

Harness Engineered LLM Agents for Material Science: A Case Study on Perovskite Materials with Machine Learning and Materials Project

Lei Zhang ^{1,2,*} and Tianle Wang ¹

¹ Key Laboratory of Spectral Imaging Technology, Xi'an Institute of Optics and Precision Mechanics of the Chinese Academy of Sciences, Xi'an 710119, China

² Department of Internet Engineering, School of Software, Nanjing University of Information Science & Technology, Nanjing 210044, China

* Correspondence: 002699@nuist.edu.cn

How To Cite: Zhang, L.; Wang, T. Harness Engineered LLM Agents for Material Science: A Case Study on Perovskite Materials with Machine Learning and Materials Project. *AI for Materials* **2026**, *1*(1), 13. <https://doi.org/10.53941/aimat.2026.100013>

Received: 9 June 2026

Revised: 17 August 2026

Accepted: 4 September 2026

Published: 11 September 2026

Abstract: The integration of large-language-model (LLM) agents into materials science requires a balance between adaptive reasoning and reliable scientific execution. Here, we present a Harness framework that converts agent proposals into structured and validated actions, connects them to the Materials Project and deterministic machine-learning tools, and records settings and computational evidence. The present perovskite band-gap case study uses an expert-defined oxide/halide partition as an initial subclass strategy; it does not claim autonomous discovery of an optimal material taxonomy. Fourteen regressors are compared under pooled and subclass-specific training using identical test entries across 50 matched stratified splits. Subclass-specific modeling directly improves several linear, nearest-neighbor, and ensemble regressors, but the effect is model dependent and is negative for SVR and LightGBM. A direct training-only K-means baseline produces a different partition; the chemical rule performs better for several regressors, whereas K-means performs better for LightGBM. In the current implementation, Harness provides the reproducible execution pathway for evaluating this scientific hypothesis. More broadly, the framework is designed to compare and refine candidate subclasses, descriptors, models, and physical-validation tools using downstream performance feedback. This work therefore provides a foundational demonstration of adaptive, tool-integrated workflow construction for materials science.

Keywords: harness engineering; agent; large language model; LLM; perovskite materials

1. Introduction

The convergence of data-driven approaches and materials science has reshaped the research landscape [1–3]. Materials informatics applies data science and machine learning techniques to materials data for property prediction, pattern discovery, and accelerated materials design [4–6]. In addition, the Materials Genome concept [7–9] inspired by the Human Genome Project envisions the acceleration of materials discovery through high-throughput computation, databases, and data-enabled prediction [10,11]. Structural databases such as the Materials Project [12,13], the Open Quantum Materials Database (OQMD) [14], the Automatic Flow for Materials Discovery (AFLOW) [15], 2DMatPedia [16], and the Joint Automated Repository for Various Integrated Simulations (JARVIS) [17] enable high-throughput screening of candidate materials [18–20]. In addition, Machine learning has become an accelerating



Copyright: © 2026 by the authors. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Publisher's Note: Scilight stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

force in this ecosystem [21,22], with transfer learning, inverse design through generative models such as variational autoencoders, and genetic algorithms for synthesis optimization integrated into existing workflows [23,24]. On the other hand, AI for science, as a broader paradigm, seeks to embed artificial intelligence into the full research lifecycle, from hypothesis generation to experiment design and data interpretation [25–27].

Large language models (LLMs) are a class of autoregressive models trained on massive text corpora to process, generate, and reason with natural language [28–30]. Notable examples include ChatGPT developed by OpenAI, Claude developed by Anthropic, DeepSeek developed by DeepSeek Company, and the LLaMA series developed by Meta [31,32]. These models have demonstrated capabilities in code generation, context understanding, multi-step reasoning [33–35], and tool use, which make them suitable as reasoning engines for autonomous agents. Furthermore, AI agents are autonomous or semi-autonomous systems that use LLMs as their reasoning core to perceive environments, plan actions, execute tasks, and adapt to feedback [36–38]. The tool-use capability distinguishes agents from standalone language models, as it allows agents to interact with software, databases, and hardware. In the context of computational materials science, tools can include density functional theory [39] codes such as VASP, CASTEP, and Quantum ESPRESSO, molecular dynamics packages, machine learning interatomic potential libraries such as MACE, DeepMD, and SevenNet, materials databases such as the Materials Project, OQMD, AFLOW, 2DMPedia, and JARVIS, and analysis libraries such as Pymatgen and ASE. The materials agent list for materials modeling includes ChemCrow [40], MDAgent [41], DREAMS [42], VASPilot [43], El Agente [44], MatSciAgent [39], Crystalyse [45], MOFGen [46], CrossMatAgent [47], Masgent [48], MatLLMSearch [49], CrystalPlasticitySim [50], and T2MAT [51], to name a few. For example, ChemCrow employs a chain-of-thought reasoning loop to plan and execute chemistry tasks using 18 specialized tools [40]. LLaMP is designed to use hierarchical retrieval-augmented generation to query the Materials Project and run simulations for property predictions [52]. HoneyComb combines a curated knowledge base called MatSciKB with inductive tool creation, outperforming baseline models [53]. LLaMat is a domain-adapted foundation model for materials science, and its LLaMat-CIF variant demonstrates tool-integrated crystal structure generation when paired with CrystaLLM, achieving high coverage across the periodic table [2]. MatSciAgent deploys a master agent that delegates data retrieval, crystal generation, and simulation tasks to specialized agents using the Materials Project and MatWeb [39]. Crystalyse is a multi-tool agent that integrates compositional screening, structure generation, and force-field evaluation with provenance to eliminate hallucinations [45]. The agent-based learning framework Eunomia uses tools such as online dataset search and document search to extract information [54]. LLMatDesign translates natural language into API calls and machine learning tool invocations for iterative material modification [55]. Challenges persist in alignment between generation intention and physical feasibility, data scarcity, and multi-scale modeling, emphasizing the need of materials AI agent for deeper integration of physical constraints [56,57]. Related materials-science applications include LLM-assisted evidence auditing of solid-state electrolyte data [58] and collaborative AI for image interpretation [59].

However, a critical bottleneck for AI agents in materials research is the tension between flexibility and reliability. Fully autonomous agents can propose unsupported or physically inconsistent actions, whereas fully hardcoded pipelines are reliable but difficult to adapt to new tasks. Harness engineering addresses this gap by defining the executable layer between LLM-generated proposals and scientific tools. In this work, Harness is not treated as prompt wording or passive logging. It structures proposed actions, validates inputs and tool availability, routes accepted actions to deterministic software, records evidence, and can expose downstream model performance as feedback for later workflow refinement [60]. Related scientific-agent studies likewise emphasize grounded tool use, domain constraints, verification, provenance, and recovery from failed actions rather than unconstrained text generation [61–64].

In this study, we apply Harness engineering to a materials-informatics case study on perovskite band-gap prediction (Figure 1). The Materials Project is selected because it provides standardized, provenance-linked crystal structures and computed properties through reproducible API access [12,13,65]. Within the workflow, these records ground agent responses in identifiable materials, computed values, and calculation metadata, while the retrieved band gaps are explicitly treated as database-computed rather than experimental measurements. The quantitative comparison uses a chemically defined oxide/halide partition as an initial scientific hypothesis. Pooled and subclass-specific regressors are evaluated under matched data, descriptors, model settings, and test entries. Subclass-specific modeling is the direct mechanism behind the observed numerical changes. Harness provides the structured route for evaluating that strategy and establishes a framework in which alternative partitions, descriptors, models, and validation tools can later be compared and refined using performance feedback. The present work is therefore a framework demonstration, not a claim that the LLM has already discovered the optimal taxonomy.

Database band-gap screening is only an initial filter because it does not test whether a candidate remains geometrically and thermodynamically stable after relaxation or at finite temperature. Machine-learning interatomic

potentials (MLIPs), including universal models and system-specific potentials trained by on-the-fly Gaussian process regression, provide a practical route to structural relaxation and molecular dynamics, while active learning and model committees can flag configurations that require new first-principles labels [66–71]. The broader relationship among data-driven screening, atomistic accuracy, and energy-materials validation is reviewed by Kim [72]. In a fuller Harness workflow, the LLM would not decide that a structure is physically valid. Instead, a shortlisted material would be routed to an MLIP relaxation or molecular-dynamics tool, checked against predefined convergence and structural criteria, and escalated to density functional theory (DFT) when committee disagreement or another out-of-distribution score exceeds a validated threshold. This physical-validation loop is proposed as an extension; it was not executed in the present band-gap case study.

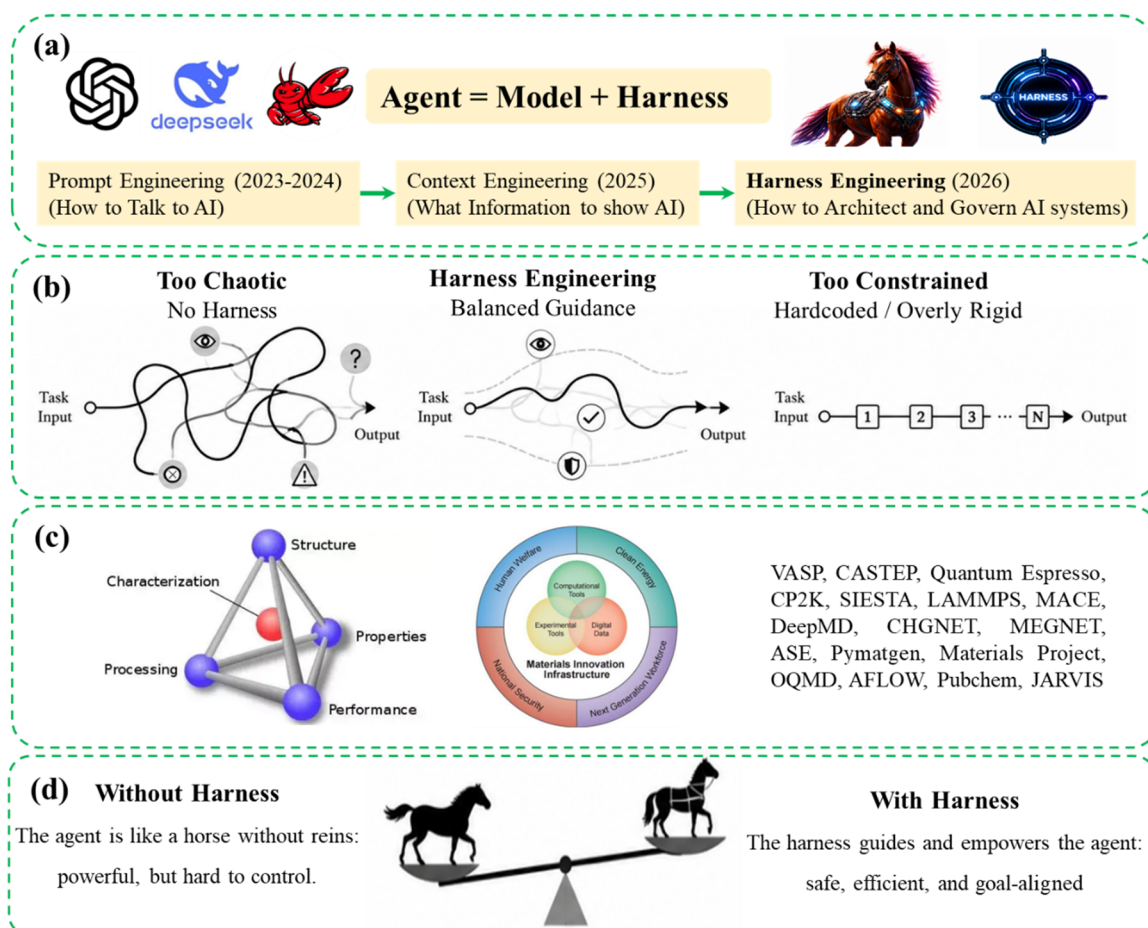


Figure 1. Schematic illustration of the Harness paradigm for materials science and the broader AI domain. **(a)** Evolution from prompt engineering (2023–2024) to context engineering (2025) to Harness engineering (2026), following the paradigm Agent = Model + Harness. **(b)** Comparison of three approaches related to Harness concept for AI agent. Left: an agent without Harness relying entirely on the LLM, leading to chaotic and unpredictable behavior. Right: a fully hardcoded pipeline with no LLM involvement, which is too rigid to adapt to novel scenarios. Middle: the harnessed agent balances LLM-driven flexibility with structured constraints. **(c)** The materials science tetrahedron (composition, structure, processing, properties) and the Materials Genome framework, with typical computational materials science tools listed as examples. **(d)** Comparison between an agent without Harness and an agent with Harness. The Harness guides and empowers the agent, ensuring safe, efficient, and goal-aligned operation that is important in AI for materials.

2. Methods

2.1. Agent Framework

Figure 2 summarizes the workflow and matched comparison. The agent workflow is driven by Claude Sonnet 4 (Anthropic), accessed through the Messages API with temperature = 0.2 and a maximum of 2048 tokens per call. Five agent roles operate at research design, taxonomy proposal and initialization, descriptor recommendation, adaptive scheduling, and result diagnosis (Table 1). Each agent receives a structured system prompt and a JSON-formatted user prompt. Free-form text is not executed directly: proposed actions must conform to predefined fields

and are checked against the data, descriptors, models, and tools available to the workflow. Stage 1: Task decomposition. The LLM translates a natural-language research goal into a candidate material query, a candidate taxonomy or routing strategy, and a scientific rationale. For the reported quantitative benchmark, the oxide/halide composition rule is supplied as the initial validated partition; candidate LLM taxonomies are not credited with the reported R^2 changes. Stage 2: Data acquisition. Candidate formulae are submitted to the Materials Project API. The workflow retrieves crystallographic and computed electronic-structure properties, and removes entries with energy above hull greater than 0.15 eV or non-positive band gap values under the stated screening criteria. Stage 3: Taxonomy initialization and routing. The benchmark assigns 78 oxide and 51 halide records using an explicit chemical composition rule. The validated label routes each sample to the corresponding specialist model. For a direct unsupervised baseline, K-means with $k = 2$ and 20 initializations is fitted only to standardized training descriptors in each split. Test records are assigned to the nearest training-derived centroid, and the same regressor configuration is fitted separately within each cluster. In a future adaptive implementation, alternative partitions may be proposed using clustering, learned representations, hierarchical rules, or agent-generated candidates and then scored using matched predictive feedback. Stage 4: Descriptor design and recommendation. Fourteen physical descriptors are computed, including volume, density, symmetry, formation energy, magnetization, and the Goldschmidt tolerance factor, octahedral factor, and A/B-site radius ratio derived from Shannon ionic radii. The agent may recommend candidate subsets, but the pooled and subclass routes in the quantitative comparison use the same validated descriptor set so that the effect of material-space partitioning is isolated. Stage 5: Pooled and subclass-specific training. Fourteen regressors spanning linear, tree, ensemble, kernel, nearest-neighbor, neural-network, and gradient-boosted-tree families are trained with identical configurations in the pooled and subclass routes. For the representative single-split analyses discussed in Section 3, the sample-count-weighted subclass coefficient of determination is defined as:

$$R_{\text{harness}}^2 = \frac{\sum_s n_s R_s^2}{\sum_s n_s}$$

where n_s and R_s^2 are the test-set sample count and coefficient of determination for subclass s , respectively. This weighted metric is used only for the representative single-split analyses discussed in Section 3. In the 50-split analysis, predictions from the specialist models are recombined over the complete matched test set and a global R^2 is calculated for both routes. All 14 regressors are executed in every reported split; the adaptive scheduler does not early-stop or tune the benchmark. In interactive use, the scheduler may propose whether to continue, tune, stop, or evaluate a different workflow configuration, but no test-set-guided tuning is used here. Stage 6: Result diagnosis. The final agent summarizes the stored material identifiers, settings, predictions, metrics, subclass behavior, and limitations. Interpretations are presented as evidence-linked hypotheses rather than as replacements for physical validation.

Table 1. Agent modules and their functions.

Module	Function
Module 1: Task Decomposition	Produces a structured material query and candidate workflow plan.
Module 2: Taxonomy Proposal, Initialization, and Evaluation	Uses the expert oxide/halide rule for the current benchmark and supports future generation and feedback-based evaluation of alternative partitions.
Module 3: Descriptor Recommendation	Recommends descriptors from a validated pool; matched quantitative routes use the same descriptor set.
Module 4: Result Diagnosis	Interprets stored prediction patterns and returns evidence-linked, physics-informed hypotheses and limitations.
Module 5: Adaptive Scheduling and Feedback	Proposes whether to continue, tune, stop, or test another configuration; no test-guided tuning is used in the reported benchmark.

2.2. Agent Tools

Scikit-learn provides feature standardization, data partitioning, supervised regressors, and the R^2 and mean absolute error metrics (Table 2). Fourteen regressors are evaluated. Ridge uses an L2 penalty of 0.5; Lasso uses an L1 penalty of 0.01 with a maximum of 10,000 iterations; and ordinary least squares is the unregularized linear baseline. Decision Tree uses maximum depth 8 and minimum 2 samples per leaf. Random Forest and Extra Trees use 300 estimators, maximum depth 12, and minimum 2 samples per leaf. Gradient Boosting uses 200 estimators, learning rate 0.05, maximum depth 5, subsampling 0.8, and minimum 2 samples per leaf. AdaBoost uses 100 estimators and learning rate 0.1. SVR uses an RBF kernel with $C = 15$ and $\gamma = 0.05$. KNN uses 7 neighbors with

distance weighting. The MLP contains one hidden layer of 50 ReLU units and is trained with Adam for at most 500 iterations. XGBoost, LightGBM, and CatBoost use 200 estimators, maximum depth 5, and learning rate 0.05 when available. These values are predefined configurations used in this study; they are not claimed to be software defaults or globally optimal settings. They were fixed before the repeated analysis and were not selected by examining test-set performance. The same configuration is used for the pooled model and the corresponding specialist models. Model random state is fixed at 42 where supported, while the split seed is varied to generate 50 stratified 80/20 partitions. Within each split, pooled and subclass routes use the same test entries. Standardization is fitted only on the training data and then applied to the matched test data. K-means uses $k = 2$, $n_init = 20$, and $random_state = 42$. The expert-rule and K-means routes are compared using the same bootstrap interval, paired Wilcoxon test, and Holm adjustment as the pooled comparison. Empirical 95% confidence intervals are calculated from 5000 bootstrap resamples of the 50 paired ΔR^2 values, and paired Wilcoxon tests are adjusted across the 14 regressors using the Holm procedure.

Table 2. Tools used in the agent framework.

Tools	Descriptions
Materials Project API	Crystallographic and electronic-structure data: e.g., band gap, symmetry, volume, density, formation energy, magnetic ordering.
Machine Learning Algorithms	Scikit-learn and XGBoost/LightGBM/CatBoost (11 + 3 supervised regressors): linear, tree, ensemble, kernel, nearest-neighbor, neural network, and gradient-boosted trees.
Shannon Ionic Radii	Reference table of 30+ ionic radii (Shannon) for computing perovskite Structural descriptors: Goldschmidt tolerance factor, octahedral factor, and A/B-site radius ratio.
NumPy/Pandas/Matplotlib	Numerical computation, data management, and publication-quality visualization.

The Materials Project API is accessed through the MPRester client. The query retrieves material identifiers, formulae, band gaps, energy above hull, symmetry, volume, density, formation energy per atom, number of elements, and electronic and magnetic fields. Materials Project is used as the grounding resource because its structured records keep each returned property linked to an identifiable material and computational provenance [12,13,65]. This allows Harness to validate that requested fields are available, pass them directly to descriptor and regression tools, and avoid substituting unsupported LLM estimates for database records. The values used in this study are computed database properties and are not presented as experimental measurements. Perovskite geometric descriptors are calculated from Shannon ionic radii. The Goldschmidt tolerance factor is $t = (r_A + r_O)/[\sqrt{2}(r_B + r_O)]$, the octahedral factor is $\mu = r_B/r_O$, and the A/B-site radius ratio is r_A/r_B , where $r_O = 1.40 \text{ \AA}$ for oxygen in the descriptor convention used here.

2.3. Harness Engineering Principles

Harness is implemented as an executable scientific-workflow framework between LLM-generated proposals and deterministic materials tools [60]. Its present functions are: (1) structured proposal handling, in which candidate queries, taxonomies, descriptor sets, model choices, and schedules are represented in machine-readable fields; (2) validation gates, which check schema, tool availability, allowed options, sample adequacy, and the current expert-defined oxide/halide rule before execution; (3) deterministic tool routing, through which accepted actions call the Materials Project API, preprocessing code, regressors, metrics, and visualizations; (4) evidence recording, including identifiers, prompts, settings, split seeds, outputs, statistical summaries, and stated limitations; and (5) performance-feedback exposure, which makes partition, descriptor, model, and validation choices explicit variables that can be retained, rejected, or refined. In the present study, the oxide/halide partition is supplied as the initial strategy and automatic taxonomy search is not performed. The framework contribution is therefore not an LLM-discovered classification or an additional numerical transformation after a fixed model has been specified. Its intended predictive role is to support systematic search over alternative subclass definitions and tool combinations. The present benchmark now exposes the expert chemical rule and training-only K-means as directly comparable routing choices. Future implementations may extend the same interface to learned representations, additional databases, active learning, and MLIP validation. A candidate that passes the band-gap screen could be relaxed or propagated by an MLIP; convergence, unphysical contacts, force or stress criteria, phase retention, and committee disagreement could then be evaluated by deterministic checks. High-disagreement or failed cases would be rejected or sent to DFT for labeling and later retraining, with the model version, threshold, structure, and decision recorded [66–72].

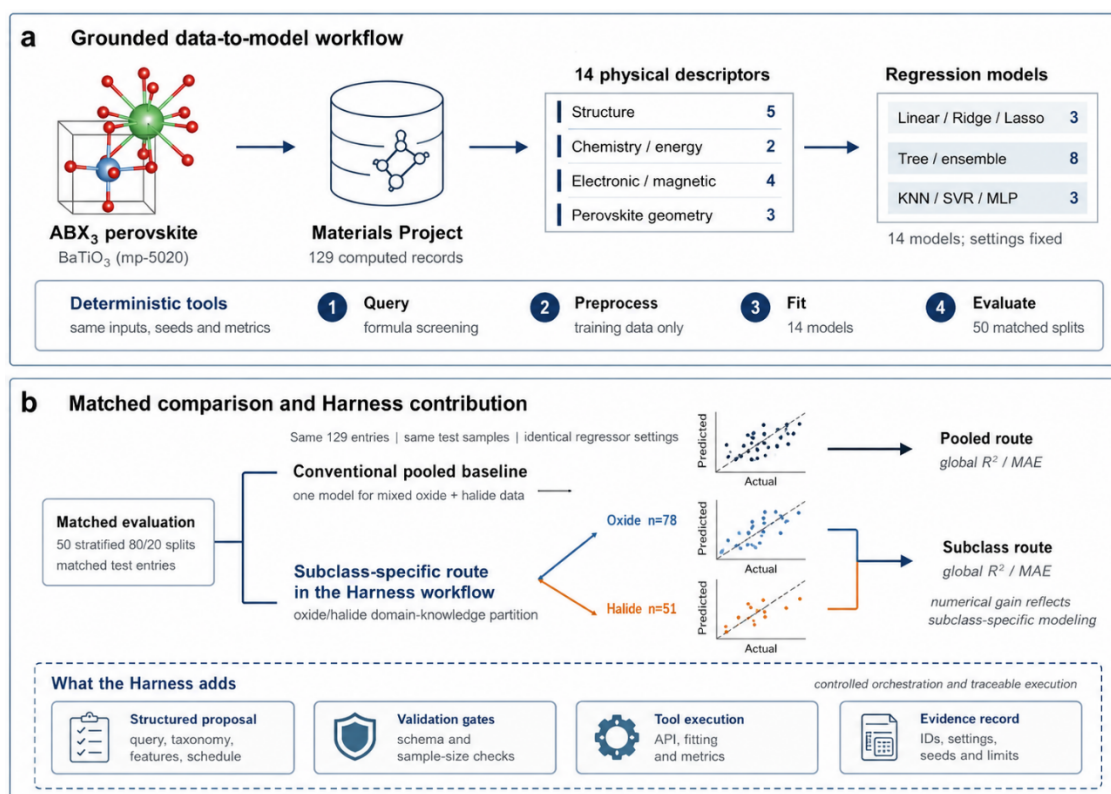


Figure 2. Grounded workflow, matched comparison, and Harness functions. **(a)** Materials Project records are converted into 14 physical descriptors and evaluated with 14 fixed-configuration regressors; query, preprocessing, fitting, and evaluation are executed by deterministic tools. **(b)** The pooled and expert-defined oxide/halide subclass routes use the same 129 entries, matched test samples, descriptor set, and regressor settings. The current numerical change therefore reflects subclass-specific modeling rather than an LLM-discovered taxonomy. The lower band summarizes the Harness functions used to structure proposals, validate actions, route tools, expose model feedback, and record evidence. The oxide/halide rule is the initial scientific strategy in this demonstration; comparison and refinement of alternative taxonomies are reserved for future work.

3. Results and Discussion

In Figures 3–6, the in-figure labels “No Harness” and “With Harness” denote the pooled route and the expert-defined oxide/halide subclass route, respectively. These are route labels only; the numerical differences arise from subclass-specific training rather than from an additional transformation performed by the Harness control layer. Figures 3–6 present a representative stratified 80/20 train-test split to illustrate model-fitting and prediction behavior; robustness across data partitions is evaluated separately using the 50 matched splits in Table 3. In that split, the expert-defined oxide/halide subclass route improves test R^2 for 11 of the 14 regressors, but the result is not universal. Gradient Boosting increases from pooled $R^2 = 0.665$ to a sample-count-weighted subclass $R^2 = 0.754$. XGBoost increases from 0.684 to 0.736. Several linear models show large relative gains but remain below the best ensemble models in absolute R^2 . Random Forest, CatBoost, and especially LightGBM do not improve in this split. These values illustrate the direct effect of the selected material partition; they do not isolate an additional LLM accuracy effect.

Gradient Boosting and XGBoost fit the training data almost perfectly under both routes, whereas their test scores are substantially lower, indicating overfitting risk in this small dataset. KNN also reaches training $R^2 = 1.000$ but generalizes poorly after the effective neighborhood is reduced by subclass partitioning. LightGBM falls from pooled test $R^2 \approx 0.39$ to subclass $R^2 \approx 0.29$ in the representative single-split analysis and shows an even stronger negative mean change in the repeated analysis. This negative case demonstrates that a chemically meaningful partition is not automatically appropriate for every regressor and must be assessed together with subclass sample size and model family.

In the present implementation, Harness evaluates the expert-defined partition under controlled and reproducible conditions: it keeps the candidate strategy explicit, applies the same inputs and model settings to the matched routes, records the resulting evidence, and exposes prediction performance for diagnosis. Harness does not change the deterministic fitted model after the partition and settings have been fixed. Its broader predictive

value is the ability to use such feedback to compare, reject, or refine alternative subclasses and tool chains in future adaptive workflows. Relative to a fixed script, the LLM can translate a new research request into candidate query, descriptor, routing, and scheduling actions and can propose a next action when tool outputs change; the Harness restricts those proposals to valid interfaces before execution. Figure 7 illustrates this grounded query handling. The present study does not claim a measured accuracy advantage for this interaction layer.

To test sensitivity to data partitioning, all 14 regressors were rerun on 50 matched stratified 80/20 splits. Within each split, the pooled and subclass-specific routes use the same test entries, and preprocessing is fitted on the training portion only. The mean paired ΔR^2 is +0.120 for Lasso (95% CI +0.095 to +0.147), +0.115 for linear regression (+0.085 to +0.145), +0.113 for Ridge (+0.087 to +0.138), +0.080 for KNN (+0.052 to +0.109), +0.061 for AdaBoost (+0.043 to +0.077), and +0.054 for CatBoost (+0.040 to +0.067). These six models also retain significant paired changes after Holm adjustment. Decision Tree has a positive mean change and confidence interval but does not remain significant after multiplicity adjustment. The intervals for Gradient Boosting, Extra Trees, Random Forest, XGBoost, and MLP include zero. SVR and LightGBM decrease significantly. Tables 3 and 4 are therefore the principal statistical evidence: neither subclass routing nor a particular partition is universally beneficial across model families.

A direct comparison with unsupervised routing was performed over the same 50 matched splits. K-means was fitted only to the training descriptors within each split, and its test labels showed weak agreement with the expert oxide/halide labels (mean adjusted Rand index = 0.214). The expert rule had significantly higher R^2 for Ridge, Lasso, linear regression, KNN, and CatBoost after Holm adjustment. K-means had significantly higher R^2 for LightGBM, while the remaining eight regressors showed no adjusted paired difference. For example, expert-minus-K-means ΔR^2 was +0.139 for Ridge (95% CI +0.108 to +0.171) and −0.194 for LightGBM (−0.252 to −0.142). These results do not support a claim that one partition is uniformly best. They instead show why a Harness should treat taxonomy as an explicit workflow choice that is tested with the downstream model and rejected when it performs poorly. In this study, the LLM did not discover either partition.

Table 3. Repeated-split comparison of pooled training and the expert-defined oxide/halide subclass strategy (50 matched stratified 80/20 splits; global test-set R^2 for each route).

Model	Pooled R^2	Subclass R^2	Paired ΔR^2 [95% CI]	Wins	Holm p
Lasso	0.443	0.563	+0.120 [+0.095, +0.147]	47/50	8.6×10^{-11}
Linear Reg	0.420	0.535	+0.115 [+0.085, +0.145]	46/50	1.59×10^{-8}
Ridge	0.446	0.559	+0.113 [+0.087, +0.138]	47/50	4.08×10^{-10}
Decision Tree	0.253	0.358	+0.105 [+0.009, +0.212]	30/50	0.444
KNN (k = 7)	0.328	0.408	+0.080 [+0.052, +0.109]	40/50	1.84×10^{-6}
AdaBoost	0.460	0.521	+0.061 [+0.043, +0.077]	41/50	9.58×10^{-8}
CatBoost	0.607	0.661	+0.054 [+0.040, +0.067]	43/50	1.76×10^{-8}
Gradient Boost	0.610	0.624	+0.014 [−0.002, +0.032]	32/50	0.616
Extra Trees	0.592	0.603	+0.012 [−0.004, +0.028]	25/50	0.664
Random Forest	0.577	0.587	+0.010 [−0.005, +0.026]	31/50	0.664
XGBoost	0.607	0.615	+0.008 [−0.005, +0.020]	25/50	0.664
MLP (1L-50)	0.456	0.461	+0.005 [−0.029, +0.037]	29/50	0.664
SVR (rbf)	0.526	0.456	−0.070 [−0.096, −0.044]	7/50	2.79×10^{-6}
LightGBM	0.490	0.279	−0.211 [−0.239, −0.185]	0/50	2.49×10^{-14}

Table 4. Direct comparison of the expert-defined oxide/halide rule and training-only K-means routing (50 matched stratified 80/20 splits; global test-set R^2). ΔR^2 = expert rule – K-means; wins count splits in which the expert rule has higher R^2 .

Model	Expert R^2	K-Means R^2	Paired ΔR^2 [95% CI]	Expert Wins	Holm p
Lasso	0.563	0.425	+0.138 [+0.104, +0.174]	47/50	6.54×10^{-12}
Linear Reg	0.535	0.087	+0.448 [+0.226, +0.725]	46/50	5.84×10^{-12}
Ridge	0.559	0.420	+0.139 [+0.108, +0.171]	48/50	6.22×10^{-13}
Decision Tree	0.358	0.283	+0.076 [−0.009, +0.169]	30/50	0.658
KNN (k = 7)	0.408	0.365	+0.043 [+0.021, +0.067]	38/50	0.00023
AdaBoost	0.521	0.536	−0.015 [−0.059, +0.023]	25/50	1
CatBoost	0.661	0.611	+0.050 [+0.029, +0.070]	40/50	0.000178
Gradient Boost	0.624	0.631	−0.007 [−0.029, +0.014]	25/50	1
Extra Trees	0.603	0.596	+0.008 [−0.010, +0.026]	25/50	1
Random Forest	0.587	0.576	+0.011 [−0.010, +0.032]	30/50	1
XGBoost	0.615	0.612	+0.003 [−0.021, +0.024]	28/50	1
MLP (1L-50)	0.461	0.392	+0.069 [+0.010, +0.137]	26/50	1
SVR (rbf)	0.456	0.432	+0.024 [−0.006, +0.058]	29/50	1
LightGBM	0.279	0.473	−0.194 [−0.252, −0.142]	9/50	4.75×10^{-9}

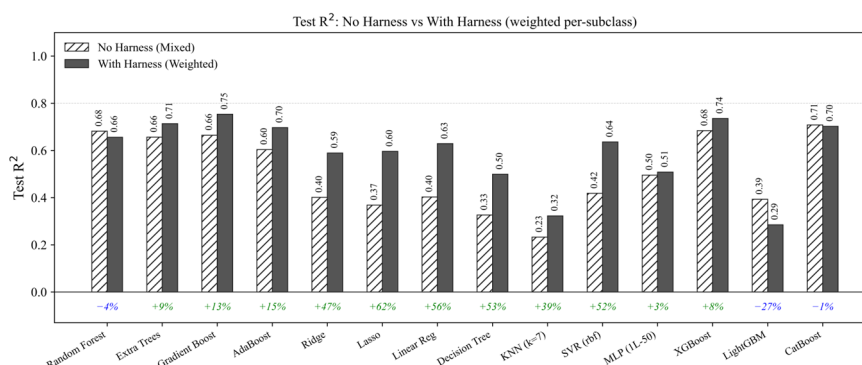


Figure 3. Test-set R^2 for pooled and expert-defined oxide/halide subclass-specific training in a representative stratified 80/20 split. Table 3 reports the matched 50-split robustness analysis.

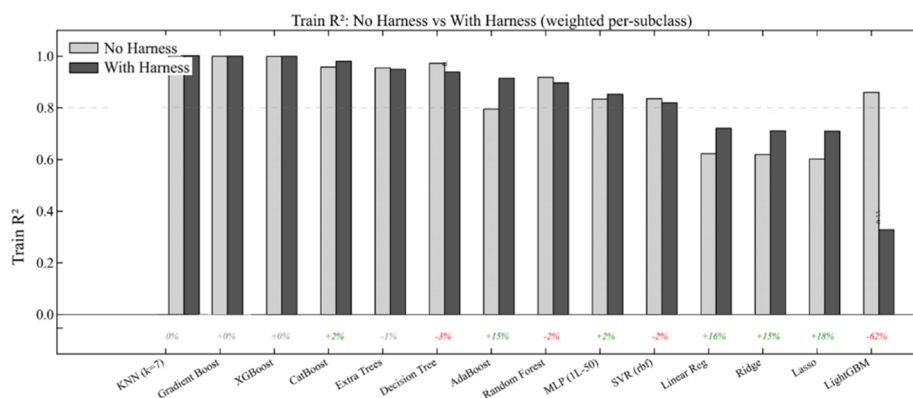


Figure 4. Training-set R^2 for pooled and expert-defined oxide/halide subclass-specific training in the representative stratified 80/20 split. The figure illustrates model-fitting behavior; robustness across data partitions is assessed in Table 3.

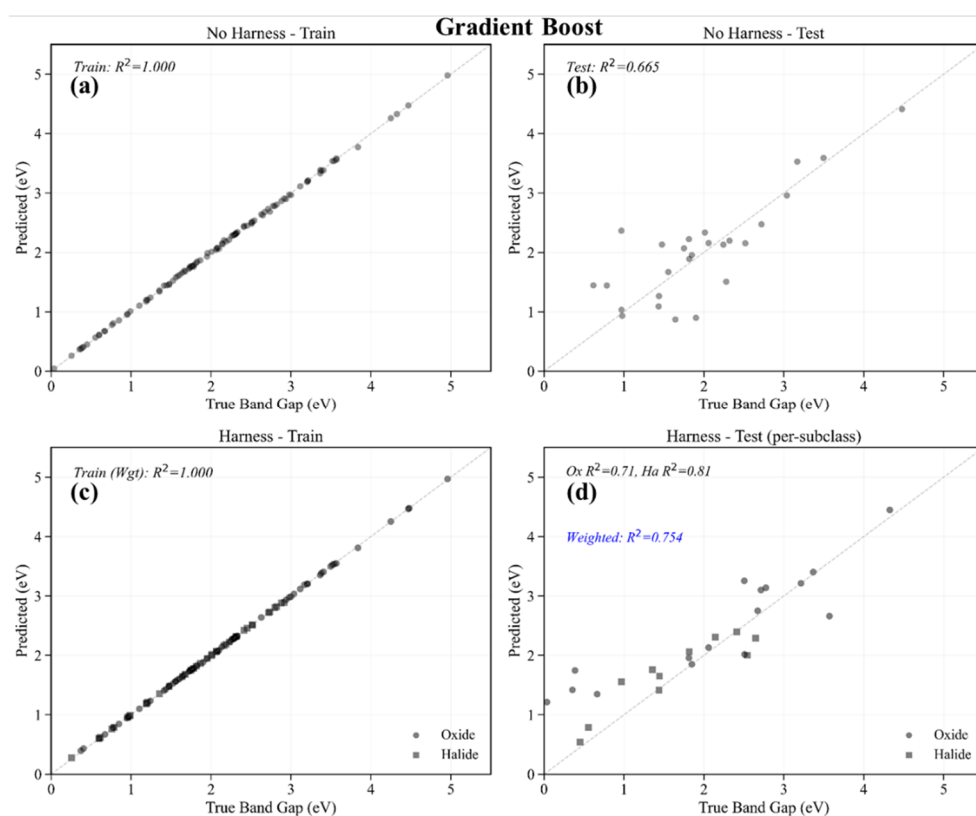


Figure 5. Gradient Boosting predictions for a representative stratified 80/20 split. **(a,b)** Pooled training and test predictions. **(c,d)** Expert-defined oxide/halide subclass-specific training and test predictions.

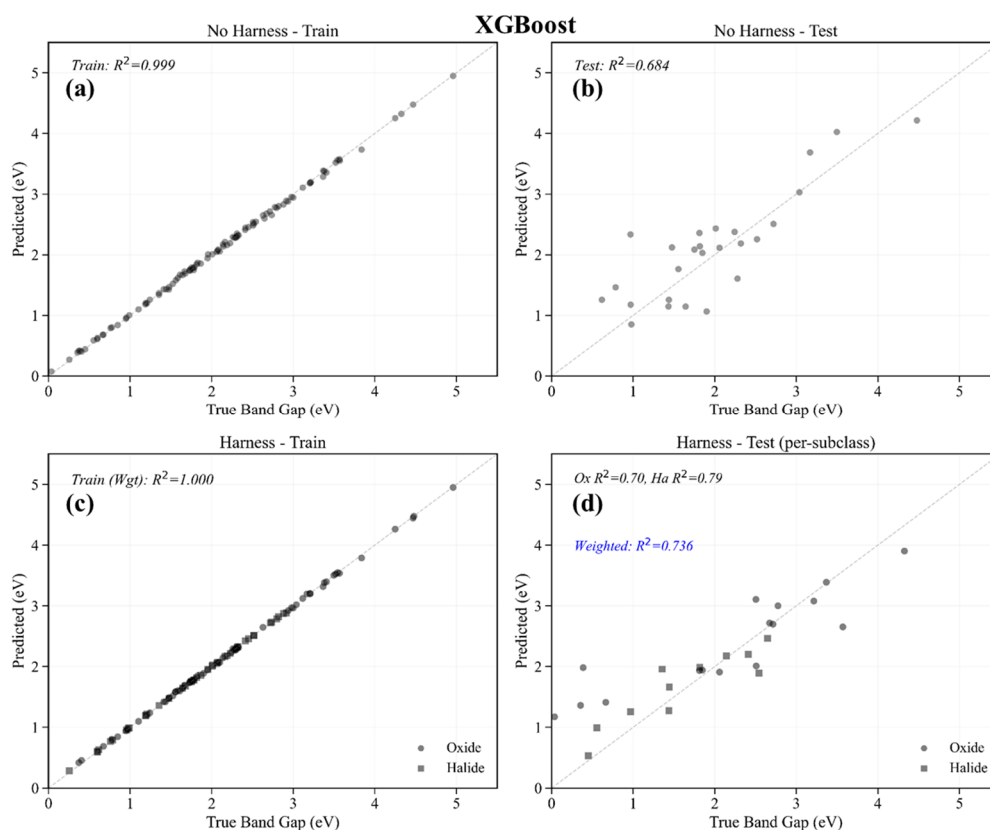


Figure 6. XGBoost predictions for a representative stratified 80/20 split. **(a,b)** Pooled training and test predictions. **(c,d)** Expert-defined oxide/halide subclass-specific training and test predictions.

Three representative queries (Figure 7) illustrate the evidence-grounding and diagnostic functions of the Harness workflow rather than an independent taxonomy benchmark. For CsPbI_3 , the standalone LLM gives an unsourced approximate value, whereas the workflow queries Materials Project and returns three polymorph records with computed band gaps, energies above hull, material identifiers, and phase information. For $\text{Cs}_2\text{AgInBr}_6$, the workflow retrieves the Materials Project-computed band gap of 0.26 eV for mp-1113554 and compares it with the halide specialist prediction of 0.31 eV, while noting that the double perovskite is outside the dominant ABX_3 training distribution. For the Decision Tree query, the workflow retrieves the stored single-split metrics and separates oxide and halide behavior. The reported interpretation is described as a pattern consistent with limited halide data, a wider target range, and the fixed tree depth; it is not presented as a causal physical proof. These examples show how Harness replaces unsupported text-only responses with retrievable records, model evidence, and explicit limitations.

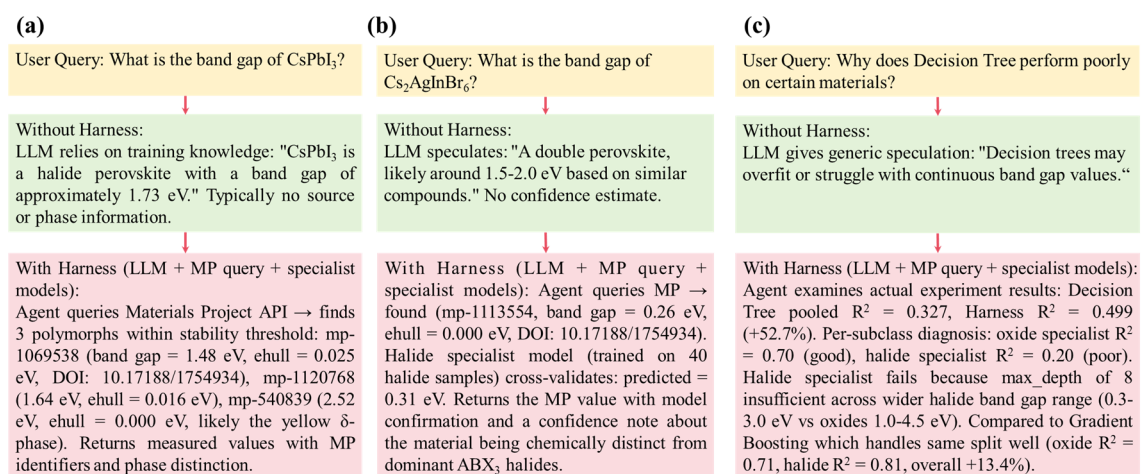


Figure 7. Evidence-grounded query examples comparing a standalone LLM with the Harness workflow. **(a)** Materials Project polymorph retrieval for CsPbI_3 . **(b)** Database retrieval and specialist-model comparison for $\text{Cs}_2\text{AgInBr}_6$. **(c)** Evidence-linked diagnosis of Decision Tree performance for the representative single-split analysis.

Pearson correlation matrices are calculated separately for the oxide and halide subclasses (Figures 8 and S1), together with their difference matrix. The oxide data show strong coupling among several structural descriptors and a pronounced negative correlation between the tolerance and octahedral factors. The halide data show a different correlation pattern, including a positive formation-energy–volume relationship in this dataset. The difference matrix contains many feature pairs with $|\Delta r| > 0.5$. The features `nelements`, `is_magnetic_int`, and `ordering_int` have zero variance within one subclass, so their within-subclass Pearson correlations are undefined and they cannot add information to a within-subclass regressor. These descriptor differences support the use of the expert-defined oxide/halide partition as an initial modeling hypothesis, but they do not establish automatic taxonomy discovery or prove that the partition is optimal.

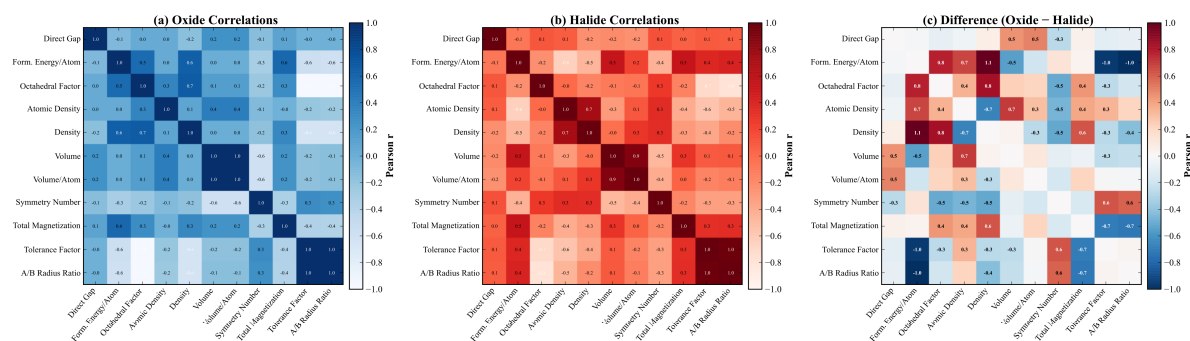


Figure 8. Feature correlation structure comparison between Oxide and Halide subclasses. **(a)** Oxide correlation matrix. **(b)** Halide correlation matrix. **(c)** Difference matrix (Oxide minus Halide).

Figure 9 compares feature-importance ranks from the oxide and halide Gradient Boosting specialists for the representative split used in Figures 3–6 (weighted subclass $R^2 = 0.754$). Importance is calculated independently

for each specialist model, and connecting lines show rank changes. The ranking differences indicate that the two models use the shared descriptor set differently. Formation energy per atom ranks first for the oxide specialist, whereas the octahedral factor is more prominent in the halide specialist. The tolerance factor and A/B-site radius ratio also change rank between the two models. These patterns are consistent with different geometric, energetic, and structural influences in the two material families, but feature importance alone should not be interpreted as a causal mechanism. The result explains why subclass-specific fitting can be useful for this initial partition and provides a diagnostic signal that a future Harness workflow could use when comparing alternative partitions.

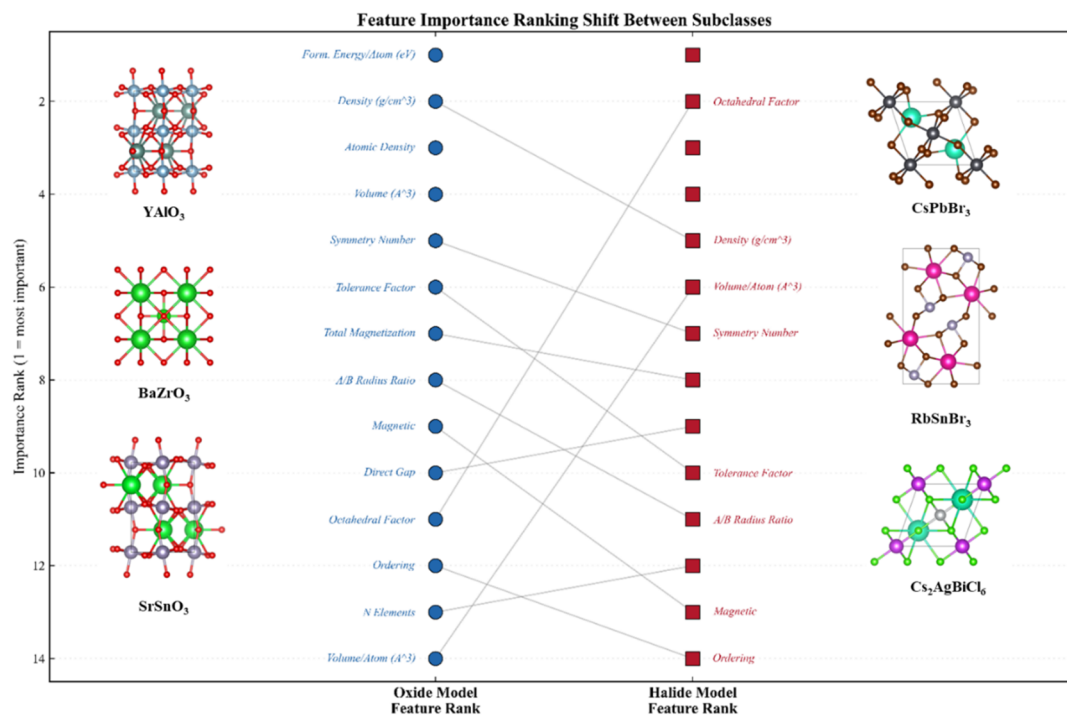


Figure 9. Feature importance ranking shift between subclasses.

The subclass-stratified residual analysis in Figure 10 compares pooled and subclass-specific Ridge models under five-fold cross-validation. In the pooled model, the oxide and halide mean signed residuals are -0.043 and $+0.057$ eV, respectively, giving a difference of approximately 0.10 eV and a point-biserial correlation of $+0.062$ between subclass identity and signed residual. With separate specialist models, the oxide–halide mean-residual difference decreases to -0.027 eV and the point-biserial correlation decreases to $+0.019$. The robust residual spread (MAD) decreases from 0.816 to 0.592 eV, and the overall cross-validated R^2 increases from 0.433 to 0.568 . These results show that the selected subclass strategy reduces, but does not prove complete elimination of, subclass-correlated error in this Ridge example.

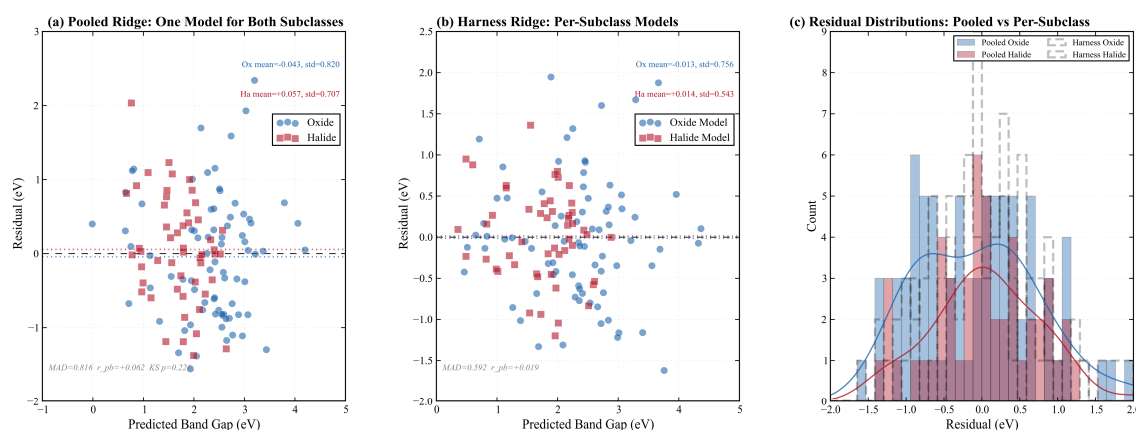


Figure 10. Residual comparison for pooled and expert-defined oxide/halide subclass-specific Ridge models. (a,b) Signed residuals by material family. (c) Residual distributions. The subclass route narrows the residual distribution and reduces subclass-correlated bias in this five-fold cross-validation example.

Figure 11 presents the same pooled-versus-subclass-specific Ridge comparison as parity plots and residual maps on a common PCA projection. The pooled model has cross-validated $R^2 = 0.433$ and $MAD = 0.816$ eV; the specialist route has $R^2 = 0.568$ and $MAD = 0.592$ eV. Each PCA point is colored by signed residual and scaled by absolute residual magnitude. In the pooled route, residual sign is more visibly organized along the oxide–halide separation, whereas the subclass-specific route shows a weaker association and fewer large residuals. The point-biserial correlation decreases from $+0.062$ to $+0.019$. This is visual evidence that the current partition can reduce systematic family-dependent error for Ridge, not evidence that Harness has already discovered the best possible partition.

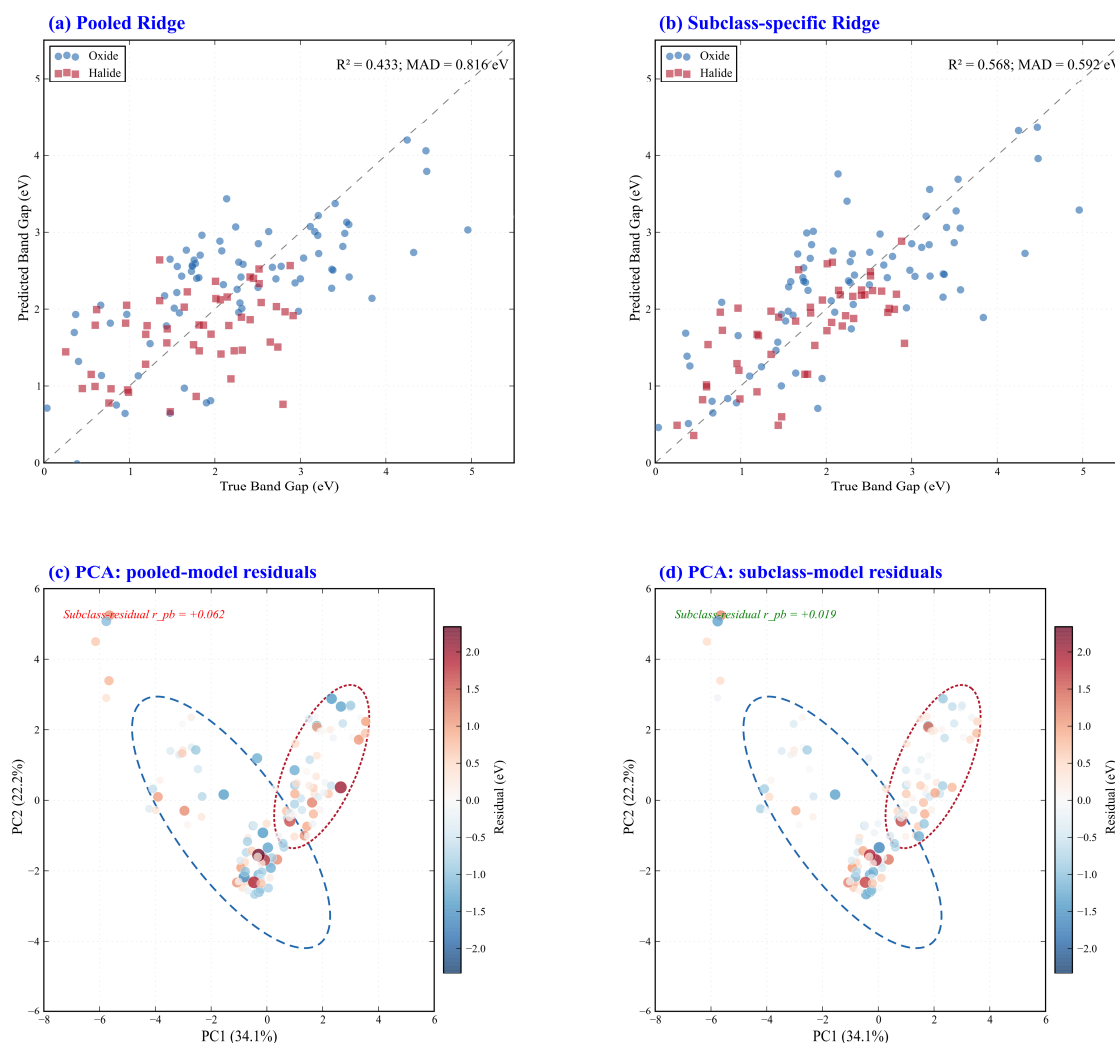


Figure 11. Pooled versus expert-defined oxide/halide subclass-specific Ridge regression under five-fold cross-validation. (a,b) Parity plots. (c,d) Signed residuals mapped onto a common PCA projection; marker size scales with absolute residual. Dashed and dotted ellipses indicate the oxide and halide regions, respectively.

The PCA projection of the 14-dimensional descriptor space (Figures S2 and S3) shows that oxide and halide records occupy largely separable regions in the first two components, which account for 34.1% and 22.2% of the variance. The centroid separation and loading analysis provide feature-level support for using the chemical partition as an initial hypothesis. Density, A/B-site radius ratio, tolerance factor, and octahedral factor contribute strongly to PC1, while volume, volume per atom, formation energy per atom, and symmetry number contribute strongly to PC2. The loading magnitudes are distributed across many descriptors rather than dominated by a single feature. This analysis does not demonstrate that the oxide/halide rule is optimal. Instead, it identifies measurable structure in descriptor space that can be used by a future adaptive Harness to propose and compare expert rules, clustering solutions, learned representations, or hierarchical taxonomies using matched prediction feedback.

4. Limitations

Several limitations define the scope of this study. First, the dataset contains only 129 records and the oxide/halide split leaves still smaller specialist training sets. The 50 matched splits reduce dependence on one

favorable partition, but they do not replace validation on a larger independent dataset. Second, the chemical partition is fixed and supplied from domain knowledge. The added $k = 2$ K-means baseline is a direct unsupervised comparison, but it does not exhaust clustering choices and does not constitute autonomous taxonomy discovery. Representation-based, hierarchical, soft, and agent-generated partitions remain untested. Third, the effect is model dependent: several regressors show small or inconclusive changes, and SVR and LightGBM become worse under the expert-rule split. The fixed LightGBM configuration decreases from mean pooled $R^2 = 0.490$ to expert-rule $R^2 = 0.279$ (paired $\Delta R^2 = -0.211$; 95% CI -0.239 to -0.185 ; 0/50 wins). K-means routing gives a higher mean R^2 of 0.473 for this model. These results are consistent with sensitivity to the smaller routed training sets and partition choice, but the present experiment does not identify a unique cause or establish an optimal LightGBM configuration. Partition quality and hyperparameters must therefore be evaluated jointly with sample adequacy, model family, uncertainty, and physical interpretability. Fourth, the 14 descriptors and database-computed band gaps do not capture all electronic-structure or finite-temperature physics. Materials Project grounding improves traceability but does not replace experiment or atomistic validation. Future work will treat subclass definition as a workflow variable rather than a fixed label. A more complete Harness could connect learned representations, additional databases, automated model selection, uncertainty estimation, active learning, and MLIPs [66–72]. After a database screen, candidate structures could be relaxed or simulated at finite temperature, screened for failed convergence or implausible geometry, and escalated to DFT when an uncertainty or committee-disagreement threshold is exceeded. Newly labeled configurations could then enter an active-learning update. Candidate workflows should be compared using nested or independent validation and rejected or refined when they reduce accuracy, create undersized subclasses, increase uncertainty, or fail physical checks. The current work does not perform this atomistic validation loop. Such adaptive taxonomy and tool-chain optimization is the intended next step and is not claimed as a completed result of the present case study.

5. Conclusions

This study provides a foundational demonstration of Harness engineering for materials science using perovskite band-gap prediction. The expert-defined oxide/halide partition is a conventional scientific strategy and is the direct cause of the observed difference between pooled and specialist models. Across 50 matched splits, the benefit is substantial and reproducible for several regressors, negligible for others, and negative for SVR and LightGBM. The training-only K-means comparison further shows that the best partition depends on the downstream model: the chemical rule is stronger for several regressors, but K-means is stronger for LightGBM. Harness contributes by making the partition and the surrounding descriptor, model, validation, and evidence choices explicit, executable, and evaluable within one agent workflow. In the current implementation, it provides a controlled and reproducible path for testing the initial chemical hypothesis; it does not autonomously discover an optimal taxonomy or create an additional accuracy gain after the workflow has been fixed. The practical value of the broader framework is that the same controlled route can screen database candidates, compare alternative material partitions, call structural-relaxation or finite-temperature tools, and send uncertain MLIP configurations to DFT without allowing an LLM judgment to replace a physical check. The present results therefore establish the workflow and its scientific motivation, while adaptive search for higher-performing material taxonomies remains an important direction for future work.

Supplementary Materials

The additional data and information can be downloaded at: <https://media.scilit.com/articles/others/2609111437548415/AIMat-26060080-SI.pdf>. Supplementary materials accompanying this article include Figure S1: feature correlation comparison between oxide and halide subclasses; Figure S2: PCA projection of the perovskite descriptor space; and Figure S3: PCA loading analysis.

Author Contributions

L.Z.: conceptualization, supervision, writing—original draft preparation. T.W.: software, visualization, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported by the Chinese Academy of Sciences Pioneer Initiative (CAS Leading Action Plan).

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The data supporting the findings of this study are provided in the article and its supplementary materials. The software and code are available at <https://github.com/Zhang-NJ-Lab/HarnessPerovskiteAgent> (accessed on 9 September 2026).

Acknowledgments

This work was supported by the Chinese Academy of Sciences Pioneer Initiative (CAS Leading Action Plan) (corresponding author: L. Zhang).

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

Claude Sonnet 4 was used as the reasoning engine within the agent workflow, as described in Section 2.1. AI-assisted tools were also used for language editing, figure preparation, and proof correction. The authors are responsible for the text, figures, references, and scientific content of the article.

References

1. Zhang, L.; Liu, Z.; Ni, B.; et al. Large Language Models (LLMs) for Materials Design. *Adv. Funct. Mater.* **2026**, *36*, e25897. <https://doi.org/10.1002/adfm.202525897>.
2. Ahlawat, D.; Mishra, V.; Singh, S.; et al. A Family of Large Language Models for Materials Research with Insights into Model Adaptability in Continued Pretraining. *Nat. Mach. Intell.* **2026**, *8*, 435–448. <https://doi.org/10.1038/s42256-026-01199-8>.
3. Wang, X.; Zeng, Q.; Xu, D.-H.; et al. Accelerating Materials Discovery via AI-Agent Integration of Large Language Models and Simulation Tools. *J. Mater. Informatics* **2026**, *6*, 9. <https://doi.org/10.20517/jmi.2025.69>.
4. Huang, L.; Zhou, Y.; Yang, J. Integrated Sharing Platform for Genetic Data of Rare and Precious Metal Materials. *Comput. Mater. Contin.* **2025**, *85*, 4587–4606. <https://doi.org/10.32604/cmc.2025.068370>.
5. Vergara, D.; del Bosque, A.; Fernández-Arias, P. A Decade of Digital Twins in Materials Science and Engineering. *Comput. Mater. Contin.* **2025**, *85*, 41–64. <https://doi.org/10.32604/cmc.2025.067881>.
6. Xu, P.; Ji, X.; Li, M.; et al. Virtual Sample Generation in Machine Learning Assisted Materials Design and Discovery. *J. Mater. Informatics* **2023**, *3*, 16. <https://doi.org/10.20517/jmi.2023.18>.
7. Deng, H.; Wu, Y.; Wang, Y.; et al. PVmatAgent: A Large Language Model (LLM) Agent for Photovoltaic and Perovskite Material Design and Analysis. *ChemRxiv* **2026**.
8. Qiu, J.; Wang, Y.; Jiang, F.; et al. STALAgent: A Multi-Agent System Based on Large Language Model (LLM) for Steel and Alloy Design. *ChemRxiv* **2026**.
9. Zhang, L.; Zhou, J.; Chen, X. Data-Driven Exploration and First-Principles Analysis of Perovskite Material. *J. Mater. Informatics* **2024**, *4*, 13. <https://doi.org/10.20517/jmi.2024.20>.
10. Xue, X.; Huang, X.; Wang, G. Materials Genome Engineering: A Promising Approach for the Development of High-Performance Metal–Organic Frameworks. *Sci. Bull.* **2022**, *67*, 1197–1200. <https://doi.org/10.1016/j.scib.2022.05.003>.
11. Hegde, V.I.; Borg, C.K.H.; del Rosario, Z.; et al. Quantifying Uncertainty in High-Throughput Density Functional Theory: A Comparison of AFLOW, Materials Project, and OQMD. *Phys. Rev. Mater.* **2023**, *7*, 053805. <https://doi.org/10.1103/PhysRevMaterials.7.053805>.
12. Jain, A.; Ong, S.P.; Hautier, G.; et al. Commentary: The Materials Project: A Materials Genome Approach to Accelerating Materials Innovation. *APL Mater.* **2013**, *1*, 011002. <https://doi.org/10.1063/1.4812323>.
13. Park, Y.J.; Jerng, S.E.; Yoon, S.; et al. 1.5 Million Materials Narratives Generated by Chatbots. *Sci. Data* **2024**, *11*, 1060. <https://doi.org/10.1038/s41597-024-03886-w>.

14. Kirklin, S.; Saal, J.E.; Meredig, B.; et al. The Open Quantum Materials Database (OQMD): Assessing the Accuracy of DFT Formation Energies. *npj Comput. Mater.* **2015**, *1*, 15010. <https://doi.org/10.1038/npjcompumats.2015.10>.
15. Curtarolo, S.; Setyawan, W.; Hart, G.L.W.; et al. AFLOW: An Automatic Framework for High-Throughput Materials Discovery. *Comput. Mater. Sci.* **2012**, *58*, 218–226. <https://doi.org/10.1016/j.commatsci.2012.02.005>.
16. Zhou, J.; Shen, L.; Costa, M.D.; et al. 2DMatPedia, an Open Computational Database of Two-Dimensional Materials from Top-down and Bottom-up Approaches. *Sci. Data* **2019**, *6*, 86. <https://doi.org/10.1038/s41597-019-0097-3>.
17. Choudhary, K. The JARVIS Infrastructure Is All You Need for Materials Design. *Comput. Mater. Sci.* **2025**, *259*, 114063. <https://doi.org/10.1016/j.commatsci.2025.114063>.
18. Zhou, Z.; Zhang, Y.; Wang, S.; et al. A Knowledge Graph Attention Network for the Cold-start Problem in Intelligent Manufacturing: Interpretability and Accuracy Improvement. *Mater. Genome Eng. Adv.* **2025**, *3*, e85. <https://doi.org/10.1002/mgea.85>.
19. Mannan, S.; Bihani, V.; Krishnan, N.M.A.; et al. Navigating Energy Landscapes for Materials Discovery: Integrating Modeling, Simulation, and Machine Learning. *Mater. Genome Eng. Adv.* **2024**, *2*, e25. <https://doi.org/10.1002/mgea.25>.
20. Chen, X.; He, S.; Paik, K.-W.; et al. Machine Learning-Driven Design and Optimization of Electronic Packaging: Applications and Future Developments. *J. Mater. Informatics* **2025**, *5*, 47. <https://doi.org/10.20517/jmi.2025.26>.
21. Yang, Y.; Cao, L.; Ren, J.; et al. Artificial Intelligence for Fibrous Network Design and Mechanics. *AI Sci.* **2025**, *1*, 012001. <https://doi.org/10.1088/3050-287X/adfaf7>.
22. Wang, W.Y.; Zhang, S.; Li, G.; et al. Artificial Intelligence Enabled Smart Design and Manufacturing of Advanced Materials: The Endless Frontier in AI + Era. *Mater. Genome Eng. Adv.* **2024**, *2*, e56. <https://doi.org/10.1002/mgea.56>.
23. Zhang, L.; Deng, H. NJmat 2.0: User Instructions of Data-Driven Machine Learning Interface for Materials Science. *Comput. Mater. Contin.* **2025**, *83*, 1–11. <https://doi.org/10.32604/cmc.2025.062666>.
24. He, M.; Zhang, L. Machine Learning and Symbolic Regression Investigation on Stability of MXene Materials. *Comput. Mater. Sci.* **2021**, *196*, 110578. <https://doi.org/10.1016/j.commatsci.2021.110578>.
25. Butler, K.T.; Davies, D.W.; Cartwright, H.; et al. Machine Learning for Molecular and Materials Science. *Nature* **2018**, *559*, 547–555. <https://doi.org/10.1038/s41586-018-0337-2>.
26. Muuga, K.; Knijff, L.; Zhang, C. Molecular Electrostatic Potentials from Machine Learning Models for Dipole and Quadrupole Predictions. *AI Sci.* **2026**, *2*, 025002. <https://doi.org/10.1088/3050-287X/ae531a>.
27. Fang, Z.; Hsu, T.-W.; Yan, Q. Learning Atomic Representations for Data-Driven Materials Design. *AI Sci.* **2026**, *2*, 013001. <https://doi.org/10.1088/3050-287X/ae45bf>.
28. Lin, X.; Wang, Y.; Zhang, Z.; et al. Scientific Claim Recognition via Staged Fine-Tuning with LoRA. *Data Intell.* **2025**, *7*, 303–335. <https://doi.org/10.3724/2096-7004.di.2025.0009>.
29. Fu, F.; Li, Q.-Q.; Wang, F.; et al. Synergizing a Knowledge Graph and Large Language Model for Relay Catalysis Pathway Recommendation. *Natl. Sci. Rev.* **2025**, *12*, nwaf271. <https://doi.org/10.1093/nsr/nwaf271>.
30. Zhang, C.; Lin, Q.; Zhu, B.; et al. SynAsk: Unleashing the Power of Large Language Models in Organic Synthesis. *Chem. Sci.* **2025**, *16*, 43–56. <https://doi.org/10.1039/D4SC04757E>.
31. Soares, F.M.; Saraiva, A.M.; Pires, L.F.; et al. Exploring a Large Language Model for Transforming Taxonomic Data into OWL: Lessons Learned and Implications for Ontology Development. *Data Intell.* **2025**, *7*, 265–302. <https://doi.org/10.3724/2096-7004.di.2025.0020>.
32. Xu, K.; Lu, Y.; Liu, L.; et al. MatterMind: A User-Friendly Material Design Platform Assisted by Large Language Model. *Mater. Genome Eng. Adv.* **2026**, *4*, e70075. <https://doi.org/10.1002/mgea.70075>.
33. Yan, Z.; Liang, H.; Wang, J.; et al. PDGPT: A Large Language Model for Acquiring Phase Diagram Information in Magnesium Alloys. *Mater. Genome Eng. Adv.* **2024**, *2*, e77. <https://doi.org/10.1002/mgea.77>.
34. Lv, S.; Peng, L.; Wu, W.; et al. Bridging Language Models and Computational Materials Science: A Prompt-driven Framework for Material Property Prediction. *Mater. Genome Eng. Adv.* **2025**, *3*, e70013. <https://doi.org/10.1002/mgea.70013>.
35. Xu, Q.; Yu, Q.; Qian, L. Analysis and Evaluation of Open Source Scientific Entity Datasets. *Data Intell.* **2025**, *7*, 667–691. <https://doi.org/10.3724/2096-7004.di.2024.0009>.
36. Yao, T.; Xu, A.; Ou, P.; et al. Let Materials Science Data Learn to Reason. *AI Mater.* **2026**, *1*, 4. <https://doi.org/10.53941/aimat.2026.100004>.
37. He, J.; Guan, H.; Feng, W.; et al. Controlling Risks of AI in Chemical Science with Agents. *AI Sci.* **2025**, *1*, 015002. <https://doi.org/10.1088/3050-287X/adfee5>.
38. Wang, W.; Rignanes, G.-M. AI for Science to Serve the AI-Driven Scientific Innovation. *AI Sci.* **2025**, *1*, 010201. <https://doi.org/10.1088/3050-287X/add0cb>.
39. Chaudhari, A.; Ock, J.; Barati Farimani, A. Modular Large Language Model Agents for Multi-Task Computational Materials Science. *Commun. Mater.* **2026**, *7*, 131. <https://doi.org/10.1038/s43246-025-00994-x>.

40. Bran, A.M.; Cox, S.; Schilter, O.; et al. Augmenting Large Language Models with Chemistry Tools. *Nat. Mach. Intell.* **2024**, *6*, 525–535. <https://doi.org/10.1038/s42256-024-00832-8>.
41. Shi, Z.; Xin, C.; Huo, T.; et al. A Fine-Tuned Large Language Model Based Molecular Dynamics Agent for Code Generation to Obtain Material Thermodynamic Parameters. *Sci. Rep.* **2025**, *15*, 10295. <https://doi.org/10.1038/s41598-025-92337-6>.
42. Wang, Z.; Huang, H.; Zhao, H.; et al. DREAMS: Density Functional Theory Based Research Engine for Agentic Materials Simulation. *arXiv* **2025**, arXiv:2507.14267.
43. Liu, J.; Zhu, T.; Ye, C.; et al. VASPilot: MCP-Facilitated Multi-Agent Intelligence for Autonomous VASP Simulations. *arXiv* **2025**, arXiv:2508.07035.
44. Zou, Y.; Cheng, A.H.; Aldossary, A.; et al. El Agente: An Autonomous Agent for Quantum Chemistry. *Matter* **2025**, *8*, 102263. <https://doi.org/10.1016/j.matt.2025.102263>.
45. Nduma, R.; Park, H.; Walsh, A. Crystallise: A Multi-Tool Agent for Materials Design. *arXiv* **2025**, arXiv:2512.00977.
46. Inizan, T.J.; Yang, S.; Kaplan, A.; et al. System of Agentic AI for the Discovery of Metal-Organic Frameworks. *arXiv* **2025**, arXiv:2504.14110.
47. Tian, J.; Sobczak, M.T.; Patil, D.; et al. CrossMatAgent: AI-Assisted Design of Manufacturable Metamaterial Patterns via Multi-Agent Generative Framework. *Adv. Intell. Discov.* **2025**, *2*, e202500063. <https://doi.org/10.1002/aidi.202500063>.
48. Liu, G.; Yang, S.; Zhong, Y. Masgent: An AI-Assisted Materials Simulation Agent. *arXiv* **2025**, arXiv:2512.23010.
49. Gan, J.; Zhong, P.; Du, Y.; et al. MatLLMSearch: Crystal Structure Discovery with Evolution-Guided Large Language Models. *arXiv* **2025**, arXiv:2502.20933.
50. Yang, J.; Kobayashi, Y.; Demura, M. AI Agents for Automating Materials Research: A Case Study of Crystal Plasticity Simulations. *Sci. Technol. Adv. Mater. Methods* **2026**, *6*, 2630445. <https://doi.org/10.1080/27660400.2026.2630445>.
51. Song, Z.; Lu, S.; Zhou, Q.; et al. T2MAT (Text-to-Material): A Universal Agent for Generating Material Structures with Goal Properties from a Single Sentence. *Acta Physico-Chimica Sin.* **2026**, *42*, 100213. <https://doi.org/10.1016/j.actphy.2025.100213>.
52. Chiang, Y.; Hsieh, E.; Chou, C.-H.; et al. LLaMP: Large Language Model Made Powerful for High-Fidelity Materials Knowledge Retrieval. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 25200–25232. <https://doi.org/10.18653/v1/2025.emnlp-main.1280>.
53. Zhang, H.; Song, Y.; Hou, Z.; et al. HoneyComb: A Flexible LLM-Based Agent System for Materials Science. In *Findings of the Association for Computational Linguistics: EMNLP 2024*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp 3369–3382. <https://doi.org/10.18653/v1/2024.findings-emnlp.192>.
54. Ansari, M.; Moosavi, S.M. Agent-Based Learning of Materials Datasets from Scientific Literature. *Digit. Discov.* **2024**, *3*, 2607–2617. <https://doi.org/10.1039/D4DD00252K>.
55. Jia, S.; Zhang, C.; Fung, V. LLMatDesign: Autonomous Materials Discovery with Large Language Models. *arXiv* **2024**, arXiv:2406.13163.
56. Ralph, J.; Martynov, P.; Ma, X.; et al. Identifier Service in the Mindat Database: Persistent and Structured Access to Massive Records of Minerals and Other Natural Materials. *Data Intell.* **2024**, *7*, 692–711. <https://doi.org/10.3724/2096-7004.di.2024.0021>.
57. Hu, Z.; You, J.; Huang, X.; et al. Automatic Construction of Knowledge Graphs from Scientific Literature for Copper-Based Composites. *Data Intell.* **2024**, *6*, 1168–1189. <https://doi.org/10.3724/2096-7004.di.2024.0038>.
58. Li, L.; Xu, K.; Su, R.; et al. Data Speaks: LLM-Enabled Evidence Auditing Resets the “Migration Barrier” Playbook for Solid-State Electrolytes. *AI Mater.* **2026**, *1*, 2. <https://doi.org/10.53941/aimat.2026.100002>.
59. Yin, R.A.; Ren, Z.; Yin, Z.; et al. Collaborative AI Enhances Image Understanding in Materials Science. *AI Mater.* **2026**, *1*, 6. <https://doi.org/10.53941/aimat.2026.100006>.
60. Zhong, H.; Zhu, S. AI Harness Engineering: A Runtime Substrate for Foundation-Model Software Agents. *arXiv* **2026**, arXiv:2605.13357.
61. Miret, S.; Krishnan, N.M.A. Enabling Large Language Models for Real-World Materials Discovery. *Nat. Mach. Intell.* **2025**, *7*, 991–998. <https://doi.org/10.1038/s42256-025-01058-y>.
62. Ding, K.; Yu, J.; Huang, J.; et al. SciToolAgent: A Knowledge-Graph-Driven Scientific Agent for Multitool Integration. *Nat. Comput. Sci.* **2025**, *5*, 962–972. <https://doi.org/10.1038/s43588-025-00849-y>.
63. Li, X.; Huang, Z.; Quan, S.; et al. SLM-MATRIX: A Multi-Agent Trajectory Reasoning and Verification Framework for Enhancing Language Models in Materials Data Extraction. *npj Comput. Mater.* **2025**, *11*, 241. <https://doi.org/10.1038/s41524-025-01719-x>.
64. Soleymanibrojeni, M.; Aydin, R.; Guedes-Sobrinho, D.; et al. GENIUS: An Agentic AI Framework for Autonomous Design and Execution of Simulation Protocols. *Commun. Mater.* **2026**, *7*, 115. <https://doi.org/10.1038/s43246-026-01167-0>.

65. Horton, M.K.; Huck, P.; Yang, R.X.; et al. Accelerated Data-Driven Materials Science with the Materials Project. *Nat. Mater.* **2025**, *24*, 1522–1532. <https://doi.org/10.1038/s41563-025-02272-0>.
66. Batatia, I.; Benner, P.; Chiang, Y.; et al. A Foundation Model for Atomistic Materials Chemistry. *J. Chem. Phys.* **2025**, *163*, 184110. <https://doi.org/10.1063/5.0297006>.
67. Zhang, L.; Lin, D.-Y.; Wang, H.; et al. Active Learning of Uniformly Accurate Interatomic Potentials for Materials Simulation. *Phys. Rev. Mater.* **2019**, *3*, 023804. <https://doi.org/10.1103/PhysRevMaterials.3.023804>.
68. Kulichenko, M.; Barros, K.; Lubbers, N.; et al. Uncertainty-Driven Dynamics for Active Learning of Interatomic Potentials. *Nat. Comput. Sci.* **2023**, *3*, 230–239. <https://doi.org/10.1038/s43588-023-00406-5>.
69. Yoo Willow, S.; Kim, D.G.; Sundheep, R.; et al. Active Sparse Bayesian Committee Machine Potential for Isothermal–Isobaric Molecular Dynamics Simulations. *Phys. Chem. Chem. Phys.* **2024**, *26*, 22073–22082. <https://doi.org/10.1039/D4CP01801J>.
70. Liang, X.; Klarbring, J.; Baldwin, W.J.; et al. Structural Dynamics Descriptors for Metal Halide Perovskites. *J. Phys. Chem. C* **2023**, *127*, 19141–19151. <https://doi.org/10.1021/acs.jpcc.3c03377>.
71. Liang, X.; Klarbring, J.; Walsh, A. Phase Stability and Transformations in Lead Mixed Halide Perovskites from Machine Learning Force Fields. *Chem. Mater.* **2025**, *37*, 9392–9405. <https://doi.org/10.1021/acs.chemmater.5c01730>.
72. Kim, K.S. Machine Learning for Accelerating Energy Materials Discovery: Bridging Quantum Accuracy with Computational Efficiency. *Adv. Energy Mater.* **2026**, *16*, e03356. <https://doi.org/10.1002/aenm.202503356>.