PhysAgent: A Multi-Agent Large Language Model Framework for Physics Data Analysis and Simulation

Xiao-Qi Han[1,2,†], Ze-Feng Gao[1,2,†,*], Peng-Jie Guo[1,2], and Zhong-Yi Lu[1,2,*]
1. School of Physics, Renmin University of China, Beijing, 100872, P.R. China and
2. AI-Driven Quantum Materials Research Center, School of Physics, Renmin University of China, Beijing, 100872, P.R. China

The inference of physical laws has traditionally relied on human intuition, analytical reasoning, and experimental observation. However, modern physics research is increasingly constrained by challenges such as high specialization, fragmented workflows, and limited computational resources, which impede scientific progress. To address these issues, we introduce **PhysAgent**, a novel multi-agent system powered by large language models (LLMs), designed to execute end-to-end scientific workflows—from hypothesis generation and computational modeling to data analysis and reporting. PhysAgent's innovation lies in its specialized agent collaboration: a Mentor Agent guides scientific reasoning through Socratic questioning, a Student Agent handles technical execution (e.g., code implementation, DFT calculations), and a Leader Agent dynamically optimizes task scheduling and resource allocation. Integrating domain-specific tools like first-principles simulations (e.g., Quantum ESPRESSO, VASP), PhysAgent aims to improve reproducibility while maintaining human-in-the-loop refinement. We demonstrate PhysAgent as a proof-of-concept framework for recovering known physical relations from controlled data, including Kepler-law recovery from orbital trajectories and Newton-law recovery from force-motion data. Furthermore, it extends to ab initio materials modeling, automating electronic structure workflows (e.g., GaAs band-structure and density-of-states calculations). In addition, PhysAgent simulates complex real-world phenomena, such as raindrop flow on train windows. By harmonizing LLM-driven planning with domain-specific tools, PhysAgent provides a reproducible multi-agent workflow for physics data analysis and simulation. We provide a demonstration video of *PhysAgent* to illustrate its workflow and capabilities: https://b23.tv/emOilkC.

Keywords: multi-agent system, large language models, first-principles simulations, automated scientific inference

† These authors contributed equally to this work.
* Corresponding authors. E-mail: zfgao@ruc.edu.cn and zlu@ruc.edu.cn.

INTRODUCTION

Physics plays a central role across diverse domains, including energy [1], materials [2–5], information [6], and life sciences [7]. In recent years, advances in experimental techniques and computational capabilities have led to an unprecedented surge in data volume [8–11], model complexity [12–14], and interdisciplinary challenges [15]. However, the progress of physics research is increasingly constrained by high specialization, fragmented workflows, and limitations in computational resources. From theoretical model development and numerical simulation [16–18] to experimental validation, modern physics research demands the integration of multidisciplinary knowledge [19–21], the orchestration of intricate workflows [22–24], and exhaustive exploration of vast parameter spaces. The growing disparity between the complexity of physical systems [25] and the limited cognitive and computational bandwidth available to researchers highlights the urgent need for a new research paradigm—an intelligent system capable of integrating knowledge, autonomously exploring models, and accelerating scientific inference.

In recent years, the rapid advancement of large language models (LLMs), such as GPT-4 [6] and DeepSeek [26], has led to the emergence of AI agents [27–30]—autonomous or semi-autonomous systems powered by LLMs as their central reasoning engines. Representative frameworks include MetaGPT [28], which model software engineering processes through role-based agent collaboration; AutoGen [29], which enables customizable multi-agent conversations with enhanced reasoning capabilities; and LangChain [30], a widely adopted toolkit for building LLM-based agents that can interact with tools, APIs, and memory. These AI agents are profoundly transforming multiple scientific and engineering domains. In particular, fields such as law [31], biomedicine [32], software engineering [28, 33], and materials science [34] have witnessed significant progress, as these agents automate repetitive tasks, greatly enhance research efficiency, and enable breakthroughs that were previously considered unattainable. As a result, AI agents are becoming indispensable tools for accelerating scientific discovery and engineering innovation.

However, existing agent frameworks (such as MetaGPT [28] and AutoGen [29]) face significant limitations in the context of physics research. They lack deep causal reasoning capabilities grounded in physical theory [35, 36], are unable to natively integrate special-

ized tools like first-principles simulations, and struggle with dynamic task scheduling across multiple scales [37]. Specifically, these systems often rely on predefined operation sequences (e.g., parameter sweeps [38, 39]), making it difficult to construct goal-oriented problem-solving workflows dynamically. Moreover, their limited integration with the physics computing ecosystem (e.g., Linux/Python/MPI) hinders seamless execution from electronic structure [22] calculations to molecular dynamics [37, 40, 41]. Critically, most frameworks lack built-in mechanisms for enforcing physical consistency, such as energy conservation checks, which may lead to unphysical results. Overcoming these challenges requires the integration of three key capabilities: scientific agents with causal reasoning, execution frameworks natively compatible with domain-specific computing platforms, and safety mechanisms that enforce physical consistency.

In this work, we introduce **PhysAgent**, a multi-agent system powered by LLMs [6, 26, 42], designed to execute end-to-end scientific workflows—from hypothesis generation and computational modeling to data analysis and reporting. PhysAgent's core innovation lies in its specialized agent collaboration: a Mentor Agent guides scientific reasoning via Socratic questioning, a Student Agent handles technical execution (e.g., code implementation, density functional theory (DFT) calculations), and a Leader Agent dynamically optimizes task scheduling and resource allocation. This triad operates within a physics-grounded paradigm, integrating first-principles simulations (e.g., Quantum ESPRESSO, VASP) and data-driven validation to improve reproducibility. In the present work, PhysAgent is evaluated as a proof-of-concept framework for controlled law recovery rather than as proof of de novo discovery of unknown physical laws. We demonstrate Kepler-law recovery from orbital data (Figure 2) and Newton-law recovery from force-motion data (Figure 3), extend the workflow to ab initio materials modeling (e.g., GaAs electronic-structure calculations, Figure 4), and simulate raindrop flow on train windows (Figure 5). By harmonizing LLM-driven planning with domain-specific tools, PhysAgent provides a reproducible multi-agent workflow for physics data analysis and simulation.

METHODS

Architecture and Workflow of the PhysAgent System

As depicted in Figure 1, the PhysAgent system is powered by an LLM that serves as the agent's brain, enabling capabilities such as experimental planning, tool usage, and action execution. This brain is implemented through direct API calls, supporting a range of LLMs, including GPT-4, Kimi, Doubao, Llama, and DeepSeek. Similar to existing agents [27], PhysAgent is also equipped with capabilities such as learning [44, 45], reasoning [46, 47], memory [48, 49], reward [50], perception [51], and action [52].

At the initiation of a task, PhysAgent autonomously manages project planning, tool installation, and action execution. These actions encompass operating Linux systems, writing and executing Python scripts, performing web searches and downloads, debugging code, conducting data analysis, invoking software packages such as Quantum ESPRESSO or VASP for electronic structure calculations of materials, and drafting experimental reports and academic papers. The PhysAgent multi-agent system integrates human users and specialized agents, each tasked with distinct responsibilities within the scientific workflow, the detail of these prompt can be see in Supplementary Materials.

- **Human User**: Provides experimental data and defines scientific problems
- **Mentor Agent**: Guides hypothesis formation using Socratic questioning and bias mitigation strategies
- **Student Agent**: Performs numerical computation, code implementation, and technical problem-solving
- **Leader Agent**: Optimizes task scheduling via dynamic resource allocation

In the scientific workflow of PhysAgent, collaboration among multiple agents enables task completion. The Human user defines the scientific objective and research question, generates simulated experimental data, and provides feedback at key stages. The Mentor Agent guides the analytical process using heuristic strategies, formulates hypotheses, and offers feedback to the Student Agent to facilitate autonomous pattern inferring while avoiding prior biases. The Student Agent is responsible for building analysis pipelines, implementing and verifying code, performing numerical simulations, processing and visualizing data, and drafting reports and research papers. As illustrated in Figure 1, the Mentor Agent and Student Agent engage in multiple rounds of feedback and iterative refinement throughout the analytical process, ensuring continual hypothesis improvement and enhanced data interpretation. The Leader Agent manages system resources and orchestrates agent collaboration to ensure efficiency and order. With a closed-loop design encompassing problem formulation, data generation, analysis execution, manuscript writing, and workflow coordination, PhysAgent provides an end-to-end intelligent framework for scientific inferring, offering a new paradigm for solving complex scientific problems.

Here, Figure 1 illustrates an example of materials electronic structure calculations (GaAs) collaboratively completed by PhysAgent. Compared to traditional single-agent models and general multi-agent systems, PhysAgent demonstrates advantages in handling complex, multi-step, cross-scale physics research tasks

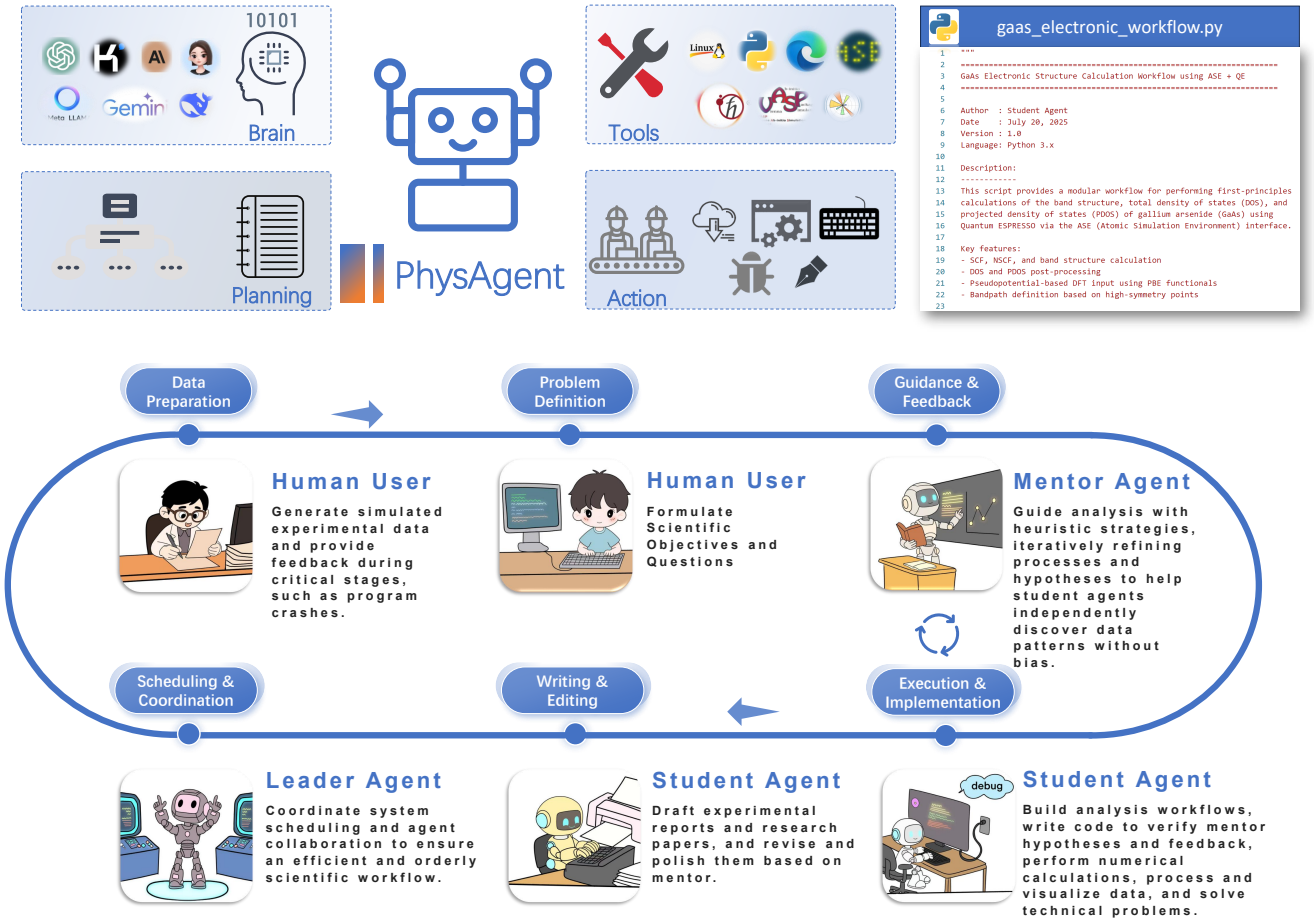


FIG. 1. **Architecture and Workflow of the PhysAgent System.** Schematic of the PhysAgent system, powered by an LLM acting as the agent's brain, enabling experimental planning, tool usage, and action execution via API calls to models like GPT-4 [6], KIMI [42], Doubao, Llama [43], and DeepSeek [26]. The multi-agent system integrates human users, who provide data and define problems, with specialized agents: the Mentor Agent guides hypothesis formation, the Student Agent handles computation and coding, and the Leader Agent optimizes task scheduling. The figure illustrates a material's electronic structure calculation script completed through multi-round agent collaboration.

through clear role division, effective knowledge sharing, and efficient strategy coordination. Experimental results show that the system can dynamically construct complete problem-solving workflows—from structure modeling to final property analysis—seamlessly integrate specialized toolchains such as ASE and DFT software, and produce key scientific outputs that meet research standards, including structural models, convergence curves, charge density distributions, and band structure/density of states plots.

To enable the autonomous execution of complex scientific workflows, PhysAgent is equipped with a set of core cognitive functions that emulate fundamental aspects of human intelligence. These capabilities include learning, reasoning, memory, reward processing, perception, and action execution. Each function plays a distinct role in enhancing the agent's adaptability, decision-making, and interaction with dynamic environments. The following provides a detailed explanation of these components.

- **Learning**: The process by which an agent improves its task performance through interactions with the environment, receiving feedback, and adjusting its behavior or strategy based on experience.
- **Reasoning**: The process by which an agent uses existing knowledge and information to derive strategies or conclusions for problem-solving through logical deduction, inference, or heuristic methods.
- **Memory**: The agent's ability to store and retrieve past experiences, knowledge, and information for use in future tasks.
- **Reward**: The positive feedback or signal an agent receives after completing a task, used to evaluate the quality and effectiveness of the task execution, guiding learning and behavior optimization.

- **Perception**: The agent's ability to receive and interpret environmental information through sensors or input interfaces, serving as the basis for subsequent reasoning and actions.
- **Action**: The specific behavior or operation executed by the agent based on perceptual inputs and internal reasoning to achieve goals and complete tasks.

In the context of multi-agent systems, automated platforms demonstrate significant potential in task planning, code generation, and API development. As shown in Supplementary Materials, conventional frameworks such as AutoGPT and LangChain primarily focus on general-purpose task execution and code processing, lacking domain-specific capabilities for scientific or engineering applications. In contrast, MetaGPT supports automatic generation of product requirement documents (PRDs), technical design plans, and API interfaces, along with comprehensive code generation, precompilation execution, and role-based task management [28]. Building upon these general capabilities, PhysAgent further integrates domain-specific toolkits, hypothesis generation, data analysis, physical-law recovery, and complex simulations (e.g., raindrop flow), demonstrating end-to-end autonomous reasoning and experimental planning from theoretical hypotheses to numerical validation.

To further clarify the novelty of PhysAgent, we compare it with a representative multi-agent framework, MetaGPT. The comparison in Supplementary highlights the key differences between the MetaGPT framework for software development and our PhysAgent framework for scientific reasoning in physics. While MetaGPT relies on roles such as product manager, architect, and engineer to coordinate software engineering workflows, PhysAgent introduces domain-specialized agents (mentor, student, leader) to enable hypothesis-driven reasoning, technical execution, and dynamic task optimization. Moreover, the workflow evolves from sequential task decomposition to one that integrates Socratic questioning and domain-specific tools (e.g., Quantum ESPRESSO, VASP). Consequently, the outputs shift from conventional software artifacts to physics workflow outputs, including controlled physical-law recovery and complex simulations. This comparison demonstrates the adaptability of multi-agent systems and highlights their potential to extend beyond software engineering into scientific research.

RESULTS

PhysAgent recovers Kepler-law relations.

To demonstrate the capability of PhysAgent in controlled physical-law recovery, we selected a task based on Kepler's laws of planetary motion. PhysAgent is required to analyze simulated trajectory data and recover regularities, but the task should not be interpreted as discovery of unknown laws without prior knowledge. As shown in Figure 2(a), the task is initiated by a human user who numerically simulates the time-series three-dimensional trajectories of five planets in the solar system (Mercury, Venus, Earth, Mars, and Jupiter) under the gravitational influence of the Sun. The user then poses a scientific objective: Can the known Kepler-law relations be recovered from these controlled data?

In the PhysAgent multi-agent system, multiple agents collaboratively recovered Kepler's laws of planetary motion. As shown in Figure 2 (a), the mentor agent employed heuristic questioning to guide the student agent in exploring planetary motion from various perspectives. For instance, in analyzing orbit shapes, the mentor agent asked, "What is the shape of planetary orbits around the Sun? Circular or elliptical?" In examining velocity variations, it inquired, "Is the orbital speed of a planet constant? Is it faster when closer to the Sun or when farther away?" In recovering the relationship between period and distance, it posed, "Do planets farther from the Sun, such as Jupiter, require more time to complete an orbit?" These questions stimulated critical thinking in the student agent. Based on exploratory analysis of the provided trajectories and general physical reasoning, the student agent formulated hypotheses: stronger gravitational forces near the Sun lead to greater acceleration and higher planetary velocities; weaker forces farther from the Sun result in slower velocities; and planets with larger orbital radii, such as Jupiter, have longer orbital paths and thus extended periods. Using Python tools, the student agent designed experiments to analyze time and position data (`time`, `x`, `y`, `z`) from the dataset, computing orbital shapes, velocity variations, and orbital periods to test these hypotheses. During the experiments, the student agent addressed challenges such as data noise and numerical computation errors and utilized visualization tools to generate orbital plots, areal velocity curves, and period-radius relationship graphs.

As shown in Figure 2 (c), the orbits of Mercury, Venus, Earth, Mars, and Jupiter are elliptical, with the Sun at one focus, confirming Kepler's First Law. To make the recovered relations quantitatively explicit, we annotated the fitted metrics directly in Fig. 2(c–e). For the orbit-shape analysis in Fig. 2(c), each trajectory was fitted using a Sun-focused elliptical model. The maximum radial normalized root-mean-square error (NRMSE) across the five planets is only 0.171%, indicating that the simulated trajectories are accurately described by ellipses. The corresponding orbit-fit coefficients of determination are high for all planets, with $R^2 = 0.9998$ for Mercury, 0.9670 for Venus, 0.9965 for Earth, 0.9999 for Mars, and 0.9999 for Jupiter. The relatively lower R^2 for Venus arises from its nearly circular orbit, where the radial variation is small and therefore more sensitive to coordinate noise. As depicted in Figure 2 (d), the areal velocity of each planet is approximately constant, indicating that the line connecting a planet to the Sun sweeps equal areas in equal times, consistent with Kepler's Second Law.

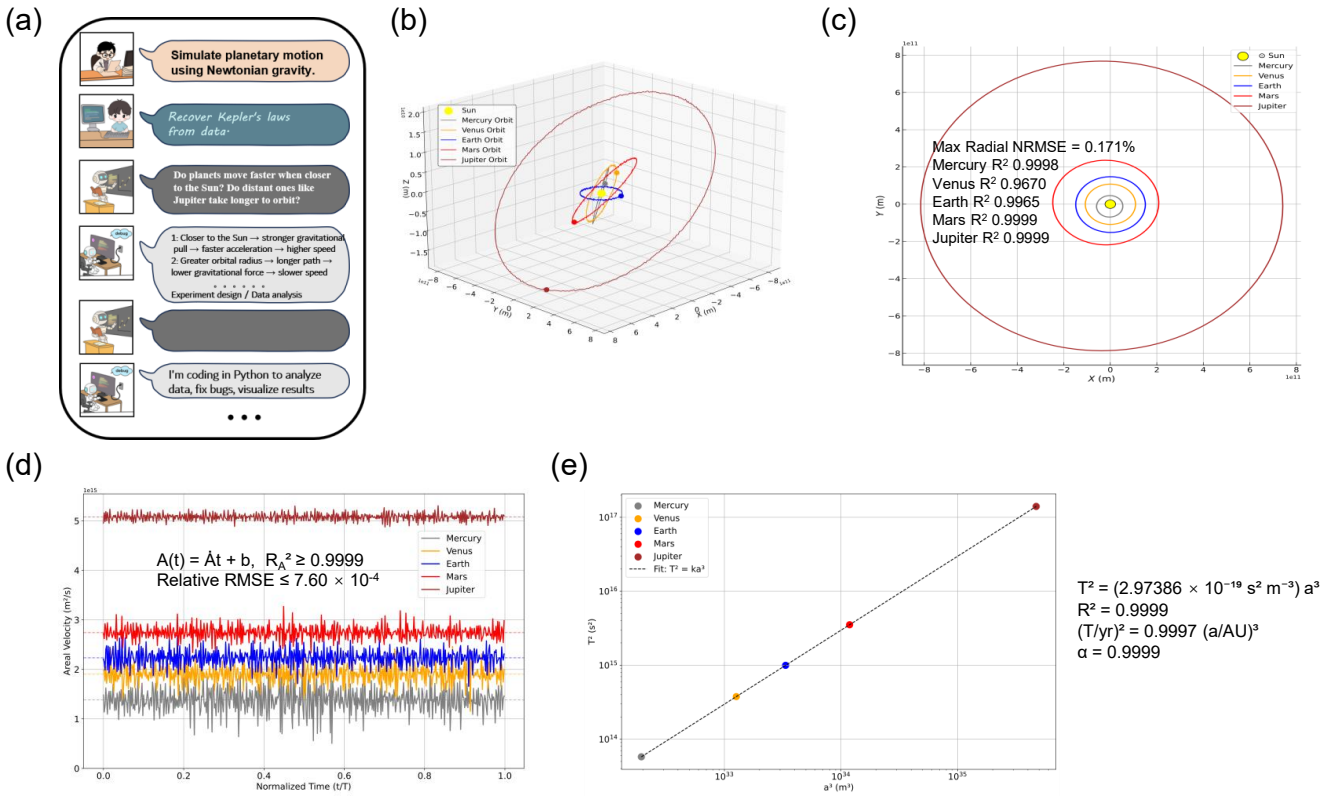


FIG. 2. **PhysAgent recovers Kepler-law relations of planetary motion.** (a) Collaborative workflow of PhysAgent recovering Kepler's planetary motion laws: Human-generated time-series three-dimensional trajectory data for five solar system planets (Mercury, Venus, Earth, Mars, and Jupiter) were analyzed through mentor-student agent collaboration in the PhysAgent system. The mentor agent provided heuristic questions to guide the student agent in exploring orbit shapes, velocity variations, and period-radius relationships, while the student agent implemented code and resolved technical challenges, producing publication-quality visualizations. (b) Planetary Motion Trajectories (Human User): Time-series 3D trajectories of five planets in the solar system—Mercury, Venus, Earth, Mars, and Jupiter—generated by human users using numerical methods. (c) Orbit Shape Visualization (Student Agent): Projections of the orbits of Mercury, Venus, Earth, Mars, and Jupiter in the XY plane, distinguished by colors (gray, orange, blue, red, brown). The orbits are elliptical with the Sun at one focus, confirming Kepler's First Law. (d) Areal Velocity Analysis (Collaborative Result): Areal velocity variations over time for each planet, using the same color scheme. The approximately constant areal velocity indicates that the line connecting a planet to the Sun sweeps equal areas in equal times, consistent with Kepler's Second Law. (e) Period-Radius Relationship (Collaborative Result): Relationship between the square of the orbital period (T^2) and the cube of the semi-major axis (a^3), with planets distinguished by the same color scheme. A linear fit demonstrates $T^2 \propto a^3$, validating Kepler's Third Law.

In the released demo artifacts, the maximum coefficient of variation of areal velocity is 2.44×10^{-5}. As illustrated in Figure 2 (e), the square of a planet's orbital period is proportional to the cube of its semi-major axis, validating Kepler's Third Law. Fitting T^2 against a^3 gives $T^2 = (2.9738 \times 10^{-19}\ \mathrm{s^2\,m^{-3}})a^3$ with $R^2 = 0.9999$. Equivalently, in astronomical units and years, the normalized relation is $(T/\mathrm{yr})^2 = 0.9997(a/\mathrm{AU})^3$, and the log-log fitted exponent is $\alpha = 0.9999$. A through-origin fit of T^2 versus a^3 gives slope 1.000 with $R^2 = 1.000$, quantitatively supporting the recovered Kepler third-law scaling. Throughout the collaborative process, the mentor agent engaged in multiple rounds of interaction with the student agent, providing feedback to refine hypotheses (e.g., adjusting from circular to elliptical orbits), optimize analytical methods (e.g., enhancing the precision of areal velocity calculations), and improve visualization designs to clearly highlight scientific patterns. Ultimately, the PhysAgent system recovered the Kepler-law relations in this controlled synthetic benchmark, thereby demonstrating the efficiency and exploratory capability of multi-agent collaboration for reproducible analysis.

PhysAgent recovers Newton-law scaling.

To further evaluate the capability of the multi-agent system PhysAgent in controlled law recovery, we designed a classical mechanics task that challenges the system to recover the quantitative relationship between

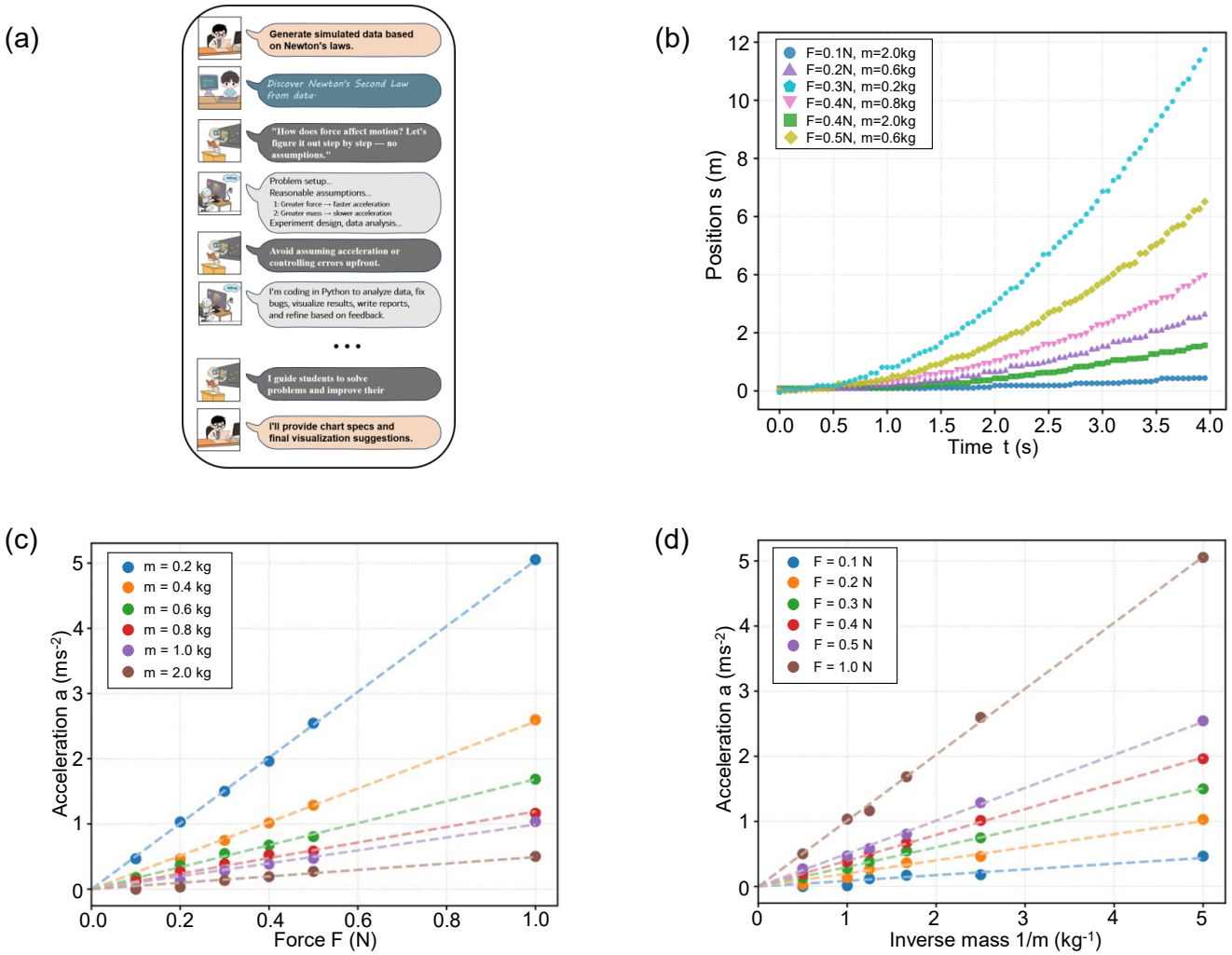


FIG. 3. **PhysAgent recovers Newton-law scaling.** (a) Collaborative workflow of PhysAgent recovering Newton's second-law relation: Human-generated position-time data of a cart under controlled force-mass conditions were analyzed through mentor-student agent collaboration. The mentor provided analysis strategies while the student implemented code and resolved technical challenges. Final visualizations were refined for publication standards. (b) Position-time visualization (Student agent): Position s versus time t under varying force-mass conditions. Distinct markers/colors represent different (F, m) pairs. Each trajectory was fitted by $s(t) = s_0 + v_0 t + \frac{1}{2} a_{\text{fit}} t^2$ to estimate acceleration and residuals; the position-fit residual RMSE ranges from 0.0239 to 0.0529 m. (c) Acceleration-force relationship (Collaborative result): Linear a-F dependence estimated through trajectory-level acceleration fitting. Color-coded masses demonstrate proportionality $a \propto F$ at constant m. Through-origin fits give $a = \beta_F F$, with $\beta_F m = 1.003$–1.045 and $R^2 \geq 0.9938$. (d) Acceleration-inverse mass relationship (Collaborative result): Linear a-$1/m$ relationship established via agent collaboration. Color-coded forces verify proportionality $a \propto 1/m$ at constant F. Through-origin fits give $a = \beta_m(1/m)$, with $\beta_m/F = 1.004$–1.019 and $R^2 \geq 0.9946$. The global fit gives $a = (1.0054 \pm 0.0034)F/m$ with $R^2 = 0.9998$.

force and motion from controlled data. Because the data are synthetic and the underlying LLMs may already be familiar with Newton's second law, this task evaluates workflow orchestration and quantitative recovery rather than de novo law discovery. As shown in Figure 3 (a), the task begins with a human user who generates simulated position-time data of a cart moving on an inclined plane, followed by posing the question: can one recover how force influences motion from these data? Subsequently, multiple agents within the PhysAgent system collaboratively engaged in the task. The mentor agent proposed scientific strategies and guided the student agent in uncovering patterns from the data, emphasizing that the target formula should not be directly supplied to the analysis workflow. Next, the student agent constructed the analysis workflow from scratch, including designing the data collection scheme, formulating reasonable hypotheses, processing the experimental data, and conducting visualization and curve fitting using Python and standard scientific libraries such as *NumPy*, *Pandas*, and

Matplotlib. The human user provided iterative feedback at key stages, particularly with respect to visualization standards, graphical presentation, and scientific writing norms. Throughout this collaborative process, the mentor and student agents engaged in multiple rounds of interaction, continuously refining their hypotheses and improving the analysis workflow. Ultimately, the agents recovered Newton-law scaling from the simulated data, completing a reproducible law-recovery workflow from data generation to fitting and reporting.

Figure 3 (b) illustrates the trajectories of a cart under six different combinations of applied force F and mass m: ($F = 0.1$ N, $m = 2.0$ kg), ($F = 0.2$ N, $m = 0.6$ kg), ($F = 0.3$ N, $m = 0.2$ kg), ($F = 0.4$ N, $m = 0.8$ kg), ($F = 0.4$ N, $m = 2.0$ kg), and ($F = 0.5$ N, $m = 0.6$ kg). The cart's position s exhibits an approximately parabolic dependence on time t, indicating uniformly accelerated motion. To better mimic experimental conditions, small amounts of measurement noise were added to the data. For each force-mass condition, acceleration was estimated by fitting the raw position-time trajectory to $s(t) = s_0 + v_0 t + \frac{1}{2} a_{\text{fit}} t^2$. This fit provides both the fitted acceleration and its uncertainty, with trajectory-level position residual RMSE values ranging from 0.0239 to 0.0529 m and a maximum 95% confidence interval of $\pm 0.0202\ \mathrm{m\,s^{-2}}$ for a_{fit}.

Under the guidance of the mentor agent, the student agent computed the acceleration a from these quadratic trajectory fits, rather than relying only on numerical differentiation of noisy data. As shown in Figure 3 (c), the acceleration of the cart under varying forces is plotted for fixed mass values. Each color represents a different mass ($m = 0.2$, 0.4, 0.6, 0.8, 1.0, and 2.0 kg). Within each group, a clear linear relationship between acceleration and force is observed, confirming that $a \propto F$ when mass is held constant. Through-origin fits of the form $a = \beta_F F$ give $\beta_F m = 1.003$–1.045, with $R^2 \geq 0.9938$, showing that the fitted slopes are consistent with the expected $1/m$ dependence.

Furthermore, Figure 3 (d) presents the variation of acceleration under fixed external force conditions, plotted against the inverse of mass $1/m$. Different colors correspond to different force values ($F = 0.1$, 0.2, 0.3, 0.4, 0.5, and 1.0 N). The results exhibit a strong linear dependence of a on $1/m$, supporting the hypothesis that $a \propto 1/m$. Through-origin fits of the form $a = \beta_m(1/m)$ give $\beta_m/F = 1.004$–1.019, with $R^2 \geq 0.9946$, confirming that the fitted slopes recover the applied force scale.

By integrating these two key relationships, PhysAgent recovered the quantitative relation:

$$a \propto \frac{F}{m} \quad \Rightarrow \quad F = kma.$$

Using all 36 force-mass conditions, the global through-origin acceleration-form fit gives $a = (1.0054 \pm 0.0034)F/m$ with $R^2 = 0.9998$ and acceleration residual RMSE $0.0125\ \mathrm{m\,s^{-2}}$. Equivalently, fitting in force form gives $F = (0.9860 \pm 0.0096)ma$ with $R^2 = 0.9975$. These fitted proportionality constants are close to unity in SI units, quantitatively validating the recovered Newton-law scaling. This controlled example shows that PhysAgent can recover the mathematical scaling of Newton's second law and document the evidence trail from data generation to fitting and visualization. From problem formulation, data acquisition, visualization, hypothesis construction, and data analysis to law recovery, the process mirrors key elements of scientific inquiry. This demonstrates PhysAgent's capability for interpretable and verifiable analysis in addressing fundamental problems in science.

PhysAgent performs quantum calculations.

In addition to the controlled recovery of classical mechanics relations, PhysAgent also enables fully automated quantum mechanical calculations for materials. To demonstrate its capability, we consider the case of gallium arsenide (GaAs), a direct band gap semiconductor with a zinc blende structure. As shown in Figure 4(a), the task begins with a human user uploading the CIF file of GaAs and requesting an analysis of its electronic structure. The objective is further clarified as: "perform electronic structure calculations of GaAs based on DFT."

Once the goal is established, the mentor agent searches for DFT workflows and requires online tools, then forwards this information to the student agent. The student agent then autonomously plans and executes a complete first-principles DFT workflow to study the electronic properties of GaAs. As illustrated in Figure 4(a), the workflow starts with setting up the computational environment. The student agent installs and configures the Atomic Simulation Environment (ASE), Quantum ESPRESSO (QE), Python, and relevant visualization packages. It also sets the paths for pseudopotential files and necessary environment variables. During the structure modeling phase (Figure 4(b)), ASE is used to parse the user-provided CIF file and construct the GaAs crystal structure. In the calculation phase, a self-consistent field (SCF) calculation is performed to obtain the ground-state charge density, with the energy convergence process shown in Figure 4(c). This is followed by band structure calculations along high-symmetry paths in the Brillouin zone (Γ–X–W–K–Γ–L–U–W–L–K), and a non-self-consistent field (NSCF) calculation to generate wavefunctions on a dense k-point mesh for density of states (DOS) analysis. The student agent invokes `dos.x` and `projwfc.x` to compute the total density of states (DOS) and the projected density of states (PDOS), respectively.

In the visualization stage, the student agent uses `pp.x` to convert the SCF-derived charge density into CUBE format. Figure 4(e) shows a logarithmic color-mapped slice in the XY plane, while Figure 4(d) illustrates 3D isosurfaces overlaid with atomic positions and lattice vectors. Furthermore, the agent constructs a combined

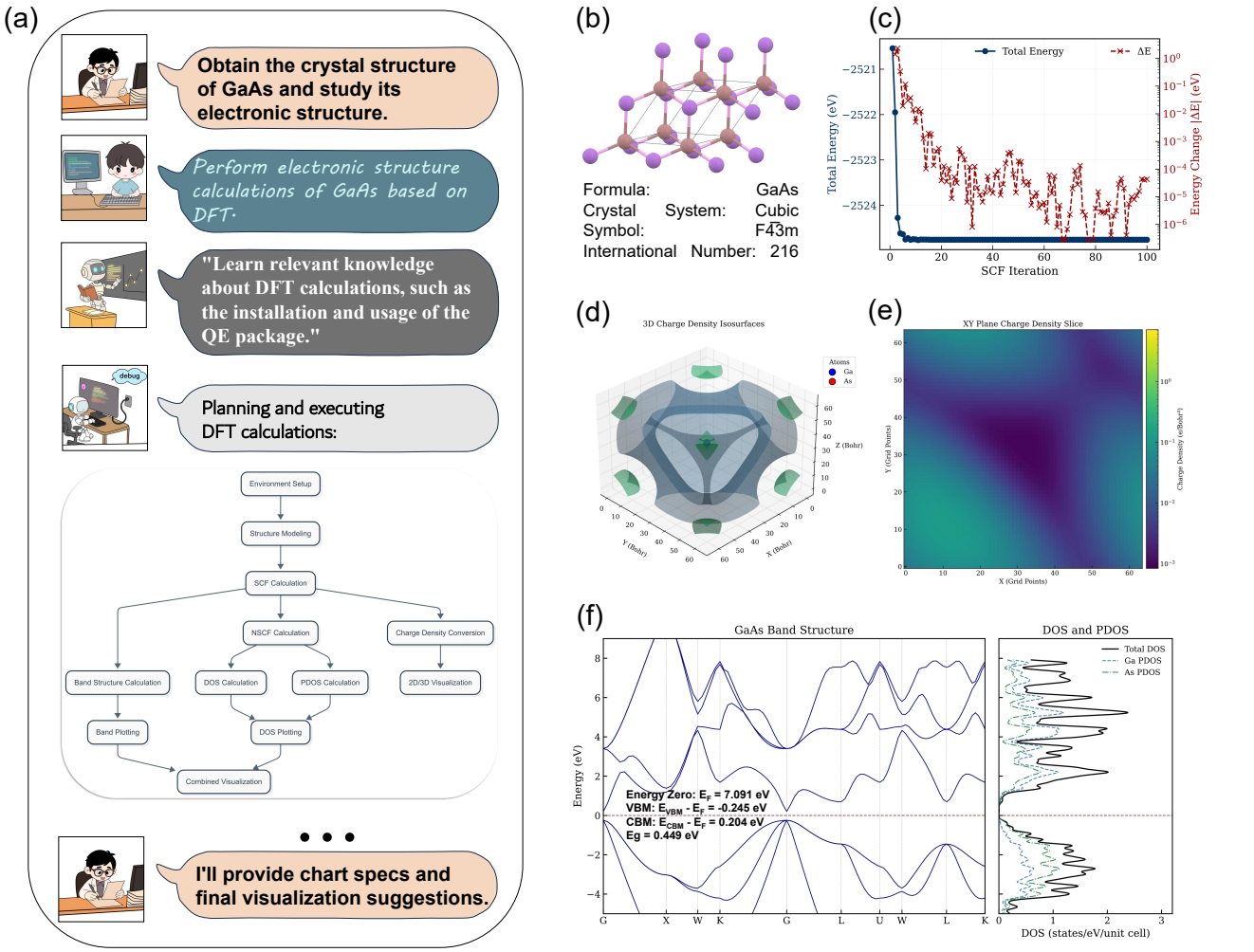


FIG. 4. **PhysAgent performs quantum calculations.** (a) Collaborative workflow of PhysAgent performing DFT electronic structure calculations for GaAs: A human user initiates the task by uploading the CIF structure file of GaAs and requesting a DFT-based electronic structure calculation. The mentor agent retrieves necessary workflows and tools, while the student agent autonomously configures the environment and executes the full DFT workflow using Quantum ESPRESSO and ASE, producing scientifically accurate results. (b) Structure Modeling (Student Agent): The crystal structure of GaAs is parsed from the CIF file and reconstructed using ASE, visualized here as a ball-and-stick model. This step prepares the input for first-principles calculations. (c) SCF Energy Convergence (Collaborative Result): Energy convergence behavior during the self-consistent field (SCF) calculation is plotted, showing systematic reduction of total energy over iterations, confirming convergence to the ground-state charge density. (d) Charge Density Isosurface Visualization (Collaborative Result): 3D isosurfaces of the SCF charge density are rendered, overlaid with atomic positions and lattice vectors. The distribution reflects electronic localization around atomic sites. (e) Logarithmic Charge Density Slice (Collaborative Result): A 2D log-scale slice of the charge density in the XY plane, illustrating fine variations in electron distribution and bonding characteristics. (f) Band Structure and Density of States (Collaborative Result): The left panel shows the band structure of GaAs along high-symmetry paths (Γ–X–W–K–Γ–L–U–W–L–K), and the right panel displays the total DOS and projected DOS onto Ga- and As-centered pseudo-atomic orbitals. The energy zero is defined as the Fermi level ($E - E_F = 0$, with $E_F = 7.091$ eV in the band-structure XML output). The VBM and CBM both occur at Γ, with $E_{\mathrm{VBM}} - E_F = -0.245$ eV and $E_{\mathrm{CBM}} - E_F = 0.204$ eV, giving a direct PBE band gap of $E_g = 0.449$ eV. The DOS is reported in states eV^{-1} per primitive cell, and the PDOS curves represent projections of Kohn–Sham states onto the atomic-like projector channels from the Ga and As pseudopotentials.

band structure and DOS plot (Figure 4(f)), where the left panel displays energy dispersion with labeled high-symmetry points and the right panel overlays total and projected DOS for Ga and As atoms. All band and DOS energies in Figure 4(f) are plotted relative to the Fermi level, i.e., $E - E_F$, with $E_F = 7.091$ eV read from the QE XML band-structure output. From the calculated eigenvalues, the valence-band maximum (VBM)

and conduction-band minimum (CBM) both occur at Γ. The VBM is located at $E_{\mathrm{VBM}} - E_F = -0.245$ eV and the CBM at $E_{\mathrm{CBM}} - E_F = 0.204$ eV, yielding a direct band gap of $E_g = 0.449$ eV. This confirms the direct-gap character of GaAs within the present PBE calculation, while the numerical gap is expectedly smaller than the experimental value because semilocal DFT underestimates semiconductor band gaps.

The total DOS was obtained from `dos.x` and is plotted in units of states eV^{-1} per primitive cell, without normalization to unit area. The integrated DOS at the Fermi level is 18 states per primitive cell, consistent with the number of valence electrons included by the Ga and As pseudopotentials. The PDOS was obtained from `projwfc.x`; the Ga PDOS curve sums the Ga-centered s, p, and d projector channels, while the As PDOS curve sums the As-centered s and p projector channels. These projected DOS curves therefore indicate the orbital and atomic character of the Kohn–Sham states, rather than separate normalized charge populations. Both panels share a common energy axis for direct comparison. The fully automated pipeline outputs key physical quantities such as charge density, band eigenvalues, and orbital-resolved DOS, enabling a comprehensive theoretical understanding of the electronic structure of GaAs.

PhysAgent simulates raindrop flow on train windows.

In this section, we apply PhysAgent to simulate and interpret experimental phenomena (Figure 5(a)). The flow of raindrops on a window is a typical thin-film flow problem influenced by gravity, surface tension, viscosity, and external driving forces such as air shear stress and raindrop deposition. When a train is moving at high speed, rainwater on the window often forms obliquely downward flow paths, sometimes with meandering trajectories that branch into tree-like patterns, accompanied by evaporation and dripping (Figure 5(c)-(d)). PhysAgent models this behavior using the thin-film approximation, employing a height evolution equation to describe the spatiotemporal dynamics of the film thickness, and incorporates stochastic raindrop deposition, branching mechanisms, as well as evaporation and dripping effects to reproduce realistic scenarios.

As show Figure 5(b), the model assumes that the film thickness $h(x, y, t)$ is much smaller than the flow length scale (thin-film approximation), that the flow is laminar, and that inertial effects are negligible. The coordinate system is defined with the x-axis along the horizontal length of the window and the y-axis along its vertical height. The flow is primarily driven by gravity and air shear stress, moving obliquely downward. The thin-film flow can be derived from the Navier–Stokes equations under the thin-film approximation. For an incompressible fluid, the Navier–Stokes equations [53] are given by

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mu\nabla^2\mathbf{u} + \rho\mathbf{g}, \tag{1}$$

where $\mathbf{u}$ is the velocity field, p is the pressure, μ is the viscosity, ρ is the density, and $\mathbf{g}$ is the gravitational acceleration. Under the thin-film approximation, assuming $h \ll L$ (where L is the characteristic length), the velocity is mainly along the film plane and inertial terms are neglected (low Reynolds number). The equations reduce to

$$\mu\frac{\partial^2 u_x}{\partial z^2} = \frac{\partial p}{\partial x} + \rho g_x, \quad \mu\frac{\partial^2 u_y}{\partial z^2} = \frac{\partial p}{\partial y} + \rho g_y, \tag{2}$$

where z is the coordinate normal to the surface, u_x and u_y are the velocity components in the x and y directions, respectively, and the gravitational components are $g_x = 0$ and $g_y = -g$ (downward). The boundary conditions are: no-slip at the bottom ($z = 0$: $u_x = u_y = 0$); at the free surface ($z = h$), the tangential stress includes the air shear τ_{air} (negative x-direction, modeling wind resistance), and the normal stress includes the curvature-induced pressure due to surface tension, $p = -\sigma\nabla^2 h$(Laplace pressure).

Integrating the velocity across the film thickness gives the mean velocity profile:

$$\begin{aligned} \bar{u}_x &= \frac{h^2}{3\mu}\left(-\frac{\partial p}{\partial x}\right) - \frac{h\tau_{\mathrm{air}}}{2\mu}, \\ \bar{u}_y &= \frac{h^2}{3\mu}\left(-\frac{\partial p}{\partial y} - \rho g_y\right). \end{aligned} \tag{3}$$

Substituting $p = -\sigma\nabla^2 h$ and treating τ_{air} as a constant, we obtain

$$\begin{aligned} \bar{u}_x &= \frac{h^2}{3\mu}\sigma\frac{\partial}{\partial x}(\nabla^2 h) - \frac{h\tau_{\mathrm{air}}}{2\mu}, \\ \bar{u}_y &= \frac{h^2}{3\mu}\left(\sigma\frac{\partial}{\partial y}(\nabla^2 h) - \rho g_y\right). \end{aligned} \tag{4}$$

The film height evolves according to mass conservation:

$$\frac{\partial h}{\partial t} + \nabla\cdot(h\bar{\mathbf{u}}) = 0, \tag{5}$$

which expands to the thin-film equation:

$$\frac{\partial h}{\partial t} = -\nabla\cdot\left(\frac{h^3}{3\mu}\nabla(\sigma\nabla^2 h - \rho g y) - \frac{h^2\tau_{\mathrm{air}}}{2\mu}\mathbf{e}_x\right), \tag{6}$$

where $\mathbf{e}_x$ is the unit vector in the x-direction. To mitigate numerical instability in resolving the fourth-order surface tension term $\nabla\cdot\left(\frac{\sigma h^3}{3\mu}\nabla\nabla^2 h\right)$, approximate it as $\frac{\alpha\sigma h^3}{\mu}\nabla^2 h$ (where α is a dimensionless damping coefficient), sacrificing some physical fidelity for computational robustness.

As show Figure 5(e)-(f), to more realistically simulate the flow of raindrops on a window, the model incorporates several additional physical mechanisms. First, the

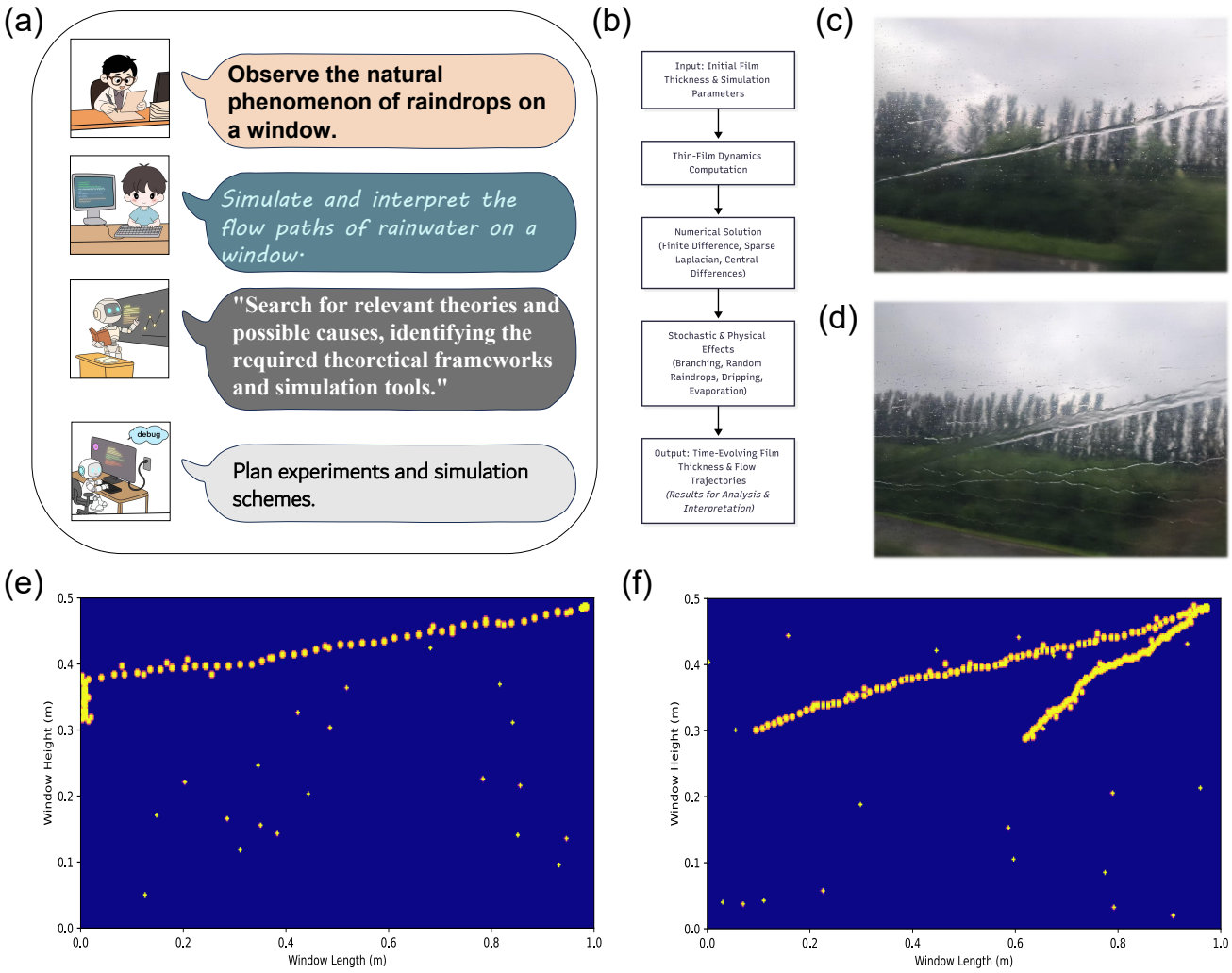


FIG. 5. **PhysAgent simulates raindrop flow on train windows.** (a) Collaborative workflow of PhysAgent simulating raindrop flow on train windows: A human user observes the natural phenomenon of raindrops on a window and defines the research objective as simulating and interpreting the flow paths of rainwater on the window. The mentor agent searches for relevant theories and possible causes, identifying the necessary theoretical frameworks and simulation tools, while the student agent plans and executes the simulation process. (b) Thin-Film Evolution Simulation workflow by the student agent: The process begins with the input of the initial film thickness and simulation parameters, followed by thin-film dynamics calculations solved using numerical methods such as finite differences, sparse Laplacians, and central differences. Stochastic and physical effects, including branching, random raindrop impacts, dripping, and evaporation, are then incorporated. Finally, the simulation outputs the time evolution of the film thickness and the flow trajectories. (c)–(d) Real-world observations: On a high-speed train window during rainfall, large streams of water are seen flowing in the opposite direction of travel, along with branched, tree-like flow paths oriented diagonally downward. (e)–(f) Thin-Film Evolution Simulation results: The simulation reproduces the flow trajectories as the collaborative outcome. The simulations in panels (e)–(f) were performed on a 400×200 grid over a 1.0 m $\times$ 0.5 m window. The main parameters were $\mu = 1.0 \times 10^{-3}$ Pa s, $\sigma = 0.072$ N m^{-1} for (e) and 0.036 N m^{-1} for (f), $\tau_{\text{air}} = 3.66 \times 10^{4}$, branching probability 0.85, evaporation rate 1.0×10^{-4}, and time steps 5.0×10^{-4} s and 2.0×10^{-4} s.

main raindrop flow follows a slanted, meandering path, with the trajectory modulated by random fluctuations to generate wave-like variations, while liquid is continuously supplied along the main flow. Second, the film can branch from the main path, forming multiple secondary streams with random deviations in direction and length, and some branches may further generate tertiary sub-branches. Third, intermittent raindrops are randomly added to simulate uneven rainfall. To capture gravity-driven dripping, liquid is partially removed when the film thickness exceeds a threshold. Finally, evaporation and film lifetime are included, allowing the film to gradually

thin and eventually disappear over time. Throughout all processes, the film thickness is maintained non-negative. The numerical solution is implemented using a finite difference method on a two-dimensional rectangular grid. The Laplacian operator is constructed via sparse matrices, and the flux divergence is computed using central differences. Boundary conditions are set as either periodic or zero-flux, implemented implicitly through truncation. To avoid numerical diffusion, the flow is computed only in regions where the film thickness exceeds a minimum threshold. The detailed settings of the Numerical Methods and Parameters are provided in the Supplementary Materials.

DISCUSSION

In this work, we introduced PhysAgent, a multi-agent LLM-driven framework that executes end-to-end scientific workflows—from hypothesis formulation and computational modeling to data analysis and reporting. By integrating specialized agents (Mentor, Student, and Leader) within a physics-grounded workflow, PhysAgent demonstrates a proof-of-concept route for recovering known physical relations from controlled data, performing ab initio materials-workflow generation (e.g., GaAs electronic structures), and simulating real-world physical phenomena such as raindrop flow on train windows. The system harmonizes LLM-based planning with domain-specific tools (e.g., Quantum ESPRESSO, VASP) and produces reproducibility artifacts including data, scripts, reports, and transcripts.

Recent frameworks, including Coscientist [54], ChemCrow [55], SciAgents [56], and AtomAgents [57], demonstrate the potential of LLM-based agents for scientific discovery through tool integration and collaborative reasoning. Meanwhile, VASPilot [58] automates end-to-end VASP workflows, while emerging AI scientist systems, such as Wang et al. [59] and Sparks [60], explore autonomous decision-making and iterative optimization. Together, these advances provide a foundation for integrating crystal generation, property evaluation, and feedback-driven optimization to improve the dynamical stability of generated materials.

Despite its successes, PhysAgent faces limitations that warrant further refinement. These examples should not be interpreted as evidence that the system discovers unknown physical laws. Because the demonstrations use synthetic data generated from known models and because the underlying LLMs may have encountered these laws during pretraining, the current experiments test workflow orchestration, quantitative analysis, and reproducible law recovery rather than de novo law discovery. Future blind tests should hide the data generator from the agents and include non-Keplerian, non-Newtonian, and artificially constructed laws. In addition, broader benchmarks, ablation studies, success/failure statistics, and uncertainty quantification are needed before making strong claims about generality, trustworthiness, or scalability. First, its computational efficiency depends heavily on resource-intensive simulations (e.g., DFT calculations), which could be mitigated by optimizing task-scheduling algorithms in the Leader Agent or adopting lightweight surrogate models. Second, while effective for well-defined problems (e.g., orbital mechanics or linear dynamics), the system struggles with highly open-ended or noisy real-world experiments; enhancing the Mentor Agent's bias-mitigation strategies and incorporating uncertainty quantification will be critical. Third, human intervention remains essential for refining outputs (e.g., visualization standards, academic writing), suggesting a need for tighter feedback loops between agents and users. Finally, reliance on simulated data (e.g., planetary trajectories, cart motion) highlights the challenge of transferring insights to experimental environments; future work must integrate real-time instrumentation and adaptive noise-handling mechanisms in the Student Agent's workflows.

Looking ahead, PhysAgent's architecture may help accelerate scientific inference by automating labor-intensive tasks (e.g., code generation, DFT setup, hypothesis testing) and by making computational workflows easier to audit and reproduce. Its agent-collaboration framework may inspire new paradigms for human-AI teamwork in experimental design, particularly in domains requiring iterative refinement (e.g., drug discovery or quantum material optimization). Furthermore, the controlled law-recovery examples may be useful in STEM education, where students can rediscover known relations through guided AI collaboration. As LLMs and scientific tools evolve, PhysAgent provides a modular starting point for more rigorous, benchmarked human-AI scientific workflows.

Data & Code availability: The *PhysAgent* framework developed in this work is implemented based on the *MetaGPT* framework: https://github.com/FoundationAgents/MetaGPT. A public reproducibility package for the four examples in this manuscript is available at https://github.com/xqh19970407/PhysAgent-release. The repository contains the executable Python package, role prompts, input data, generated scripts, analysis JSON files, figures, reports, command logs, and collaboration transcripts. The package runs in a deterministic offline mode by default and can optionally use DeepSeek through environment variables. We also provide a demonstration video of *PhysAgent* to illustrate its workflow and capabilities: https://b23.tv/emOilkC. In addition, the Supplementary Materials provide the detailed prompts and related configurations for the different agent roles in PhysAgent.

Acknowledgements: The work is supported by Beijing Natural Science Foundation(No.Z260002, No.Z250005), the National Natural Science Foundation of China (No.62476278, No.12434009), and the National Key

R&D Program of China (Grants No. 2024YFA1408601). Computational resources have been provided by the Physical Laboratory of High Performance Computing at Renmin University of China.

Corresponding authors: Correspondence and requests for materials should be addressed to Ze-Feng Gao (zfgao@ruc.edu.cn) and Zhong-Yi Lu (zlu@ruc.edu.cn).

[1] Xun Xiao, Niansheng Xu, Xueyu Tian, Tiankai Zhang, Bingzheng Wang, Xiaoming Wang, Yeming Xian, Chunyuan Lu, Xiangyu Ou, Yanfa Yan, Licheng Sun, Fengqi You, and Feng Gao. Aqueous-based recycling of perovskite photovoltaics. *Nature*, 638(8051):670–675, 2025.

[2] Zirui Gao, Aowen Li, Xingwu Liu, Mi Peng, Shixiang Yu, Maolin Wang, Yuzhen Ge, Chengyu Li, Tie Wang, Zhaohua Wang, Wu Zhou, and Ding Ma. Shielding Pt/γ-Mo_2N by inert nano-overlays enables stable H_2 production. *Nature*, 638(8051):690–696, 2025.

[3] Hualei Sun, Mengwu Huo, Xunwu Hu, Jingyuan Li, Zengjia Liu, Yifeng Han, Lingyun Tang, Zhongquan Mao, Pengtao Yang, Bosen Wang, Jinguang Cheng, Dao-Xin Yao, Guang-Ming Zhang, and Meng Wang. Signatures of superconductivity near 80 K in a nickelate under high pressure. *Nature*, 621(7979):493–498, 2023.

[4] Guangdi Zhou, Wei Lv, Heng Wang, Zihao Nie, Yaqi Chen, Yueying Li, Haoliang Huang, Wei-Qiang Chen, Yu-Jie Sun, Qi-Kun Xue, and Zhuoyu Chen. Ambient-pressure superconductivity onset above 40 K in $(La,Pr)_3Ni_2O_7$ films. *Nature*, 640(8059):641–646, 2025.

[5] He Jiang, Xiankun Zhang, Kuanglei Chen, Xiaoyu He, Yihe Liu, Huihui Yu, Li Gao, Mengyu Hong, Yunan Wang, Zheng Zhang, and Yue Zhang. Two-dimensional Czochralski growth of single-crystal MoS2. *Nature Materials*, 24(2):188–196, 2025.

[6] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024.

[7] Josh Abramson, Jonas Adler, Jack Dunger, and Richard Evans. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024.

[8] Jonathan Schmidt, Tiago F.T. Cerqueira, Aldo H. Romero, Antoine Loew, Fabian Jäger, Hai-Chen Wang, Silvana Botti, and Miguel A.L. Marques. Improving machine-learning models in materials science through large datasets. *Materials Today Physics*, 48:101560, 2024.

[9] Amil Merchant, Simon Batzner, Samuel S. Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023.

[10] Luis Barroso-Luque, Muhammed Shuaibi, Xiang Fu, Brandon M. Wood, Misko Dzamba, Meng Gao, Ammar Rizvi, C. Lawrence Zitnick, and Zachary W. Ulissi. Open materials 2024 (omat24) inorganic materials dataset and models. *arXiv preprint arXiv:2410.12771*, 2024.

[11] Xiao-Qi Han, Peng-Jie Guo, Ze-Feng Gao, Hao Sun, and Zhong-Yi Lu. Invdesflow-al: Active learning-based workflow for inverse design of functional materials. *arXiv preprint arXiv:2505.09203*, 2025.

[12] Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, Roberto Sordillo, Lixin Sun, Jake Smith, Bichlien Nguyen, Hannes Schulz, Sarah Lewis, Chin-Wei Huang, Ziheng Lu, Yichi Zhou, Han Yang, Hongxia Hao, Jielan Li, Chunlei Yang, Wenjie Li, Ryota Tomioka, and Tian Xie. A generative model for inorganic materials design. *Nature*, 639(8055):624–632, 2025.

[13] Rui Jiao, Wenbing Huang, Peijia Lin, Jiaqi Han, Pin Chen, Yutong Lu, and Yang Liu. Crystal structure prediction by joint equivariant diffusion on lattices and fractional coordinates. *ICLR*, 2023.

[14] Xiao-Qi Han, Zhenfeng Ouyang, Peng-Jie Guo, Hao Sun, Ze-Feng Gao, and Zhong-Yi Lu. InvDesFlow: An AI-driven materials inverse design workflow to explore possible high-temperature superconductors. *Chin. Phys. Lett.*, 42(4):047301, 2025.

[15] Xiao-Qi Han, Xin-De Wang, Meng-Yuan Xu, Zhen Feng, Bo-Wen Yao, Peng-Jie Guo, Ze-Feng Gao, and Zhong-Yi Lu. AI-Driven Inverse Design of Materials: Past, Present, and Future. *Chin. Phys. Lett.*, 42(2):027403, 2025.

[16] Tian Xie and Jeffrey C. Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.*, 120:145301, Apr 2018.

[17] Kamal Choudhary and Kevin Garrity. Designing high-T_C superconductors with BCS-inspired screening, density functional theory, and deep-learning. *Npj Comput. Mater.*, 8(1):244, 2022.

[18] Fernando Gómez-Ortiz, Aldo H. Romero, and Eric Bousquet. Pathways to crystal chirality: An algorithm to identify displacive chiral phase transitions. *Phys. Rev. B*, 112:014107, Jul 2025.

[19] Chengkai Zhu, Yin Mo, Yu-Ao Chen, and Xin Wang. Reversing unknown quantum processes via virtual combs for channels with limited information. *Phys. Rev. Lett.*, 133:030801, Jul 2024.

[20] Yi-Jun Chang, Jia-Hui Zhang, Yong-Heng Lu, Ying-Yue Yang, Feng Mei, Jie Ma, Suotang Jia, and Xian-Min Jin. Observation of photonic mobility edge phases. *Phys. Rev. Lett.*, 134:053601, Feb 2025.

[21] Kaiwen Chen, Xiangqi Liu, Jiachen Jiao, Muyuan Zou, Chengyu Jiang, Xin Li, Yixuan Luo, Qiong Wu, Ningyuan Zhang, Yanfeng Guo, and Lei Shu. Evidence of spin density waves in $la_3ni_2o_{7-\delta}$. *Phys. Rev. Lett.*, 132:256503, Jun 2024.

[22] Yi-Heng Tian, Yin Chen, Jia-Ming Wang, Rong-Qiang He, and Zhong-Yi Lu. Correlation effects and concomitant two-orbital $s_\pm$-wave superconductivity in $la_3ni_2o_7$ under high pressure. *Phys. Rev. B*, 109:165154, Apr 2024.

[23] Jia-Ming Wang, Yin Chen, Yi-Heng Tian, Rong-Qiang He, and Zhong-Yi Lu. Low-energy interband kondo bound states in orbital-selective mott phases. *Phys. Rev. B*, 111:155107, Apr 2025.

[24] Ru Zheng, Rong-Qiang He, and Zhong-Yi Lu. Natural orbitals renormalization group approach to a spin-$\frac{1}{2}$ impurity interacting with two helical liquids. *Phys. Rev. B*,

107:115149, Mar 2023.

[25] P. W. Anderson. More is different. *Science*, 177(4047):393–396, 1972.

[26] DeepSeek-AI. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2025.

[27] Bang Liu et al. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint arXiv:2504.01990*, 2025.

[28] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. *ICLR*, 2024.

[29] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

[30] Keivalya Pandya and Mehfuza Holia. Automating customer service using langchain: Building custom open-source gpt chatbot for organizations. *arXiv preprint arXiv:2310.05421*, 2023.

[31] Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *CoRR*, abs/2306.16092, 2023.

[32] Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Yusuf Roohani, Ryan Li, Lin Qiu, Gavin Li, Junze Zhang, Di Yin, Shruti Marwaha, Jennefer N. Carter, Xin Zhou, Matthew Wheeler, Jonathan A. Bernstein, Mengdi Wang, Peng He, Jingtian Zhou, Michael Snyder, Le Cong, Aviv Regev, and Jure Leskovec. Biomni: A general-purpose biomedical ai agent. *bioRxiv*, May 2025.

[33] Tal Ridnik, Dedy Kredo, and Itamar Friedman. Code generation with alphacodium: From prompt engineering to flow engineering. *arXiv preprint arXiv:2401.08500*, 2024.

[34] Lianhao Zhou, Hongyi Ling, Keqiang Yan, Kaiji Zhao, Xiaoning Qian, Raymundo Arróyave, Xiaofeng Qian, and Shuiwang Ji. Toward greater autonomy in materials discovery agents: Unifying planning, physics, and scientists. *arXiv preprint arXiv:2506.05616*, 2025.

[35] Chengping Rao, Pu Ren, Qi Wang, Oral Buyukozturk, Hao Sun, and Yang Liu. Encoding physics to learn reaction–diffusion processes. *Nat. Mach. Intell.*, 5(7):765–779, Jul 2023.

[36] Zeyu Li, Wang Han, Yue Zhang, Qingfei Fu, Jingxuan Li, Lizi Qin, Ruoyu Dong, Hao Sun, Yue Deng, and Lijun Yang. Learning spatiotemporal dynamics with a pretrained generative model. *Nat. Mach. Intell.*, 6(12):1566–1579, Dec 2024.

[37] Duo Zhang, Xinzijian Liu, Xiangyu Zhang, et al. DPA-2: a large atomic model as a multi-task learner. *Npj Comput. Mater.*, 10(1):293, 2024.

[38] Antonio Sanna, Tiago F. T. Cerqueira, Yue-Wen Fang, Ion Errea, Alfred Ludwig, and Miguel A. L. Marques. Prediction of ambient pressure conventional superconductivity above 80 K in hydride compounds. *Npj Comput. Mater.*, 10(1):44, 2024.

[39] Kapildeb Dolui, Lewis J. Conway, Christoph Heil, Timothy A. Strobel, Rohit P. Prasankumar, and Chris J. Pickard. Feasible Route to High-Temperature Ambient-Pressure Hydride Superconductivity. *Phys. Rev. Lett.*, 132:166001, Apr 2024.

[40] Chao Shen, Xiaoqi Han, Heng Cai, Tong Chen, Yu Kang, Peichen Pan, Xiangyang Ji, Chang-Yu Hsieh, Yafeng Deng, and Tingjun Hou. Improving the reliability of language model-predicted structures as docking targets through geometric graph learning. *J. Med. Chem.*, 68(2):1956–1969, Jan 2025.

[41] Heng Cai, Chao Shen, Tianye Jian, Xujun Zhang, Tong Chen, Xiaoqi Han, Zhuo Yang, Wei Dang, Chang-Yu Hsieh, Yu Kang, Peichen Pan, Xiangyang Ji, Jianfei Song, Tingjun Hou, and Yafeng Deng. Carsidock: a deep learning paradigm for accurate protein–ligand docking and screening based on large-scale pre-training. *Chem. Sci.*, 15:1449–1471, 2024.

[42] Kimi Team. Kimi k1.5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[43] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[44] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2022.

[45] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. *ICML*, 97:2790–2799, 09–15 Jun 2019.

[46] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2023.

[47] Wen Yang, Minpeng Liao, and Kai Fan. Markov chain of thought for efficient mathematical reasoning. *arXiv preprint arXiv:2410.17635*, 2025.

[48] Kieran C. R. Fox, Nicholas S. Fitz, and Peter B. Reiner. The multiplicity of memory enhancement: Practical and ethical implications of the diverse neural substrates underlying human memory systems. *Neuroethics*, 10(3):375–388, 2017.

[49] Alan Baddeley. Working memory. *Science*, 255(5044):556–559, 1992.

[50] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *NeurIPS*, 36:2011, 2022.

[51] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2020.

[52] Lu Wang, Fangkai Yang, Chaoyun Zhang, Junting Lu, Jiaxu Qian, Shilin He, Pu Zhao, Bo Qiao, Ray Huang, Si Qin, Qisheng Su, Jiayi Ye, Yudi Zhang, Jian-Guang Lou, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. Large action models: From inception to implementation. *arXiv preprint arXiv:2412.10047*, 2025.

[53] Andreui Nikolaevich Kolmogorov. Turbulence: the legacy of an kolmogorov. *Cambridge university press*, 1995.

[54] Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

[55] Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Chemcrow: Augmenting large-language models with chemistry tools. 2023.

[56] Alireza Ghafarollahi and Markus J. Buehler. Sciagents: Automating scientific discovery through bioinspired multi-agent intelligent graph reasoning. *Advanced Materials*, 37(22):2413523, 2025.

[57] Alireza Ghafarollahi and Markus J Buehler. Automating alloy design and discovery with physics-aware multimodal multiagent ai. *Proceedings of the National Academy of Sciences*, 122(4):e2414074122, 2025.

[58] Jiaxuan Liu, Tiannian Zhu, Caiyuan Ye, Zhong Fang, Hongming Weng, and Quansheng Wu. Vaspilot: Mcp-facilitated multi-agent intelligence for autonomous vasp simulations. *Chinese Physics B*, 34(11):117106, nov 2025.

[59] Fiona Y. Wang, Lee Marom, Subhadeep Pal, Rachel K. Luu, Wei Lu, Jaime A. Berkovich, and Markus J. Buehler. Autonomous agents coordinating distributed discovery through emergent artifact exchange, 2026.

[60] Alireza Ghafarollahi and Markus J. Buehler. Sparks: Multi-agent artificial intelligence model discovers protein design principles, 2025.