

AutoXRD: Autonomous LLM Agents and Comprehensive Evaluation for Powder Diffraction Analysis

Anonymous Author(s)

ABSTRACT

Powder X-ray diffraction (XRD) is central to materials characterization, yet reliable end-to-end automation remains challenging. An XRD agent must interpret diffraction evidence, operate refinement software, manage coupled parameters in a defensible order, and distinguish numerical improvement from physical validity. In this paper, we propose AutoXRD, an autonomous large language model (LLM) agent framework that organizes powder-XRD analysis as stepwise refinement, grounds actions in observed evidence, and applies deterministic crystallographic and physical checks before accepting results. We further introduce XRD-Bench with two complementary tracks. XRD-Bench-QA contains 100 bounded diagnostic tasks that isolate scientific reasoning and decision-making, whereas XRD-Bench-E2E contains 34 executable workflows that test whether agents can compose these capabilities into complete analyses requiring file inspection, crystallographic-software execution, iterative refinement, evidence preservation, and reporting. We evaluate ten recent LLMs across 1,340 model-task runs. Models average only 57.8 out of 100, falling from 61.9 on XRD-Bench-QA to 53.7 on XRD-Bench-E2E. They perform best on refinement-history assessment and result acceptance, but remain substantially weaker on refinement-action selection, phase quantification, indexing, and Rietveld refinement. GPT-5.6 Sol achieves the highest overall score of 81.1, GPT-5.6 Terra the highest XRD-Bench-E2E point estimate of 81.0, and GPT-5.6 Luna the best score-cost trade-off. Ablations show that all six AutoXRD components consistently improve performance, supporting the framework design. Finally, execution-trace analysis reveals recurring failures in coupled-parameter control, quantitative reasoning, evidence preservation, and workflow termination, motivating stronger scientific constraints, uncertainty-aware decisions, and more efficient planning.

 <https://autoxrd.github.io/AutoXRD/>

CCS CONCEPTS

• **Computing methodologies** → **Intelligent agents**; *Natural language processing*; • **Applied computing** → *Physical sciences and engineering*.

KEYWORDS

scientific agents, tool use, powder diffraction, Rietveld refinement, agent evaluation, AI for science

ACM Reference Format:

Anonymous Author(s). 2026. AutoXRD: Autonomous LLM Agents and Comprehensive Evaluation for Powder Diffraction Analysis. In *Proceedings of the Twentieth ACM International Conference on Web Search and Data Mining (WSDM '27)*, February 15–19, 2027, Hong Kong, China. ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

Powder X-ray diffraction (XRD) compresses three-dimensional reciprocal-space information into a one-dimensional intensity profile. Peak overlap, background, instrumental aberrations, preferred orientation, finite crystallite size, strain, and phase mixtures can therefore produce similar observations. Rietveld refinement addresses overlap by fitting the complete profile [24, 41], but it does not turn analysis into an unconstrained curve-fitting problem. The parameters governing the crystal structure, specimen, and instrument are strongly coupled; successful practice depends on releasing them in a defensible order and inspecting the observed, calculated, and difference patterns after each stage [34, 35].

These requirements expose two related gaps in current language-agent research. First, tool-using language models can interleave reasoning and external actions [28, 39], and domain agents have automated parts of chemistry and experimentation [2, 3], but a general tool loop provides no scientific guarantee. A syntactically valid action may jointly release non-identifiable parameters, a lower R_{wp} may arise from an incorrect structural or peak-profile hypothesis, and a fluent report may conceal an impossible occupancy or an unresolved phase. Second, current XRD-agent evaluation is not yet systematic. Existing studies demonstrate the feasibility of LLM-assisted Rietveld refinement [17, 30], but evaluations are often based on selected cases, isolated questions, or final fit statistics. No single criterion is sufficient: exact matching cannot accommodate valid open-ended analyses, fit metrics do not establish physical correctness, and judging only the final report cannot reveal whether a failure arose from scientific reasoning, software execution, missing artifacts, or poor termination. The field therefore lacks a unified protocol that tests both individual XRD decisions and complete executable workflows while measuring scientific quality, execution validity, efficiency, and failure behavior across models.

Our abstraction treats refinement as a stepwise graph of scientific decisions. A node stores a complete fit state; an edge stores one structured action, the evidence that supports it, and predictions that can falsify its rationale. Numerical software computes the candidate child state, but deterministic checks—not the language model—decide whether the step is accepted. Rejected edges remain first-class evidence, enabling failure analysis and future policy distillation.

Figure 1 summarizes AutoXRD. At runtime, the language model maintains session context and memory, plans with ten XRD-specific skills, and invokes permissioned tools within a sandbox. The skills translate domain procedures into executable scientific plans that specify the refinement stage, parameters or code, numerical bounds, required evidence, and expected outcomes. The scientific action layer parses diffraction patterns and structures, enforces parameter constraints, executes FullProf, GSAS-II, or task-appropriate numerical routines, and stores the resulting artifacts. Pattern-quality,

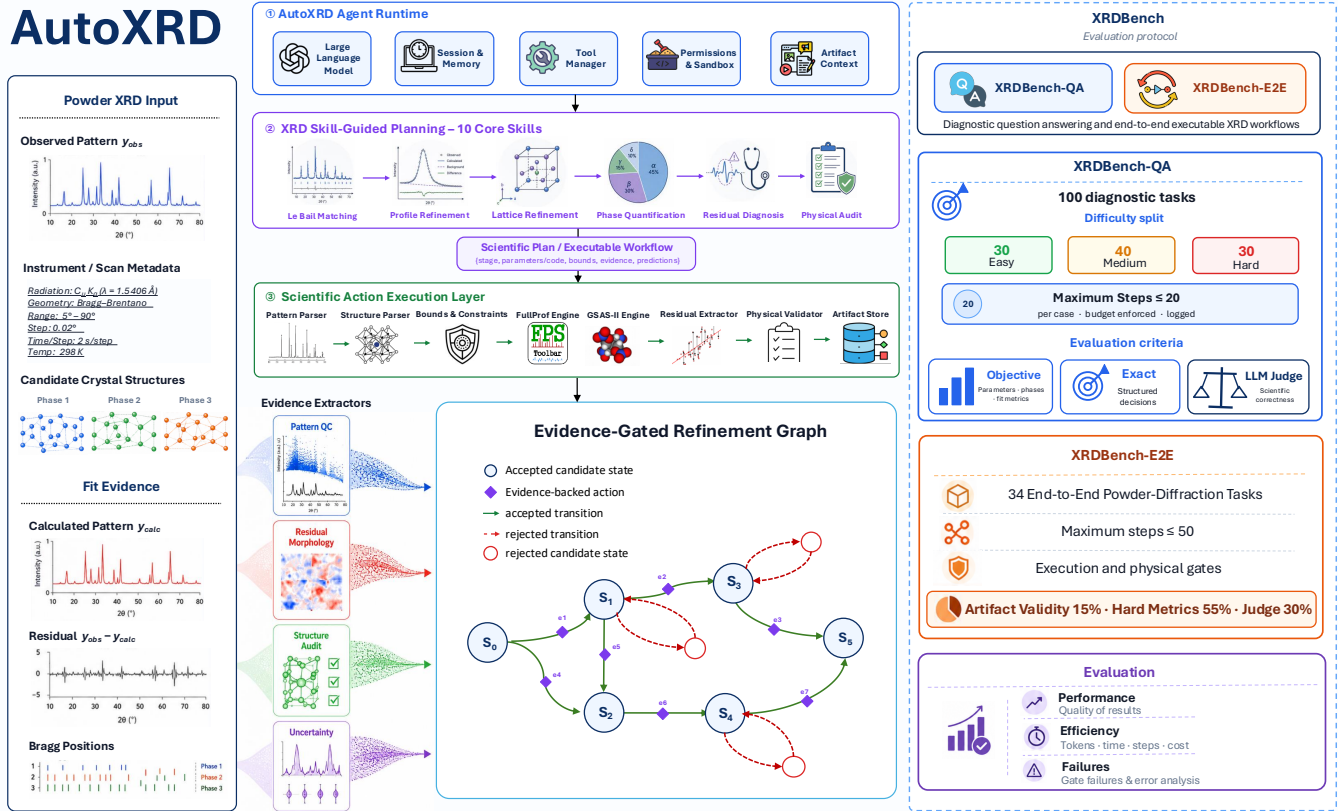


Figure 1: Overview of AutoXRD and the XRD-Bench evaluation protocol. AutoXRD combines an autonomous agent runtime, skill-guided powder-diffraction workflows, executable crystallographic backends, and evidence-based scientific validation. XRD-Bench evaluates diagnostic reasoning through XRDBench-QA and autonomous end-to-end XRD analysis through XRDBench-E2E.

residual, structural, and uncertainty evidence is then used to accept or reject candidate transitions in an evidence-gated refinement graph.

The right panel of Figure 1 presents XRDBENCH, designed to address the evaluation gap through two complementary tracks. XRDBENCH-QA contains 100 bounded diagnostic tasks that isolate scientific reasoning and decisions, enabling reliable exact, objective, and reference-guided assessment. XRDBENCH-E2E contains 34 executable workflows over measured patterns, requiring agents to inspect inputs, implement and run numerical analyses, validate intermediate results, and deliver auditable artifacts. Neither track is sufficient alone: the former diagnoses what an agent understands, whereas the latter tests whether those capabilities can be composed into a functioning analysis. Together they evaluate the same XRD process at different degrees of integration and report performance, efficiency, and failure behavior under a common protocol. Using XRDBENCH, we evaluate ten recent LLMs in 1,340 agent runs. Average performance decreases from 61.9 on XRDBENCH-QA to 53.7 on XRDBENCH-E2E, and the two track scores are only moderately correlated ($r = 0.56$, $\rho = 0.65$), showing that diagnostic competence does not reliably translate into autonomous execution. Models perform best on refinement-history assessment and result acceptance,

but remain substantially weaker on refinement-action selection, phase quantification, indexing, and Rietveld refinement.

To summarize, our main contributions are:

- **A scientifically grounded autonomous XRD agent.** We introduce AUTOXRD, which combines XRD skill-guided planning, executable crystallographic backends, and deterministic scientific validation. Rather than accepting a step merely because the software runs or the fit residual decreases, AUTOXRD requires the predicted evidence change and crystallographic checks to pass, while preserving both accepted and rejected actions in an auditable refinement trajectory.
- **A two-track benchmark for XRD reasoning and execution.** We introduce XRDBENCH, comprising 100 bounded XRDBENCH-QA diagnostic tasks and 34 executable XRDBENCH-E2E workflows across eleven powder-diffraction task families. Its complementary evaluation protocols measure scientific decisions, numerical results, physical validity, artifact completeness, efficiency, and failure behavior.
- **A large-scale empirical study of LLMs for XRD.** We evaluate ten recent LLMs across 1,340 agent runs, 22,135 model turns, and 625.6M selected-trajectory tokens. Our analysis shows that

diagnostic performance does not reliably transfer to autonomous execution and identifies refinement-action selection, phase quantification, indexing, and Rietveld refinement as major remaining bottlenecks.

2 BACKGROUND AND RELATED WORK

2.1 Rietveld Refinement and Its Failure Modes

Let x_i denote the diffraction coordinate (e.g., 2θ , time of flight, or d -spacing), y_i the observed intensity, σ_i its standard uncertainty, and $\hat{y}_i(\theta)$ the profile calculated from parameters θ . Classical Rietveld refinement minimizes weighted least squares [24]:

$$\theta^* = \arg \min_{\theta \in \Theta} \chi^2(\theta), \quad \chi^2(\theta) = \sum_{i=1}^n w_i [y_i - \hat{y}_i(\theta)]^2, \quad (1)$$

where $w_i = \sigma_i^{-2}$ when uncertainties are available. The vector θ includes scale, background, zero shift, lattice, profile, atomic coordinates, displacement factors, occupancies, preferred orientation, and microstructure terms. Le Bail profile matching can align the cell, background, and peak profile before atomic parameters are exposed [1]. For a fully crystalline mixture with all relevant phases included in the refinement model, phase weight fractions can be derived from refined scale factors and phase-specific ZMV terms, where Z , M , and V denote formula units per cell, formula mass, and unit-cell volume [13]. Preferred orientation and line shape require specialized models [9, 32].

No scalar fit index establishes structural correctness. For example, the weighted-profile factor

$$R_{wp} = 100 \sqrt{\frac{\sum_i w_i (y_i - \hat{y}_i)^2}{\sum_i w_i y_i^2}} \quad (2)$$

depends on data quality and uncertainty conventions; a wrong model on weak data may achieve a lower value than an excellent model on precise data [34]. This motivates explicit checks of residual structure, parameter correlations, chemistry, bounds, and unresolved peaks.

2.2 Automation of XRD Analysis

Crystallographic engines such as FullProf [26] and GSAS-II [36] provide mature numerical capabilities. Automation has progressed from high-throughput wrappers (SrRietveld) [33], to expert systems (AutoFP) [8], reinforcement learning (PowderBot) [11], black-box optimization (BBO-Rietveld) [23], derivative-guided recipe selection [35], and learned parameter prediction [22]. Robotic PXRD systems have integrated sample preparation, measurement, and automated analysis [40].

Phase identification is related but distinct. Deep models can learn multi-phase labels from synthetic patterns [16, 29]; CrystalShift uses symmetry-constrained pseudo-refinement and Bayesian comparison [6]; Dara searches and refines multiple phase combinations [10]; and RADAR-PD couples mismatch-tolerant prediction with Rietveld verification [31]. Recent end-to-end pipelines broaden this scope: AGAPI-XRD combines database matching, a fine-tuned structure generator, and GSAS-II/BGMN refinement [4], while MatDiffraction couples hierarchical retrieval with full-pattern refinement

and quantitative phase fitting [37]. COD [12], opXRD [14], SimXRD-4M [5], and SIMPOD [25] improve data availability. These advances automate engines, prediction, or search. Our focus is complementary: constraining and evaluating the sequence of decisions made by a language agent operating scientific software.

Table 1 compares observable functionality rather than project-specific terminology. Prior systems cover important but separated parts of the problem: SrRietveld, AutoFP, PowderBot, and BBO-Rietveld automate refinement; CrystalShift and Dara address phase ambiguity; and Rongzai Agent and AgentBuild introduce LLM-based control. Rongzai compares four base models on five neutron-diffraction samples and records iterative refinement, while AgentBuild evaluates reports and trajectories with an expert-authored rubric and curriculum. AGAPI-XRD and MatDiffraction benchmark automated structure identification and refinement over hundreds of patterns. None provides a reusable benchmark spanning bounded scientific diagnosis and heterogeneous executable workflows with cross-model analysis of decisions, tool trajectories, and execution behavior. In contrast, XRDBENCH evaluates one process through two complementary parts: 100 QA tasks isolate scientific decisions and outcomes, while 34 state-changing workflows test their composition into executable analyses. Together they measure reasoning, scientific outcomes, execution evidence, physical validity, artifacts, trajectories, token use, wall time, and tool efficiency under a shared ten-model comparison.

2.3 Tool-Using and Scientific Agents

ReAct interleaves language reasoning with actions [39], while Toolformer learns when and how to call APIs [28]. AgentBench [18], τ -bench [38], and MLAGentBench [15] show that endpoint success alone can hide long-horizon failures and inconsistency. Scientific-agent work includes tool-augmented reasoning [20], chemistry agents [3], autonomous laboratory orchestration [2], and broad research automation [19]. ScienceAgentBench argues for authentic, task-level assessment before claims of end-to-end discovery [7].

AUTOXRD applies this lesson to a narrow scientific domain with unusually sharp validity constraints. It differs from generic agents by making action schemas, falsifiable predictions, stage order, coupled-parameter exclusions, mandatory physical checks, and rejected steps part of the executable protocol.

3 PROBLEM FORMULATION

3.1 Refinement States and Structured Actions

An input instance is

$$\mathcal{I} = (\mathcal{D}, \mathcal{M}, \mathcal{C}), \quad \mathcal{D} = \{(x_i, y_i, \sigma_i)\}_{i=1}^n, \quad (3)$$

where $\mathcal{M}$ is instrument metadata and $\mathcal{C}$ is a set of candidate structures. At step t , the agent observes

$$s_t = (\theta_t, q_t, r_t, v_t, c_t, O_t), \quad (4)$$

where $q_t = (R_{wp,t}, R_{exp,t}, \text{GoF}_t)$ contains fit scalars, r_t contains residual features, v_t is a set of physical violations, c_t describes complexity and correlations, and O_t indexes backend output files. Here R_{wp} is the weighted-profile residual, R_{exp} is the statistically expected residual, and $\text{GoF} = R_{wp}/R_{exp}$ denotes goodness of fit.

Table 1: Functional comparison with representative automated powder-diffraction systems. ✓ denotes a capability explicitly provided or evaluated in the cited work; ✗ means that the work does not report it. Residual-guided iteration uses localized or structured residual evidence; physical-validity checks go beyond a scalar fit index or optimizer bound. Public suites require reusable tasks and scoring protocols; complete telemetry reports tokens, wall time, and tool- or step-level measurements.

System	Scientific analysis capabilities				Agent and evaluation capabilities					
	Automated Rietveld refinement	Residual-guided iteration	Physical-validity checks	Multiple-hypothesis analysis	LLM tool agent	Auditable action trajectory	Public diagnostic suite	Public executable workflow suite	Cross-model evaluation	Complete agent telemetry
SrRietveld [33]	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
AutoFP [8]	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
PowderBot [11]	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
BBO-Rietveld [23]	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
CrystalShift [6]	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
Dara [10]	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗
AGAPI-XRD [4]	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗
MatDiffact [37]	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗
Rongzai Agent [17]	✓	✓	✓	✗	✓	✓	✗	✗	✓	✗
AgentBuild [30]	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗
AutoXRD (ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

AutoXRD supports two complementary execution modes. In the validated refinement path, the agent proposes a structured action

$$a_t = (k_t, g_t, P_t, B_t, \mathcal{Z}_t, \Pi_t), \quad (5)$$

where k_t is an action kind, g_t a refinement stage, P_t a parameter subset, B_t optional bounds, $\mathcal{Z}_t$ cited evidence, and Π_t falsifiable predictions. Each evidence item is (f, v, τ, src) : feature name, observed value, decision threshold, and data source. A prediction $\pi = (m, d, \delta)$ specifies metric m , required direction $d \in \{-1, +1\}$, and minimum effect $\delta \geq 0$. In the native code-agent path, actions may instead read, create, edit, or execute task files. These broader actions are retained verbatim in the trajectory and are evaluated through execution, artifact, and physical gates.

The deterministic backend produces a candidate

$$(s'_{t+1}, \mathcal{O}_{t+1}) = F(s_t, a_t; \mathcal{D}), \quad (6)$$

but the transition is not automatically accepted. This distinguishes three questions: (i) did the optimizer find a numerically better point; (ii) did the intervention change the feature predicted by its stated mechanism; and (iii) is the child state physically and statistically admissible?

3.2 Refinement Decision Graph

A refinement decision graph is $G = (V, \mathcal{E}^+, \mathcal{E}^-)$. Each $s \in V$ is a fit state, $\mathcal{E}^+$ contains accepted actions, and $\mathcal{E}^-$ contains rejected actions. Let $H(s, a, s') \in \{0, 1\}$ denote hard admissibility and

$$M(s, a, s') = \prod_{\pi \in \Pi(a)} \mathbb{I}[d_\pi(m_\pi(s') - m_\pi(s)) \geq \delta_\pi] \quad (7)$$

denote support for the action's expected outcome. The candidate child is accepted only if

$$\text{Accept}(s, a, s') = H(s, a, s') M(s, a, s') \mathbb{I}[U(s') - U(s) > 0] = 1. \quad (8)$$

Crucially, utility cannot compensate for $H = 0$ or $M = 0$.

4 AUTOXRD

4.1 Layered Agent Architecture

As shown in Figure 1, AutoXRD separates five responsibilities. The agent controller manages the model, tools, sessions, and budgets. An XRD procedure library provides ten skills: pattern quality control, structure validity checking, executable GSAS-II workflows, Le Bail initialization, validated PCR generation, staged refinement, residual extraction, residual diagnosis, result acceptance, and final scientific review. The XRD execution layer parses patterns and structures, invokes GSAS-II, FullProf, or task-appropriate numerical crystallography code, and extracts reproducible measurements. Pre-execution and post-execution checks enforce parameter, fit, residual, and physical constraints. Finally, a run-history store and report generator expose the complete sequence of accepted and rejected steps.

The language model is therefore a planner, implementer, and explainer, not the numerical optimizer or final scientific authority. This division follows a capability principle: use language models for context-dependent hypothesis selection, mature crystallographic engines for nonlinear optimization, and deterministic code for invariants that can be expressed exactly.

4.2 Deterministic Residual Evidence

For residuals $e_i = y_i - \hat{y}_i$, the feature extractor estimates robust scale

$$\hat{\sigma}_e = \frac{\text{median}_i |e_i - \text{median}(e)|}{0.67448975} \quad (9)$$

and standardized residuals $z_i = (e_i - \text{median}(e)) / \hat{\sigma}_e$. Let $\tilde{e}_i = e_i - \bar{e}$, where $\bar{e} = n^{-1} \sum_i e_i$. The extractor reports normalized absolute residual, lag-one autocorrelation

$$\rho_1 = \frac{\sum_{i=1}^{n-1} \tilde{e}_i \tilde{e}_{i+1}}{\sum_{i=1}^n \tilde{e}_i^2}, \quad (10)$$

low/high-angle signed bias, the fraction of points in contiguous regions with $|z_i| \geq 5$, and an unexplained-peak ratio. These features make diagnoses testable. For example, a nearly angle-independent peak-position offset can motivate zero-shift refinement. Geometry-dependent specimen-displacement and calibration effects must first

be excluded, and the action must predict a minimum decrease in a position-bias feature. If R_{wp} decreases but the predicted feature does not, Equation 7 rejects the causal explanation.

4.3 Validated PCR Generation

FullProf PCR files are positional and contain codewords that can couple multiple parameters. The PCR validator and generator parses comment-anchored semantic slots into a catalog, freezes catalogued codeword groups, and releases only selectors permitted by k_t . It validates stage membership, selector coverage, constraints, phase count, pattern count, and mode before writing a new file. The source is never overwritten.

Let $\Gamma(k)$ be the allowed semantic selector set for action kind k and $\text{cl}(P)$ the closure of P under existing constraint groups. Compilation is permitted only when

$$P \subseteq \Gamma(k), \quad \text{cl}(P) \setminus P = \emptyset, \quad (11)$$

unless the expansion is explicitly declared. This prevents a seemingly local edit from silently releasing unrelated parameters.

4.4 Scientific Validity Checks and Candidate Ranking

The pre-execution check rejects actions outside their stage and high-risk actions without bounds. It also rejects or requires explicit constraints for strongly correlated parameter combinations, such as simultaneous refinement of occupancy and displacement parameters or poorly identifiable combinations of zero shift, lattice parameters, and wavelength. After execution, $H(s_t, a_t, s'_{t+1}) = 0$ if a new physical violation appears, R_{wp} regresses by more than 2%, the unexplained-peak ratio increases by more than 0.02, the maximum absolute correlation exceeds 0.98, or any declared prediction fails.

Among states that pass all mandatory checks, the current implementation ranks candidates with

$$U(s) = - \left[\frac{R_{wp}(s)}{\max(R_{exp}(s), \epsilon)} + 2R(s) + 3Q(s) + 0.002K(s) + 10[C_{\max}(s) - 0.90]_+ + 10|v(s)| \right], \quad (12)$$

where R is residual score, Q unexplained-peak ratio, K parameter count, and $C_{\max}$ maximum absolute correlation. Here $\epsilon > 0$ is a numerical stabilizer. Weights are transparent policy constants rather than learned scientific truth. Equation 8 uses this score only after all mandatory checks pass.

4.5 Complete Refinement History

Each run record contains $(a_t, s_t, s'_{t+1}, \text{dec}_t, \mathcal{O}_{t+1})$, where dec_t is the acceptance decision and $\mathcal{O}_{t+1}$ lists the corresponding code, back-end inputs, outputs, and evidence artifacts. The history records the proposed action, supporting observations, before/after states, validation results, and output-file paths for every accepted or rejected step. Retaining rejected attempts makes the final result reproducible and allows scientists to review which hypotheses were tried, why they failed, and how the accepted refinement path was obtained.

5 XRDBENCH

5.1 From Bounded Questions to Executable Workflows

An autonomous diffraction analysis combines local scientific judgments with implementation, backend use, evidence preservation, and termination. Endpoint-only evaluation reveals whether a workflow succeeded but obscures why; bounded questions isolate the failure but cannot establish autonomy. XRDBENCH therefore contains two complementary parts that evaluate the same process at different levels of integration:

$$\underbrace{\mathcal{B}_Q}_{\text{bounded XRD questions}} \rightarrow \underbrace{\mathcal{B}_E}_{\text{state-changing workflows}}, \quad (13)$$

where the arrow denotes increasing task integration, not information flow.

The first part is XRDBENCH-QA. Here QA denotes question answering under bounded evidence and output requirements, not only multiple choice. Thirty cases require exact option sets, 40 require structured evidence-grounded reports, and 30 require quantitative outcomes plus reports. XRDBENCH-E2E contains 34 open workflows in which the agent must inspect files, implement and execute an analysis, revise failures, and package the evidence supporting its answer. Figure 2 summarizes the two tracks, their capability strata, and their task families; the complete task inventory is provided in the supplementary material.

The QA track uses three evaluation forms: exact matching for 30 scientific decisions, reference-guided assessment for 40 structured diagnoses, and recomputed objective metrics plus report assessment for 30 quantitative outcomes. The end-to-end track instead evaluates executed reports and artifacts through execution and physical gates, task-specific numerical metrics, and scientific judging. The complete protocol table is included in the supplementary material.

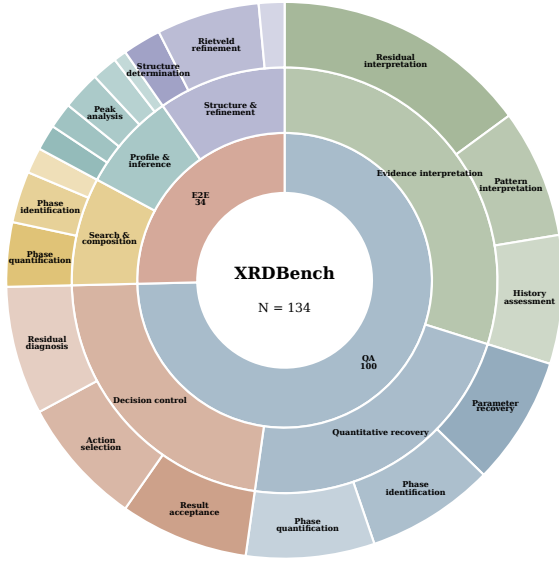
5.2 XRDBench-QA

XRDBench-QA evaluates whether an agent can make scientifically grounded decisions from bounded diffraction evidence. Its 100 tasks cover nine capabilities spanning refinement decisions, evidence interpretation, residual diagnosis, and quantitative recovery. Seventy tasks use deterministically generated inputs and ground truth, while the remaining 30 are constructed from experimental quantitative-phase-analysis data reported by the IUCr and phase-identification cases adapted from Dara [10, 21]. Each public task provides the relevant pattern or refinement-trajectory evidence together with a required response format, whereas ground-truth answers, numerical targets, and scoring rubrics are retained by the evaluator.

For an objective metric m , baseline b , and oracle o , higher- and lower-is-better utilities are

$$u_{\uparrow}(m) = \text{clip}\left(\frac{m-b}{o-b}, 0, 1\right), \quad u_{\downarrow}(m) = \text{clip}\left(\frac{b-m}{b-o}, 0, 1\right). \quad (14)$$

Parameter recovery combines mechanism detection and normalized errors; QPA combines phase-set F1, fraction error/closure, amorphous accuracy, and report quality; phase identification combines phase-set F1, indexed coverage, unknown handling, and report

**Inner Ring: Evaluation Track**

- XRDBench-QA: 100 tasks
- XRDBench-E2E: 34 tasks

Difficulty

- QA: 30 Easy / 40 Medium / 30 Hard
- E2E: 11 Medium / 23 Hard

Middle Ring: Capability Stratum

- QA - Evidence interpretation: 40
- QA - Quantitative recovery: 30
- QA - Decision control: 30
- E2E - Search & composition: 11
- E2E - Profile & inference: 10
- E2E - Structure & refinement: 13

Outer Ring: Task Family**XRDBench-QA**

- History assessment: 10
- Pattern interpretation: 10
- Residual interpretation: 20
- Parameter recovery: 10
- QA phase identification: 10
- QA phase quantification: 10
- Action selection: 10
- Residual diagnosis: 10
- Result acceptance: 10

XRDBench-E2E

- Database Retrieval: 2
- Phase Identification: 4
- Phase Quantification: 5
- Crystallite Analysis: 2
- Defect Disorder: 2
- Material Property Prediction: 1
- Pattern Prediction: 2
- Peak Analysis: 3
- Indexing: 2
- Rietveld Refinement: 8
- Structure Determination: 3

Ring areas encode task counts.

Figure 2: Composition of XRDBench. The inner ring separates the 100 diagnostic tasks from the 34 executable end-to-end workflows; the middle ring groups related capabilities; and the outer ring shows nine XRDBench-QA capability categories and eleven XRDBench-E2E task families.

quality. With normalized case scores $u_j^{(Q)} \in [0, 1]$, the QA track score is $S_Q = \sum_{j=1}^{100} u_j^{(Q)}$.

5.3 XRDBench-E2E

The end-to-end track evaluates whether an agent can transform diffraction inputs into a complete, executable, and scientifically supported analysis. It contains 34 workflows spanning eleven powder-diffraction task families, including phase identification and quantification, pattern prediction, indexing, peak and microstructure analysis, structure determination, database retrieval, and Rietveld refinement. The workflows use 30 Cu K α patterns derived from two opXRD sources [14] and four neutron-diffraction examples distributed with FullProf, with 11 categorized as Medium and 23 as Hard. Each task provides the diffraction data, instrument information, and executable workspace required by the agent, while target results and scoring criteria remain evaluator-side. For detailed analysis, we report all eleven task families separately;

Each fresh, networkless workspace provides public inputs, native file/code tools, task-relevant XRD procedures, GSAS-II, FullProf, and numerical libraries. The agent must deliver `final_report.md` plus task-specific executed code and evidence. An *execution gate* checks input inspection, state change, successful execution, evidence, deliverables, and oracle isolation; an independent *physical gate* rejects numerical or crystallographic contradictions. Let $g_j = g_j^{\text{exec}} g_j^{\text{phys}} \in \{0, 1\}$ be their joint outcome. Let $\phi_{A,j}, \phi_{M,j}, \phi_{J,j} \in [0, 1]$ denote artifact integrity, task-specific numerical correctness, and evidence-grounded scientific judging, respectively. The case

and track scores are

$$S_{E,j} = 100g_j (0.15\phi_{A,j} + 0.55\phi_{M,j} + 0.30\phi_{J,j}),$$

$$S_E = \frac{1}{34} \sum_{j=1}^{34} S_{E,j}. \quad (15)$$

5.4 Aggregation and Reproducibility

We primarily report (S_Q, S_E) and use

$$S_{\text{overall}} = 0.5S_Q + 0.5S_E \quad (16)$$

All runs record trajectories, commands, artifacts, time, model turns, tool calls, and tokens; public cases enable audit, while future claims require sequestered extensions [27].

6 EXPERIMENTS

6.1 Experimental Setup

We evaluate GPT-5.6 Sol, Terra, and Luna; MiMo-V2.5-Pro and MiMo-V2.5; DeepSeek-V4-Pro and Flash; MiniMax-M2.7; Claude Haiku 4.5; and Gemini 3.1 Pro Low under the AutoXRD agent framework. On XRDBENCH-QA, each case permits at most 20 tool calls and 8,192 output tokens per turn; On XRDBENCH-E2E, each workflow permits at most 50 agent steps and 32,768 output tokens per turn; GPT-5.6 Sol supplies the scientific-judgment component after execution and physical gates and objective metrics are evaluated. The complete protocol is provided in the supplementary material.

6.2 Research Questions and Statistical Analysis

We ask: **RQ1**, which XRD capabilities current models master; **RQ2**, whether they compose into reliable workflows; **RQ3**, how ranks



Figure 3: Overall performance, ordered by the overall score. The wider, hatched bar denotes Overall, followed by XRDBench-QA and XRDBench-E2E.

and failure modes change with integration; and **RQ4**, how efficiency and component choices affect performance. Our evaluation comprises 1,340 runs across ten models, involving 22,135 model turns. All intervals use 20,000 stratified task-bootstrap replicates; model differences use paired resampling within track. Thus intervals describe task variation, not decoding stochasticity.

7 RESULTS

7.1 Scientific Capability and Workflow Integration

Figure 3 compares model performance across the two tracks and the overall benchmark. GPT-5.6 Sol and GPT-5.6 Terra lead overall with 81.09 and 79.41, followed by GPT-5.6 Luna at 68.87. The ten-model mean falls by 8.22 points, from 61.89 on XRDBENCH-QA to 53.67 on XRDBENCH-E2E, and seven models score lower on the end-to-end track. Although the tracks contain different tasks, the consistent decrease and the gate failures below show that composing scientific decisions into executable analyses remains substantially more difficult than answering bounded questions.

On XRDBENCH-QA, models are strongest at refinement-history assessment (88.0) and result acceptance (81.0), but weak at next-action selection (44.0) and quantitative phase analysis (33.0). They therefore judge completed evidence more reliably than they control coupled parameters or recover phase fractions. All models miss the high-risk occupancy/zero-cell coupling cases; nine also accept a lower- R_{wp} state that introduces an unexplained peak, allowing a familiar scalar metric to override contradictory residual evidence.

End-to-end workflows expose different bottlenecks. Figure 5 contrasts workflow validity and completion with result quality across the eleven task families. Indexing is weakest (37.72) despite 90%

scoring-gate passage, 80% complete deliverables, and 86.2 artifact validity: its objective-metric score is only 26.9 and its scientific-judge score is 33.2, showing that agents often execute and document a workflow but recover inaccurate cells or HKL assignments. Rietveld refinement instead combines a low final score (48.58) with the lowest scoring-gate rate (71.25%) and only 50% complete deliverables, indicating that workflow validity and evidence completion are major bottlenecks in addition to result quality. Crystallite and defect/disorder analyses score only 48.30 and 47.52, although their small family sizes make these comparisons descriptive. Peak analysis passes the scoring gate in 96.7% of runs and produces complete deliverables in 90.0%, yet scores only 57.60, further separating successful execution from scientific correctness. Figure 4 places the nine XRDBench-QA capabilities and nine end-to-end reporting domains on one profile per model.

7.2 Rank Transfer, Completion, and Efficiency

XRDBench-QA and XRDBench-E2E scores are moderately correlated (Pearson $r = 0.557$; Spearman $\rho = 0.648$), but rankings change materially. DeepSeek-V4-Pro and Gemini 3.1 Pro Low gain 16.46 and 20.58 points and rise to third and fifth on XRDBench-E2E, whereas MiMo-V2.5-Pro, MiMo-V2.5, and DeepSeek-V4-Flash lose 21.8–27.6 points. These reversals show that diagnostic scores do not fully predict autonomous workflow performance. The overall estimates of Sol (81.09) and Terra (79.41) differ by 1.68 points with paired 95% interval $[-1.46, 4.80]$; neither is significantly ahead.

Figure 7 compares performance with resource use across all 134 tasks per model. Terra approaches Sol’s score while using 298.7K rather than 349.2K tokens and costing \$0.68 rather than \$1.97 per task. Luna offers the strongest score–cost trade-off (68.87 at \$0.07),

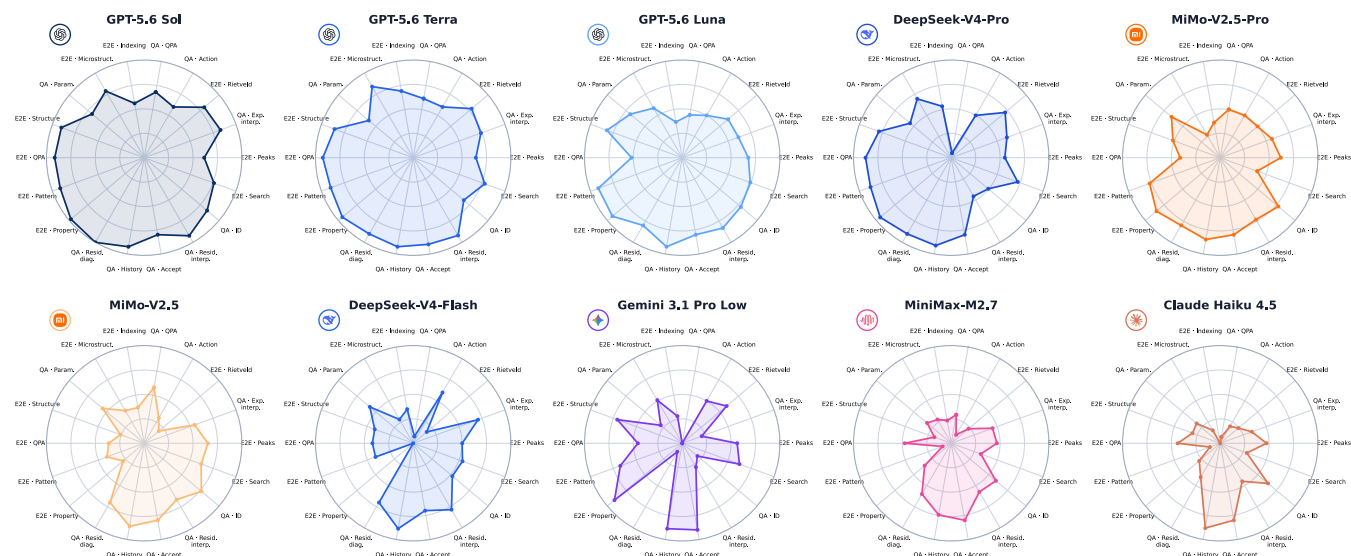


Figure 4: Fine-grained performance across 18 dimensions: nine XRDBench-QA capabilities and nine end-to-end reporting domains. Each model has a separate 0–100 radar; numeric radial labels are omitted.

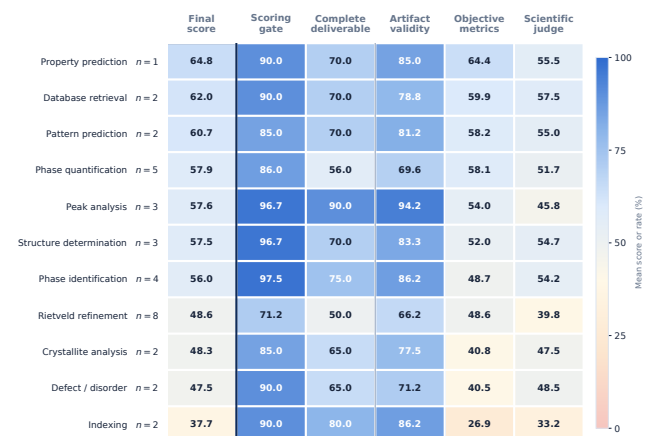


Figure 5: XRDBench-E2E bottlenecks across eleven task families, averaged over ten models. Rows are ordered by final score; gate and completion columns are pass rates, while the remaining columns are effective 0–100 components.

but its 435-second duration exceeds Terra’s 239 seconds, showing that monetary and latency efficiency need not coincide.

7.3 Failure Analysis

Trace inspection exposes three recurring failure types in end-to-end workflows. First, *scientific outcome failures* occur after genuine execution: indexing often produces inaccurate cells or HKL assignments despite passing the gate, while refinement may improve fit statistics without a valid structural result. Second, *execution and evidence failures* prevent scoring or completion: 46 of 340 runs (13.5%) fail at least one execution or physical gate, while 117 (34.4%) fail

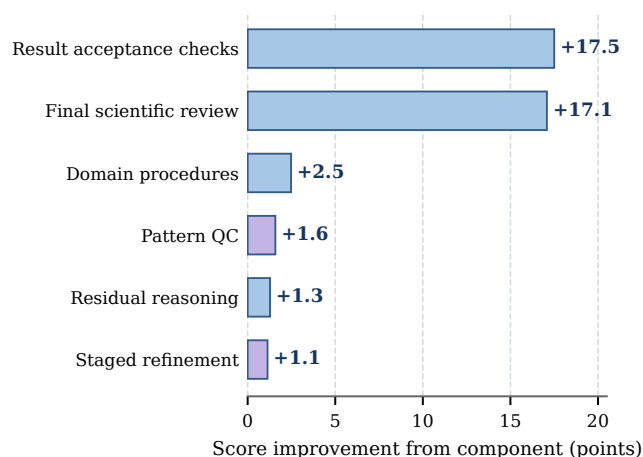


Figure 6: Component ablations on XRDBench-QA. Bars show the full-system score minus component removal.

at least one full-completion requirement, commonly because code, backend outputs, or required evidence is missing or invalid. Third, *termination failures* consume almost the full 50-step budget without producing a valid deliverable. These traces explain why aggregate QA competence transfers only partially to autonomous workflows and separate runnable-but-wrong science from incomplete execution.

7.4 Ablation and Component Analysis

Figure 6 summarizes the component effects on GPT-5.6 Luna. Each of the six tested components improves downstream performance. Result-acceptance checks and final scientific review yield the largest

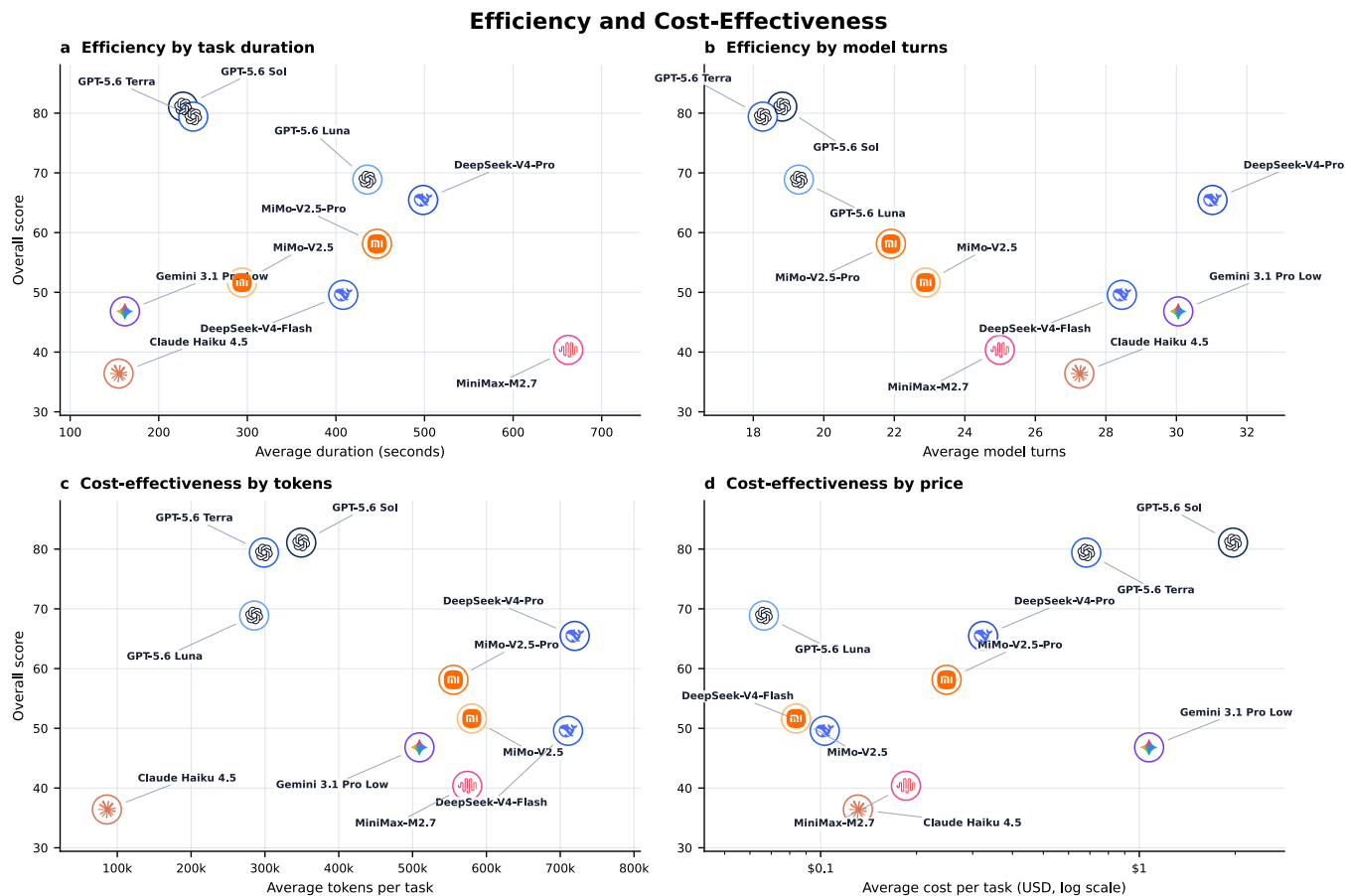


Figure 7: Full-benchmark efficiency and cost-effectiveness. Resource means pool all 134 task runs per model; the vertical axis is the overall score.

gains (17.5 and 17.1 points). Domain procedures add 2.5 points, while pattern quality control, residual reasoning, and staged refinement add 1.1–1.6 points. The consistent gains support combining procedural guidance with explicit verification.

8 DISCUSSION

XRDBench-QA and XRDBench-E2E expose complementary failure modes. The former diagnoses whether a model recognizes a valid action or scientific outcome; the latter adds implementation, backend operation, evidence preservation, physical validation, and delivery. Low Rietveld joint gate passage indicates incomplete or physically invalid workflows, whereas low Indexing accuracy despite high joint gate passage indicates incorrect science after genuine execution. The former calls for stronger execution contracts and stopping controls; the latter for better candidate generation and uncertainty-aware algorithms.

Even the strongest models occasionally accept an invalid refinement because a single fit statistic improves, despite clear deterioration in the residual pattern. In end-to-end workflows, models may also produce convincing reports without retaining the output files and intermediate results needed to verify their conclusions. These

findings show that reliable XRD agents must evaluate numerical fit, physical validity, and reproducible execution evidence together. Accordingly, XRD-agent evaluation should report not only final performance, but also execution success, physical-check passage, artifact completeness, and computational cost.

9 CONCLUSION

AutoXRD combines LLM planning, crystallographic tools, and deterministic validation; XRDBench evaluates diagnostic reasoning and end-to-end workflows. Across ten models and 1,340 runs, diagnostic competence only moderately predicts autonomous performance: indexing is executable but inaccurate, while Rietveld refinement is limited by completion and validity. Reliable diffraction agents must expose the evidence and invariants supporting each conclusion, not merely produce plausible reports.

ETHICAL CONSIDERATIONS

Incorrect autonomous diffraction conclusions can affect materials decisions and downstream experiments. AutoXRD is decision support rather than expert certification: it exposes uncertainty and

validation failures, while users remain responsible for provenance, safety, calibration, licensing, and independent scientific review.

REFERENCES

- [1] A. Le Bail, H. Duroy, and J.L. Fourquet. 1988. Ab-initio structure determination of LiSbWO₆ by X-ray powder diffraction. *Materials Research Bulletin* 23, 3 (1988), 447–452. [https://doi.org/10.1016/0025-5408\(88\)90019-0](https://doi.org/10.1016/0025-5408(88)90019-0) Verified by Crossref.
- [2] Danil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous Chemical Research with Large Language Models. *Nature* 624 (2023), 570–578. <https://doi.org/10.1038/s41586-023-06792-0> Potential Hallucination.
- [3] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. 2024. Augmenting Large Language Models with Chemistry Tools. *Nature Machine Intelligence* 6 (2024), 525–535. <https://doi.org/10.1038/s42256-024-00832-8> Ambiguous match: Augmenting Large Language Models for Automated Discovery of Element Extractants.
- [4] Charles Rhys Campbell, Justin Ely, Jaehyung Lee, Frank M. Abel, and Kamal Choudhary. 2026. Hybrid DiffractGPT–Rietveld Refinement Framework for Automated X-ray Diffraction Analysis. arXiv:2607.08890 [cond-mat.mtrl-sci] Potential Hallucination.
- [5] Bin Cao, Yang Liu, Zinan Zheng, Ruifeng Tan, Jia Li, and Tong-Yi Zhang. 2025. SimXRD-4M: Big Simulated X-ray Diffraction Data and Crystal Symmetry Classification Benchmark. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=mkuB677eMM> Potential Hallucination.
- [6] Ming-Chiang Chang, Sebastian Ament, Maximilian Amsler, Duncan R. Sutherland, Lan Zhou, John M. Gregoire, Carla P. Gomes, R. Bruce van Dover, and Michael O. Thompson. 2025. Probabilistic phase labeling and lattice refinement for autonomous materials research. *npj Computational Materials* 11 (2025). <https://doi.org/10.1038/s41524-025-01627-0> Verified by Crossref.
- [7] Ziru Chen, Shijie Chen, Yuting Ning, et al. 2025. ScienceAgentBench: Toward Rigorous Assessment of Language Agents for Data-Driven Scientific Discovery. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=6z4YKrOGK6> Ambiguous match: Adaptive agents for information retrieval and data-driven knowledge discovery.
- [8] Xiaopeng Cui, Zhenjie Feng, Yuan Jin, Yiming Cao, Dongmei Deng, Hao Chu, Shixun Cao, Cheng Dong, and Jincang Zhang. 2015. <i>-i>AutoFP<i>-i>: a GUI for highly automated Rietveld refinement using an expert system algorithm based on<i>-i>FullProf<i>-i>. *Journal of Applied Crystallography* 48 (2015), 1581–1586. <https://doi.org/10.1107/S1600576715015125> Verified by Crossref.
- [9] W. A. Dollase. 1986. Correction of intensities for preferred orientation in powder diffractometry: application of the March model. *Journal of Applied Crystallography* 19 (1986), 267–272. <https://doi.org/10.1107/S0021889886089458> Verified by Crossref.
- [10] Yuxing Fei, Matthew J. McDermott, Christopher L. Rom, Shilong Wang, and Gerbrand Ceder. 2026. Dara: Automated Multiple-Hypothesis Phase Identification and Refinement from Powder X-ray Diffraction. *Chemistry of Materials* 38, 3 (2026), 1364–1376. <https://doi.org/10.1021/acs.chemmater.5c02820> Verified by Crossref.
- [11] Zhenjie Feng, Qiang Hou, Yonglei Zheng, Wei Ren, Jun-Yi Ge, Tao Li, Cheng Cheng, Wencong Lu, Shixun Cao, Jincang Zhang, and Tongyi Zhang. 2019. Method of artificial intelligence algorithm to improve the automation level of Rietveld refinement. *Computational Materials Science* 156 (2019), 310–314. <https://doi.org/10.1016/j.commatsci.2018.10.006> Verified by Crossref.
- [12] Saulius Gražulis, Adriana Daškevič, Andrius Merkys, Daniel Chateigner, Luca Lutterotti, Miguel Quirós, Nadezhda R. Serebryanaya, Peter Moeck, Robert T. Downs, and Armel Le Bail. 2012. Crystallography Open Database (COD): an open-access collection of crystal structures and platform for world-wide collaboration. *Nucleic Acids Research* 40, D1 (2012), D420–D427. <https://doi.org/10.1093/nar/gkr900> Verified by Crossref.
- [13] R. J. Hill and C. J. Howard. 1987. Quantitative phase analysis from neutron powder diffraction data using the Rietveld method. *Journal of Applied Crystallography* 20, 6 (1987), 467–474. <https://doi.org/10.1107/S0021889887086199> Verified by Crossref.
- [14] Daniel Hollarek, Henrik Schoppmans, Jona Östreicher, Jonas Teufel, Bin Cao, Adie Alwen, Simon Schweidler, Miganka Singh, Tim Kodalle, Hanlin Hu, Gregoire Heymans, Maged Abdelsamie, Arthur Hardiagan, Alexander Wiecezorek, Siarhei Zhuk, Ruth Schwaiger, Sebastian Siol, François-Xavier Couder, Moritz Wolf, Carolin M. Sutter-Fella, Ben Breitung, Andrea M. Hodge, Tong-yi Zhang, and Pascal Friederich. 2026. opXRD: Open Experimental Powder X-Ray Diffraction Database. *Advanced Intelligent Discovery* (2026). <https://doi.org/10.1002/aidi.202500044> Verified by Crossref.
- [15] Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2024. MAgentBench: Evaluating Language Agents on Machine Learning Experimentation. In *Proceedings of the 41st International Conference on Machine Learning*, Vol. 235. PMLR, 20271–20309. Ambiguous match: Conversational Agents, Natural Language Processing, and Machine Learning for Psychotherapy.
- [16] Jin-Woong Lee, Woon Bae Park, Jin Hee Lee, Satendra Pal Singh, and Kee-Sun Sohn. 2020. A deep-learning technique for phase identification in multiphase inorganic compounds using synthetic XRD powder patterns. *Nature Communications* 11 (2020), 86. <https://doi.org/10.1038/s41467-019-13749-3> Verified by Crossref.
- [17] Qingmeng Li, Hao Wang, Dongbo Xiong, Jiajun Zhong, Wenhai Ji, et al. 2026. Rongzai Agent: A Large Language Model-Based Autonomous Assistant for Rietveld Refinement of Neutron Diffraction Data. arXiv:2605.13911 [cond-mat.mtrl-sci] Potential Hallucination.
- [18] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Yuxian Gu, Han Ding, Kai Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Shengqi Shen, Tianjun Zhang, Sheng Shen, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. AgentBench: Evaluating LLMs as Agents, In *International Conference on Learning Representations*. *International Conference on Learning Representations*. <https://openreview.net/forum?id=zADUB0aCTQ> Verified by Semantic Scholar.
- [19] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. arXiv:2408.06292 [cs.AI]
- [20] Yubo Ma, Zhibin Gou, Junheng Hao, Ruochen Xu, Shuohang Wang, Liangming Pan, Yujun Yang, Yixin Cao, and Aixin Sun. 2024. SciAgent: Tool-augmented Language Models for Scientific Reasoning, In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 15701–15723. <https://doi.org/10.18653/v1/2024.emnlp-main.880> Verified by Crossref.
- [21] Ian C. Madsen, Nicola V. Y. Scarlett, Lachlan M. D. Cranswick, and Thang Lwin. 2001. Outcomes of the International Union of Crystallography Commission on Powder Diffraction Round Robin on Quantitative Phase Analysis: samples 1<i>-i> to 1<i>-i>. *Journal of Applied Crystallography* 34, 4 (2001), 409–426. <https://doi.org/10.1107/S0021889801007476> Verified by Crossref.
- [22] Suk Jin Mun, Yoonsoo Nam, and Sungkyun Choi. 2026. Automation of Rietveld refinement through machine learning. *Journal of Applied Crystallography* 59, 2 (2026), 564–577. <https://doi.org/10.1107/S1600576726001494> Verified by Crossref.
- [23] Yoshihiko Ozaki, Yuta Suzuki, Takafumi Hawaii, Kotaro Saito, Masaki Onishi, and Kanta Ono. 2020. Automated crystal structure analysis based on blackbox optimisation. *npj Computational Materials* 6 (2020), 75. <https://doi.org/10.1038/s41524-020-0330-9> Verified by Crossref.
- [24] H. M. Rietveld. 1969. A profile refinement method for nuclear and magnetic structures. *Journal of Applied Crystallography* 2, 2 (1969), 65–71. <https://doi.org/10.1107/S0021889869000658> Verified by Crossref.
- [25] Sergio Rincón, Gabriel González, Mario A. Macías, and Pablo Arbeláez. 2025. A new benchmark for machine learning applied to powder X-ray diffraction. *Scientific Data* 12 (2025), 1186. <https://doi.org/10.1038/s41597-025-05534-3> Verified by Crossref.
- [26] Juan Rodríguez-Carvajal. 1993. Recent advances in magnetic structure determination by neutron powder diffraction. *Physica B: Condensed Matter* 192, 1–2 (1993), 55–69. [https://doi.org/10.1016/0921-4526\(93\)90108-1](https://doi.org/10.1016/0921-4526(93)90108-1) Verified by Crossref.
- [27] Oscar Sainz, Jon Campos, Iker García-Ferrero, Julien Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP Evaluation in trouble: On the Need to Measure LLM Data Contamination for each Benchmark. arXiv:2310.18018 [cs.CL] Verified by Crossref.
- [28] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools, In *Advances in Neural Information Processing Systems*. *Advances in Neural Information Processing Systems* 36 36, 68539–68551. Verified by Crossref.
- [29] Jan Schuetzke, Alexander Benedix, Ralf Mikut, and Markus Reischl. 2021. Enhancing deep-learning training for phase identification in powder X-ray diffractograms. *IUCrJ* 8 (2021), 408–420. <https://doi.org/10.1107/S2052252521002402> Verified by Crossref.
- [30] Woong Shin, Craig A. Bridges, Marshall T. McDonnell, and Rafael Ferreira da Silva. 2026. Fantastic Scientific Agents and How to Build Them: AgentBuild for Rietveld Refinement. arXiv:2606.12834 [cs.AI] Ambiguous match: Fantastic Agents and How to Build Them.
- [31] Tatsunori Tanai et al. 2026. Automated Multiphase Identification and Refinement in Powder Diffraction Using Mismatch-Tolerant Machine Learning. arXiv:2605.12478 [cond-mat.mtrl-sci] Ambiguous match: Dara: Automated Multiple-Hypothesis Phase Identification and Refinement from Powder X-ray Diffraction.
- [32] P. Thompson, D. E. Cox, and J. B. Hastings. 1987. Rietveld Refinement of Debye–Scherrer Synchrotron X-ray Data from Al₂O₃. *Journal of Applied Crystallography* 20 (1987), 79–83. <https://doi.org/10.1107/S0021889887087090> Ambiguous match: Rietveld refinement of Debye–Scherrer synchrotron X-ray data from Al₂O₃.
- [33] P. Tian, W. Zhou, J. Liu, Y. Shang, C. L. Farrow, P. Juhás, and S. J. L. Billinge. 2013. <i>-i>SrRietveld<i>-i>: a program for automating Rietveld refinements for high-throughput powder diffraction studies. *Journal of Applied Crystallography* 46 (2013), 255–258. <https://doi.org/10.1107/S0021889812045967> Verified by Crossref.
- [34] Brian H. Toby. 2006. R Factors in Rietveld Analysis: How Good Is Good Enough? *Powder Diffraction* 21, 1 (2006), 67–70. <https://doi.org/10.1154/1.2179804> Ambiguous match: <i>-i>R<i>-i> factors in Rietveld analysis: How good is good enough?
- [35] B. H. Toby. 2024. A simple solution to the Rietveld refinement recipe problem. *Journal of Applied Crystallography* 57 (2024), 175–180. <https://doi.org/10.1107/S1600576723011032> Verified by Crossref.
- [36] Brian H. Toby and Robert B. Von Dreele. 2013. <i>-i>GSAS-II<i>-i>: the genesis of a modern open-source all purpose crystallography software package. *Journal of Applied Crystallography* 46, 2 (2013), 544–549. <https://doi.org/10.1107/S0021889813003531> Verified by Crossref.
- [37] Honggang Wang, Mingwei Chen, Hongjie Luo, Yin Wen, Fazhi Qi, Xuqing Chai, and Fengyao Hou. 2026. MatDiffact: A Material-Informed Automated Analysis Platform for X-ray Powder Diffraction. arXiv:2607.20880 [cond-mat.mtrl-sci] Potential Hallucination.
- [38] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. r-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains, In *International Conference on Learning Representations*. arXiv.org. <https://openreview.net/forum?id=roNSXZpUDN> Verified by Semantic Scholar.
- [39] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafraan, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=WE_vluYUL-X Potential Hallucination.
- [40] Yuto Yotsumoto, Yusaku Nakajima, Ryusei Takamoto, Yasuo Takeichi, and Kanta Ono. 2024. Autonomous robotic experimentation system for powder X-ray diffraction. *Digital Discovery* 3 (2024), 2523–2532. <https://doi.org/10.1039/D4DD00190G> Verified by Crossref.
- [41] Robert A. Young (Ed.). 1993. *The Rietveld Method*. Oxford University Press. <https://doi.org/10.1093/oso/9780198555773.001.0001> Ambiguous match: Introduction to the Rietveld method.