambiguity is detected, the agent explicitly asks for clarification, ensuring transparency and preventing inappropriate or unsafe execution.

The AI agent is also designed for multi-turn, context–aware interaction. It maintains lightweight session memory that tracks active structures, working directories, previous outputs, and user preferences. This allows users to reference prior results implicitly, for example: "use the relaxed structure from earlier" or "repeat that calculation with a denser *k*-point grid. To minimize latency and token consumption, the agent retains only contextual information required to execute ongoing tasks, while the full conversation history is still stored for each session and can be accessed or reviewed later.

The AI agent supports a broad range of capabilities, including but not limited to structure manipulation, VASP input preparation, construction of DFT workflows, fast MLP simulations, and various ML tasks. A complete list of current functionalities is provided in Appendix A. All interactions, tool calls, and workflow decisions are automatically logged in a structured format to ensure reproducibility and traceability. Sessions can be saved, reviewed, or shared with collaborators, facilitating transparency, knowledge transfer, and consistent computational practices across users and research groups.

To illustrate how Masgent integrates natural-language interaction with structured workflow automation, Fig. 3 presents a complete end-to-end example of the AI agent generating a full simulation workflow from a single conversational instruction. In this demonstration, a user requests a simulation workflow for $LaCoO_3$ involving three sequential tasks: (1) retrieve a pristine structure from the Materials Project; (2) introduce an oxygen vacancy; and (3) estimate the resulting defect energy using an MLP model.

As shown in Fig. 3, the left panel displays a representative interaction between the user and the agent. The agent interprets the request, constructs a multi-step execution plan, resolves ambiguous details (*e.g.*, defect type, number of vacancies, choice of MLP engine), and requests confirmation before proceeding. This transparent, iterative planning process ensures that all parameters are fully specified, physically meaningful, and aligned with user intent before execution.

The right panel shows the automatically generated workflow directory produced by Masgent. All intermediate and final outputs, including pristine and defective structures, MLP input and output files, and conversation transcripts, are organized into a standardized directory layout. As indicated by the timestamps, the full interaction required only about three minutes, including the user's initial request, the agent's intent parsing and planning, user confirmation, and the agent's execution and summarization of results.

This example demonstrates how Masgent transforms a high-level instruction, "Prepare a $LaCoO_3$ structure with an oxygen vacancy and estimate its energy using MLP", into a fully configured, reproducible simulation pipeline within seconds to minutes. It highlights the practical utility of AI-assisted workflow construction and its potential to significantly accelerate early-stage computational exploration.

It is important to note that while the AI agent simplifies the setup, configuration, and post-processing of simulations, it does not execute VASP calculations directly. Instead, it orchestrates the workflow surrounding these calculations. Certain expert-level tasks, such as diagnosing subtle VASP error messages or optimizing HPC job scheduling, may still require manual intervention and domain expertise. Nevertheless, by automating routine operations and providing an intuitive natural-language interface, Masgent significantly lowers the barrier to entry and enables a broader range of users to leverage advanced computational simulation techniques.

3.2 Structure manipulation and input generation

Masgent provides a comprehensive suite of tools for structure manipulation and automated input generation for DFT simulations. These capabilities form the foundation of streamlined workflow construction, allowing users to perform complex structural operations and prepare high-quality simulation inputs with minimal manual effort. All structure-related functionalities are built upon robust foundations provided by ASE, Pymatgen, and Masgent's own utility modules, ensuring compatibility with widely used computational materials workflows.

3.2.1. Structure preparation, conversion, and visualization. Masgent supports all major structure file formats used in atomic simulations, including VASP POSCAR/CONTCAR, CIF, and XYZ, and enables conversion between them. Structures may be loaded from local files or retrieved directly from the Materials Project *via* a user-provided API key, giving researchers rapid access to experimentally validated and computationally optimized crystal prototypes.

For VASP workflows, Masgent offers convenient tools to switch between Direct and Cartesian coordinates, facilitating compatibility with downstream simulation codes and analysis tools. As illustrated in Fig. 4, Masgent also provides interactive, browser-based 3D visualization using 3Dmol.js,[58] allowing users to inspect structural geometries, identify potential issues, and verify structural configurations before launching simulations. The viewer supports full rotation and zooming, and uses default atomic radii derived from VESTA[59,60] (Table 3) to ensure physically meaningful and visually consistent representations of atomic structures.

Conversations Histroy

```
================================================================================
New AI Session Started at 2025-12-18 11:49:22
================================================================================

[2025-12-18 11:49:45] User:

Prepare a LaCoO3 structure with oxygen vacancy, and estimate its energy using MLPs.
--------------------------------------------------------------------------------

[2025-12-18 11:50:19] Masgent AI:

PHASE I — PLANNING

1) Generate pristine LaCoO3 POSCAR from Materials Project database
- formula: LaCoO3
- note: default settings for other options
2) Create an oxygen vacancy in the structure
- original_element: O
- defect_amount: 1
- poscar_path (base): POSCAR_LaCoO3
3) Estimate the energy with a machine-learning potential
- poscar_path: (from previous step)
- mlps_type: CHGNet
- task_type: single
- fmax: 0.1
- max_steps: 500
- (defaults used for other parameters)

Please review the PLAN and confirm to proceed.
--------------------------------------------------------------------------------

[2025-12-18 11:50:50] User:

Confirm
--------------------------------------------------------------------------------

[2025-12-18 11:52:45] Masgent AI:

Phase II EXECUTION

- Vacancy structure created
  - POSCAR with oxygen vacancy generated at: defects/vacancies/POSCAR
- MLP energy estimation
  - ML model: CHGNet
  - Task: single-point energy (MLP-based)
  - Output:
    - total_energy (eV): -151.77938842773438
    - energy_per_atom (eV/atom): -7.988389015197754
  - CONTCAR for the relaxed/target structure: mlps_simulation/CHGNet/single/CONTCAR
  - Simulation log: mlps_simulation/CHGNet/ single/masgent_mlps_single.log

Notes and potential next steps
- If you want to compare with another ML model (e.g., SevenNet, Orb-v3, MatterSim) or adjust
parameters (temperature, md_steps, etc.), I can run parallel or sequential tests.
- Would you like me to proceed with another model or additional vacancy/types (e.g., Co or La
vacancies, different vacancy concentrations)?
```

Workflow Directory

```
masgent_projects
└── runs_20251218114914
    ├── POSCARs
    │   └── LaCoO3
    │       ├── POSCAR_Cubic_Pm-3m_mp-510611
    │       ├── POSCAR_Monoclinic_C2_c_mp-1180725
    │       ├── POSCAR_Orthorhombic_Imma_mp-1504268
    │       ├── POSCAR_Triclinic_P-1_mp-561922
    │       ├── POSCAR_Trigonal_R-3c_mp-19051
    │       └── ...
    ├── defects
    │   └── vacancies
    │       └── POSCAR
    ├── mlps_simulation
    │   └── CHGNet
    │       └── single
    │           ├── CONTCAR
    │           └── masgent_mlps_single.log
    ├── POSCAR
    ├── POSCAR_LaCoO3
    └── conversation_history.txt
```

Fig. 3 End-to-end example of Masgent's AI agent generating a complete workflow from a single natural-language request. (Left) The agent interacts with the user *via* clarifying questions, confirmation steps, and progress updates to build a valid simulation workflow. (Right) The resulting directory structure includes the fetched crystal structure, vacancy defect, and MLP energy evaluation.

3.2.2. Structure construction and modification. Masgent provides first-class support for constructing and modifying atomistic structures, enabling users to generate a wide range of configurations commonly required in materials simulations.

3.2.2.1. Defect generation. As shown in Fig. 5, Masgent automates the creation of vacancy, substitutional, and interstitial defects. Vacancy creation allows users to specify either the number of atoms to remove or a target concentration, after which Masgent randomly selects atomic sites for removal. Substitutional defects are generated by randomly replacing host atoms with dopant species according to user-specified compositions. For interstitials, Masgent employs the *VoronoiInterstitialGenerator* from pymatgen-analysis-defects,[61] which identifies plausible interstitial positions using Voronoi analysis and places the desired species accordingly. Together, these tools allow users to rapidly prototype defect chemistries without manually editing structure files.

3.2.2.2. Supercell generation. Fig. 6a illustrates a $2 \times 2 \times 2$ $LaCoO_3$ supercell generated using Masgent's supercell module. Users provide an integer 3×3 scaling matrix, and Masgent automatically replicates lattice vectors and atomic positions while preserving structural symmetry. Such supercells are essential for modeling dilute defects, finite-size effects, lattice dynamics, and long-range structural distortions.

3.2.2.3. SQS generation. To simulate chemically disordered systems, Masgent integrates with the Icet package[62] to generate SQS structures. As shown in Fig. 6b, users specify the target composition, supercell size, and correlation functions to match. Icet's Monte Carlo optimization framework then produces an atomic arrangement whose statistics approximate those of a perfectly random alloy, ideal for modeling high-entropy oxides, doped perovskites, and solid solutions.

3.2.2.4. Surface slab generation. Masgent also supports automated surface slab generation from bulk structures. Fig. 7a

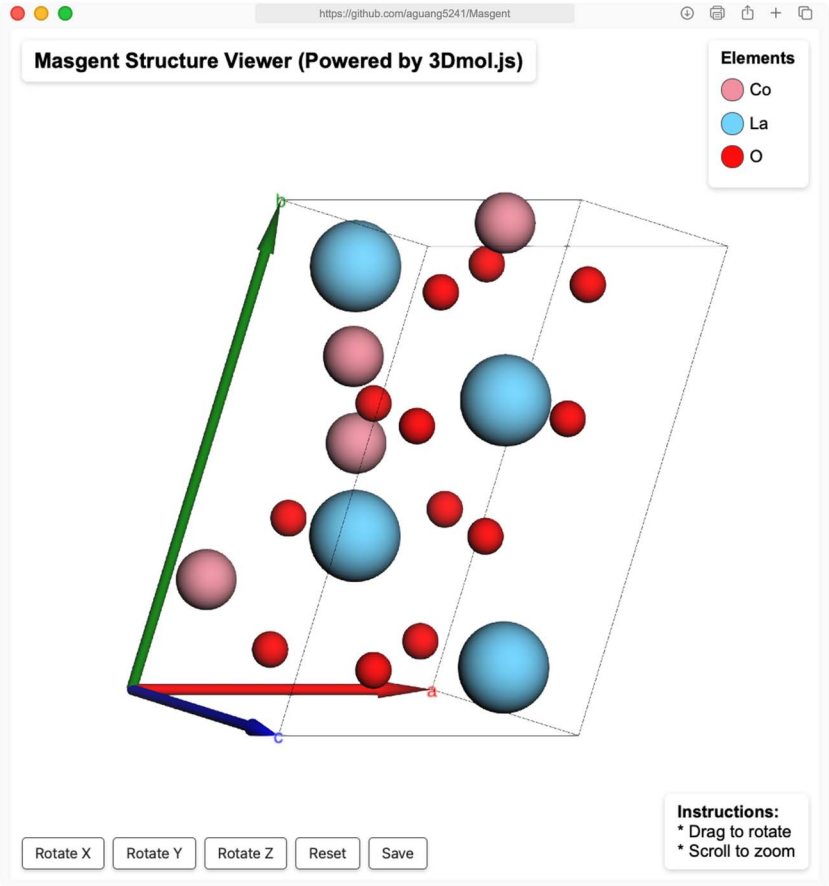

Fig. 4 Interactive crystal structure visualization in Masgent. The viewer supports rotation and zooming, displays the unit cell and lattice vectors, and uses VESTA-consistent atomic radii for standardized visualization.

and b show examples of $LaCoO_3$ and La_2NiO_4 (001) slabs generated through the framework. Users specify Miller indices, slab thickness, and vacuum spacing, and Masgent constructs a periodic slab geometry with user-defined terminations. These slabs are suitable for surface energy calculations, adsorption studies, and catalysis simulations.

Table 3 Default atomic radii used in Masgent for structure visualization

Element	Radius	Element	Radius	Element	Radius	Element	Radius	Element	Radius
H	0.46	Sc	1.64	Nb	1.47	Pm	1.81	Tl	1.71
He	1.22	Ti	1.47	Mo	1.40	Sm	1.81	Pb	1.75
Li	1.57	V	1.35	Tc	1.35	Eu	2.06	Bi	1.82
Be	1.12	Cr	1.29	Ru	1.34	Gd	1.79	Po	1.77
B	0.81	Mn	1.37	Rh	1.34	Tb	1.77	At	0.62
C	0.77	Fe	1.26	Pd	1.37	Dy	1.77	Fr	1.00
N	0.74	Co	1.25	Ag	1.44	Ho	1.76	Ra	2.35
O	0.74	Ni	1.25	Cd	1.52	Er	1.75	Ac	2.03
F	0.72	Cu	1.28	In	1.67	Tm	1.00	Th	1.80
Ne	1.60	Zn	1.37	Sn	1.58	Yb	1.94	Pa	1.63
Na	1.91	Ga	1.53	Sb	1.41	Lu	1.72	U	1.56
Mg	1.60	Ge	1.22	Te	1.37	Hf	1.59	Np	1.56
Al	1.43	As	1.21	I	1.33	Ta	1.47	Pu	1.64
Si	1.18	Se	1.04	Xe	2.18	W	1.41	Am	1.73
P	1.10	Br	1.14	Cs	2.71	Re	1.37	Others	0.80
S	1.04	Kr	1.98	Ba	2.24	Os	1.35		
Cl	0.99	Rb	2.50	La	1.88	Ir	1.36		
Ar	1.92	Sr	2.15	Ce	1.82	Pt	1.39		
K	2.35	Y	1.82	Pr	1.82	Au	1.44		
Ca	1.97	Zr	1.60	Nd	1.82	Hg	1.55		

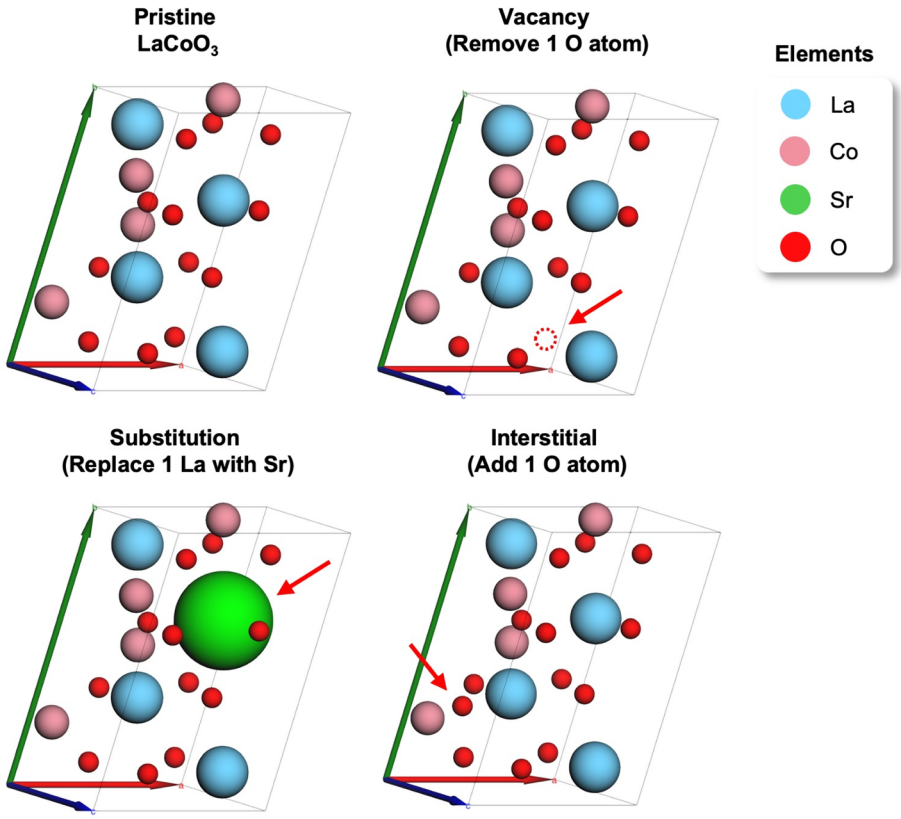

Fig. 5 Defect generation in Masgent, illustrating pristine $LaCoO_3$ and representative vacancy, substitution, and interstitial defect configurations generated from the parent structure.

3.2.2.5. Interface structure generation. For heterostructure and multilayer simulations, Masgent incorporates a lattice-matching algorithm[63] to construct interfaces between different materials. As demonstrated in Fig. 7c, users specify the two bulk crystals, orientations, and matching tolerances. Masgent identifies compatible surface–pair combinations and generates a strain-minimized interface supercell. This functionality is particularly useful for studying epitaxial growth, grain boundaries, oxide heterostructures, and interfacial transport.

3.2.3. Automatic VASP input generation. Masgent provides turnkey utilities for generating high-quality VASP input files that follow best practices and established parameter standards. Users may rely on built-in templates for common simulation types or customize settings to meet specific workflow requirements. These tools significantly reduce manual setup effort while improving reproducibility and consistency across calculations.

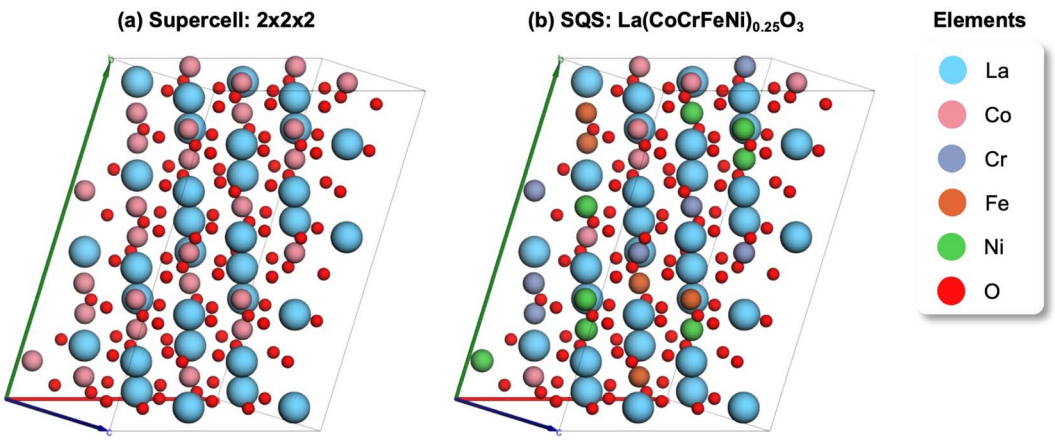

Fig. 6 Supercell and SQS generation in Masgent, showing (a) a 2 × 2 × 2 supercell constructed from the primitive $LaCoO_3$ cell and (b) an SQS generated using the Icet package to model compositional disorder.

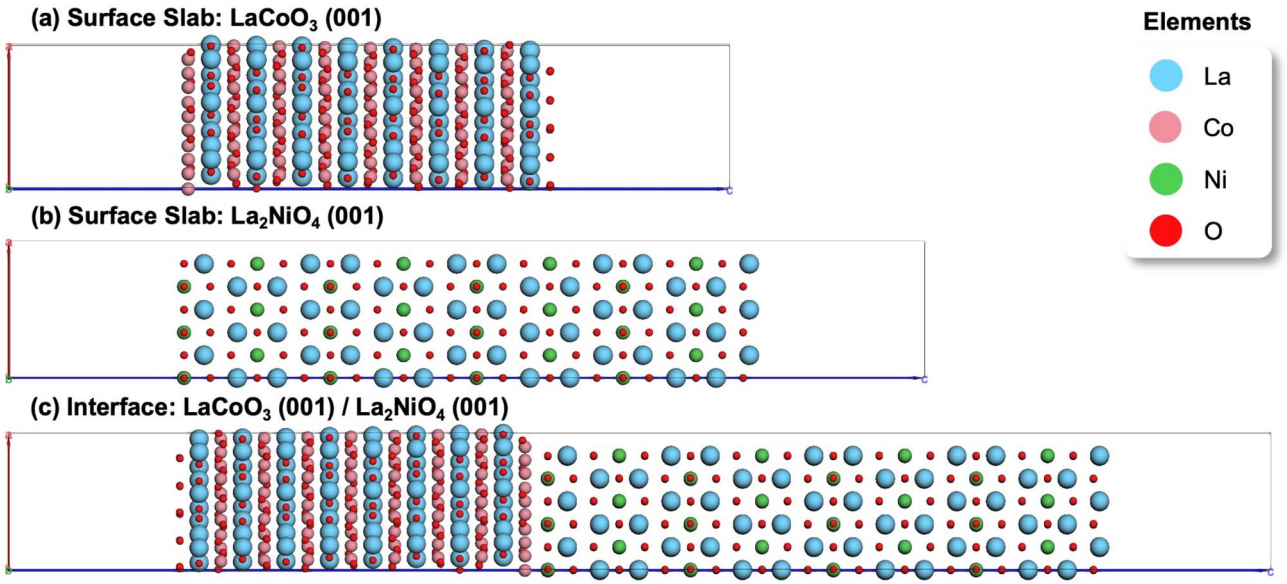

Fig. 7 Surface slab and interface construction in Masgent, showing (a) $LaCoO_3$ (001) and (b) La_2NiO_4 (001) surface slabs generated with specified slab thickness and vacuum spacing, and (c) a $LaCoO_3$ (001)/La_2NiO_4 (001) interface constructed using lattice-matching algorithms.

3.2.3.1. INCAR generation. Masgent includes predefined INCAR templates adapted from Pymatgen and optimized for various calculation types, including *MPMetalRelaxSet* (metallic relaxations), *MPRelaxSet* (general structure relaxations), *MPStaticSet* (static energy calculations), *MPNonSCFBandSet* (non-self-consistent band structure), *MPNonSCFDOSSet* (non-self-consistent density-of-states), and *MPMDSet* (*ab initio* molecular dynamics). Each template provides sensible defaults derived from established Materials Project workflows while allowing advanced users to override parameters for finer control.

3.2.3.2. KPOINTS generation. The KPOINTS generator creates either Monkhorst–Pack or Γ-centered meshes depending on user preference and lattice symmetry. Users can specify target accuracy levels (low, medium, or high) or provide explicit *k*-point densities. Masgent then automatically constructs an appropriate grid that balances computational cost and convergence quality, ensuring adequate Brillouin-zone sampling across diverse system types.

3.2.3.3. POTCAR generation. Pseudopotentials are assembled from locally stored PAW datasets. Masgent automatically selects the correct POTCAR files based on the chemical elements present in the structure, ensuring compatibility and consistency with VASP's recommended datasets and preventing manual errors. This greatly simplifies setup for multi-element compounds and large defect or interface supercells.

3.2.3.4. HPC job script generation. For HPC execution, Masgent generates ready-to-submit job scripts compatible with schedulers such as SLURM. Users can specify computational resources, including node count, tasks per node, wall time, queue/partition, and module environments, and Masgent automatically constructs a script that loads the appropriate modules and launches VASP with correct MPI settings. These scripts follow best-practice conventions for parallel performance, traceability, and job reproducibility.

3.3 DFT workflow automation and analysis

Masgent provides a unified framework for constructing, organizing, and analyzing standard DFT workflows. To ensure reliability, Masgent relies on standardized workflow templates and validated parameter presets derived from commonly adopted computational protocols, including structure relaxations, static energy calculations, elasticity calculations, EOS fitting, AIMD simulations, and NEB pathways. During the planning stage, the AI agent constructs workflows using these predefined protocols and exposes key simulation parameters (*e.g.*, exchange-correlation functional, *k*-point density, energy cutoff, and convergence thresholds) to the user for verification and adjustment. By integrating the automated input-generation tools described in Section 3.2 with modular workflow templates and dedicated post-processing routines, Masgent ensures that workflow construction remains transparent and consistent with established simulation methodologies, enabling users to conduct reproducible, systematic, and efficient DFT studies while minimizing manual intervention and reducing opportunities for human errors.

As illustrated in Fig. 8, Masgent automatically generates a complete VASP workflow directory with clearly labeled subfolders, template input files, structured output locations, and ready-to-submit HPC job scripts. This standardized directory architecture ensures immediate usability on both local workstations and large HPC clusters.

Masgent currently provides predefined workflow templates for the following classes of DFT simulations:

3.3.1. Convergence tests. Convergence tests for key numerical parameters, such as the plane-wave energy cutoff

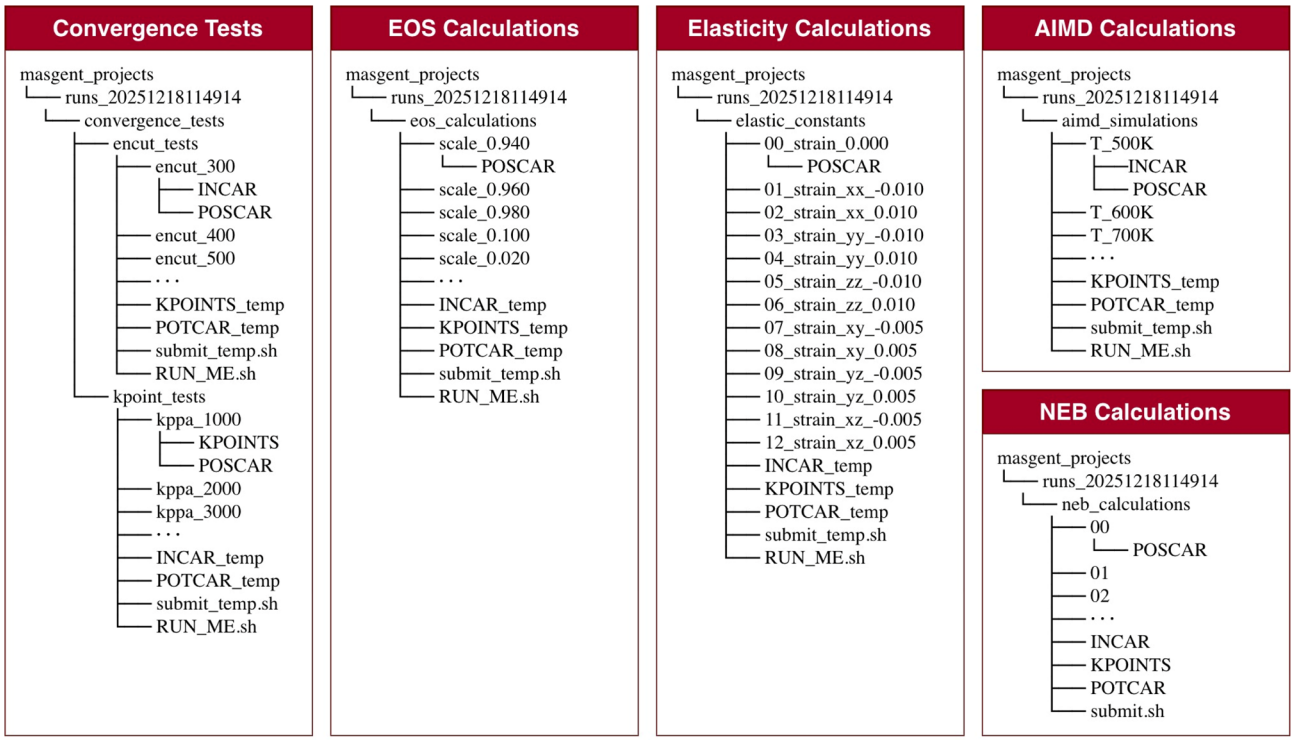

Fig. 8 Example VASP workflow structure generated by Masgent, illustrating automatically organized directory trees for convergence tests, EOS, elasticity, AIMD, and NEB calculations within a unified and reproducible DFT simulation pipeline.

(ENCUT) and *k*-point density, are critical for ensuring accuracy and reproducibility. Masgent automates these tests by generating a series of calculations with systematically varied ENCUT values or *k*-point meshes, preparing all required VASP inputs, and organizing the runs into structured directories. After execution, Masgent collects and analyzes energy differences per atom to identify convergence thresholds and recommend optimal simulation settings.

3.3.2. EOS calculations. Masgent provides automated tools to compute EOS curves by generating a sequence of volume-scaled structures around the equilibrium configuration. For each scaled structure, the framework prepares static calculations and organizes them into a consistent workflow directory. Once energies are collected, Masgent fits the resulting energy–volume data to standard EOS models, such as the Birch–Murnaghan equation,[64] to extract equilibrium volume and generate the corresponding optimized structure.

3.3.3. Elasticity calculations. To evaluate elastic properties, Masgent constructs a set of symmetry-preserving strained structures by applying small deformations to the equilibrium lattice. Each strain state is placed in a separate directory. After the static calculations are completed, the framework extracts stress tensors and constructs the elastic stiffness matrix using linear elasticity theory.

3.3.4. AIMD simulations. Masgent automates the preparation and organization of AIMD simulations by generating input decks for user-specified temperatures, time steps, and simulation durations. AIMD workflows are organized by temperature. Post-processing utilities compute trajectory statistics such as temperature equilibration, energy fluctuations, mean-squared displacement (MSD), diffusion coefficients, and Arrhenius behavior.

3.3.5. NEB calculations. For migration-barrier and reaction-pathway studies, Masgent automates the NEB workflow by generating intermediate images between initial and final states using interpolation algorithms. Each image is placed into a numbered directory. The framework produces complete job scripts and input templates suitable for climbing-image NEB (CI-NEB) or standard NEB calculations. After execution, Masgent provides tools to extract reaction-coordinate positions, compute image energies, construct the full energy profile, and evaluate the corresponding migration barrier.

3.4 Fast machine learning potential simulations

Masgent integrates several state-of-the-art MLPs to enable rapid, low-cost simulations that approximate DFT-level accuracy. These capabilities dramatically accelerate preliminary screening, large-scale simulations, and workflow prototyping, often reducing compute time by orders of magnitude relative to conventional DFT. As illustrated in Fig. 9, Masgent provides a unified MLP framework that abstracts away the differences between individual models, allowing users to switch seamlessly between potentials while preserving a consistent workflow interface.

3.4.1. Unified MLP interface. Masgent's MLP module wraps multiple widely used models, including SevenNet, CHGNet, Orb-v3, and MatterSim, under a single, standardized API.

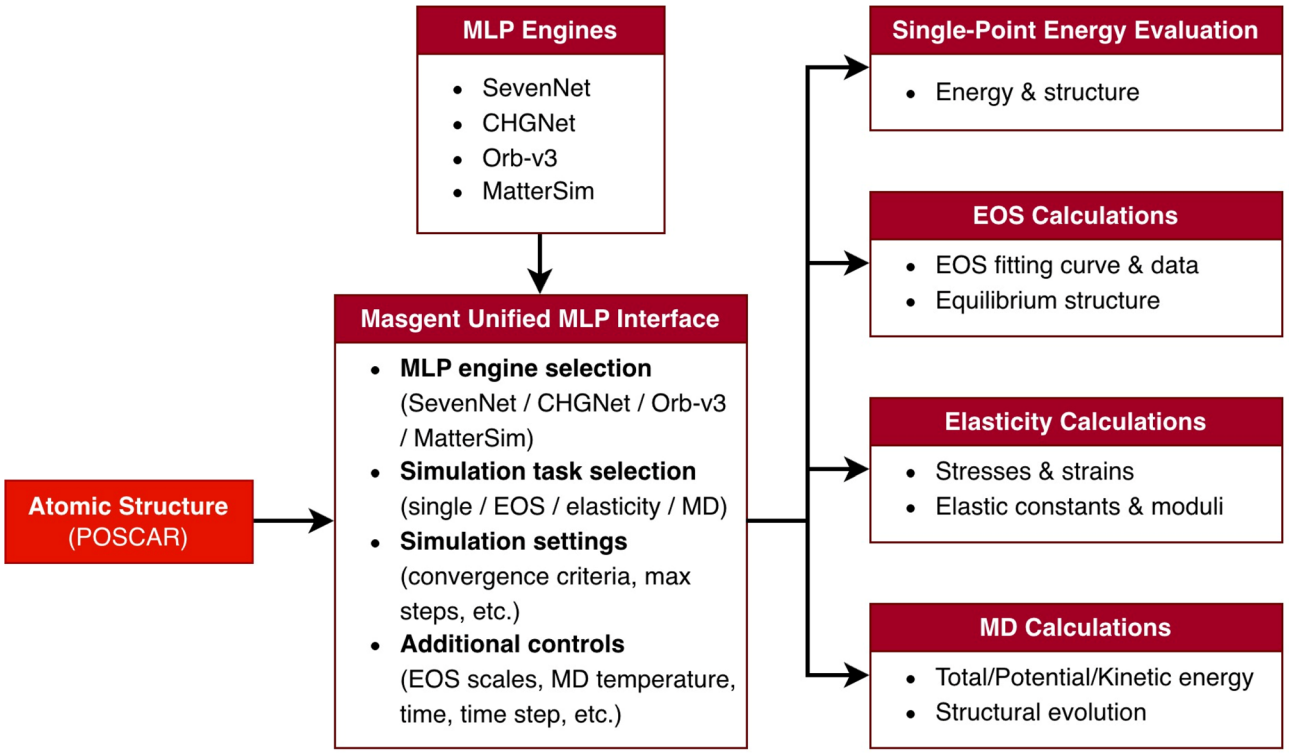

Fig. 9 Unified MLP framework in Masgent. Atomic structures are processed through a unified MLP interface that provides consistent access to multiple engines (SevenNet, CHGNet, Orb-v3, and MatterSim), enabling standardized single-point energy, EOS, elastic constant, and *ab initio* molecular dynamics calculations.

Although these models differ in architecture, training data, and target material domains, Masgent exposes a uniform set of high-level function calls for energy evaluation, EOS fitting, elasticity calculations, and MD simulations. Users simply select the desired MLP engine, and the remaining workflow logic remains unchanged. This unified interface simplifies experimentation and enables cross-model benchmarking without modifying code or directory structures.

3.4.2. Single-point and EOS calculations. For rapid energetic assessments, Masgent supports single-point energy and force evaluations using any supported MLP engines. For EOS calculations, Masgent automatically generates volume-scaled structures, evaluates their energies with the selected potential, and fits the resulting energy-volume curve to standard EOS models. This yields equilibrium volumes and optimized structures at a fraction of the cost of DFT, enabling quick exploration of structural stability and material trends before committing to more computationally intensive first-principles calculations.

3.4.3. Elasticity calculation. Masgent provides MLP-accelerated elasticity workflows by applying small strains to a structure, computing the resulting stress tensors *via* the chosen potential, and extracting elastic constants using linear elasticity theory. These calculations typically complete in seconds to minutes, allowing high-throughput screening of mechanical properties across large compositional or configurational design spaces.

3.4.4. MD simulations. Using the same unified interface, Masgent enables MLP-driven MD simulations in the NVT ensemble using a Nosé-Hoover chain thermostat. Users may specify temperature, timestep, and total simulation duration. The framework handles trajectory generation and provides post-processing tools to compute MSD, diffusion coefficients, and temperature and energy evolution. These MLP-based MD simulations offer orders of magnitude faster than AIMD, making them particularly useful for exploratory phase-space sampling, transport property analysis, and preliminary stability assessments.

3.5 Lightweight machine learning utilities

In addition to automated DFT and MLP workflows, Masgent provides a suite of lightweight ML utilities designed to support materials informatics tasks such as from-scratch model development and pre-trained model applications. As shown in Fig. 10, these tools enable researchers and students to rapidly prototype ML workflows directly within Masgent, eliminating the need for external environments and offering a unified platform that integrates both physics-based and data-driven methodologies.

3.5.1. Feature exploration and correlation analysis. Masgent provides utilities for loading structured datasets (*e.g.*, CSV files containing descriptors and target properties) and performing exploratory feature analysis. Users can compute

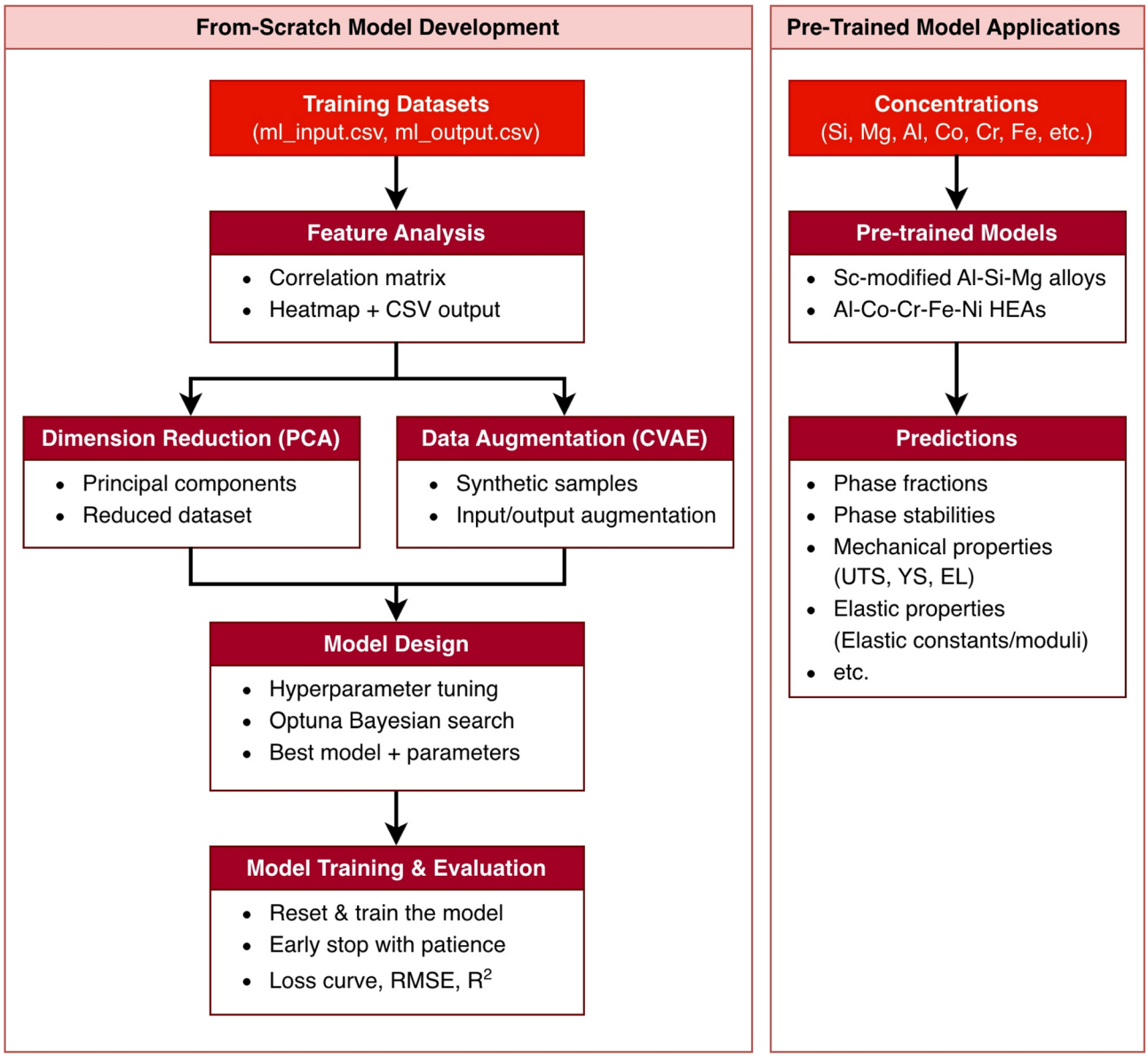

Fig. 10 Overview of lightweight ML utilities in Masgent, illustrating two workflows: from-scratch model development (left), including feature analysis, PCA-based dimensionality reduction, CVAE-based data augmentation, hyperparameter optimization, and model training/evaluation; and pre-trained model applications (right), enabling property prediction from alloy compositions using existing models.

a correlation matrix heatmap to visualize linear relationships, redundant descriptors, and potential descriptor groups relevant for model construction. All plots are produced in publication-quality formats and may be exported for further use in manuscripts, presentations, or reports.

3.5.2. Dimensionality reduction. To help interpret high-dimensional datasets, common in composition-property mappings, defect thermodynamics studies, and doping analyses, Masgent incorporates a built-in PCA module. PCA projects data into a lower dimension while preserving dominant variance directions, allowing users to visualize clustering behavior, flag outliers, and identify low-dimensional manifolds that may inform subsequent modeling choices.

3.5.3. Data augmentation. For scenarios where training data are sparse, a frequent challenge in materials science research, Masgent provides a lightweight CVAE implementation. The CVAE module generates synthetic samples conditioned on existing descriptors, producing augmented datasets that more accurately reflect the distribution of the training set. This helps mitigate overfitting and improve the robustness of downstream predictive models.

3.5.4. Hyperparameter tuning. Masgent integrates Optuna-based hyperparameter optimization to streamline model selection. Users define the hyperparameter search spaces for a given ML model, and Masgent employs Bayesian optimization, *via* the Tree-Structured Parzen Estimator (TPE),[65] to efficiently identify the best-performing configurations. This

automation automates the search procedure, reduces manual trial-and-error, and improves overall predictive performance across diverse modeling tasks.

3.5.5. Model training and evaluation. Masgent provides simplified interfaces for training, validating, and evaluating ML models. Standardized routines support train-test splitting, early stopping with patience, loss and accuracy tracking, and evaluation metrics such as RMSE and R^2. The framework also generates training-curve visualizations that help diagnose model convergence and generalization behavior. These tools allow users to construct baseline predictive models for properties such as formation energies, lattice distortions, diffusion barriers, and defect energetics with minimal setup and without requiring advanced ML expertise.

3.5.6. Pre-trained model applications. In addition to from-scratch model development, Masgent supports direct property prediction using pre-trained ML models. These models are adapted from established studies on Sc-modified Al–Mg–Si alloys[66,67] and Al–Co–Cr–Fe–Ni high-entropy alloys (HEAs),[68] and are embedded within the framework. For Sc-modified Al–Mg–Si alloys, users provide elemental concentrations (*e.g.*, Mg and Si), and Masgent returns CALPHAD-derived phase fractions of the primary Al phase, eutectic phases, and the $AlSc_2Si_2$ phase, as well as predicted mechanical properties such as ultimate tensile strength (UTS), yield strength (YS), and elongation (EL). For Al–Co–Cr–Fe–Ni HEAs, users input elemental concentrations (*e.g.*, Al, Co, Cr, and Fe), and Masgent predicts phase stability metrics such as formation enthalpy, along with elastic properties including elastic constants and elastic moduli. By combining pre-trained inference with from-scratch training utilities, Masgent provides a flexible ML ecosystem that enables both rapid property screening and customized model development within a unified environment.

4. Benchmarks and case studies

To evaluate the reliability, efficiency, and practical utility of Masgent, a series of benchmarks and case studies were performed, covering structure preparation, automated DFT workflow generation, MLP simulations, and lightweight ML tasks. Detailed input files, raw results, and additional examples are available at https://github.com/IMPDGroup/Masgent/tree/main/examples. These demonstrations highlight Masgent's ability to accelerate simulation setup, reduce human error, and provide rapid scientific insight across diverse material systems.

4.1 Workflow preparation efficiency and validation

We benchmarked the preparation speed of a standard DFT simulation workflow (*e.g.* EOS) using Masgent and compared it with typical graduate-level manual practice. The EOS workflow consisting of (i) generating structures at multiple volumes, (ii) preparing VASP input files, (iii) organizing calculation directories, and (iv) performing automated post-processing and Birch–Murnaghan fitting. A representative set of materials were tested, including simple metals (Al, Cu), elemental and compound semiconductors (Si, GaAs, MoS_2), ionic and transition-metal oxides (MgO, NiO, TiO_2), and complex oxide frameworks such as perovskites and Ruddlesden–Popper phases ($LaCoO_3$, La_2NiO_4).

Masgent generated all required input files, directory structures, and analysis scripts in under 30 seconds on a standard laptop (measured as wall-clock time from user prompt to completed workflow generation, excluding HPC execution time). In contrast, manual preparation, performed without pre-existing automation scripts, required approximately 1–3 hours. This manual setup time included structure scaling, INCAR/KPOINTS configuration, directory organization, and implementation of post-processing routines. The exact duration depends on system complexity and user experience level. These results demonstrate a substantial reduction in workflow preparation time.

To evaluate the correctness of Masgent-generated inputs, we compared calculated equilibrium energies and volumes obtained from Masgent-prepared EOS workflows with reference values reported in the Materials Project database for Al, Cu, Si, MgO, TiO_2, and GaAs. As summarized in Table 4, the total energy differences (ΔE, Masgent – Materials Project) are within 0.05 eV per atom, and the relative deviations in equilibrium volume (ΔV) are below 1% for all tested systems. These deviations fall within the typical range expected from differences in convergence settings, pseudopotential versions, and numerical tolerances across independent DFT calculations.

By combining rapid workflow generation with reproducible and database-consistent results, Masgent reduces manual setup effort while maintaining scientific reliability.

4.2 MLPs performance benchmarks

To assess the accuracy and computational efficiency of Masgent's fast-simulation module, we benchmarked the supported MLPs, SevenNet, CHGNet, Orb-v3, and MatterSim, against DFT-PBE reference calculations. Test systems include metals (Al, Cu), elemental and compound semiconductors (Si, GaAs, MoS_2), binary oxides (MgO, NiO, TiO_2), and complex oxides such as $LaCoO_3$ and La_2NiO_4. Benchmark results are summarized in Fig. 11.

4.2.1. Energy accuracy. As shown in Fig. 11a, all MLPs reproduced DFT energies with chemically reasonable fidelity. To ensure a fair assessment, identical DFT-relaxed structures

Table 4 Comparison of equilibrium energies and volumes obtained from Masgent-generated EOS workflows with reference values from the Materials Project database

System	ΔE (eV per atom)	ΔV (%)
Al	−0.0039	−0.0244
Cu	−0.0057	−0.1070
Si	0.0001	−0.1910
MgO	0.0498	0.0203
TiO_2	0.0001	−0.5635
GaAs	0.0004	−0.2396

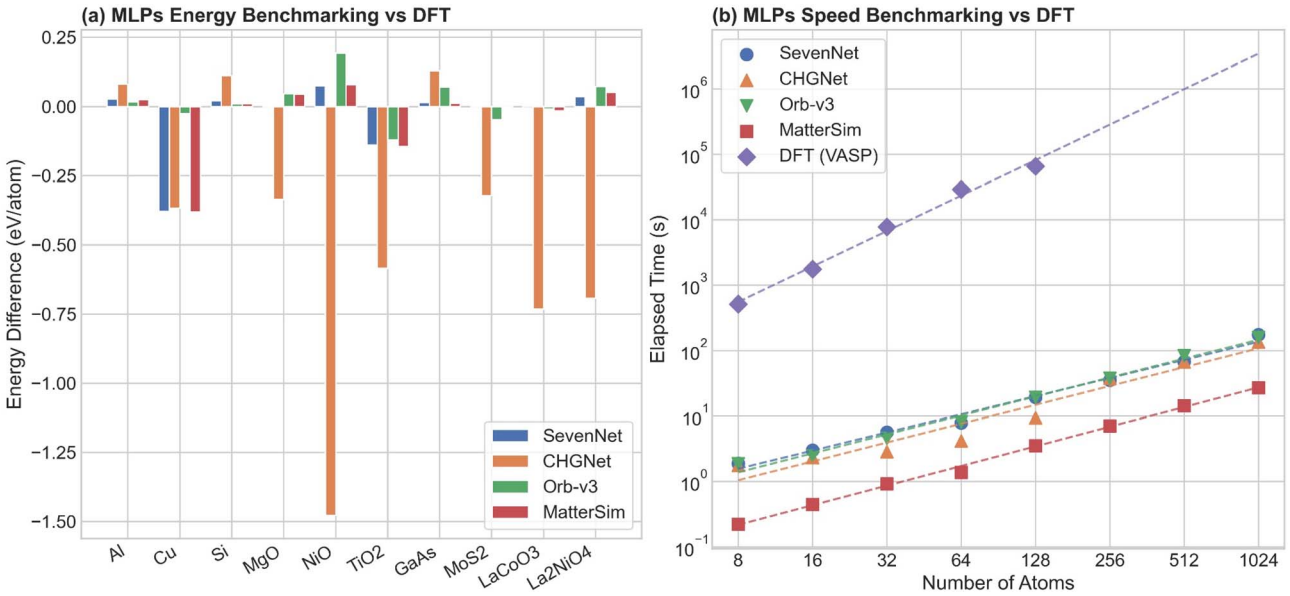

Fig. 11 Performance benchmarking of Masgent's MLPs. (a) Single-point energy errors of SevenNet, CHGNet, Orb-v3, and MatterSim compared to DFT across diverse materials. (b) Computational scaling showing 10^3–10^4 times speedups over DFT for increasing system sizes.

were used for all MLP evaluations, and Masgent did not modify model parameters beyond documented defaults. Across all systems, most models achieved typical deviations below ∼100 meV per atom, consistent with their reported literature performance. Larger deviations were observed for materials lying outside a model's primary training domain (*e.g.*, NiO for CHGNet), reflecting intrinsic model limitations, and emphasizing the importance of selecting an appropriate potential for a given material class.

4.2.2. Computational efficiency. Computational efficiency results, shown in Fig. 11b, reveal that MLPs achieve dramatic speed advantages over DFT across system sizes ranging from 8 to 1024 atoms in binary Cu–Mg supercells. All evaluated models exhibit approximately logarithmic scaling with atom count and consistently deliver 10^3–$10^4\times$ speedups relative to VASP-based DFT calculations, which incur rapidly increasing computational cost with system size and require hours, sometimes even days, for systems exceeding ∼128 atoms. Among the models tested, MatterSim provides the fastest inference, followed by CHGNet, Orb-v3, and SevenNet, though all remain orders of magnitude faster than first-principles methods. These findings demonstrate that MLPs offer an extremely efficient alternative for rapid screening, large-supercell exploration (hundreds to thousands of atoms), and MD simulations in scenarios where DFT would be computationally prohibitive.

4.3 Case studies

To further demonstrate Masgent's reliability and practical effectiveness across diverse simulation workflows, we present several representative case studies highlighting convergence analysis, EOS fitting, AIMD simulations, NEB studies, elastic property evaluation, and lightweight ML modeling. Together, these examples highlight how Masgent not only accelerates traditionally labor-intensive tasks, such as structure generation, input preparation, workflow construction, and results analysis, but also ensures that each step adheres to consistent, validated, and reproducible standards. By automatically generating schema-validated inputs, enforcing best-practice defaults, and organizing workflows into robust directory structures, Masgent enables users to perform high-fidelity simulations with confidence and minimal manual intervention. The examples presented here focus primarily on inorganic materials, reflecting common applications of periodic DFT workflows. However, the workflow construction and input-generation framework of Masgent can in principle be applied to other periodic systems, including periodic organic crystalline materials.

4.3.1. Case study 1: convergence testing for aluminum. As shown in Fig. 12a, Masgent was used to perform a systematic convergence study for bulk Al by varying the ENCUT values. With a single user request, the AI agent automatically generated all required input files, producing a series of calculations spanning ENCUT values from 300 to 700 eV. Analysis of the resulting energy difference identified convergence within the threshold of 1 meV per atom at ENCUT of approximately 400 eV. This case demonstrates not only Masgent's efficiency, but also its reliability in executing best-practice convergence protocols that yield trustworthy and reproducible results.

4.3.2. Case study 2: EOS for silicon. Fig. 12b shows an EOS calculation for bulk Si prepared entirely through Masgent. The AI agent generated a series of volume-scaled structures around the equilibrium configuration and assembled all necessary VASP input files. The resulting energy-volume data were automatically fitted to the Birch–Murnaghan EOS, yielding an equilibrium volume of 20.448 Å^3 per atom and lattice parameters $a = b = c = 5.469$ Å, $\alpha = \beta = \gamma = 90°$ at 0 K. These values are in good agreement with the experimentally measured lattice

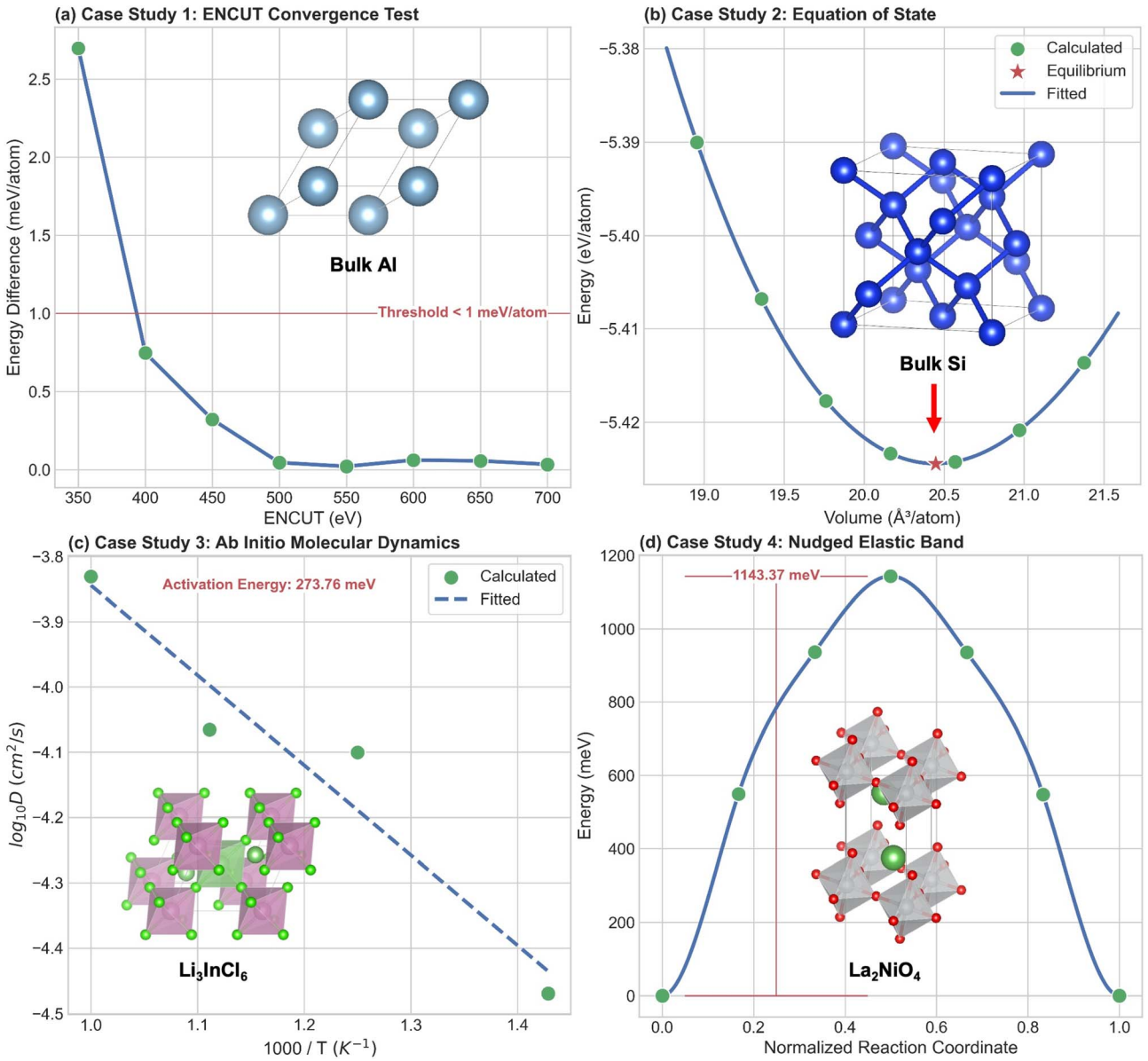

Fig. 12 Case studies demonstrating automated DFT workflows in Masgent. (a) ENCUT convergence test for bulk Al, showing the energy difference between successive plane-wave cutoffs. Convergence is reached once $\Delta E < 1$ meV per atom (red dashed line). (b) EOS calculation for bulk Si, where DFT-calculated energies (green points) are fitted using the Birch–Murnaghan (blue curve) to obtain the equilibrium volume (red star). (c) AIMD simulation for Li_2InCl_6 showing the Arrhenius behavior of the Li diffusion coefficient as a function of inverse temperature. (d) NEB calculation illustrating a migration pathway between initial and final configurations. The spline-smoothed energy profile (blue curve) shows the normalized reaction coordinate and associated energy barrier.

constant of silicon ($a_0 = 5.431$ Å)[69] and previous first-principles calculations ($a_0 = 5.44$ Å, $V_0 = 20.478$ Å^3 per atom).[70] This case demonstrates Masgent's ability to produce accurate, publication-quality EOS analyses through a fully automated and reproducible workflow.

4.3.3. Case study 3: AIMD for Li_3InCl_6. To demonstrate Masgent's capability in handling finite-temperature dynamical simulations and transport analysis, AIMD simulations were performed for the solid electrolyte Li_3InCl_6. Masgent automatically constructed the AIMD workflow, including time-step configuration and temperature scheduling across multiple simulation points. From the resulting trajectories, the AI agent computed mean squared displacements (MSDs) and extracted Li diffusion coefficients (D) at each temperature. As shown in Fig. 12c, the temperature dependence of the diffusion coefficients follows Arrhenius behavior. A linear fit to $\log_{10}D$ *versus* $1000/T$ yields an activation energy of 273.76 meV, in good agreement with reported values ($\approx$0.280 eV) for halide-based solid electrolytes.[71] This case highlights Masgent's ability to automate complex AIMD workflows, perform reliable post-processing of dynamical data, and deliver quantitatively

meaningful transport properties with minimal user intervention.

4.3.4. Case study 4: NEB for La_2NiO_4. Masgent was used to construct and analyze an NEB pathway for oxygen migration in La_2NiO_4. The AI agent interpolated an intermediate image between initial and final configurations and produced a complete NEB workflow with ready-to-submit job scripts. After the calculation, Masgent automatically parsed the outputs, reconstructed the reaction pathway using spline smoothing, and exported the reaction coordinate, image energies, and migration barrier. As shown in Fig. 12d, the resulting energy profile clearly highlights the transition-state region and yields a migration barrier of $\Delta E = 1.14$ eV, which falls within the range of previously reported simulation results (0.46–1.34 eV).[72–74] The variation in reported values arises from differences in diffusion pathways, oxygen stoichiometry, and computational methodologies used in previous studies. This case illustrates Masgent's ability to handle complex, multi-image workflows and deliver fully post-processed NEB analyses with minimal user intervention.

4.3.5. Case study 5: elastic constants for copper. Masgent was also employed to compute the elastic constants of FCC Cu by generating a complete set of symmetry-allowed strained configurations. The AI agent prepared input files for each strain

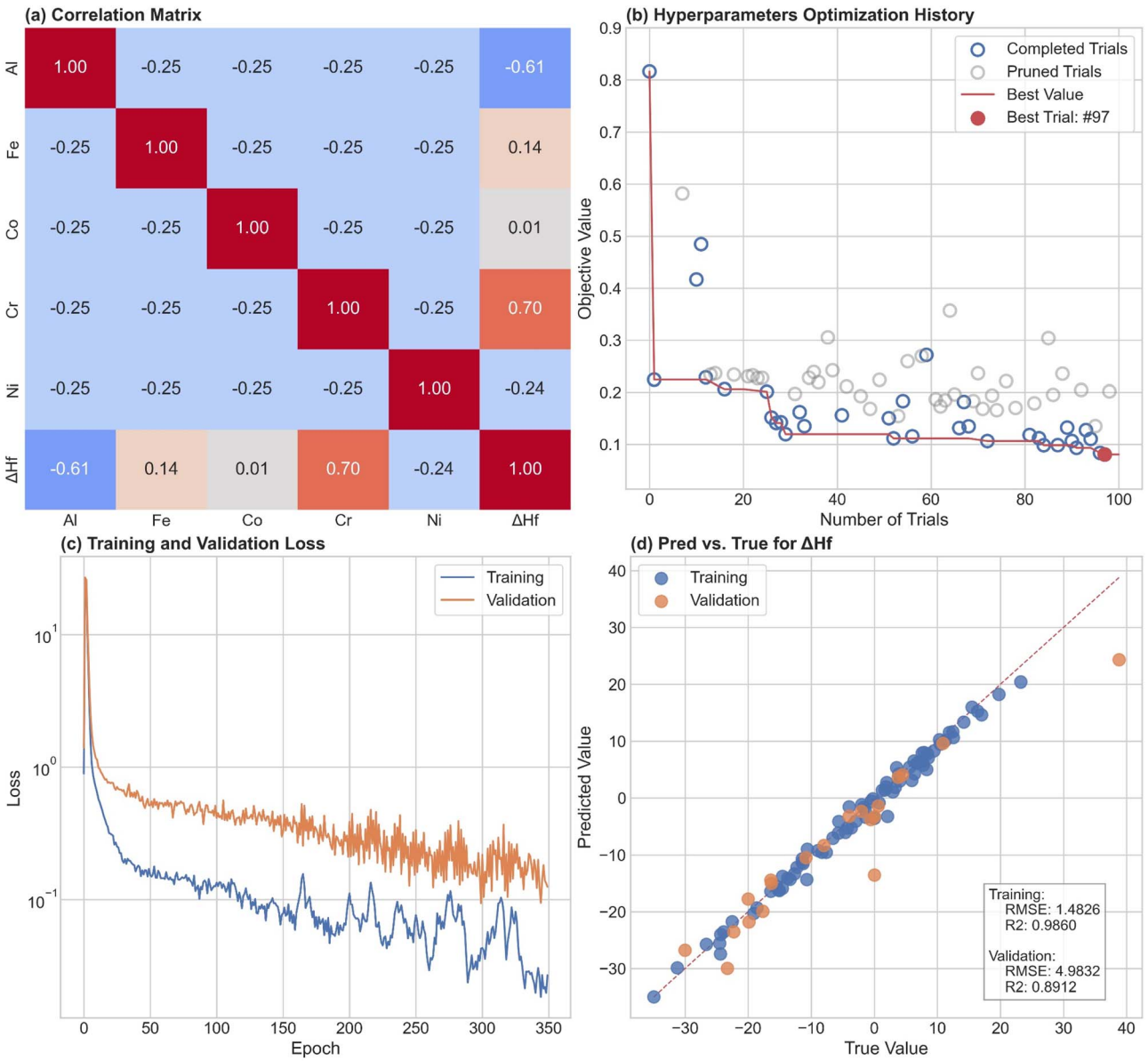

Fig. 13 Case study demonstrating ML modeling of formation enthalpies (ΔH_f) for high-entropy Al–Co–Cr–Fe–Ni alloys. (a) Correlation matrix showing linear relationships among elemental fraction descriptors and the target ΔH_f. (b) Optuna hyperparameter optimization history, illustrating completed and pruned trials and highlighting the best-performing configuration. (c) Training and validation loss curves with early stopping, demonstrating smooth convergence and model stabilization. (d) Predicted *versus* true ΔH_f for both training and validation datasets, with corresponding RMSE and R^2 metrics indicating strong model performance.

state, organized the workflow, and extracted stress tensors from the completed VASP calculations. Linear elasticity analysis yielded C_{11} = 216.62 GPa, C_{12} = 151.67 GPa, and C_{44} = 105.90 GPa at 0 K. These values follow the expected trends and compare reasonably with established experimental results at room temperature (C_{11} = 168.4 GPa, C_{12} = 121.4 GPa, and C_{44} = 75.4 GPa),[75] accounting for thermal softening effects. This example highlights Masgent's reliability in managing multi-step, tensor-level analyses that typically require substantial scripting, data extraction, and manual post-processing.

4.3.6. Case study 6: ML modeling of stability for high-entropy Al–Co–Cr–Fe–Ni alloys. Masgent was also applied to a lightweight ML task aimed at predicting the formation enthalpies (ΔH_f) of high-entropy Al–Co–Cr–Fe–Ni alloys, adapted from an established dataset calculated from DFT.[68] The correlation matrix in Fig. 13a provides initial insight into descriptor relevance, highlighting a positive correlation between Cr content and ΔH_f, as well as a negative correlation with Al content. As shown in Fig. 13b, Masgent automatically tuned the model hyperparameters using Optuna-based Bayesian optimization, efficiently identifying high-performing configurations while pruning unpromising trials. A neural network regressor was then trained using Masgent's standardized pipeline. The training and validation loss curves in Fig. 13c exhibit consistent behavior, demonstrating a well-regularized model. The predicted *versus* true ΔH_f values in Fig. 13d fall close to the ideal diagonal, with final metrics of RMSE ≈ 1.48 eV per atom (training) and RMSE ≈ 4.98 eV per atom (validation), along with $R^2 > 0.89$. This case study highlights Masgent's ability to support rapid, data-driven modeling workflows for alloy design, enabling users to explore feature relationships, optimize hyperparameters, and evaluate predictive accuracy, all within a unified platform that complements its physics-based simulation tools.

5. Discussion and future work

Masgent demonstrates that AI-assisted automation can substantially streamline the setup, execution, and analysis of materials simulations. By unifying DFT automation, MLP simulations, ML utilities, and natural-language interaction within a single platform, Masgent lowers barriers for new practitioners while accelerating the productivity of expert users. At the same time, the current implementation highlights opportunities for further refinement and expansion.

5.1 Strengths and practical impact

The integration of structured tool-calling with a natural-language interface represents a notable step forward in computational materials science. Compared with traditional scripting-based approaches, Masgent offers several practical advantages:

5.1.1 Dramatically reduced setup time. Complete simulation workflows can be prepared in seconds to minutes, replacing hours or even days of manual setup and scripting.

5.1.2 Lowered technical barriers. The AI agent enables users with minimal programming or HPC experience to construct advanced simulations reliably.

5.1.3 Enhanced reproducibility. Standardized templates and schema-validated input generation minimize human error and ensure consistent simulation protocols across users and projects.

5.1.4 Hybrid DFT + MLP capabilities. Seamless switching between fast MLPs and DFT enables rapid hypothesis testing and efficient pre-screening before expensive calculations.

5.1.5 Unified data-centric analysis tools. Integrated ML utilities support from-scratch model development and pre-trained model applications for materials science.

5.2 Limitations and future directions

Despite its advantages, Masgent still faces certain limitations that present clear opportunities for future development:

5.2.1 Domain awareness and model specialization. Masgent currently relies on general-purpose LLMs, and ambiguous or underspecified prompts may still require user clarification. Future versions will focus on domain-aware prompting strategies, fine-tuned materials-science LLMs, and task-specific instruction tuning to improve robustness, accuracy, and accessibility.

5.2.2 Standardized LLM-tool interfaces. Recent developments in the LLM ecosystem have introduced standards such as the Model Context Protocol (MCP), which provides a unified interface for connecting language models with external tools, databases, and services. Future versions may incorporate MCP-compatible interfaces to standardize connections with external resources such as materials databases and computational services.

5.2.3 Responsible scientific control. Masgent is intentionally designed to avoid autonomous decision-making in areas requiring physical judgment, such as the selection of exchange–correlation functionals, Hubbard U parameters, chemical-potential limits, or convergence settings beyond standardized protocols. These choices remain explicitly user-controlled to ensure scientific validity and reproducibility.

5.2.4 Lack of direct job execution. Although Masgent prepares complete workflows, it does not execute VASP jobs or manage HPC queues. Integrating workflow managers such as FireWorks[76] will enable automated job submission, monitoring, dependency handling, and error recovery at HPC scale.

5.2.5 Limited workflow coverage. While convergence tests, EOS fitting, elasticity calculations, AIMD, and NEB workflows are supported, more advanced routines, such as phonon calculations, defect thermodynamics, charge-state analysis, and band-structure workflows, remain under development. Extending coverage to the workflow library is a key priority for future releases.

5.2.6 Limited MLP support. Current potentials are constrained by the scope of their training datasets, reducing reliability for exotic chemistries or extreme conditions. Incorporating additional models (*e.g.*, M3GNet, MACE) and

supporting user-supplied MLPs will broaden chemical coverage and improve transferability.

5.2.7 Limited ML model support. Masgent's ML utilities focus on lightweight models for feature analysis, data augmentation, regression, and a small set of embedded pre-trained predictors. Future enhancements will expand the library to include a broader range of pre-trained models covering additional alloy systems and materials properties, as well as more advanced learning frameworks such as graph neural networks, automated feature engineering, uncertainty quantification, and improved model explainability, to support more sophisticated data-driven research.

6. Summary

Masgent provides a unified, AI-driven framework that streamlines computational materials research by integrating automated DFT workflow construction, fast MLP simulations, lightweight ML tools, and natural-language interaction. Through schema-validated tool-calling, modular workflow templates, and intelligent parameter handling, Masgent reduces manual effort and user error while enabling rapid, reproducible, and scalable simulation pipelines. Benchmark tests show that Masgent can prepare intricate workflows within seconds to minutes, reproduce established DFT methodologies, and deliver accurate, low-cost approximations *via* MLP-based calculations.

While certain capabilities, such as phonon workflows and deeper autonomous reasoning, remain under development, the current platform already offers substantial improvements in usability, efficiency, and accessibility. Looking ahead, Masgent provides a robust foundation for incorporating active learning, autonomous optimization loops, expanded workflow coverage, and deeper integration with HPC workflow managers, positioning it as a next-generation computational assistant capable of accelerating both exploratory and production-level materials simulations.

Author contributions

During the preparation of this work the author(s) used ChatGPT to improve the clarity, grammar, and overall readability of the English writing. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Conflicts of interest

The authors declare no competing interests.

Data availability

All data generated or analyzed during this study are included in this published article and its associated online examples, available at https://github.com/IMPDGroup/Masgent/tree/main/examples. The Masgent code developed in this work, along with detailed instructions for the usage, is publicly available *via* GitHub at https://github.com/IMPDGroup/Masgent and archived on Zenodo at https://doi.org/10.5281/zenodo.19456832.

Appendix A: Current functional overview

- Density functional theory (DFT) simulations
  - ○ Structure preparation & manipulation
  - – Generate POSCAR from chemical formula
  - – Convert POSCAR coordinates (direct ↔ Cartesian)
  - – Convert structure file formats (CIF, POSCAR, XYZ)
  - – Generate structures with defects (vacancies, substitutions, interstitials)
  - – Generate supercells
  - – Generate special quasirandom structures (SQS)
  - – Generate surface slabs
  - – Generate interface structures
  - – Visualize structures
  - ○ VASP input file preparation
  - – Prepare full VASP input files (INCAR, KPOINTS, POTCAR, POSCAR)
  - – Generate INCAR templates
  - • MPMetalRelaxSet: suggested for metallic structure relaxation
  - • MPRelaxSet: suggested for structure relaxation
  - • MPStaticSet: suggested for static calculations
  - • MPNonSCFBandSet: suggested for non-self-consistent field calculations (Band structure)
  - • MPNonSCFDOSSet: suggested for non-self-consistent field calculations (density of states)
  - • MPMDSet: suggested for molecular dynamics simulations
  - – Generate KPOINTS with specified accuracy
  - – Generate HPC job submission script
  - ○ Standard VASP workflow preparation
  - – Convergence test (ENCUT, KPOINTS)
  - – Equation of state (EOS)
  - – Elastic constants calculations
  - – *Ab initio* molecular dynamics (AIMD)
  - – Nudged elastic band (NEB) calculations
  - ○ Standard VASP workflow output analysis
  - – Convergence test analysis
  - – Equation of state (EOS) analysis
  - – Elastic constants analysis
  - – *Ab initio* molecular dynamics (AIMD) analysis
  - – Nudged elastic band (NEB) analysis
  - • Fast simulations using machine learning potentials (MLPs)
  - ○ Supported MLPs:
  - – SevenNet
  - – CHGNet
  - – Orb-v3
  - – MatterSim
  - ○ Implemented simulations for all MLP:
  - – Single point energy calculation
  - – Equation of state (EOS) calculation
  - – Elastic constants calculation

– Molecular dynamics dimulation (NVT)
○ Simple machine learning for materials science
○ Data preparation & feature analysis
– Feature analysis and visualization
– Dimensionality reduction (if too many features)
– Data augmentation (if limited data)
○ Model design & hyperparameter tuning
○ Model training & evaluation
○ Model retraining with new data
○ Pre-trained models prediction
– Mechanical properties prediction in Sc-modified Al–Mg–Si alloys
– Phase stability & elastic properties prediction in Al–Co–Cr–Fe–Ni high-entropy alloys

Acknowledgements

The authors acknowledge the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program for computational resources provided under Award No. MAT240062. The authors thank ACCESS for their support, resources, and infrastructure that enabled this research. Masgent builds on the open-source materials ecosystem, including ASE, Pymatgen, Icet, and modern machine learning potentials. We thank the developers of these tools for making advanced materials simulation possible.

References

1 K. Gubaev, E. V. Podryabinkin, G. L. Hart and A. V. Shapeev, Accelerating high-throughput searches for new alloys with active learning of interatomic potentials, *Comput. Mater. Sci.*, 2019, **156**, 148–156.
2 P. Korotaev, I. Novoselov, A. Yanilkin and A. Shapeev, Accessing thermal conductivity of complex compounds by machine learning interatomic potentials, *Phys. Rev. B*, 2019, **100**(14), 144308.
3 B. Mortazavi, E. V. Podryabinkin, I. S. Novikov, S. Roche, T. Rabczuk, X. Zhuang and A. V. Shapeev, Efficient machine-learning based interatomic potentials for exploring thermal conductivity in two-dimensional materials, *J. Phys.: Mater.*, 2020, **3**(2), 02LT02.
4 V. Ladygin, P. Y. Korotaev, A. Yanilkin and A. Shapeev, Lattice dynamics simulation using machine learning interatomic potentials, *Comput. Mater. Sci.*, 2020, **172**, 109333.
5 B. Mortazavi, E. V. Podryabinkin, S. Roche, T. Rabczuk, X. Zhuang and A. V. Shapeev, Machine-learning interatomic potentials enable first-principles multiscale modeling of lattice thermal conductivity in graphene/borophene heterostructures, *Mater. Horiz.*, 2020, **7**(9), 2359–2367.
6 E. V. Podryabinkin, E. V. Tikhonov, A. V. Shapeev and A. R. Oganov, Accelerating crystal structure prediction by machine-learning interatomic potentials with active learning, *Phys. Rev. B*, 2019, **99**(6), 064114.
7 S. Yang, G. Liu and Y. Zhong, Revisit the VEC criterion in high entropy alloys (HEAs) with high-throughput ab initio calculations: A case study with Al-Co-Cr-Fe-Ni system, *J. Alloys Compd.*, 2022, 165477.
8 S. Yang, G. Liu and Y. Zhong, Ab initio investigations on the electronic properties and stability of Cu-substituted lead apatite (LK-99) family with different doping concentrations (x= 0, 1, 2), *Mater. Today Commun.*, 2023, **37**, 107379.
9 G. Liu, S. Yang and Y. Zhong, A Computational Framework for Interface Design Using Lattice Matching, Machine Learning Potentials, and Active Learning: A Case Study on $LaCoO_3/La_2NiO4$, *Mater. Today Phys.*, 2025, 101940.
10 G. Kresse and J. Furthmüller, Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1996, **54**(16), 11169.
11 G. Kresse and J. Furthmüller, Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set, *Comput. Mater. Sci.*, 1996, **6**(1), 15–50.
12 S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson and G. Ceder, Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis, *Comput. Mater. Sci.*, 2013, **68**, 314–319.
13 A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer and C. Hargus, The atomic simulation environment—a Python library for working with atoms, *J. Phys.: Condens. Matter*, 2017, **29**(27), 273002.
14 V. Wang, N. Xu, J.-C. Liu, G. Tang and W.-T. Geng, VASPKIT: A user-friendly interface facilitating high-throughput computing and analysis using VASP code, *Comput. Phys. Commun.*, 2021, **267**, 108033.
15 W. Yi, G. Tang, X. Chen, B. Yang and X. Liu, qvasp: A flexible toolkit for VASP users in materials simulations, *Comput. Phys. Commun.*, 2020, **257**, 107535.
16 K. Mathew, J. H. Montoya, A. Faghaninia, S. Dwarakanath, M. Aykol, H. Tang, I.-h. Chu, T. Smidt, B. Bocklund and M. Horton, Atomate: A high-level interface to generate, execute, and analyze computational materials science workflows, *Comput. Mater. Sci.*, 2017, **139**, 140–152.
17 A. M. Ganose, H. Sahasrabuddhe, M. Asta, K. Beck, T. Biswas, A. Bonkowski, J. Bustamante, X. Chen, Y. Chiang and D. C. Chrzan, Atomate2: Modular workflows for materials science, *Digital Discovery*, 2025, 1944–1973.
18 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss and V. Dubourg, Scikit-learn: Machine learning in Python, *J. Mach. Learn. Res.*, 2011, **12**, 2825–2830.
19 M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving and M. Isard, TensorFlow: a system for large-scale machine learning, *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016, pp. 265–283.
20 A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein and L. Antiga, PyTorch: An imperative style, high-performance deep

learning library, *Advances in Neural Information Processing Systems*, 2019, vol. 32.

21 A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner and G. Ceder, Commentary: The Materials Project: A materials genome approach to accelerating materials innovation, *APL Mater.*, 2013, **1**(1), 011002.

22 S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl and C. Wolverton, The Open Quantum Materials Database (OQMD): assessing the accuracy of DFT formation energies, *npj Comput. Mater.*, 2015, **1**(1), 1–15.

23 J. E. Saal, S. Kirklin, M. Aykol, B. Meredig and C. Wolverton, Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD), *JOM*, 2013, **65**, 1501–1509.

24 Y. Park, J. Kim, S. Hwang and S. Han, Scalable parallel algorithm for graph neural network interatomic potentials in molecular dynamics simulations, *J. Chem. Theory Comput.*, 2024, **20**(11), 4857–4868.

25 B. Deng, P. Zhong, K. Jun, J. Riebesell, K. Han, C. J. Bartel and G. Ceder, CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling, *Nat. Mach. Intell.*, 2023, **5**(9), 1031–1041.

26 M. Neumann, J. Gin, B. Rhodes, S. Bennett, Z. Li, H. Choubisa, A. Hussey and J. Godwin, Orb: A fast, scalable neural network potential, *arXiv*, 2024, preprint, arXiv:2410.22570, DOI: 10.48550/arXiv.2410.22570.

27 B. Rhodes, S. Vandenhaute, V. Šimkus, J. Gin, J. Godwin, T. Duignan and M. Neumann, Orb-v3: atomistic simulation at scale, *arXiv*, 2025, preprint, arXiv:2504.06231, DOI: 10.48550/arXiv.2504.06231.

28 H. Yang, C. Hu, Y. Zhou, X. Liu, Y. Shi, J. Li, G. Li, Z. Chen, S. Chen and C. Zeni, MatterSim: A deep learning atomistic model across elements, temperatures and pressures, *arXiv*, 2024, preprint, arXiv:2405.04967, DOI: 10.48550/arXiv.2405.04967.

29 C. Chen and S. P. Ong, A universal graph deep learning interatomic potential for the periodic table, *Nat. Comput. Sci.*, 2022, **2**(11), 718–728.

30 I. Batatia, D. P. Kovacs, G. Simm, C. Ortner and G. Csányi, MACE: Higher order equivariant message passing neural networks for fast and accurate force fields, *Adv. Neural Inf. Process. Syst.*, 2022, **35**, 11423–11436.

31 S. Takamoto, C. Shinagawa, D. Motoki, K. Nakago, W. Li, I. Kurata, T. Watanabe, Y. Yayama, H. Iriguchi and Y. Asano, Towards universal neural network potential for material discovery applicable to arbitrary combination of 45 elements, *Nat. Commun.*, 2022, **13**(1), 2991.

32 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, *Advances in Neural Information Processing Systems*, 2017, vol. 30.

33 T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry and A. Askell, Language models are few-shot learners, *Adv. Neural Inf. Process. Syst.*, 2020, **33**, 1877–1901.

34 J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu and D. Amodei, Scaling laws for neural language models, *arXiv*, 2020, preprint, arXiv:2001.08361, DOI: 10.48550/arXiv.2001.08361.

35 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, *Adv. Neural Inf. Process. Syst.*, 2023, **36**, 68539–68551.

36 H. Yang, S. Yue and Y. He, Auto-gpt for online decision making: Benchmarks and additional opinions, *arXiv*, 2023, preprint, arXiv:2306.02224, DOI: 10.48550/arXiv.2306.02224.

37 S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan and Y. Cao, React: Synergizing reasoning and acting in language models, *The Eleventh International Conference on Learning Representations*, 2022.

38 G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan and A. Anandkumar, Voyager: An open-ended embodied agent with large language models, *arXiv*, 2023, preprint, arXiv:2305.16291, DOI: 10.48550/arXiv.2305.16291.

39 S. Gao, R. Zhu, P. Sui, Z. Kong, S. Aldogom, Y. Huang, A. Noori, R. Shamji, K. Parvataneni and T. Tsiligkaridis, Democratizing AI scientists using ToolUniverse, *arXiv*, 2025, preprint, arXiv:2509.23426, DOI: 10.48550/arXiv.2509.23426.

40 K.-D. Inc., *Claude Scientific Skills: A Comprehensive Collection of Scientific Tools for Claude AI*, 2026. https://github.com/K-Dense-AI/claude-scientific-skills.

41 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.

42 J. Liu, T. Zhu, C. Ye, Z. Fang, H. Weng and Q. Wu, VASPilot: MCP-facilitated multi-agent intelligence for autonomous VASP simulations, *Chin. Phys. B*, 2025, **34**(11), 117106.

43 M. Soleymanibrojeni, R. Aydin, D. Guedes-Sobrinho, A. C. Dias, M. J. Piotrowski, W. Wenzel and C. R. C. Rêgo, GENIUS: An Agentic AI Framework for Autonomous Design and Execution of Simulation Protocols, *arXiv*, 2025, preprint, arXiv:2512.06404, DOI: 10.48550/arXiv.2512.06404.

44 H. Chen, J. Qiu, Y. S. Tew and X. Wang, CatMaster: An Agentic Autonomous System for Computational Heterogeneous Catalysis Research, *arXiv*, 2026, preprint, arXiv:2601.13508, DOI: 10.48550/arXiv.2601.13508.

45 F. Yang and J. D. Evans, QUASAR: A Universal Autonomous System for Atomistic Simulation and a Benchmark of Its Capabilities, *arXiv*, 2026, preprint, arXiv:2602.00185, DOI: 10.48550/arXiv.2602.00185.

46 Y. Zou, A. H. Cheng, A. Aldossary, J. Bai, S. X. Leong, J. A. Campos-Gonzalez-Angulo, C. Choi, C. T. Ser, G. Tom and A. Wang, El Agente: An autonomous agent for quantum chemistry, *Matter*, 2025, **8**(7), 102263.

47 T. D. Pham, A. Tanikanti and M. Keçeli, ChemGraph as an agentic framework for computational chemistry workflows, *Commun. Chem.*, 2026, **9**(1), 33.

48 A. D. McNaughton, G. K. Sankar Ramalaxmi, A. Kruel, C. R. Knutson, R. A. Varikoti and N. Kumar, Cactus: Chemistry agent connecting tool usage to science, *ACS Omega*, 2024, **9**(46), 46563–46573.
49 X. Huang, J. Chen, Y. Fei, Z. Li, P. Schwaller and G. Ceder, Cascade: Cumulative agentic skill creation through autonomous development and evolution, *arXiv*, 2025, preprint, arXiv:2512.23880, DOI: 10.48550/arXiv.2512.23880.
50 J. Hu, H. Nawaz, Y. Rui, L. Chi, A. Ullah and P. O. Dral, Aitomia: Your Intelligent Assistant for AI-Driven Atomistic and Quantum Chemical Simulations, *arXiv*, 2025, preprint, arXiv:2505.08195, DOI: 10.48550/arXiv.2505.08195.
51 J. Bai, A. Aldossary, T. Swanick, M. Müller, Y. Kang, Z. Zhang, J. W. Lee, T. W. Ko, V. Bernales and A. Aspuru-Guzik, El Agente Gráfico: Structured Execution Graphs for Scientific Agents, *arXiv*, 2026, preprint, arXiv:2602.17902, DOI: 10.48550/arXiv.2602.17902.
52 M. Ding, C. Huang, Y. Hu, Y. Li, Z. Lu, X. Yu, D. Zhang, W. Zhai, T. Zhu and Q. Gu, Automating Computational Chemistry Workflows via OpenClaw and Domain-Specific Skills, *arXiv*, 2026, preprint, arXiv:2603.25522, DOI: 10.48550/arXiv.2603.25522.
53 A. Maćkiewicz and W. Ratajczak, Principal components analysis (PCA), *Comput. Geosci.*, 1993, **19**(3), 303–342.
54 C. Doersch, Tutorial on variational autoencoders, *arXiv*, 2016, preprint, arXiv:1606.05908, DOI: 10.48550/arXiv.1606.05908.
55 D. P. Kingma and M. Welling, An introduction to variational autoencoders, *Found. Trends® Mach. Learn.*, 2019, **12**(4), 307–392.
56 X. Gong, B. Tang, R. Zhu, W. Liao and L. Song, Data augmentation for electricity theft detection using conditional variational auto-encoder, *Energies*, 2020, **13**(17), 4291.
57 T. Akiba, S. Sano, T. Yanase, T. Ohta and M. Koyama, Optuna: A next-generation hyperparameter optimization framework, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2623–2631.
58 N. Rego and D. Koes, 3Dmol.js: molecular visualization with WebGL, *Bioinformatics*, 2015, **31**(8), 1322–1324.
59 K. Momma and F. Izumi, VESTA 3 for three-dimensional visualization of crystal, volumetric and morphology data, *Appl. Crystallogr.*, 2011, **44**(6), 1272–1276.
60 K. Momma and F. Izumi, VESTA: a three-dimensional visualization system for electronic and structural analysis, *Appl. Crystallogr.*, 2008, **41**(3), 653–658.
61 J.-X. Shen and J. Varley, Pymatgen-analysis-defects: A Python package for analyzing point defects in crystalline materials, *J. Open Source Softw.*, 2024, **9**(93), 5941.
62 M. Ångqvist, W. A. Muñoz, J. M. Rahm, E. Fransson, C. Durniak, P. Rozyczko, T. H. Rod and P. Erhart, ICET–a Python library for constructing and sampling alloy cluster expansions, *Adv. Theory Simul.*, 2019, **2**(7), 1900015.
63 A. Zur and T. McGill, Lattice match: An application to heteroepitaxy, *J. Appl. Phys.*, 1984, **55**(2), 378–386.
64 F. D. Murnaghan, Finite deformations of an elastic solid, *Am. J. Math.*, 1937, **59**(2), 235–260.
65 S. Watanabe, Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance, *arXiv*, 2023, preprint, arXiv:2304.11127, DOI: 10.48550/arXiv.2304.11127.
66 J. Gao, J. Zhong, G. Liu, S. Zhang, J. Zhang, Z. Liu, B. Song and L. Zhang, Accelerated discovery of high-performance Al-Si-Mg-Sc casting alloys by integrating active learning with high-throughput CALPHAD calculations, *Sci. Technol. Adv. Mater.*, 2023, **24**(1), 2196242.
67 G. Liu, J. Gao, C. Che, Z. Lu, W. Yi and L. Zhang, Optimization of casting means and heat treatment routines for improving mechanical and corrosion resistance properties of A356-0.54 Sc casting alloy, *Mater. Today Commun.*, 2020, **24**, 101227.
68 G. Liu, S. Yang and Y. Zhong, High-Entropy Materials Design by Integrating the First-Principles Calculations and Machine Learning: A Case Study in the Al-Co-Cr-Fe-Ni System, *High Entropy Alloys & Mater.*, 2024, 1–14.
69 W. Bond and W. Kaiser, Interstitial versus substitutional oxygen in silicon, *J. Phys. Chem. Solids*, 1960, **16**(1–2), 44–45.
70 *Materials Data on Si by Materials Project*, United States, 2020, DOI: 10.17188/1190959.
71 C. Zhang, H. Zhuang, Z. Qi, X. Liu and Y. Ren, The effect of defects for the ion transport of Li_3ScCl_6 and Li_3InCl_6 with the interface of lithium metal anode: A first-principles study, *Mater. Today Commun.*, 2023, **35**, 105764.
72 Z. Du, Z. Zhang, A. Niemczyk, A. Olszewska, N. Chen, K. Świerczek and H. Zhao, Unveiling the effects of A-site substitutions on the oxygen ion migration in $A_{2-x}A'_xNiO_{4+\delta}$ by first principles calculations, *Phys. Chem. Chem. Phys.*, 2018, **20**(33), 21685–21692.
73 S. Yang, G. Liu, Y.-L. Lee, J.-M. Bassat, J. Gamon, A. Villesuzanne, J. Pietras, X.-D. Zhou and Y. Zhong, A systematic ab initio study of vacancy formation energy, diffusivity, and ionic conductivity of $Ln_2NiO_{4+\delta}$ (Ln= La, Nd, Pr), *J. Power Sources*, 2023, **576**, 233200.
74 S. Yang, G. Liu, W. Li, E. M. Sabolsky, X. Liu and Y. Zhong, Ab initio study on the effect of A-site doping on the stability, equilibrium volume, activation energy barrier, and oxygen diffusivity in La2-xAxNiO4+δ, *Int. J. Hydrogen Energy*, 2025, **119**, 239–251.
75 H. Ledbetter and E. Naimon, Elastic properties of metals and alloys. II. Copper, *J. Phys. Chem. Ref. Data*, 1974, **3**(4), 897–935.
76 A. Jain, S. P. Ong, W. Chen, B. Medasani, X. Qu, M. Kocher, M. Brafman, G. Petretto, G. M. Rignanese and G. Hautier, FireWorks: a dynamic workflow system designed for high-throughput applications, *Concurrency Comput. Pract. Ex.*, 2015, **27**(17), 5037–5059.