

ChemGraph-XANES: An Agentic Framework for XANES Simulation and Curation

Vitor F. Grizzi¹, Thang Duc Pham², Luke N. Pretzie¹, Jiayi Xu¹, Murat Keceli², and Cong Liu^{1*}

¹Chemical Sciences and Engineering Division
Argonne National Laboratory, Lemont, IL 60439, USA

²Computational Science Division
Argonne National Laboratory, Lemont, IL 60439, USA

August 24, 2026

Abstract

Computational X-ray absorption near-edge structure (XANES) is widely used to interpret local coordination environments, oxidation states, and electronic structure, but large computational campaigns are often limited by workflow complexity. We present ChemGraph-XANES, a large language model (LLM)-based agentic framework that combines documentation-grounded parameter retrieval via retrieval-augmented generation (RAG), schema-constrained tool execution, deterministic FDMNES input generation, Parsl-backed execution, and provenance-aware data curation. Scripted and natural-language interfaces share a common scientific backend for structure handling, parameterization, execution, spectral extraction, and optional post-processing. We evaluate three workflow modes: documentation-grounded parameter propagation, structure-file-based execution, and composition-based execution from a chemistry-level request. Repeated trials yielded end-to-end completion in 10/10 composition-based runs, 10/10 structure-file-based runs, and 9/10 documentation-grounded RAG runs. In every RAG run, the energy-grid specification retrieved from the FDMNES manual was correctly propagated, with the single end-to-end failure occurring downstream during multi-structure handling. In a separate task-parallel demonstration, the framework retrieved 21 TiO₂ structures from the Materials Project and submitted one FDMNES calculation per structure. All calculations completed successfully, with Parsl distributing the independent tasks across the user-configured worker pool. Together, these results show that ChemGraph-XANES provides a constrained and reproducible orchestration layer for computational spectroscopy, supporting consistent execution of representative tasks, documentation-linked parameter selection, and task-parallel generation of structure-linked XANES collections.

1 Introduction

Computational X-ray absorption near-edge structure (XANES) has become an important complement to experimental X-ray absorption spectroscopy for interpreting local coordination environments, oxidation states, and electronic structure in chemically complex systems [1, 2, 3, 4, 5]. In practice, however, the primary obstacle to routine, large-scale XANES calculations often lies not only in the electronic-structure method or scattering formalism, but also in the workflow required to

*Corresponding author. Email: congliu@anl.gov

execute each calculation in a reproducible manner. A typical XANES campaign may require retrieving or preparing candidate structures, identifying absorber sites, specifying calculation parameters, generating code-specific inputs, organizing many independent runs, parsing spectral outputs, applying consistent normalization, and preserving the provenance linking each spectrum to its parent structure. When these steps are handled manually or through ad hoc scripts, the workflow becomes difficult to reproduce, reuse, and scale.

This workflow bottleneck becomes especially important in settings that require many calculations rather than a single forward simulation. Such settings are common across computational chemistry tasks, including high-throughput screening of structural databases, comparison of DFT-optimized candidate geometries, sampling of thermally accessible configurations from molecular dynamics or *ab initio* molecular dynamics trajectories, iterative refinement against experiments, materials discovery, and dataset generation for machine-learning (ML) models [6, 7, 8, 9, 10, 11, 12, 13, 14]. In computational spectroscopy, these same pressures appear when large collections of spectra must be generated and curated consistently for screening, ensemble analysis, experiment-theory comparison, or ML applications [15, 16]. In such settings, the practical value of computational XANES depends not only on the fidelity of the underlying physics-based method, but also on whether structure preparation, parameter specification, execution, post-processing, and data organization can be carried out consistently across large batches of jobs. Because individual XANES calculations are naturally independent once their inputs are defined, these workloads are also well suited to task-parallel execution on high-performance computing (HPC) systems.

Recent progress in large language models (LLMs) and LLM-based agents has created new opportunities for scientific workflow automation and for making scientific workflows more machine-actionable, accessible, and reusable [17, 18, 19, 20, 21, 22, 23, 24, 25]. By exposing established domain software as callable tools, high-level natural-language requests can be translated into structured computational actions, reducing the need for users to manually compose each step of the simulation pipeline [26, 27, 28, 29]. In computational spectroscopy, this paradigm enables an agent to select the appropriate operation, populate its parameters through typed interfaces, and orchestrate multi-step workflows spanning structure retrieval, simulation setup, execution, and post-processing. Rather than replacing the underlying spectroscopy engine, the agent functions as an orchestration layer over conventional scientific software, improving accessibility while preserving transparent execution logic [30].

In this work, we develop **ChemGraph-XANES** as a domain-specific extension of the original ChemGraph framework [27] for computational XANES simulation and curation. Built on ASE, FDMNES, Parsl, and a LangGraph/LangChain-based tool interface [31, 32, 33, 34], the framework exposes XANES workflow operations as typed Python tools, enabling language-model agents to interpret user requests, map relevant quantities onto structured function arguments, and execute the resulting spectroscopy workflow. The central contribution is a documentation-grounded, schema-constrained, provenance-aware orchestration layer that connects structure acquisition, absorbing-species specification, FDMNES input generation, execution, spectral parsing, normalization helpers, and structure-spectrum linkage within a single Python framework. The workflow can be invoked either directly from scripts or through natural-language requests mapped onto typed tool calls, thereby preserving a common deterministic scientific backend across multiple user-facing interaction modes. We demonstrate documentation-grounded parameter retrieval and propagation, support for both local structure-file inputs and chemistry-level requests, task-specific consistency across ten separate executions of each of three representative workflows, and a 21-structure task-parallel TiO₂ calculation on an HPC system. The present work combines proof-of-capability demonstrations with a limited repeated-execution evaluation and one task-parallel collection. However, it is not a comprehensive benchmark of spectral fidelity, retrieval performance, general workflow reliability, or

distributed-execution scaling. Taken together, these capabilities establish **ChemGraph-XANES** as a machine-actionable workflow layer for reproducible and extensible XANES simulation and data curation.

2 Methods

2.1 Workflow overview and agentic orchestration

We developed ChemGraph-XANES as a modular, agent-compatible workflow for automated XANES calculations. The framework organizes natural-language task specification, structure acquisition, FDMNES input preparation, task-parallel execution, spectral post-processing, and provenance tracking as callable Python stages. The underlying scientific operations are implemented as Python functions and exposed to the execution layer as typed tools with schema-defined arguments. In agentic mode, a large language model (LLM) interprets a user query, selects the relevant tool, and maps user-specified quantities, such as the chemical system, absorbing species, cluster radius, electronic configuration settings, or output location, onto structured function arguments. These arguments are then passed to the same underlying Python routines used in direct scripted execution, thereby preserving a single source of scientific logic while enabling the workflow to be driven from natural-language requests. In the current implementation, execution, database expansion, and normalization/plotting remain separate callable stages, so an agent trace should not be interpreted as automatically completing every downstream curation step unless those tools are explicitly invoked.

The current implementation supports both single-agent and multi-agent execution modes. In the single-agent setting, the model iteratively alternates between reasoning and tool invocation until the requested workflow is completed. In the multi-agent setting, a planner first decomposes the user request into subtasks, worker agents execute those subtasks through the same tool abstraction, and an aggregator combines the resulting outputs into a final response. Further details of the underlying ChemGraph framework and single- or multi-agent settings are available in Ref. [27].

The multi-agent mode can additionally incorporate a retrieval-augmented expert agent that grounds XANES parameter selection in external documentation, such as a text-converted software manual, allowing executor agents to receive documentation-informed guidance without hard-coding all decision rules directly into the executor prompt or the simulation backend. This design enables the same XANES workflow to be invoked from interactive natural-language sessions, notebooks, or conventional Python scripts without changing the underlying spectroscopy routines. At a high level, the pipeline begins with either user-provided structures or structures retrieved from a materials database, converts them into a unified atomistic representation, generates FDMNES input files, executes the calculations, parses the resulting convolution spectra, and stores the spectra back alongside the originating structures for downstream analysis. Figure 1 provides a schematic overview of the ChemGraph-XANES workflow.

It is useful to distinguish three levels of capability provided by the current implementation. First, at the deterministic workflow level, the framework supports conventional scripted execution of structure handling, FDMNES input generation, execution, parsing, and optional post-processing without requiring any language-model interaction. Second, at the orchestration level, a natural-language request can be translated into schema-validated tool calls that populate the same underlying scientific routines. Third, at the documentation-grounded assistance level, external reference material can be retrieved and used to inform parameter selection before execution. This layered design is intended to separate the benefits of language-based interaction from the underlying scientific logic, which remains implemented in explicit Python functions and constrained tool schemas rather than in free-form model output. Accordingly, ChemGraph-XANES should not be interpreted as

granting unrestricted autonomy to a language model over arbitrary FDMNES inputs. Instead, the agent operates within constrained interfaces that specify which parameters can be set, how they are validated, and how they are passed to the backend. This distinction is important for reproducibility and trustworthiness, particularly in scientific settings where undocumented parameter changes can compromise later interpretation or reuse.

2.2 Structure acquisition and curation

For database-driven workflows, bulk structures are retrieved programmatically from the Materials Project through its Python API [35, 36]. The current implementation queries the materials summary endpoint using a user-specified chemical formula together with an energy-above-hull threshold in order to restrict the search to low-energy candidate structures. The retrieval tool returns all structures satisfying the filter and does not itself reduce a formula-level query to a single structure. When a request requires one structure, selection from the returned candidates occurs during orchestration unless the user supplies an explicit structure identifier or selection rule; requests that explicitly ask for all qualifying structures retain the complete returned list. Retrieved structures are converted from the native Pymatgen representation into `ase.Atoms` objects, which are used as the internal structural representation throughout the workflow. Each structure is annotated with its Materials Project identifier, thereby preserving traceability to the original source and simplifying later aggregation of results.

To support both interoperability and local reuse, structures retrieved from the Materials Project are stored in two complementary formats. Each structure is exported as an individual `.cif` file, preserving compatibility with standard crystallographic tools, and the full retrieved collection is serialized as a pickled list of `ase.Atoms` objects in `atoms_db.pkl`. In the `create_fdmnes_inputs` helper, this serialized collection is used as the fallback source for FDMNES input generation when an explicit in-memory structure list is not supplied. Other execution paths can also operate directly on supported structure files, directories, or ASE databases.

2.3 Automatic FDMNES input generation

Automatic FDMNES input generation in ChemGraph-XANES combines deterministic script-based file construction with an agentic parameter-selection layer. Within the broader ChemGraph architecture, a retrieval-augmented agent can ingest text or PDF-derived documentation, index its contents into a queryable knowledge base, and retrieve grounded guidance when an FDMNES calculation must be parameterized. The knowledge base is constructed locally at runtime from user-supplied documentation rather than stored as a fixed internal resource; users can reconstruct and query it using the shared code and the corresponding source document. In the demonstration trace reported below, the RAG agent graph exposes both documentation-retrieval tools and XANES execution tools, allowing a retrieved `Range` value to be passed into a schema-validated XANES tool call. This should be interpreted as a configured tool-level coupling demonstration rather than as a fully general expert/executor controller that automatically runs all downstream normalization and data-curation helpers. Agent decisions are validated against structured schemas, which translate high-level reasoning into the standardized parameters required by the backend.

Once the calculation arguments are fixed, FDMNES input files are generated directly from the `ase.Atoms` objects. For each structure, the workflow writes both the steering file `fdmfile.txt` and the main input file `fdmnes_in.txt`. The absorber can be specified either by atomic species or by site index. If the user provides `absorber_idx`, the workflow writes the FDMNES `Absorber` keyword with that 1-based atom index, enabling a site-specific calculation for the selected atom in

ChemGraph-XANES Workflow

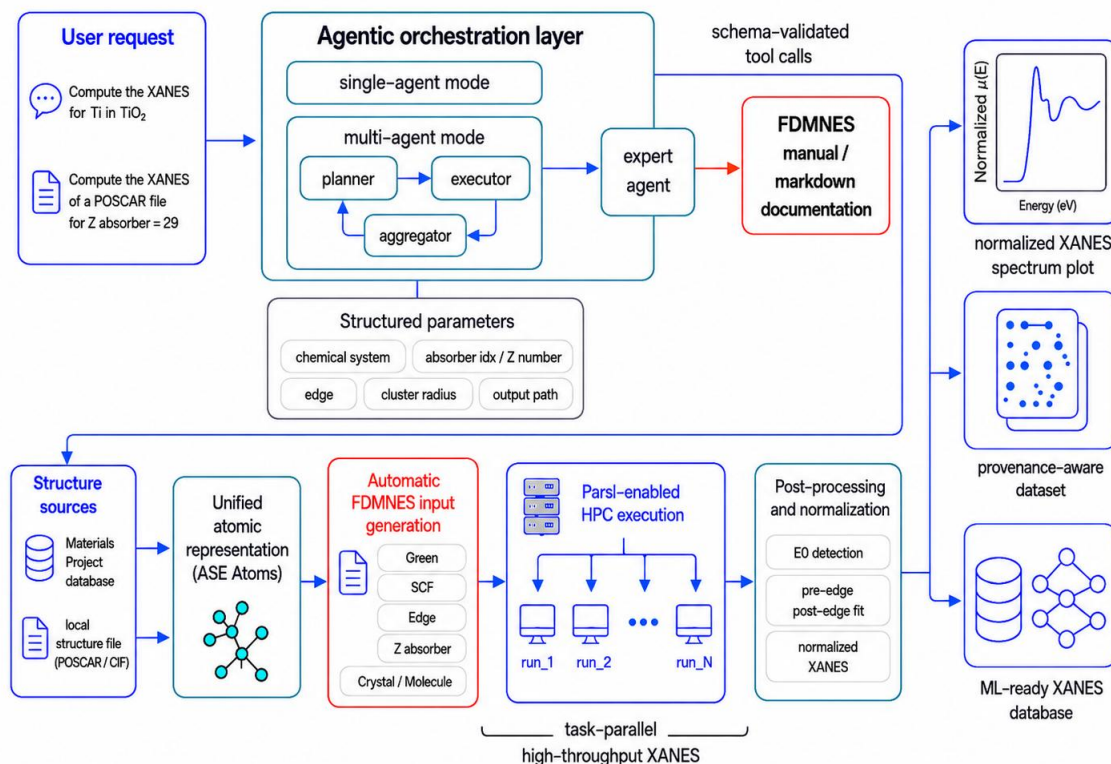


Figure 1: Schematic overview of the ChemGraph-XANES workflow. User requests can be provided either as chemistry-level natural-language queries or as explicit local structure files. These inputs are interpreted by an agentic orchestration layer that supports both single-agent and multi-agent execution, with an optional documentation-grounded expert agent that consults the FDMNES manual to inform parameter selection. The resulting schema-validated tool calls are mapped onto structured arguments, enabling automated structure retrieval, conversion to a unified ASE Atoms representation, FDMNES input generation, and task-parallel execution through Parsl on HPC systems. After successful calculations, separate post-processing helpers can parse spectra, apply normalization when the energy mesh is compatible with the selected fitting windows, and attach spectra back to the corresponding structures for downstream analysis and ML-oriented database generation.

the structure ordering. If `absorber_idx` is not provided but `z_absorber` is specified, the workflow writes `Z_absorber`, which selects all atoms with that atomic number as the absorbing species in the FDMNES sense. If neither argument is supplied, the workflow chooses the element with the largest atomic number present in the structure as a deterministic fallback and writes it through `Z_absorber`. Edge selection is exposed through the `edge` argument, which is written to the generated input with the FDMNES `Edge` keyword and defaults to the K edge when no other edge is specified.

The FDMNES energy mesh can be supplied through the schema-validated `energy_range` argument, either explicitly by the user or through documentation-grounded retrieval from the FDMNES manual. More generally, the exposed FDMNES controls, including `energy_range`, `radius`, `edge`, `magnetism`, `Green`, `Density_all`, `Quadrupole`, `Spherical`, and `SCF`, have implementation-level defaults but can be overridden through schema-validated arguments when specified by the user or selected with documentation-grounded expert-agent guidance.

The current ChemGraph-XANES implementation defaults to a cluster radius of 6.0 Å and a variable energy mesh of [-55, 1, -10, 0.01, 5, 0.1, 150] eV. Calculations default to the K edge, with `Green`, `Density_all`, `Quadrupole`, `Spherical`, and `SCF` enabled and `Magnetism` disabled. These values are ChemGraph-XANES implementation defaults rather than defaults supplied by FDMNES. Unless otherwise specified, they were used for the single-structure calculations reported here. Spectral convolution was requested automatically without overriding the broadening parameters supplied internally by the installed FDMNES version. The documentation-grounded RAG calculations used the manual-derived FDMNES `Range` setting, corresponding to an energy mesh of [-5.0, 0.5, 60.0] eV, instead of the ChemGraph-XANES default mesh. The task-parallel TiO₂ batch used a 5.0 Å cluster radius and an energy mesh of [-20.0, 1.0, -10.0, 0.05, 10.0, 0.1, 50.0] eV. For this batch, `Green`, `Quadrupole`, and `SCF` were enabled, whereas `Density_all`, `Spherical`, and `Magnetism` were disabled.

The workflow distinguishes automatically between periodic and non-periodic structures. If all periodic boundary condition flags are enabled, the structure is written in `Crystal` mode using the cell parameters and fractional atomic coordinates. Otherwise, the structure is written in `Molecule` mode using Cartesian coordinates and a cubic bounding cell constructed from the atomic positions. This branching enables a common front-end representation to be used for both crystalline and molecular systems while still emitting valid FDMNES inputs for each case.

More broadly, the parameter-exposure strategy in ChemGraph-XANES is deliberately conservative. Only a defined subset of FDMNES controls is surfaced through schema-validated interfaces, with implementation-level defaults used when explicit user input or documentation-grounded guidance is not provided. This restriction is intentional: it preserves flexibility for common XANES setup choices while reducing the risk that unconstrained language-model output could introduce unsupported or poorly justified simulation settings. In this sense, the framework uses the agent as a controlled orchestration layer over deterministic input-generation logic, rather than as an unrestricted author of full simulation files.

2.4 Task-parallel execution, Parsl compatibility, and provenance tracking

To support batch studies, the pipeline includes a batch input generator that creates a dedicated run directory for each structure under a common `fdmnes_batch_runs` root. Each directory contains the corresponding FDMNES input files together with a serialized copy of the associated `ase.Atoms` object. The filename of the serialized structure includes the absorbing-species atomic number, the Materials Project identifier when available, and the chemical formula. This naming convention provides a lightweight traceability layer that links each calculation directory to its structural origin.

Single-run execution is handled by a core routine that reads an input structure through ASE,

creates the output directory, writes the FDMNES inputs, and launches the external FDMNES executable through a subprocess call. The path to the executable is provided through the `FDMNES_EXE` environment variable. Standard output and standard error streams are redirected to `fdmnes_stdout.txt` and `fdmnes_stderr.txt`, respectively, which facilitates debugging and post hoc inspection of failed jobs. After execution, the workflow searches the run directory for FDMNES convolution output files matching `*conv.txt`. All matching spectra are parsed and stored as NumPy arrays. A scientifically successful run requires at least one convolution output file; ensemble-level summaries should therefore check this file count explicitly rather than treating completion of a Parsl task alone as evidence of a completed XANES calculation.

Because each structure gives rise to an independent FDMNES calculation once the input parameters are specified, the workflow is naturally task-parallel across structures. This property makes it compatible with distributed execution through `Parsl`, where independent structure-level jobs can be dispatched on HPC systems while retaining a uniform interface for inputs, logs, and outputs. Because the focus of the present work is workflow construction and representative validation, we evaluate execution and curation under one task-parallel configuration rather than undertaking a dedicated performance-scaling study. Quantitative throughput, speedup, scaling efficiency, and operational failure rates across broader deployments remain targets for future work.

To assemble the results into a dataset suitable for downstream analysis, a separate helper reloads the serialized structures, attaches the extracted spectral data to the corresponding `ase.Atoms` objects under the `atoms.info["FDMNES-xanes"]` field, and writes the expanded collection to `atoms_db_expanded.pkl`. This optional post-processing step preserves the association between structural inputs and spectral outputs in a form that can be reused for subsequent analysis, data curation, and ML-oriented dataset construction.

2.5 Evaluation protocols

To evaluate task-specific repeated-execution consistency, we performed ten separate executions of each of three representative workflows using Gemini 2.5 Flash with a temperature of 0 and a maximum output of 6000 tokens. The evaluated tasks were (i) composition-based execution using the original prompt “*Compute the XANES for Ti in TiO₂.*”, (ii) structure-file-based execution from the supplied POSCAR with $Z_{\text{absorber}} = 29$, and (iii) a documentation-grounded RAG workflow that retrieved the default FDMNES `Range` values from the manual and used them for TiO₂ and Fe₂O₃ calculations. The RAG executions used the document-indexing, embedding, and retrieval configuration described in the documentation-grounded parameter-selection subsection. A trial was counted as an end-to-end success only when every calculation requested by the prompt produced at least one parseable convolution spectrum and manual inspection confirmed that the generated FDMNES inputs contained the requested parameters. Retrieval and propagation of the manual-derived `Range` setting were also assessed separately from end-to-end completion. This protocol measures observed consistency for the three fixed tasks under the reported configuration rather than general workflow reliability.

The task-parallel evaluation used Gemini 2.5 Flash in the multi-agent XANES configuration. The prompt template is reproduced below; at execution time, `{RUNS_DIR}` was replaced by the absolute output-directory path.

```
Fetch all non-deprecated TiO2 structures from Materials Project with
energy_above_hull <= 0.10 eV/atom. Pass exactly the returned
structure_files list to one run_xanes_ensemble call; do not scan
the output directory or call run_xanes_single.
```

```

Run Ti K-edge calculations with:
output_dir="{RUNS_DIR}", z_absorber=22, edge="K", radius=5.0,
energy_range=[-20.0, 1.0, -10.0, 0.05, 10.0, 0.1, 50.0],
green=true, density_all=false, quadrupole=true, spherical=false,
scf=true, magnetism=false, and skip_completed=false.

```

Do not plot or modify the spectra. Report the retrieved, successful, and failed structure counts and the absolute path to `xanes_results.jsonl`.

The exact structure-file list returned by the Materials Project retrieval tool was supplied to one `run_xanes_ensemble` call. For this experiment, Parsl was configured with a maximum of ten one-node worker blocks, with one worker assigned to each 128-core compute node and one FDMNES calculation executed per worker. This configuration permitted up to ten structure-level calculations to run concurrently.

2.6 Spectral normalization

Raw FDMNES convolution outputs are further processed through a normalization routine designed to place spectra on a common scale. Let E denote the energy grid and $\mu(E)$ the raw absorption signal read from a convolution file. The workflow first assigns the edge reference energy as $E_0 = 0$ when the computed energy grid spans zero, consistent with the FDMNES convention that energies are reported relative to the edge.

Two linear baselines, $\mu_{\text{pre}}(E)$ and $\mu_{\text{post}}(E)$, are fitted to the pre-edge and post-edge regions, respectively. With the default settings, the pre-edge region contains points satisfying $E \leq E_0 - 20$ eV, whereas the post-edge region contains points satisfying $E \geq E_0 + 50$ eV. The pre-edge baseline is subtracted from the spectrum, and the result is normalized by the difference between the two fitted baselines at E_0 :

$$\mu_{\text{norm}}(E) = \frac{\mu(E) - \mu_{\text{pre}}(E)}{\mu_{\text{post}}(E_0) - \mu_{\text{pre}}(E_0)}.$$

Both fitting regions must contain sufficient points for a linear fit. If the supplied energy mesh does not cover either region or a fit fails, the workflow instead applies maximum-intensity normalization:

$$\mu_{\text{norm}}(E) = \frac{\mu(E)}{\max_E \mu(E)}.$$

When the linear fits are valid, the baseline-subtracted edge-step procedure yields a normalized absorption curve that is less sensitive to baseline offsets and differences in absolute intensity scale. The maximum-intensity fallback places spectra on a common intensity scale but does not perform pre-edge subtraction or edge-step normalization. Because normalization is not scientifically neutral, ChemGraph-XANES preserves the raw convolution output alongside the normalized spectrum so that later analyses can revisit the fitting windows, edge reference, or normalization convention. Standardized normalization is useful for comparative studies and ML-oriented datasets, but it should be treated as a validated post-processing step rather than as an automatic consequence of every FDMNES run. For convenience, a plotting routine iterates over completed run directories, applies the normalization procedure, and writes an `xanes_plot.png` file for each successfully parsed calculation.

3 Results

3.1 Documentation-grounded parameter selection and propagation into execution

A key feature of ChemGraph-XANES is a documentation-grounded expert agent that consults external reference material before downstream tool invocation. Rather than relying exclusively on latent model knowledge, the expert agent retrieves evidence from a local knowledge base constructed from the FDMNES manual and uses that evidence to guide parameter selection. This functionality is especially important for XANES workflows because input parameters often depend on code-specific keywords, defaults, and conventions that are easy to misstate in free-form language.

To illustrate this behavior, we evaluated the expert agent on three representative parameterization queries concerning default absorbing-species selection, substitutional doping, and default energy-range settings. The qualitative retrieval examples in Table 1 were generated with GPT-4o, while the executable provenance trace in Table 2 was generated with Gemini 2.5 Flash. The use of two providers was intended to demonstrate model portability rather than comparative performance. Document retrieval used the Hugging Face sentence-transformers/all-MiniLM-L6-v2 embedding model on CPU and a FAISS index, with a chunk size of 1,000 characters and an overlap of 200 characters.

In all three cases, the relevant FDMNES guidance was obtained through retrieval rather than supplied directly in the prompt. For readability, Table 1 presents condensed traces containing the user query, the explicit retrieval call, the salient evidence extracted from the manual, and a shortened excerpt from the final agent response. The raw interaction logs for the qualitative examples are available in the accompanying repository notebook, and the raw Gemini trace for the range-propagation example is provided separately.¹

Across all three cases, the agent first invoked the `query_knowledge_base` tool with a targeted search phrase and then produced a response consistent with the retrieved manual content. The first example is especially instructive because it highlights both the usefulness and the limitations of agent-mediated summarization: the final response captures the practical default behavior and correctly points the user to the `Z_absorber` and `Absorber` keywords, while the retrieved manual evidence provides the more precise FDMNES convention that the calculated spectra correspond to atoms sharing the atomic number of the first species listed under `Crystal` or `Molecule`. The second and third examples show that the same retrieval mechanism can recover both procedural instructions, such as the use of the `Doping` keyword, and default numerical settings, such as the standard energy range and step size. Although these examples are qualitative demonstrations rather than a formal retrieval benchmark, they show that the expert-agent layer can consult external documentation before answering parameterization questions, improving transparency and reducing reliance on unsupported free-form reasoning.

To further test whether retrieved documentation was propagated into an executable simulation input, we ran a documentation-grounded XANES calculation in which the agent was asked to use the FDMNES manual to set the `Range` keyword before launching calculations for TiO_2 and Fe_2O_3 . The recorded Gemini 2.5 Flash trace shows that the agent first loaded `FDMNES_manual.txt` using HuggingFace embeddings, queried the knowledge base for the default `Range` values, retrieved the manual passage specifying an energy range from -5 to 60 eV with a 0.5 eV step, and then passed this value to `run_xanes_single` as `energy_range=[-5.0, 0.5, 60.0]`. This trace provides a direct link between retrieved documentation and the numerical parameter used in the execu-

¹https://github.com/vitorgrizzi/ChemGraph_xanes/blob/main/notebooks/Demo_rag_agent_Argo.ipynb;
https://github.com/vitorgrizzi/ChemGraph_xanes/blob/main/notebooks/rag_range_fill.txt

Table 1: Representative documentation-grounded expert-agent retrieval examples. For readability, the retrieved evidence and final answer fields are condensed from the raw execution traces. Full interaction is available at https://github.com/vitorgrizzi/ChemGraph_xanes/blob/main/notebooks/Demo_rag_agent_Argo.ipynb.

Example 1	
User query	How does the program determine the absorbing species by default?
Tool call	<code>query_knowledge_base(query="default absorbing species determination", file_path=FDMNES_manual.txt)</code>
Retrieved evidence	By default, the calculated spectra correspond to the sum of the scattering or absorption produced by all atoms with the same atomic number as the first one listed under Crystal or Molecule . Explicit overrides are available through Z_absorber or Absorber .
Agent answer (excerpt)	“By default, the calculated spectra correspond to atoms with the same atomic number as the first species listed under Crystal or Molecule . If a different absorbing species is intended, the user can explicitly define the absorbing-species atomic number using Z_absorber .”
Example 2	
User query	How can a user model substitutional doping in a crystal structure?
Tool call	<code>query_knowledge_base(query="model substitutional doping in a crystal structure", file_path=FDMNES_manual.txt)</code>
Retrieved evidence	The manual recommends the Doping keyword followed by the dopant atomic number and the index of the substituted site, for example Doping 27 1 . It also states that the dopant is treated as a low-concentration substitution and that crystal symmetries are preserved.
Agent answer (excerpt)	“To model substitutional doping in a crystal structure using FDMNES, you can use the Doping keyword. This allows you to substitute a dopant species in place of another atom within the crystal structure.”
Example 3	
User query	What are the default values for the energy range and step?
Tool call	<code>query_knowledge_base(query="default values for the energy range and step", file_path=FDMNES_manual.txt)</code>
Retrieved evidence	The retrieved manual passage states that the default energy range is -5 to 60 eV with a step size of 0.5 eV. It also notes that the Range keyword can be used to modify these values or define variable steps.
Agent answer (excerpt)	“The default values for the energy range and step in the FDMNES software are as follows: Energy Range: -5 to 60 eV. Step: 0.5 eV. These values can be modified by specifying the Range keyword in the input file.”

tion call. Importantly, the default numerical values for the **Range** keyword were not included in the user query, system prompt, or tool schema; they appeared only in the retrieved manual passage and in the subsequent execution argument. This example is therefore used as a parameter-propagation trace; routine ChemGraph executions can instead use the built-in ChemGraph mesh when a normalization-ready energy grid is desired.

These retrieval demonstrations should be interpreted as proof-of-capability examples rather than as a comprehensive benchmark of documentation-question answering accuracy. The repeated-

execution study reported below evaluates consistency for one documentation query and one retrieved parameter, but it does not measure retrieval recall, answer accuracy across a broad parameter set, or robustness to ambiguous software-manual passages. Such evaluations would be valuable if documentation-grounded parameter assistance becomes a larger part of autonomous spectroscopy workflows.

Table 2: Documentation-grounded propagation of the FDMNES **Range** setting into the XANES execution call. Full trace is available at https://github.com/vitorgrizzi/ChemGraph_xanes/blob/main/notebooks/rag_range_fill.txt.

Step	Recorded trace evidence
Document loading	<code>load_document(file_path=FDMNES_manual.txt, embedding_provider=huggingface)</code>
Retrieval query	<code>query_knowledge_base(query="default values of the FDMNES Range keyword")</code>
Retrieved manual evidence	Default energy range from -5 to 60 eV with a 0.5 eV step.
Execution argument	<code>run_xanes_single(..., energy_range=[-5.0, 0.5, 60.0], ...)</code>

3.2 Structure-file-based execution

We first evaluated the file-based execution mode of the framework using the natural-language query: *“Compute the XANES of the POSCAR file located at /lcrc/globalscratch/vferreiragrizzi/agents/poscar_run/POSCAR for Z absorber = 29.”* The agent must recognize that the query refers to a user-supplied atomistic input, extract both the file path and the absorber atomic number from the text, and map these quantities onto the structured arguments required by the XANES execution tool. In contrast to the documentation-grounded retrieval cases discussed above, this example probes the agent’s ability to interpret an execution-oriented request and route a local structure file into the simulation workflow.

In this run, the POSCAR file was parsed into an `ase.Atoms` object, the corresponding FDMNES steering and input files were generated, the FDMNES calculation was launched, and the resulting convolution output was parsed, normalized, and plotted by the workflow. Figure 2 shows the resulting normalized spectrum. The calculation was performed for the Cu K edge by setting the absorbing species to Cu ($Z = 29$) on a MnO_2 (010) slab obtained by cleaving MnO_2 along the (010) facet and passivating undercoordinated surface sites with H atoms. The inset in Fig. 2 shows the corresponding structure, with the Cu species highlighted in blue. No matching experimental spectrum or independent reference calculation was available for the specific MnO_2 -based structure. This example is therefore presented as a demonstration of structure-file-based workflow execution, which does not require a composition-based search or database lookup, rather than as a validated spectral benchmark. This capability is important in practice because many computational spectroscopy workflows begin from local structures generated in prior DFT, molecular dynamics, slab-construction, or structural-relaxation steps rather than from canonical bulk entries in a materials database.

3.3 Natural-language specification of a chemical system

We next examined the natural-language interface using the query: *“Compute the XANES for Ti in TiO_2 .”* In contrast to the file-based example discussed above, this request does not provide an

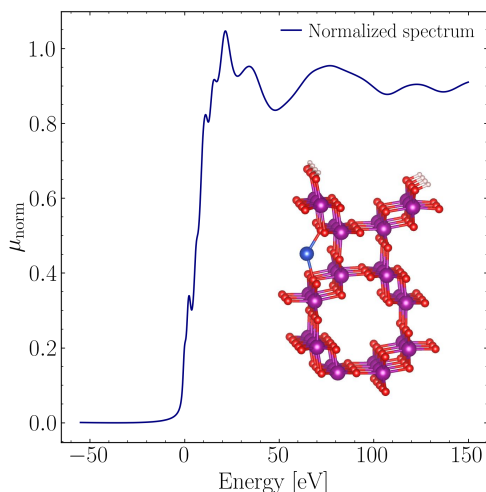


Figure 2: Normalized Cu K-edge XANES spectrum computed by setting the absorbing species to Cu on a MnO_2 (010) slab. The slab structure was constructed by cleaving MnO_2 along the (010) facet, and H atoms were added to passivate undercoordinated surface atoms. In the inset structure, the Cu species is shown in dark blue.

explicit local structure file. Instead, the agent must infer that the user is specifying a chemical system and absorbing species at a higher semantic level, identify the appropriate workflow for structure retrieval, and map the request onto the structured arguments needed for the downstream XANES tools. In practice, this means that the agent must interpret the chemistry-level request, issue a Materials Project query for TiO_2 , identify Ti as the absorbing element, infer that this corresponds to absorbing-species atomic number $Z = 22$, and pass these structured parameters to the downstream XANES workflow before triggering the same FDMNES input-generation and execution routines used in direct scripted mode. Spectral normalization and dataset expansion can then be applied through the separate post-processing helpers described above.

For the calculation reported here, the Materials Project query, using an energy-above-hull threshold of 0.001 eV/atom, resolved the formula-level request to anatase TiO_2 (Materials Project identifier `mp-390`). The identifier was retained in the calculation metadata to preserve the connection between the resulting spectrum and its source structure. This selection should not be interpreted as reproducing the experimental polymorph stability: rutile is the stable bulk phase under ambient conditions, whereas GGA and uncorrected $r^2\text{SCAN}$ calculations can incorrectly place anatase below rutile in energy [37].

Figure 3 shows the resulting normalized Ti K-edge spectrum for bulk TiO_2 . The inset displays the corresponding crystal structure used in the calculation, with Ti atoms shown in light blue and O atoms in red. To assess the qualitative physical plausibility of the calculated spectrum, we compared it with an experimental Ti K-edge spectrum of anatase TiO_2 measured in transmission at SPring-8 BL14B2 and distributed through the NIMS Materials Data Repository [38]. Because the FDMNES output used a relative energy scale, the calculated axis was shifted rigidly by +4965.52 eV to align the highest-intensity main-edge peaks; no other energy adjustment or spectral fitting was applied. The calculation used the `mp-390` structure, a 6.0 Å cluster radius, self-consistency, and the `Green` multiple-scattering mode, with both dipole and quadrupole transitions included.

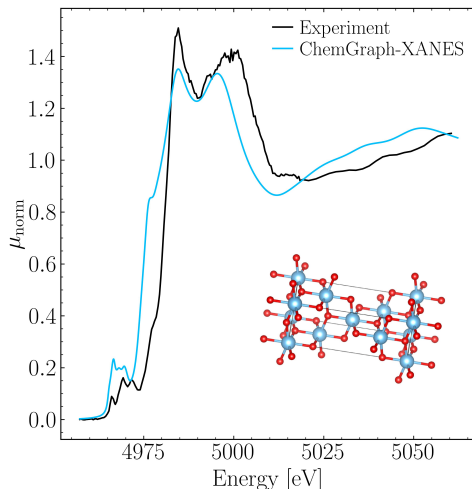


Figure 3: Calculated Ti K-edge XANES spectrum for anatase TiO_2 (mp-390) compared with the experimental anatase standard from the NIMS/SPRING-8 XAFS database [38]. Both spectra were normalized using the procedure described in the text. The calculation used a 6.0 Å cluster radius in the FDMNES **Green** multiple-scattering mode with self-consistency and both dipole and quadrupole transitions enabled. A single rigid shift of +4965.52 eV was applied to align the highest-intensity main-edge peaks; no additional energy alignment or fitting was performed.

The remaining differences may reflect the muffin-tin approximation associated with **Green**, finite-cluster effects, and non-optimized broadening [33]. Full-potential calculations further demonstrate the sensitivity of the anatase pre-edge to p - d hybridization and experimental geometry [39].

Together, the file-based and composition-based examples show that different structure-specification modes can use the same scientific backend. Once the structure and calculation parameters have been resolved, both entry points use the same routines for FDMNES input generation and execution, with normalization available as a separate post-processing step. Structure selection from an underspecified composition-level request nevertheless remains part of the upstream orchestration, as examined in the repeated-execution results below.

3.4 Repeated-execution consistency for representative workflows

Under the repeated-execution protocol described in Methods, the composition-based and structure-file-based tasks each completed successfully in all ten trials. The RAG-grounded task completed both requested calculations in nine of ten trials. Across the 30 separate task executions, 29 completed end-to-end, corresponding to an observed aggregate success rate of 96.7%. The task-specific outcomes are reported separately in Table 3 because the three workflows differ in complexity and number of requested calculations.

In the unsuccessful RAG trial, the agent retrieved 14 TiO_2 structures instead of selecting the structure with the lowest energy above hull, as occurred in the other successful trials. It initialized a calculation for one structure, after which the trace recorded an error consistent with an attempt to process another structure in the same output directory. The Fe_2O_3 calculation completed successfully in that trial. Manual inspection showed that the default **Range** values retrieved from the manual were correctly propagated to the generated FDMNES input in all ten RAG trials, including the trial that failed during structure handling. Thus, parameter retrieval and propagation

Table 3: Observed repeated-execution outcomes for three representative ChemGraph-XANES workflows under the fixed evaluation protocol described in Methods.

Task	Completed trials	Observed success rate
Composition-based TiO ₂ calculation	10/10	100%
POSCAR-based calculation	10/10	100%
RAG-grounded TiO ₂ /Fe ₂ O ₃ calculation	9/10	90%
Overall	29/30	96.7%

succeeded in 10/10 trials, whereas the complete two-material workflow succeeded in 9/10. The trace supports the interpretation that the failure occurred during multi-structure resolution and output management rather than during documentation retrieval or parameter propagation. This outcome also shows that schema validation constrains tool arguments but does not by itself determine how many structures an underspecified composition-level request should select. The scripts used for these experiments are available at https://github.com/vitorgrizzi/ChemGraph_xanes/tree/main/examples/python_scripts.

3.5 Task-parallel TiO₂ polymorphs

To evaluate the task-parallel execution and data-curation pathway, we performed a Ti K-edge XANES batch using the prompt template and calculation parameters described in Methods. The Materials Project retrieval tool returned 21 non-deprecated TiO₂ structures satisfying the specified energy-above-hull cutoff of 0.10 eV/atom. The returned structure list was used directly as the ensemble input, preserving the correspondence between the retrieved structures and submitted calculations. This produced 21 independent structure-level Parsl tasks, which were dispatched on the LCRC Improv cluster using ten one-node worker blocks. Each worker occupied one 128-core compute node and executed one FDMNES calculation at a time. Execution records confirmed that all ten worker nodes were active simultaneously and that the observed peak concurrency was ten FDMNES calculations. Tasks beyond the first ten remained in the Parsl queue and were assigned as workers became available.

Table 4: Configuration and outcome of the task-parallel TiO₂ XANES batch.

Batch property	Value
Materials Project structures retrieved	21
Worker nodes used	10
Maximum concurrent calculations	10
Ensemble tasks executed	21
Successful calculations	21
Failed calculations	0
Existing outputs reused	0
Convolution outputs generated	21

All 21 calculations completed successfully, and each run produced one convolution spectrum. The framework retained the retrieved structures in two forms: 21 individual CIF files named according to their Materials Project identifiers and compositions, and an `atoms_db.pkl` file containing the complete collection as a pickle-serialized list of `ase.Atoms` objects, each of which retains its Materials Project identifier in its associated metadata. For each calculation, the framework re-

tained the source structure, generated FDMNES input, execution logs, metadata, raw outputs, and convolution spectrum. These records preserve the link between each Materials Project identifier and its calculation parameters, execution status, and spectrum. A separate execution trace records the original request and the sequence of structure retrieval and ensemble execution. Manual inspection of the generated inputs for all 21 calculations confirmed that the absorber, edge, cluster radius, energy mesh, and FDMNES options were propagated as specified. Together, these records form a structure-linked XANES collection in which every spectrum can be traced to its source structure, calculation settings, and execution record. The complete collection is available at https://github.com/vitorgrizzi/ChemGraph_xanes/tree/main/examples/parallel_task.

4 Discussion

Automated XAS generation and spectroscopy workflow management predate the present work. The Materials Project/atomate effort established high-throughput FEFF-based XANES generation and structure-linked reference databases [40, 41]; Lightshow provides a common input-generation abstraction across several XAS codes [42]; Corvus coordinates dependency-driven workflows across spectroscopy and electronic-structure packages [43]; and AiiDA QE Xspec provides provenance-aware workchains for core-level spectroscopy [44]. ChemCrow established the broader utility of LLM agents equipped with chemistry tools [45], while the original ChemGraph framework supplies the general agent and tool infrastructure extended here [27]. We therefore do not claim novelty for automated XAS, high-throughput execution, provenance tracking, or chemistry-tool-using agents individually. The contribution of ChemGraph-XANES is its integration in an FDMNES-specific interface in which documentation retrieval can supply numerical parameters to typed, schema-validated execution tools, while retaining an explicit link among retrieved evidence, final parameters, structure identity, execution records, and spectral outputs.

The main contribution of ChemGraph-XANES lies in formalizing a machine-actionable structure-to-spectrum workflow around established simulation software. Structure acquisition, absorber selection, FDMNES input generation, execution, parsing, normalization, and provenance tracking are organized within a common Python framework and exposed through typed tool interfaces. The LLM interprets requests and supplies schema-validated arguments, while FDMNES and deterministic backend routines perform the scientific calculations. Once structures and parameters have been resolved, local-file and chemistry-level requests therefore share the same execution pathway. Upstream interpretation is not fully deterministic, however: composition-level requests can leave structure cardinality unspecified, as demonstrated by the failed repeated RAG execution. The framework should consequently be understood as controlled orchestration rather than unrestricted or fully autonomous scientific decision-making.

General-purpose coding agents constitute another practical route to FDMNES automation. Given the FDMNES manual, shell access, and a correctly configured software environment, such an agent could likely construct scripts and complete the individual demonstrations presented here. ChemGraph-XANES therefore does not claim exclusive capability or benchmarked superiority. Its distinction lies instead in the execution contract: agent actions are restricted to predefined, schema-validated tools, scientific operations are delegated to a deterministic backend, and structures, resolved parameters, generated inputs, execution records, and spectra are retained through a common workflow. The constrained action space also reduces the orchestration burden because the model selects tools and supplies typed arguments rather than generating and debugging complete execution scripts. This design may reduce token usage and opportunities for unsupported actions, although token efficiency was not quantified here. Comparable safeguards can be built

around general-purpose coding agents, but ChemGraph-XANES incorporates them directly into the XANES execution pathway. A systematic comparison remains an important direction for future work.

The repeated RAG trials distinguish the documentation-grounded parameter-selection step from the broader orchestration workflow. The retrieved FDMNES **Range** setting was propagated correctly in all ten RAG trials, whereas the complete two-material workflow succeeded in nine; the single failure occurred later during multi-structure handling. Thus, the observed failure did not arise from documentation retrieval or parameter transfer. The significance of this example is not merely that the agent recovered a default numerical value from the manual. Rather, it shows that a simulation-relevant parameter can be made externally attributable to a retrieved documentation source and then propagated through a typed execution interface into a concrete XANES calculation. This distinction matters because one of the main risks in language-model-mediated scientific workflows is that parameter choices may otherwise appear plausible without being traceable to either the user, the software documentation, or an explicit default in the implementation. In ChemGraph-XANES, the retrieval trace therefore functions as a lightweight form of parameter provenance: the numerical value used in execution can be linked to a documented source rather than to unsupported free-form model generation.

The 21-structure TiO_2 batch extends the evidence beyond architectural compatibility with distributed execution. A single ensemble call created 21 structure-level tasks, all 21 calculations produced convolution spectra, and execution records showed a peak of ten concurrent FDMNES calculations. The retained structures, generated inputs, logs, metadata, raw outputs, and spectra form a traceable collection in which each spectrum remains linked to its source structure and calculation settings. This is directly relevant to comparative spectroscopy and ML-oriented dataset construction, for which hidden differences in structures, absorbers, simulation settings, or normalization can compromise later reuse. The experiment demonstrates task-parallel execution and structure-linked curation under one configuration, but it does not establish scaling efficiency because no serial reference or worker-count sweep was performed.

The present results retain several important limitations. First, this is a workflow study rather than a comprehensive benchmark of spectral fidelity; the examples do not systematically validate spectral accuracy across a broad materials set. Second, the repeated-execution evaluation comprises three fixed tasks, ten executions per task, one model, and one configuration. The resulting 29/30 outcome is therefore a task-specific observation rather than an estimate of general workflow reliability. Third, although the manual-derived **Range** setting was propagated in all ten RAG trials, this evaluates only one documentation query and one parameter; retrieval recall, answer accuracy across a broader parameter set, and robustness to ambiguous or incomplete manual passages were not measured. Finally, the TiO_2 collection was generated in one Parsl configuration. The observed concurrency demonstrates simultaneous execution, but no serial comparison, strong- or weak-scaling analysis, worker-count sweep, or systematic throughput study was undertaken.

There are also deliberate scope limitations in the current agent interface. The framework exposes a selected subset of FDMNES-relevant controls through validated schemas, rather than allowing the agent to modify arbitrary input keywords. This restriction limits flexibility for some advanced expert use cases, but it is central to the design: constraining the action space reduces the risk that an LLM will hallucinate plausible-looking but unsupported parameters, undocumented settings, or otherwise unintended simulation inputs. As LLMs and documentation-grounded tool-use methods improve, the set of exposed controls can be expanded progressively. In our view, however, such expansion should remain grounded in documentation and mediated through structured validation rather than unconstrained text generation. The current design therefore prioritizes trustworthy orchestration over maximal agent freedom.

Finally, although the present implementation centers on FDMNES, the architectural ideas are broader than a single spectroscopy backend. Structure handling, schema-validated orchestration, task organization, output parsing, and provenance-aware structure-spectrum linkage are conceptually separable from the specific simulation engine used to generate the spectra. This suggests a path toward future extensions in which similar adapters could be built for other XANES or spectroscopy codes while preserving a common workflow and data-curation layer. Such extensibility would be valuable for comparative-method studies and for broader efforts to build interoperable computational spectroscopy infrastructure.

5 Conclusion

We introduced ChemGraph-XANES, a documentation-grounded and schema-constrained framework for XANES simulation and data curation. The framework couples natural-language task specification with structured tool interfaces, deterministic FDMNES input generation, explicit post-processing helpers, and provenance-aware data organization. By acting as an orchestration layer over established physics-based software, ChemGraph-XANES preserves transparent execution logic while making common XANES workflows easier to access, reproduce, and reuse. Across ten separate executions of each of three representative workflows, 29 of 30 tasks completed end-to-end, and the manual-derived FDMNES **Range** setting was propagated correctly in all ten RAG trials. In the task-parallel evaluation, all 21 retrieved TiO_2 structures produced convolution spectra while retaining links to their structural inputs, execution parameters, logs, and raw outputs. These experiments demonstrate task-specific execution consistency and task-parallel collection generation, while broader reliability, retrieval, spectral-accuracy, and scaling behavior remain to be quantified. ChemGraph-XANES therefore provides a foundation for future high-throughput XANES campaigns, comparative spectroscopy studies, and ML-oriented structure-spectrum dataset generation, with an emphasis on constrained orchestration, reproducibility, and traceable data generation.

Conflicts of interest

There are no conflicts to declare.

Data Availability

The XANES-specific implementation used for the experiments in this work is available at https://github.com/vitorgrizzi/ChemGraph_xanes. This repository contains the research fork associated with the present manuscript. Ongoing development of the general ChemGraph framework is maintained in the upstream repository at <https://github.com/argonne-lcf/ChemGraph>. The qualitative documentation-retrieval examples are provided in https://github.com/vitorgrizzi/ChemGraph_xanes/blob/main/notebooks/Demo_rag_agent_Argo.ipynb, and the range-propagation trace is provided in https://github.com/vitorgrizzi/ChemGraph_xanes/blob/main/notebooks/rag_range_fill.txt. Scripts used for the repeated-execution experiments are available at https://github.com/vitorgrizzi/ChemGraph_xanes/tree/main/examples/python_scripts. The task-parallel prompt trace, Materials Project-derived structures, `xanes_results.jsonl` summary, generated FDMNES inputs and logs, raw outputs, and convolution spectra are available at https://github.com/vitorgrizzi/ChemGraph_xanes/tree/main/examples/parallel_task.

Acknowledgments

This work was supported by the U.S. Department of Energy (DOE), Office of Science (SC), Office of Basic Energy Sciences (BES), Division of Chemical Sciences, Geosciences, and Biosciences (CSGB), Catalysis Science Program at Argonne National Laboratory under contract no. DE-AC02-06CH11357. L.P. acknowledges U.S. DOE Office of Science Graduate Student Research (SCGSR) award under contract number DE-SC0014664. The authors gratefully acknowledge the computing resources provided on Swing and Improv, a high-performance computing cluster operated by the Laboratory Computing Resource Center (LCRC) at Argonne National Laboratory. This research used resources of the Argonne Leadership Computing Facility, which is a U.S. Department of Energy Office of Science User Facility operated under contract DE-AC02-06CH11357.

References

- [1] Jiayi Xu, Prajay Patel, A Jeremy Kropf, David Kaphan, Massimiliano Delferro, and Cong Liu. Theoretical investigation of the hydrogenation of cyclohexene catalyzed by supported single-atom sites on redox noninnocent LiMn_2O_4 and LiMn_2O_4 surfaces. *The Journal of Physical Chemistry C*, 128(12):4946–4957, 2024.
- [2] Jing Zhu, Zhenhua Zeng, and Wei-Xue Li. K-edge xanes investigation of Fe-based oxides by density functional theory calculations. *The Journal of Physical Chemistry C*, 125(47):26229–26239, 2021.
- [3] Cristina Piquer, MA Laguna-Marco, Alejandro G Roca, R Boada, C Guglieri, and Jesús Chaboy. Fe k-edge x-ray absorption spectroscopy study of nanosized nominal magnetite. *The Journal of Physical Chemistry C*, 118(2):1332–1346, 2014.
- [4] Jiayi Xu, Colton Lund, Prajay Patel, Yu Lim Kim, and Cong Liu. Recent advances on computational modeling of supported single-atom and cluster catalysts: characterization, catalyst-support interaction, and active site heterogeneity. *Catalysts*, 14(4):224, 2024.
- [5] Jiayi Xu, Yu Lim Kim, Rishu Khurana, Shana Havenridge, Prajay Patel, and Cong Liu. Recent advances in x-ray absorption near edge structure (xanes) simulations for catalysis: Theories and applications. *Annual Reports in Computational Chemistry*, 20:157–187, 2024.
- [6] Abdulrahman Aldossary, Jorge Arturo Campos-Gonzalez-Angulo, Sergio Pablo-García, Shi Xuan Leong, Ella Miray Rajaonson, Luca Thiede, Gary Tom, Andrew Wang, Davide Avagliano, and Alán Aspuru-Guzik. In silico chemical experiments in the age of ai: From quantum chemistry to machine learning and back. *Advanced Materials*, 36(30):2402369, 2024.
- [7] Pascal Friederich, Florian Häse, Jonny Proppe, and Alán Aspuru-Guzik. Machine-learned potentials for next-generation matter simulations. *Nature Materials*, 20(6):750–761, 2021.
- [8] Vitor Ferreira Grizzi, Benjamin Nebgen, and Y Z. Active-learned potentials reveal the interplay between the structure and dynamics in thorium molten salts. *Physical Review Materials*, 10(6):065606, 2026.
- [9] Kamal Choudhary, Kevin F Garrity, Andrew CE Reid, Brian DeCost, Adam J Biacchi, Angela R Hight Walker, Zachary Trautt, Jason Hattrick-Simpers, A Gilad Kusne, Andrea Centrone, et al. The joint automated repository for various integrated simulations (jarvis) for data-driven materials design. *npj computational materials*, 6(1):173, 2020.

- [10] Dipendra Jha, Kamal Choudhary, Francesca Tavazza, Wei-keng Liao, Alok Choudhary, Carelyn Campbell, and Ankit Agrawal. Enhancing materials property prediction by leveraging computational and experimental data using deep transfer learning. *Nature communications*, 10(1):5316, 2019.
- [11] Zeeshan Ahmad, Tian Xie, Chinmay Maheshwari, Jeffrey C Grossman, and Venkatasubramanian Viswanathan. Machine learning enabled computational screening of inorganic solid electrolytes for suppression of dendrite formation in lithium metal anodes. *ACS central science*, 4(8):996–1006, 2018.
- [12] Henry Sprueill, Carl Edwards, Mariefel Olarte, Udishnu Sanyal, Heng Ji, and Sutanay Choudhury. Monte carlo thought search: Large language model querying for complex scientific reasoning in catalyst design. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8348–8365, 2023.
- [13] Jiayi Xu, Rishu Khurana, Taylor Harville, Vitor Ferreira Grizzi, Luke Pretzie, Anik Biswas, Prajay Patel, and Cong Liu. Rational design of heterogeneous single-site catalysts via surface organometallic chemistry. *Coordination Chemistry Reviews*, 566:218186, 2026.
- [14] Jenny G Vitillo, Alán Aspuru-Guzik, Eric Dskocil, Omar K Farha, Timur Islamoglu, Heather J Kulik, Peter M Margl, Stuart Miller, Jordan Reddel, Aayush R Singh, et al. Accelerating catalytic advancements through the precision of high-throughput experiments & calculations. *Digital Discovery*, 5(2):497–509, 2026.
- [15] Vitor F. Grizzi, Luke N. Pretzie, Jiayi Xu, and Cong Liu. XANE(3): An E(3)-equivariant graph neural network for accurate prediction of XANES spectra from atomic structures. *arXiv preprint arXiv:2604.12140*, 2026.
- [16] Jacob W Smith and Richard J Saykally. Soft x-ray absorption spectroscopy of liquids and solutions. *Chemical reviews*, 117(23):13909–13934, 2017.
- [17] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.
- [18] Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, et al. Large language model agent: A survey on methodology, applications and challenges. *arXiv preprint arXiv:2503.21460*, 2025.
- [19] Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*, 2024.
- [20] Cheng Yang, Jiaxuan Lu, Haiyuan Wan, Junchi Yu, and Feiwei Qin. From what to why: A multi-agent system for evidence-based chemical reaction condition reasoning. *arXiv preprint arXiv:2509.23768*, 2025.
- [21] Yunheng Zou, Austin H Cheng, Abdulrahman Aldossary, Jiaru Bai, Shi Xuan Leong, Jorge Arturo Campos-Gonzalez-Angulo, Changhyeok Choi, Cher Tian Ser, Gary Tom, Andrew Wang, et al. El agente: An autonomous agent for quantum chemistry. *Matter*, 8(7), 2025.
- [22] Edward O Pyzer-Knapp, Matteo Manica, Peter Staar, Lucas Morin, Patrick Ruch, Teodoro Laino, John R Smith, and Alessandro Curioni. Foundation models for materials discovery—current state and future directions. *Npj Computational Materials*, 11(1):61, 2025.

- [23] Yixi Shen, Ledu Wang, Yan Huang, Xiaolong Zhang, Meng Huang, Huirong Li, Jing He, Aoran Cai, Yang Wang, Pieter ES Smith, et al. Unlocking azobenzene isomerization mechanisms via an llm agent-driven workflow integrating simulation, experiment, and machine learning. *Chemical Science*, 17(19):9663–9673, 2026.
- [24] Kamal Choudhary. Atomgpt: Atomistic generative pretrained transformer for forward and inverse materials design. *The Journal of Physical Chemistry Letters*, 15(27):6909–6917, 2024.
- [25] Shi Xuan Leong, Caleb E Griesbach, Rui Zhang, Kourosh Darvish, Yuchi Zhao, Abhijoy Mandal, Yunheng Zou, Han Hao, Varinia Bernales, and Alán Aspuru-Guzik. Steering towards safe self-driving laboratories. *Nature Reviews Chemistry*, 9(10):707–722, 2025.
- [26] Izumi Takahara, Teruyasu Mizoguchi, and Bang Liu. Accelerated inorganic materials design with generative ai agents. *Cell Reports Physical Science*, 6(12), 2025.
- [27] Thang D. Pham, Aditya Tanikanti, and Murat Keçeli. ChemGraph as an agentic framework for computational chemistry workflows. *Communications Chemistry*, 9(1):33, 2026.
- [28] Ziqi Wang, Hongshuo Huang, Hancheng Zhao, Changwen Xu, Shang Zhu, Jan Janssen, and Venkatasubramanian Viswanathan. Dreams: Density functional theory based research engine for agentic materials simulation. *arXiv preprint arXiv:2507.14267*, 2025.
- [29] Andrew D McNaughton, Gautham Krishna Sankar Ramalaxmi, Agustin Kruel, Carter R Knutson, Rohith A Varikoti, and Neeraj Kumar. Cactus: Chemistry agent connecting tool usage to science. *ACS omega*, 9(46):46563–46573, 2024.
- [30] Chris Lu, Cong Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. Towards end-to-end automation of ai research. *Nature*, 651(8107):914–919, 2026.
- [31] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dulak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, Eric D Hermes, Paul C Jennings, Peter Bjerre Jensen, James Kermode, John R Kitchin, Esben Leonhard Kolsbjerg, Joseph Kubal, Kristen Kaasbjerg, Steen Lysgaard, Jón Bergmann Maronsson, Tristan Maxson, Thomas Olsen, Lars Pastewka, Andrew Peterson, Carsten Rostgaard, Jakob Schiøtz, Ole Schütt, Mikkel Strange, Kristian S Thygesen, Tejs Vegge, Lasse Vilhelmsen, Michael Walter, Zhenhua Zeng, and Karsten W Jacobsen. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, 2017.
- [32] O Bunău and Yves Joly. Self-consistent aspects of x-ray absorption calculations. *Journal of Physics: Condensed Matter*, 21(34):345501, 2009.
- [33] Y Joly. X-ray absorption near-edge structure calculations beyond the muffin-tin approximation. *Physical Review B*, 63(12):125120, 2001.
- [34] Yadu Babuji, Anna Woodard, Zhuozhao Li, Daniel S Katz, Ben Clifford, Rohan Kumar, Lukasz Lacinski, Ryan Chard, Justin M Wozniak, Ian Foster, et al. Parsl: Pervasive parallel programming in python. In *Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing*, pages 25–36, 2019.

- [35] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1), 2013.
- [36] Shyue Ping Ong, Shreyas Cholia, Anubhav Jain, Miriam Brafman, Dan Gunter, Gerbrand Ceder, and Kristin A Persson. The materials application programming interface (api): A simple, flexible and efficient api for materials data based on representational state transfer (rest) principles. *Computational Materials Science*, 97:209–215, 2015.
- [37] Jared Pohlmann, Manjula Raman, Lily Bonds, and Kenneth Park. The stability of tio2 phases studied using r2scan in the hubbard-corrected density functional theory. *Molecules*, 30(3):560, 2025.
- [38] Masashi Ishii and Industrial Application and Partnership Division. XAFS spectrum of Titanium(IV) dioxide (Anatase). NIMS Materials Data Repository, 2021. Measured at SPring-8 BL14B2.
- [39] Tiago Caneu Rossi, Daniel Grolimund, Maarten Nachtegaal, Oliviero Cannelli, Giulia Federica Mancini, Camila Bacellar, D Kinschel, J  r  my R Rouxel, N Ohannessian, Daniele Pergolesi, et al. X-ray absorption linear dichroism at the ti k edge of anatase tio 2 single crystals. *Physical Review B*, 100(24):245207, 2019.
- [40] Kiran Mathew, Chen Zheng, Donald Winston, Chi Chen, Alan Dozier, John J Rehr, Shyue Ping Ong, and Kristin A Persson. High-throughput computational x-ray absorption spectroscopy. *Scientific data*, 5(1):180151, 2018.
- [41] Ying Zhang and Chen Ling. A strategy to apply machine learning to small datasets in materials science. *Npj Computational Materials*, 4(1):25, 2018.
- [42] Matthew R Carbone, Fanchen Meng, Christian Vorwerk, Benedikt Maurer, Fabian Peschel, Xiaohui Qu, Eli Stavitski, Claudia Draxl, John Vinson, and Deyu Lu. Lightshow: a python package for generating computational x-ray absorption spectroscopy input files. *Journal of Open Source Software*, 8(87):5182, 2023.
- [43] SM Story, FD Vila, JJ Kas, KB Raniga, CD Pemmaraju, and JJ Rehr. Corvus: a framework for interfacing scientific software for spectroscopic and materials science applications. *Journal of Synchrotron Radiation*, 26(5):1694–1704, 2019.
- [44] Xing Wang, Edan Bainglass, Miki Bonacci, Andres Ortega-Guerrero, Lorenzo Bastonero, Marnik Bercx, Pietro Bonf  , Roberto De Renzi, Dou Du, Peter NO Gillespie, et al. Making atomistic materials calculations accessible with the aiidalab quantum espresso app. *npj Computational Materials*, 12(1):72, 2026.
- [45] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature machine intelligence*, 6(5):525–535, 2024.