TritonDFT: Automating DFT with a Multi-Agent Framework

Anonymous Authors[1]

Abstract

Density Functional Theory (DFT) is a cornerstone of materials science, yet executing DFT in practice requires coordinating a complex, multi-step workflow. Existing tools and LLM-based solutions automate parts of the steps, but lack support for full workflow automation, diverse task adaptation, and accuracy–cost trade-off optimization in DFT configuration. To this end, we present TRITONDFT, a multi-agent framework that enables efficient and accurate DFT execution through an expert-curated, extensible workflow design, Pareto-aware parameter inference, and multi-source knowledge augmentation. We further introduce DFTBENCH, a benchmark for evaluating the agent's multi-dimensional capabilities, spanning science expertise, trade-off optimization, HPC knowledge, and cost efficiency.

TRITONDFT provides an open user interface for real-world usage. Our source code and benchmark suite are available at https://anonymous.4open.science/r/TritonDFT-43C7/.

1. Introduction

Density Functional Theory (DFT) (Hohenberg & Kohn, 1964; Kohn & Sham, 1965) stands as the computational cornerstone of modern materials science. As a first-principles method, DFT provides high-fidelity predictions to validate theoretical hypotheses and reduce experimental cost.

Executing DFT in practice involves a complex, multi-step workflow. Practitioners must search for structural information, configure input parameters, write DFT software-specific scripts, launch and monitor HPC jobs, and interpret and analyze execution results. As shown in Figure 1, such steps require distinct areas of expertise, including physics and materials science, DFT library software details, and High-Performance-Computing (HPC). Each step takes minutes to hours of manual effort. This imposes substantial overhead and slows down the discovery process. While existing DFT tools can handle certain low-level details, such as script generation (Mathew et al., 2017; Larsen et al., 2017) and HPC resource management (Pizzi et al., 2016), users still need to manually handle most of the steps and coordinate the overall workflow.

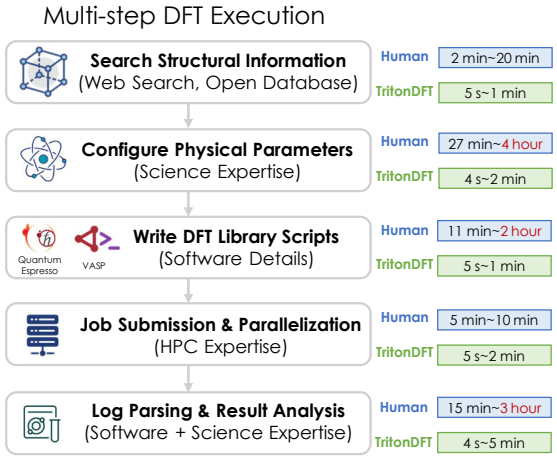

Figure 1. DFT execution is a complex, multi-step process requiring heterogeneous domain expertise. Based on an internal survey conducted with 19 domain researchers at the PhD level or above, manually handling each step typically takes minutes to hours. TRITONDFT reduces the per-step time to the scale of seconds to minutes, and provides automation across the entire workflow.

Such manual overhead gives rise to a natural question: can we leverage Large Language Model (LLM)-based agents to orchestrate these steps and enable automation? LLM agents have been successfully applied across materials science, including domain-specific tool augmentation (M. Bran et al., 2024; Zhang et al., 2024), theoretical hypothesis loop (Ding et al., 2024; Kumbhar et al., 2025), and autonomous laboratory (Szymanski et al., 2023; Dai et al., 2025). However, applying LLM-based agents to DFT execution, a highly complex, large-scale, and cross-domain theoretical tool, remains challenging and underexplored. These challenges directly motivate our design of TRITONDFT.

First, the complexity of DFT increases the difficulty to implement a robust and generalizable agent framework. Mod-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

ern DFT libraries like `Quantum Espresso` (Giannozzi et al., 2009) comprise $>$ 10 executables and $>$ 50 commonly used parameters, with completely different invocation patterns and analyzing methods across tasks. While prior work have demonstrated feasibility on specific tasks such as structural relaxation or adsorption (Wang et al., 2025; Hafner, 2008), they typically rely on static, task-specific workflows. In contrast, TRITONDFT adopts an expert-informed Plan–Execute–Refine workflow design, coupled with an explicit task–to–executable mapping mechanism for extensibility. TRITONDFT now supports a broad set of tasks, ranging from structural optimization to complex tasks such as phonon properties.

Second, as a numerical simulation, DFT requires results to be obtained both accurately and efficiently. Thus, DFT parameter configurations simultaneously affect numerical fidelity and computational workload, introducing an inherent accuracy–cost trade-off. Existing agent study on parameter configuration (Xia et al., 2025) primarily focuses on accuracy while leaving this trade-off problem unsolved. TRITONDFT introduces a Pareto-aware parameter inference method, which enables the LLM to estimate the accuracy–cost Pareto frontier and iteratively refine configurations. We further incorporate augmentation including domain-specific tools, historical memory mechanism, and an interactive human-in-the-loop interface, to improve the agent's reliability on the parameter configurations.

Third, despite extensive benchmarks like graduate-level materials-domain knowledge (Zaki et al., 2024; Mirza et al., 2024), key capabilities in end-to-end DFT workflows, including numerical accuracy, Pareto-optimality, HPC parallelization, and cost efficiency, remain unevaluated. We present DFTBENCH to evaluate these capabilities. DFTBENCH comprises 100 materials spanning 10 distinct types to ensure diversity in physical properties and computational complexity, with expert-curated convergence tests totaling over 500 CPU-hours to obtain Pareto-optimal parameter configurations and ground-truth calculation results.

In summary, our contribution can be summarized as follows:

- We present TRITONDFT, an expert-informed automation framework for complete DFT workflow execution.
- We present DFTBENCH, a benchmark suite for multi-dimensional capabilities to automate DFT workflows, spanning science, HPC and accuracy-cost tradeoffs.
- Comprehensive experiments demonstrate the potential of TRITONDFT as a practical building block for real-world research. We also reveal substantial capability differences of LLMs across different task steps.

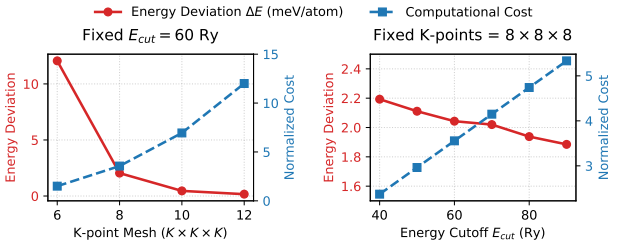

Figure 2. Energy Deviation and Computational Cost Variations with different DFT Parameters for Silicon (Space Group $Fd\bar{3}m$).

2. Background and Related Work

2.1. DFT Execution and Challenges

DFT predicts material properties from first principles, based on the atom identities and structural information. Its practical execution entails a multi-step workflow. We formulate a DFT query $\mathcal{Q}$ as a chain of computational steps $\{C_i\}_{i=1}^{N}$, where each step $C_i = \langle \mathcal{M}_i, \boldsymbol{\theta}_i, \mathcal{O}_i \rangle$ transforms the input state of a specific material via method $\mathcal{M}_i$ (e.g., structure relaxation, self consistent field) into output $\mathcal{O}_i$. $\mathcal{O}_i$ encompasses the updated physical state (e.g., atomic geometry, electron density) and computed observables (e.g., total energy, band gap, density of states).

The parameter space $\boldsymbol{\theta}_i$ includes two components: *Physical parameters* $\boldsymbol{\theta}_{\text{phy}}$ (e.g., kinetic energy cutoffs, k-point densities, and more advanced parameters like spin-orbit coupling) govern numerical accuracy and convergence behavior. It also determines computational workload. *HPC parameters* $\boldsymbol{\theta}_{\text{hpc}}$ (e.g., parallelization over k-points or images) determine how parallel tasks are mapped onto hardware resources, influencing execution time and resource efficiency. Thus, efficient DFT execution requires expertise in both materials science and high-performance computing.

Complexity in Manual DFT Execution. Widely used DFT packages such as `Quantum ESPRESSO` (Giannozzi et al., 2009) and `VASP` (Hafner, 2008) require substantial manual effort in practice. Researchers must retrieve structural data from external sources, construct software-specific input scripts, submit and monitor HPC jobs, and interpret lengthy execution logs. Existing DFT tools partially automate this workflow. For example, Atomate (Mathew et al., 2017) and ASE (Larsen et al., 2017) support structure-to-script generation, and AiiDA (Pizzi et al., 2016) manages HPC job execution and scheduling. However, existing tools typically operate on parts of the steps and do not provide intelligent guidance for parameter configuration or execution. As a result, DFT researchers still need to manually make decisions and coordinate the overall workflow.

Parameters with Accuracy–Cost Tradeoff. DFT execution time can range from minutes to hours or even days,

Table 1. Comparison of TRITONDFT with state-of-the-art agentic DFT frameworks.

Method	Framework Architecture			Evaluation Dataset & Metrics				
	Supported Task Types	Parameter Configuration	Knowledge Augmentation	Number of Material Types	Ground Truth Curation	Accuracy-Cost Tradeoff	Parallel Efficiency	Monetary Cost
DREAMS (Wang et al., 2025)	Surface Chemistry (Adsorption)	Physics Only	Open Database	2 (Metal, Insulator)	Public Dataset	✗	✗	✗
VASPilot (Liu et al., 2025)	Electronic Structure (Band, DOS)	Physics Only	Open Database	1 (Semiconductor)	Public Dataset	✗	✗	✗
AgenticDFT (Xia et al., 2025)	Geometry & Energetics (Relaxation, Band)	Physics Only	Open Database	2 (Metal, Semiconductor)	Public Dataset	✗	✗	✗
TRITONDFT (Ours)	**General QE Usage (>10 Types)**	**Physics + HPC (Pareto-aware)**	**Open Database + Memory + Human Interact.**	**10 (Metal, Insulator, Superconductor, Topological, ...)**	**Expert Curated Calculation**	✓	✓	✓

which is largely determined by parameters $\boldsymbol{\theta}$. Figure 2 presents one example: increasing some physical parameters like k-point and energy cutoff reduces calculation error but incurs higher computational cost. This introduces a tradeoff: DFT practitioners need to identify *Pareto-optimal* configurations, to maximize the efficiency while meeting the required accuracy. More importantly, the non-trivial coupling among multiple parameters further exacerbates the complexity of this tradeoff. Thus, users must either conduct expensive convergence tests or rely on extensive experience.

2.2. LLM Agents for Automated Material Discovery

LLM agents are fundamentally reshaping scientific research by automating complex discovery cycles (Wang et al., 2023), enabled by the capabilities of tool-usage (Yao et al., 2024) and multi-step workflow coordination (Hong et al., 2023).

Domain-Specific Tool Augmentation. Prior work augments agents with domain-specific tools, including paper extraction (Cheung et al., 2024; Hira et al., 2024; Song et al., 2023), retrieval augmentation (Schilling-Wilhelmi et al., 2025; Chiang et al., 2024; McNaughton et al., 2024), and surrogate models (Liu et al., 2024). Tool-hub-based agents like HoneyComb (Zhang et al., 2024) and ChemCrow (M. Bran et al., 2024) assemble multiple tools to enable coordinated augmentation. These tools typically follow simple input–output protocols and can be effectively encapsulated as single function calls. In contrast, using DFT as a tool require more complex workflows, involving physical parameter configuration, HPC job parallelization and management, result analysis and iterative refinement.

Automated Agent Framework. On the theoretical side, LLM agents have demonstrated promising capabilities in knowledge organization (Tang et al., 2025; Ye et al., 2024; Shetty et al., 2023), property prediction (Song et al., 2025; Yao et al., 2025), hypothesis proposal (Ding et al., 2024; Kumbhar et al., 2025), and novel material generation (Gruver et al., 2024; Qi et al., 2025; Ghafarollahi & Buehler, 2025; Wang et al., 2024; Jia et al., 2024). On the experimental side, agent-driven embodied systems enable increasingly autonomous wet-lab workflows for synthesis (Szymanski et al., 2023; Delgado-Licona et al., 2025), characterization (Dai et al., 2025), and measurement (Olowe & Chitnis, 2025; Boiko et al., 2023), substantially expanding the scale of materials discovery (Merchant et al., 2023).

DFT as a cornerstone should naturally be incorporated into such discovery automation. Recent efforts demonstrate the feasibility of DFT agents (Wang et al., 2025; Liu et al., 2025; Xia et al., 2025). However, these works are primarily case-study driven, focusing on the correctness on specific tasks and materials, and lack evaluation of accuracy, efficiency and monetary cost. Table 1 summarizes the key differences between TRITONDFT and existing DFT agents.

Benchmarking Agents in Materials Science. A growing number of LLM benchmarks has been proposed in material science, including graduate-level question answering (Zaki et al., 2024; Mirza et al., 2024; Zhang et al., 2025; Cheung et al., 2025), research-level task completion (Miret & Krishnan, 2024; Guo et al., 2023), specific tool usage (Huang et al., 2025), retrieval-augmented reasoning (Zhong et al., 2025), and wet-lab experimental design and analysis (Mandal et al., 2025). However, DFT requires multi-dimensional capabilities including science, HPC, and dual optimization, which are not fully captured by existing benchmarks. This motivates the design of DFTBENCH.

3. TRITONDFT: Method and Implementation

We propose TRITONDFT, a multi-agent framework designed to achieve fully automated DFT calculations. The framework integrates an LLM-driven agent workflow with `Quantum Espresso` (Giannozzi et al., 2009), one of the most prominent open-source DFT libraries. Shown in Figure 3, TRITONDFT accepts task descriptions directly in natural language and autonomously orchestrates the entire DFT workflow, allowing researchers to obtain simulation results without specialized knowledge in computational physics.

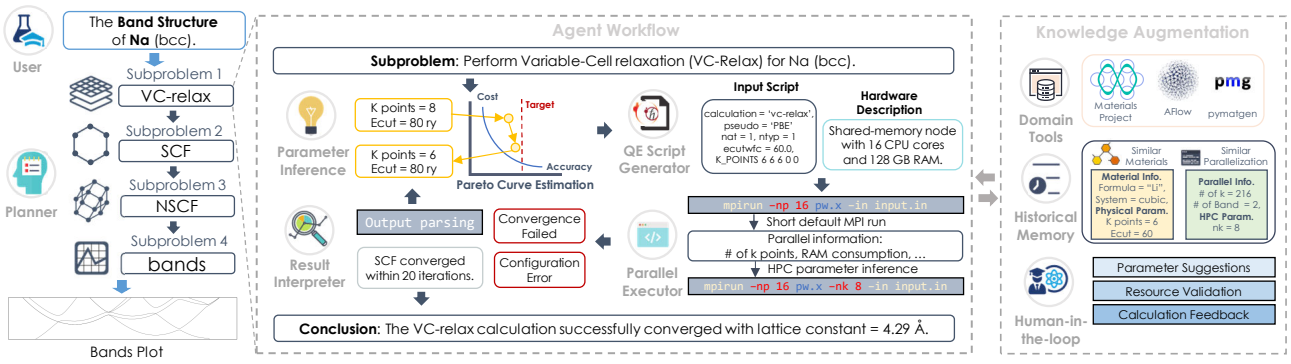

Figure 3. The overview of TritonDFT framework.

TritonDFT was developed through a close collaboration between theoretical material scientists and computer science researchers. The agent workflow encapsulates the empirical routines typically employed by experienced physicists. The input prompts are rigorously refined based on both official library documentation and expert domain knowledge.

3.1. Agentic Workflow for Automation

We adopt a Planner Agent that dynamically decomposes high-level user queries into a sequence of DFT subproblems. Each category establishes a deterministic mapping to a specific binary executable within the DFT library suite. For instance, the structural relaxation (VC-relax) and self-consistent-field calculation (SCF) are mapped to `pw.x`. This design ensures every subproblem to be solved through executable invocations. Each executable is registered with structured metadata describing input/output formats and execution constraints. This enables new executables to be added in a modular manner.

Within each subproblem, TritonDFT executes a closed-loop solving workflow: (1) **Parameter Inference** leverages Pareto-optimal-aware reasoning to identify $\boldsymbol{\theta}_{\text{phy}}$ for expected accuracy-cost tradeoff; (2) **Script Generator** synthesizes these parameters into syntactically correct input files; (3) **Executor** performs job submission and setup $\boldsymbol{\theta}_{\text{hpc}}$; (4) **Interpreter** parses the raw output to provides correctness and produces refinement suggestions or summaries.

Pareto-Aware Parameter Refinement. To navigate the inherent accuracy-cost trade-off and address the bi-objective optimization problem, we introduce a *Pareto-aware Reasoning* mechanism. Instead of static, one-shot inference, the agent is designed to iteratively infer parameter configurations based on result estimation. The agent estimates the discrepancy between the numerical accuracy of the current configuration and user expectations, and closes the feedback loop through iterative parameter self-refinement, ultimately converging toward an estimated Pareto-optimal point. The agent also leverage execution-based feedback by validating DFT outputs against correctness and convergence criteria. Configurations that fail to achieve convergence are treated as invalid points in the accuracy–cost space and excluded from Pareto consideration. Such a refinement process allows the agent to iteratively adjust parameters rather than making a single-shot guess.

Automated Parallelization. The executor automatically setup parallelization parameters to maximize execution efficiency. It takes as input a natural-language hardware specification (e.g., "32 CPU cores on a 128 GB shared-memory node"), together with the generated input scripts and an estimation of the resource cost to finish the calculation. Following common practices adopted by DFT experts, the agent estimates the resource cost via a short default MPI run (within 30 seconds), and extracts key signals such as number of k-points and total DRAM consumption. These signals provide a lightweight estimation of both parallel scalability and memory footprint, and are jointly used to guide parallelization setup and avoid oversubscription.

3.2. Domain Knowledge Augmentation

TritonDFT leverages external domain-specific tools and internal memory module for augmented generation.

Domain-Specific Tool Integration. TritonDFT integrates open materials databases, including Materials Project (Jain et al., 2013) and AFlow (Curtarolo et al., 2012), to retrieve physically grounded information that guides parameter guessing and script construction. The agent dynamically selects specific querying fields based on the current task type. For example, the agent retrieves initial atomic structures for structure relaxation and reasonable lattice constants for subsequent Self Consistent Field. We also integrate `pymatgen` (Ong et al., 2013) to enable the agent to perform symmetry analysis and space group verification, ensuring the geometric consistency of the structures.

Historical Memory Mechanism. Our system implements two forms of memory: For $\boldsymbol{\theta}_{\text{phy}}$, the agents recall param-

eters from physically similar materials. For successfully converged calculations, the agent summarizes and stores the material information and $\boldsymbol{\theta}_{\mathrm{phy}}$ in memory. During retrieval, the agent first applies structured filtering based on high-level symmetry features, including space group and crystal system. Within the filtered candidates, similarity is computed over other features such as elemental composition, electron count, and unit-cell volume. For $\boldsymbol{\theta}_{\mathrm{hpc}}$, the agent directly compares workload-related features, such as the total number of k-points, the size of the plane-wave basis, and the number of bands. Such historical execution results provides estimation guidance for the accuracy–cost trade-off.

3.3. User Interface and Interaction

We adopt a decoupled design that separates the agent backend from the DFT computation backend. This only requires users to connect their own computation platform and specify the task submission method for that environment. Currently, the system supports execution on both local servers and Slurm-managed clusters. With this design, TRITONDFT serves as a comprehensive intermediate layer between the user and the low-level hardware. This relieves users from trivial domain-specific details and tedious manual tasks, such as parameter configuration, parallelization configuration, script writing, progress monitoring, and result analysis.

We have developed an open web interface for TRITONDFT, allowing users to interact with pure natural language. We also support the human-in-the-loop feedback mechanism: Users can intervene at specific stages to provide feedback, such as reviewing $\boldsymbol{\theta}_{\mathrm{phy}}$ configurations, validating $\boldsymbol{\theta}_{\mathrm{hpc}}$ settings and submission commands, or assessing execution results. This capability enables granular control for researchers with varying levels of expertise, establishing a reliable building block for practical research.

4. DFTBENCH: Benchmarking DFT Accuracy and Efficiency with LLM Agents

Benchmark Statistics. DFTBENCH is an expert-curated benchmark suite comprising 100 unique crystalline materials, exhibiting diversity in two aspects:

Physical Diversity: DFTBENCH covers 10 distinct material categories, ranging from fundamental electronic phases (Metals, Insulators, Semiconductors) to complex functional and quantum materials (Superconductors, Topological insulators, Ferroelectrics). The material set contains 47 chemical elements, and 23 crystallographic space groups, spanning multiple electronic phases and magnetic ground states. Such diversity in composition, symmetry, and electronic structure requires the model to understand different material properties and precisely configure $\boldsymbol{\theta}_{\mathrm{phy}}$.

Computational Complexity Diversity: The dataset covers a broad range of system sizes, from single-atom primitive cells to polyatomic unit cells, with varying unit cell volumes and total valence electron counts. Such diversity requires the model to efficiently configure $\boldsymbol{\theta}_{\mathrm{hpc}}$ to handle workloads spanning multiple orders of magnitude.

Evaluation Design. Our test suite evaluates the agent in automated DFT workflows along following three dimensions:

(i) Accuracy-Cost Trade-off and Pareto-Optimal Parameter Reference . To evaluate numerical accuracy, we define three target energy deviation levels, $\Delta E < 1$, 10, and 20 meV/atom. These thresholds reflect commonly adopted DFT accuracy levels under different accuracy–cost trade-offs. The stringent threshold of 1 meV/atom is computational expensive, targeting energy-sensitive properties (e.g., relative phase stability and defect energetics). 10 meV/atom is commonly used for standard high-throughput accuracy (e.g., equilibrium structures and band properties). 20 meV/atom is coarse-grained but with lowest computational cost, commonly used for coarse-grained screening, structure filtering, and exploratory data generation.

For each material, we provide fully expert-curated input parameter configurations that satisfy these three accuracy targets, serving as reference Pareto-optimal configurations. These configurations are obtained through manual convergence testing, involving over 500 CPU-hours of DFT runs to sweep and test relevant numerical parameters, and identifying optimal configurations on the Pareto curve. We evaluate the agent by comparing its generated parameter configurations against these reference configurations, assessing its ability to identify Pareto-optimal points under different energy error tolerances.

(ii) HPC Parallelization. We evaluate the execution time for each test case under a default `mpirun` setup, where `np` denotes the total number of parallel processes, and compare it against agent-generated configurations that leverage advanced DFT parallelization parameters such as `nk` and `ntg`. This evaluation assesses the agent's capability of effective parallel acceleration for DFT workloads.

(iii) Cost evaluation. We also measure the number of tokens and the total runtime of LLM calls. These quantities are converted into the corresponding API costs and end-to-end workflow throughput, to evaluate both time and monetary costs of the agent in real-world usage.

5. Evaluation

5.1. Experimental Setup

Evaluated Models. We benchmark eight state-of-the-art models across three major families: OpenAI's GPT 5.2, GPT 5.1, GPT 4o, and GPT 4o mini; Google's Gemini 2.5 Pro and Gemini 2.5 Flash; and Anthropic's Claude Opus

Table 2. Model performance on DFT parameter configuration across different LLMs under varying error threshold.

Model	$\Delta E < 20$ meV/atom		$\Delta E < 10$ meV/atom		$\Delta E < 1$ meV/atom		Advanced Parameter Satisfaction Rate
	Pass Rate	Cost Factor	Pass Rate	Cost Factor	Pass Rate	Cost Factor	
GPT 5.2	70.5%	14.29	67.0%	8.95	47.1%	4.23	51.3%
GPT 5.1	39.3%	6.22	32.9%	4.21	9.8%	2.24	43.6%
GPT 4o	52.8%	1.85	38.2%	1.28	13.6%	0.50	28.2%
GPT 4o mini	5.7%	1.01	5.6%	1.17	4.5%	0.97	28.2%
Gemini 2.5 Pro	59.6%	3.77	53.9%	2.95	14.9%	1.24	48.7%
Gemini 2.5 Flash	23.6%	1.85	16.9%	1.68	2.3%	0.78	38.5%
Claude Opus 4.5	9.0%	1.62	5.6%	1.33	4.5%	0.58	**53.8%**
Claude Sonnet 4.5	30.3%	2.38	25.8%	1.93	21.6%	0.87	38.5%

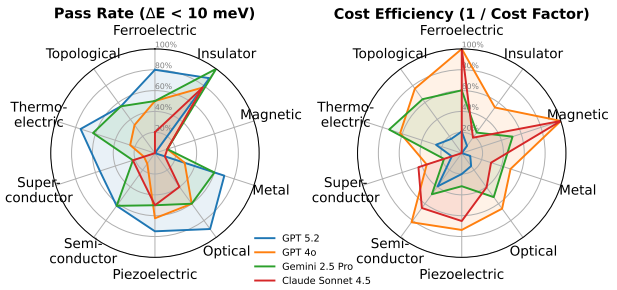

Figure 4. Performance Analysis with Pass Rate and Cost Efficiency across different material types. Cost Efficiency is measured (1 / Cost Factor), averaged over all passed cases within each type.

4.5 and Claude Sonnet 4.5. This selection spans a broad spectrum of model architectures and scales, from flagship frontier models to more cost-efficient ones.

Implementation and Platform. TRITONDFT integrates `Quantum ESPRESSO` (v7.4) with a unified API interface for accessing diverse commercial LLMs. The framework is deployed on a high-performance computing node equipped with an AMD EPYC 9534 64-Core Processor clocked at 3.48 GHz. The CPU supports advanced vector extensions like `AVX-512`, ensuring optimized execution for DFT.

Test Cases. We perform benchmarking primarily on four DFT tasks: variable-cell relaxation (VC-relax), self-consistent field (SCF), band gap, and density of states (DOS). These tasks cover the most fundamental steps in DFT workflows. We also present examples on more advanced tasks such as phonon analysis in case study.

5.2. Accuracy and Trade-off Analysis

DFT Parameter Evaluation. We evaluate the model performance in setting parameters for structural relaxation(VC-relax). It is the most fundamental step that determines the quality of the relaxed geometry and subsequent energy evaluations. Table 2 summarizes the results. Pass Rate denotes the fraction of cases where the generated configuration satisfies $\Delta E < 20, 10, 1$ meV/atom, compared to the most accurate results obtained with the most stringent configuration. Cost Factor is measured as the ratio of computational cost relative to the Pareto-optimal configuration obtained by convergence test. Values closer to 1 indicate stronger capability in identifying Pareto-optimal configurations. We also report the Advanced Requirement Satisfaction Rate, corresponding to parameters required by more complex materials such as spin polarization, Hubbard U and van der Waals corrections. In total, 39 materials in DFTBENCH require one or more such advanced parameters.

Table 3. Mean absolute error (MAE, %) across different DFT tasks, computed over successfully finished execution results.

Model	VC-relax	SCF	Band Gap	DOS
GPT 5.2	0.04	0.04	0.09	0.97
GPT 5.1	0.06	0.07	0.31	2.21
GPT 4o	0.10	1.11	2.48	9.04
Gemini 2.5 Pro	0.05	0.09	1.14	1.40
Gemini 2.5 Flash	0.06	0.83	1.21	11.17
Claude Opus 4.5	0.06	0.11	2.10	3.00
Claude Sonnet 4.5	0.09	0.14	2.00	2.12

Among the evaluated models, GPT 5.2 consistently achieves the highest pass rates across all thresholds. This advantage comes at the expense of higher computational cost (with average cost factor up to 14.29). Intermediate models, including GPT 5.1, GPT 4o, and Gemini 2.5 Pro, exhibit more balanced behavior, with moderate pass rates and lower cost factors with 20,10 meV/atom. However, under the strict 1 meV/atom threshold, pass rates of these models drop below 15%. We further observe that the Claude 4.5 family exhibits lower pass rates, with Opus underperforming Sonnet. Based on parameter-level analysis, Opus 4.5 tends to generate aggressively low-cost configurations (< 1.62). However, the insufficient accuracy estimation causes suboptimal configuration with frequent threshold violation.

Regarding advanced parameters, models with stronger reasoning capabilities generally achieve higher satisfaction rates. We also observe that models exhibit distinct strengths

Table 4. Relative speedup (%) of parallel execution of different LLMs over the default baseline across different CPU core numbers.

Model	16 Cores	32 Cores	64 Cores
GPT 5.2	+14.4%	+4.2%	+15.1%
GPT 5.1	+11.3%	-5.8%	-14.1%
GPT 4o	-21.0%	-34.0%	-23.6%
GPT 4o mini	-25.7%	-25.6%	+2.8%
Gemini 2.5 Pro	4.74%	-6.4%	-3.29%
Gemini 2.5 Flash	-20.7%	-43.7%	-32.0%
Claude 4.5 Opus	**+15.4%**	**+16.1%**	**+16.1%**
Claude 4.5 Sonnet	+13.0%	+5.1%	+2.43%

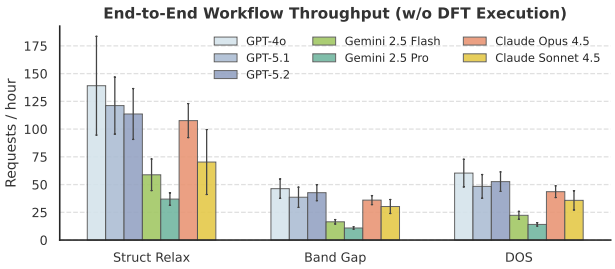

Figure 5. Comparison of TRITONDFT's workflow throughput with different LLMs across different DFT tasks.

across specific parameters. For example, Claude Opus 4.5 is the most reliable model in Hubbard U, while Gemini-2.5-Pro and Gemini-2.5-Flash are the only models that identify cases requiring van der Waals corrections. Interestingly, Opus 4.5 achieves the highest advanced parameter satisfaction rate. This demonstrates its deep domain knowledge, despite weaker cost–accuracy trade-off.

Performance across Material Types. We further compare model performance across different material types in DFTBENCH, as shown in Figure 4. We observe that relatively simple systems, such as metals, semiconductors, and insulators, achieve higher pass rates across models. These materials exhibit weak sensitivity to physical parameters and smoother convergence behavior. In contrast, performance degrades on more complex systems, particularly magnetic materials, where all models exhibit low pass rates ($< 6\%$). Magnetic systems require careful treatment of spin polarization and magnetic ordering, and are highly sensitive to the choice of initial magnetic moments and convergence parameters. We also notice that GPT 5.2 performs better on ferroelectric and optical materials, indicating higher reliability on moderately complex systems.

In terms of cost efficiency, GPT 4o performs best, while Gemini 2.5 Pro strikes a more balanced trade-off between pass rate and cost efficiency across different types. We also observe that for materials such as superconductors and insulators, cost efficiency tends to be lower. Although these systems are easier to achieve numerical convergence, models still tend to adopt more conservative parameter configurations, leading to unnecessary computational cost.

End-to-End Workflow Evaluation. We report the mean relative error of final results for different task workflow, using the maximum relative deviation of lattice parameters for VC-relax, total energy per atom for SCF, band gap for band calculations, and Fermi level for DOS. All results are compared against the most stringent human-configured results. Across all four tasks, stronger models achieve consistently high accuracy, with GPT-5.2 maintaining errors within 1%, while GPT 5.1 and Gemini 2.5 Pro remain within 3%, demonstrating the reliability of LLM-based agents for such fundamental tasks in end-to-end DFT workflows.

5.3. Time and Cost Efficiency Analysis

Automatic Parallelization. We evaluate the execution efficiency of the structure relaxation phase, which dominates over 50% of the end-to-end runtime. Table 4 shows speedups of LLM-generated parallel configurations over the default MPI baseline (`mpirun -np <cores>`).

Models with superior coding and reasoning capabilities, such as Claude 4.5 Opus and GPT 5.2, deliver consistent performance gains, achieving peak speedups of up to 16.1%. This confirms that effective HPC parallelization requires deep reasoning to map physical tasks onto hardware specifications. In contrast, mid-tier and smaller models (e.g., GPT-4o, Gemini 2.5 Flash) exhibit performance degradation. Our analysis reveals that these models attempt aggressive optimizations without fully grasping strict parallel constraints, leading to suboptimal configurations, such as over-parallelization on small-scale systems, or setting a number of k-points `nk` not divisible by the number of pools.

As the number of cores increases, the complexity of identifying valid parallel parameters rises due to stricter constraints on task divisibility and communication overhead. Most high-performing models exhibit varying degrees of performance degradation at larger scales. Notably, Claude 4.5 Opus stands out as the most robust model, maintaining consistent peak speedups ($\approx$16%) across all hardware setups.

Workflow Throughput. Figure 5 illustrates the end-to-end throughput of the DFT setup workflow. We exclude the execution time of DFT and focus only on the automation overhead. TRITONDFT achieves an effective request rate of approximately 10–100 queries/hour, depending on task complexity and API latency. In contrast, our survey on PhD-level practitioners indicates that manually setup DFT workflows sustain less than 1 request/hour. Overall, TRITONDFT delivers $> 10\times$ efficiency improvement.

Monetary Cost. We calculated the input and output tokens of TRITONDFT across three tasks and estimated the mone-

Table 5. Average cost consumption (USD) per query regarding API usage across different tasks.

Model	Struct Relax	Band Gap	DOS
GPT 5.2	0.05 ± 0.02	0.15 ± 0.04	0.13 ± 0.04
GPT 5.1	0.04 ± 0.02	0.13 ± 0.04	0.10 ± 0.03
GPT 4o	0.06 ± 0.03	0.18 ± 0.04	0.14 ± 0.03
Gemini 2.5 Pro	0.05 ± 0.03	0.13 ± 0.04	0.11 ± 0.04
Gemini 2.5 Flash	0.01 ± 0.01	0.03 ± 0.01	0.03 ± 0.01
Claude Opus 4.5	0.15 ± 0.08	0.44 ± 0.13	0.37 ± 0.11
Claude Sonnet 4.5	0.15 ± 0.06	0.34 ± 0.09	0.28 ± 0.08

tary cost based on official API pricing. The results are shown in Table 5. Different models exhibit significant divergence in economic efficiency. Gemini 2.5 Flash demonstrates exceptional cost-effectiveness, maintaining an average cost as low as \$0.01–0.03 per query. The GPT-5 series and Gemini 2.5 Pro show comparable pricing with \$0.04–\$0.18 per query. In contrast, the Claude series incurs the highest costs, reaching up to \$0.44 per query for complex calculations.

5.4. Case Study

Pareto-aware Reasoning. Figure 6 compares the accuracy–cost trade-offs obtained on representative materials under three target thresholds (20, 10, and 1 meV/atom). We contrast one-shot parameter inference with Pareto-aware iterative reasoning (Pareto). Overall, Pareto-aware reasoning enables models to adaptively navigate the accuracy–cost frontier. GPT 5.2 benefits the most, achieving up to 4.1× reductions in normalized workload while still meeting the target accuracy constraints. Gemini 2.5 Pro tends to select a fixed, relatively low-accuracy configuration, yet its choices closely follow the ground-truth frontier. In contrast, Claude Opus 4.5 tends to select suboptimal configurations on complex materials (e.g., topological systems), leading to frequent violations of the prescribed accuracy budgets.

Broader Task Types. TritonDFT is designed to handle a broad set of DFT task types. Figure 7 shows a representative example, where all intermediate steps are automatically handled by the agent except for figure plotting. With our Plan–Execute–Refine workflow, the agent autonomously plans the entire solution path (`pw.x–ph.x–q2r.x–matdyn.x` in this case) without user specification. This design eliminates the need for defining static and task-specific workflows in prior work (Wang et al., 2025).

We also observe that the agent exhibits clear adaptivity to different user intents by adjusting key numerical parameters. For example, in phonon calculations, the agent dynamically adapts the phonon q-point sampling density when performing Γ-point and full phonon dispersion calculations. In elastic energy–strain analysis that is highly sensitive to numerical noise, the agent employs stricter electronic and force convergence thresholds compared to relaxation tasks.

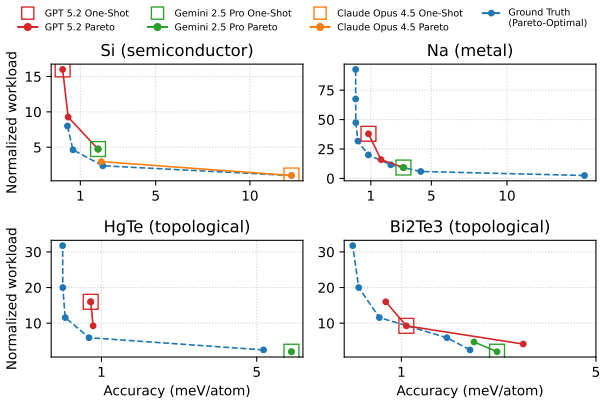

Figure 6. Accuracy-Cost Trade-offs across different models and parameter inference methods when using One-Shot inference and Pareto-aware inference, respectively.

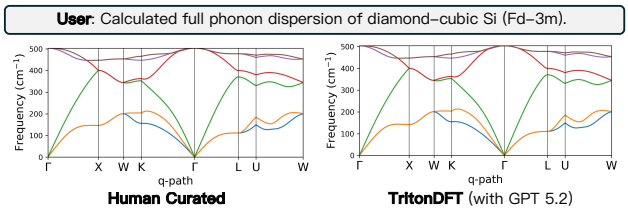

Figure 7. Comparison of TritonDFT's results with human-curated results on phonon dispersion calculation.

6. Conclusion

In this work, we presented TritonDFT, a multi-agent framework that automates Density Functional Theory (DFT) workflows through expert-informed planning, Pareto-aware parameter inference, and automated HPC orchestration. To assess multi-dimensional agentic capabilities in this domain, we introduced DFTBench, a benchmark spanning diverse materials to evaluate numerical accuracy, parallelization efficiency, and cost-effectiveness. Our evaluation confirms that TritonDFT delivers a >10× acceleration over manual expert execution. We further identify distinct strengths across models, such as GPT-5.2 excels in accuracy, Gemini 2.5 Flash provides a better accuracy–cost tradeoff, and Opus 4.5 performs better in parallelization schemes.

Future Direction. Future development will address the observed limitations in modeling complex quantum states, such as magnetic materials, where current models achieve pass rates below 6%, by integrating specialized physics-informed reasoning modules. We aim to expand the framework's modularity beyond `Quantum Espresso` to support diverse DFT solvers, enhancing portability across the materials science ecosystem. Additionally, we plan to integrate TritonDFT with autonomous wet-lab platforms for automated theoretical simulation and validation, to enable a low-cost, high-throughput in closed-loop discovery.

Impact Statement

This paper presents work whose primary goal is to accelerate scientific discovery by automating complex Density Functional Theory (DFT) workflows. By democratizing access to high-fidelity material simulations, TRITONDFT has the potential to expedite the development of critical technologies, such as clean and low-energy materials and novel semiconductors. A key societal benefit of this work lies in its emphasis on computational efficiency; the framework's Pareto-aware parameter inference and automated parallelization are specifically designed to optimize High-Performance Computing (HPC) resource usage, thereby reducing the energy footprint associated with large-scale scientific simulations.

We explicitly incorporate a human-in-the-loop mechanism to ensure expert oversight and validation of results for two reasons. First, the automation of material design may carry theoretical risks regarding dual-use applications (e.g., the design of hazardous materials). Second, while this automation democratizes access to complex simulation tools for non-experts, the reliance on probabilistic models necessitates robust verification mechanisms to prevent the propagation of scientific errors. We do not foresee other specific negative societal consequences beyond those broadly associated with the application of ML to scientific automation.

References

Boiko, D. A., MacKnight, R., Kline, B., and Gomes, G. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Cheung, J., Zhuang, Y., Li, Y., Shetty, P., Zhao, W., Grampurohit, S., Ramprasad, R., and Zhang, C. Polyie: A dataset of information extraction from polymer material scientific literature. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2370–2385, 2024.

Cheung, J. J., Shen, S., Zhuang, Y., Li, Y., Ramprasad, R., and Zhang, C. Msqa: Benchmarking llms on graduate-level materials science reasoning and knowledge. *arXiv preprint arXiv:2505.23982*, 2025.

Chiang, Y., Hsieh, E., Chou, C.-H., and Riebesell, J. Llamp: Large language model made powerful for high-fidelity materials knowledge retrieval and distillation. *arXiv preprint arXiv:2401.17244*, 2024.

Curtarolo, S., Setyawan, W., Hart, G. L., Jahnatek, M., Chepulskii, R. V., Taylor, R. H., Wang, S., Xue, J., Yang, K., Levy, O., et al. Aflow: An automatic framework for high-throughput materials discovery. *Computational Materials Science*, 58:218–226, 2012.

Dai, Y., Chan, H., Vriza, A., Fan, J., Kim, F., Wang, Y., Liu, W., Shan, N., Xu, J., Weires, M., et al. Adaptive ai decision interface for autonomous electronic material discovery. *Nature Chemical Engineering*, pp. 1–11, 2025.

Delgado-Licona, F., Alsaiari, A., Dickerson, H., Klem, P., Ghorai, A., Canty, R. B., Bennett, J. A., Jha, P., Mukhin, N., Li, J., et al. Flow-driven data intensification to accelerate autonomous inorganic materials discovery. *Nature Chemical Engineering*, 2(7):436–446, 2025.

Ding, Q., Miret, S., and Liu, B. Matexpert: Decomposing materials discovery by mimicking human experts. *arXiv preprint arXiv:2410.21317*, 2024.

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

Giannozzi, P., Baroni, S., Bonini, N., Calandra, M., Car, R., Cavazzoni, C., Ceresoli, D., Chiarotti, G. L., Cococcioni, M., Dabo, I., et al. Quantum espresso: a modular and open-source software project for quantumsimulations of materials. *Journal of physics: Condensed matter*, 21(39): 395502, 2009.

Gruver, N., Sriram, A., Madotto, A., Wilson, A. G., Zitnick, C. L., and Ulissi, Z. Fine-tuned language models generate stable inorganic materials as text. *arXiv preprint arXiv:2402.04379*, 2024.

Guo, T., Nan, B., Liang, Z., Guo, Z., Chawla, N., Wiest, O., Zhang, X., et al. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems*, 36: 59662–59688, 2023.

Hafner, J. Ab-initio simulations of materials using vasp: Density-functional theory and beyond. *Journal of computational chemistry*, 29(13):2044–2078, 2008.

Hira, K., Zaki, M., Sheth, D., Krishnan, N. A., et al. Reconstructing the materials tetrahedron: challenges in materials information extraction. *Digital Discovery*, 3(5): 1021–1037, 2024.

Hohenberg, P. and Kohn, W. Inhomogeneous electron gas. *Physical review*, 136(3B):B864, 1964.

Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S. K. S., Lin, Z., et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The twelfth international conference on learning representations*, 2023.

Huang, X., Chen, J., Fei, Y., Li, Z., Schwaller, P., and Ceder, G. Cascade: Cumulative agentic skill creation through

autonomous development and evolution. *arXiv preprint arXiv:2512.23880*, 2025.

Jain, A., Ong, S. P., Hautier, G., Chen, W., Richards, W. D., Dacek, S., Cholia, S., Gunter, D., Skinner, D., Ceder, G., et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1), 2013.

Jia, S., Zhang, C., and Fung, V. Llmatdesign: Autonomous materials discovery with large language models. *arXiv preprint arXiv:2406.13163*, 2024.

Kohn, W. and Sham, L. J. Self-consistent equations including exchange and correlation effects. *Physical review*, 140(4A):A1133, 1965.

Kumbhar, S., Mishra, V., Coutinho, K., Handa, D., Iquebal, A., and Baral, C. Hypothesis generation for materials discovery and design using goal-driven and constraint-guided llm agents. *arXiv preprint arXiv:2501.13299*, 2025.

Larsen, A. H., Mortensen, J. J., Blomqvist, J., Castelli, I. E., Christensen, R., Dułak, M., Friis, J., Groves, M. N., Hammer, B., Hargus, C., et al. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, 2017.

Liu, J., Zhu, T., Ye, C., Fang, Z., Weng, H., and Wu, Q. Vaspilot: Mcp-facilitated multi-agent intelligence for autonomous vasp simulations. *Chinese Physics B*, 34(11): 117106, 2025.

Liu, S., Wen, T., Pattamatta, A. S., and Srolovitz, D. J. A prompt-engineered large language model, deep learning workflow for materials classification. *Materials Today*, 80:240–249, 2024.

M. Bran, A., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6 (5):525–535, 2024.

Mandal, I., Soni, J., Zaki, M., Smedskjaer, M. M., Wondraczek, K., Wondraczek, L., Gosvami, N. N., and Krishnan, N. A. Evaluating large language model agents for automation of atomic force microscopy. *Nature Communications*, 16(1):9104, 2025.

Mathew, K., Montoya, J. H., Faghaninia, A., Dwarakanath, S., Aykol, M., Tang, H., Chu, I.-h., Smidt, T., Bocklund, B., Horton, M., et al. Atomate: A high-level interface to generate, execute, and analyze computational materials science workflows. *Computational Materials Science*, 139:140–152, 2017.

McNaughton, A. D., Sankar Ramalaxmi, G. K., Kruel, A., Knutson, C. R., Varikoti, R. A., and Kumar, N. Cactus: Chemistry agent connecting tool usage to science. *ACS omega*, 9(46):46563–46573, 2024.

Merchant, A., Batzner, S., Schoenholz, S. S., Aykol, M., Cheon, G., and Cubuk, E. D. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023.

Miret, S. and Krishnan, N. M. Are llms ready for real-world materials discovery? *arXiv preprint arXiv:2402.05200*, 2024.

Mirza, A., Alampara, N., Kunchapu, S., Ríos-García, M., Emoekabu, B., Krishnan, A., Gupta, T., Schilling-Wilhelmi, M., Okereke, M., Aneesh, A., et al. Are large language models superhuman chemists? *arXiv preprint arXiv:2404.01475*, 2024.

Olowe, E. A. and Chitnis, D. Labiium: Ai-enhanced zero-configuration measurement automation system. In *2025 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6. IEEE, 2025.

Ong, S. P., Richards, W. D., Jain, A., Hautier, G., Kocher, M., Cholia, S., Gunter, D., Chevrier, V. L., Persson, K. A., and Ceder, G. Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science*, 68:314–319, 2013.

Pizzi, G., Cepellotti, A., Sabatini, R., Marzari, N., and Kozinsky, B. Aiida: automated interactive infrastructure and database for computational science. *Computational Materials Science*, 111:218–230, 2016.

Qi, J., Jia, Z., Liu, M., Zhan, W., Zhang, J., Wen, X., Gan, J., Chen, J., Liu, Q., Ma, M. D., et al. Metascientist: A human-ai synergistic framework for automated mechanical metamaterial design. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, pp. 404–436, 2025.

Schilling-Wilhelmi, M., Ríos-García, M., Shabih, S., Gil, M. V., Miret, S., Koch, C. T., Márquez, J. A., and Jablonka, K. M. From text to insight: large language models for chemical data extraction. *Chemical Society Reviews*, 2025.

Shetty, P., Rajan, A. C., Kuenneth, C., Gupta, S., Panchumarti, L. P., Holm, L., Zhang, C., and Ramprasad, R. A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing. *npj Computational Materials*, 9(1):52, 2023.

Song, Y., Miret, S., and Liu, B. Matsci-nlp: Evaluating scientific language models on materials science language tasks using text-to-schema modeling. *arXiv preprint arXiv:2305.08264*, 2023.

Song, Z., Lu, S., Ju, M., Zhou, Q., and Wang, J. Accurate prediction of synthesizability and precursors of 3d crystal structures via large language models. *Nature Communications*, 16(1):6530, 2025.

Szymanski, N. J., Rendy, B., Fei, Y., Kumar, R. E., He, T., Milsted, D., McDermott, M. J., Gallant, M., Cubuk, E. D., Merchant, A., et al. An autonomous laboratory for the accelerated synthesis of novel materials. *Nature*, 624 (7990):86–91, 2023.

Tang, X., Hu, T., Ye, M., Shao, Y., Yin, X., Ouyang, S., Zhou, W., Lu, P., Zhang, Z., Zhao, Y., et al. Chemagent: Self-updating library in large language models improves chemical reasoning. *arXiv preprint arXiv:2501.06590*, 2025.

Wang, H., Fu, T., Du, Y., Gao, W., Huang, K., Liu, Z., Chandak, P., Liu, S., Van Katwyk, P., Deac, A., et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

Wang, H., Skreta, M., Ser, C.-T., Gao, W., Kong, L., Strieth-Kalthoff, F., Duan, C., Zhuang, Y., Yu, Y., Zhu, Y., et al. Efficient evolutionary search over chemical space with large language models. *arXiv preprint arXiv:2406.16976*, 2024.

Wang, Z., Huang, H., Zhao, H., Xu, C., Zhu, S., Janssen, J., and Viswanathan, V. Dreams: Density functional theory based research engine for agentic materials simulation. *arXiv preprint arXiv:2507.14267*, 2025.

Xia, Z., Ma, J., Zheng, C., Zhang, S., Li, Y., Su, H., Hu, P., Zhang, C., Gong, X., Ouyang, W., et al. An agentic framework for autonomous materials computation. *arXiv preprint arXiv:2512.19458*, 2025.

Yao, L., Samantray, S., Ghosh, A., Roccapriore, K., Kovarik, L., Allec, S., and Ziatdinov, M. Operationalizing serendipity: Multi-agent ai workflows for enhanced materials characterization with theory-in-the-loop. *arXiv preprint arXiv:2508.06569*, 2025.

Yao, S., Shinn, N., Razavi, P., and Narasimhan, K. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Ye, Y., Ren, J., Wang, S., Wan, Y., Razzak, I., Hoex, B., Wang, H., Xie, T., and Zhang, W. Construction and application of materials knowledge graph in multidisciplinary materials science via large language model. *Advances in Neural Information Processing Systems*, 37:56878–56897, 2024.

Zaki, M., Krishnan, N. A., et al. Mascqa: investigating materials science knowledge of large language models. *Digital Discovery*, 3(2):313–327, 2024.

Zhang, H., Song, Y., Hou, Z., Miret, S., and Liu, B. Honeycomb: A flexible llm-based agent system for materials science. *arXiv preprint arXiv:2409.00135*, 2024.

Zhang, J., Gan, J., Wang, X., Jia, Z., Gu, C., Chen, J., Zhu, Y., Ma, M. D., Zhou, D., Li, L., et al. Matscibench: Benchmarking the reasoning ability of large language models in materials science. *arXiv preprint arXiv:2510.12171*, 2025.

Zhong, X., Jin, B., Ouyang, S., Shen, Y., Jin, Q., Fang, Y., Lu, Z., and Han, J. Benchmarking retrieval-augmented generation for chemistry. *arXiv preprint arXiv:2505.07671*, 2025.