Towards Dynamic Benchmarks for Autonomous Materials Discovery

Shreshth A. Malik[1,2*] **Tiarnan Doherty**[2] **Panagiotis Tigas**[2]
Muhammed Razzak[2] **Aron Walsh**[3] **Yarin Gal**[1]

[1] OATML, Department of Computer Science, University of Oxford

[2] Diffractive Labs

[3] Thomas Young Centre and Department of Materials, Imperial College London

Abstract

Existing benchmarks for computational materials discovery primarily evaluate static predictive tasks or isolated computational sub-tasks. Such approaches inadequately capture the inherently iterative, exploratory, and often serendipitous nature of scientific discovery. We argue that the research community should shift evaluation practices towards including dynamic benchmarks that more realistically represent materials discovery campaigns. As a concrete example, we propose an open-ended benchmark environment designed to simulate closed-loop discovery, requiring autonomous agents or algorithms to iteratively propose, evaluate, and refine candidates under a constrained evaluation budget. Specifically, it targets the efficient discovery of new thermodynamically stable compounds within chemical systems. Multiple fidelity levels are accommodated, from machine-learned interatomic potentials to density functional theory and experimental validation. This approach emphasizes realistic elements of scientific discovery, such as iterative refinement, adaptive decision-making, handling uncertainty and traversing unknown chemical landscapes.

1 Introduction

Scientific discovery rarely progresses in a linear fashion. Researchers propose hypotheses, run experiments or simulations, and refine their ideas based on the outcomes [34]. Failures can be as informative as successes, and strategies often shift as new evidence or constraints emerge. Materials discovery is no exception: promising candidates are proposed, evaluated, and iteratively refined, often through cycles of exploration, dead ends, and serendipitous insights.

In contrast, most computational benchmarks for materials discovery assume a static, one-way process (Figure 1a). The majority are designed to measure the accuracy of predictive models such as machine learning interatomic potentials (MLIPs) [36, 11, 6, 9, 8] and related surrogates, on fixed datasets. Standard tasks include regression of formation energies, forces, band gaps, or agreement with density functional theory (DFT) calculations [10, 37, 39, 18]. These have driven real progress in ML model accuracy and efficiency and are important to further progress, but they evaluate models in isolation from the broader scientific process, where the central challenge is efficiently navigating an immense search space under uncertainty. Even recent advancements such as Matbench Discovery [37], though representing an important step forward, remain fundamentally screening benchmarks. They assess

*Correspondence to shreshth@robots.ox.ac.uk.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI4Mat: AI for Accelerated Materials Design.

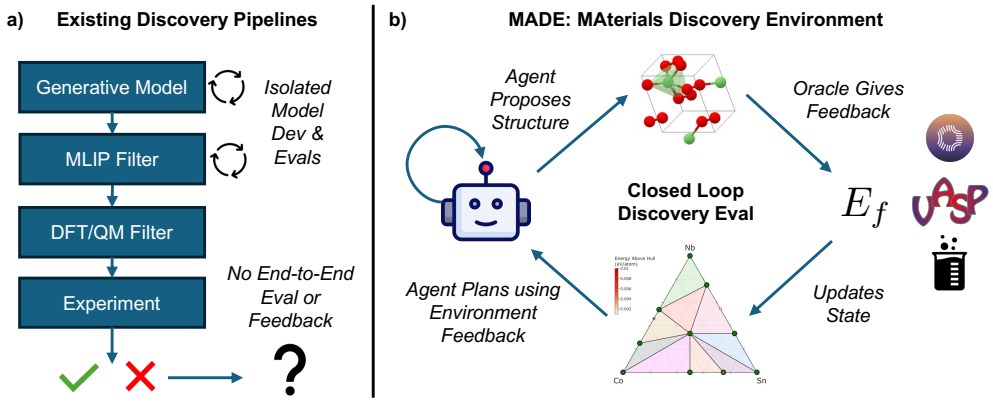


Figure 1: **a)** Existing discovery pipelines and benchmarks follow a static filtering process, moving sequentially from generative models to increasingly expensive evaluation methods, without end-to-end feedback. **b)** Our proposed benchmark MADE simulates a closed-loop discovery environment where agents iteratively propose candidates, receive oracle feedback, and update their strategy. The framework is modular, supporting different agents, oracles, and environments.

how well models rank candidates for filtering rather than how effectively agents learn and adapt in a closed design loop.

Recent advances in agentic systems, which include reinforcement learning in scientific domains, large language model (LLM) agents, and hybrid AI-scientist frameworks [24, 13, 30], highlight the potential for more adaptive, closed-loop discovery strategies. These systems can in principle integrate hypothesis generation, tool use, experimental design, and iterative refinement [4, 16, 15]. Yet the field lacks standardized benchmarks to evaluate how well they perform in realistic discovery settings. While there has been progress to create environments for evaluating LLM agents in domains like software engineering and general AI tasks [27, 17, 22], current evaluations of LLM agents for physical sciences typically test static analysis or question answering [44, 26, 2] rather than capabilities as iterative experiment designers and decision-makers.

To address this gap, we propose **MADE**: **MA**terials **D**iscovery **E**nvironment, a benchmark environment for autonomous discovery of thermodynamically stable materials. In MADE, an agent must propose candidate compounds, receive oracle feedback (e.g., formation energy), and adapt its strategy over multiple rounds. Success is measured by how efficiently the agent uncovers new stable phases relative to the convex hull of known compounds in a chemical system. The benchmark is general and modular, supporting different fidelity levels of ground truth—from machine-learned potentials to DFT to experimental datasets—making it extensible as new data and methods become available.

2 Related Work

Materials Discovery Benchmarks Benchmarking has a long history in computational materials science, from the creation of large-scale datasets such as the Materials Project and OQMD [18, 10]. These primarily focus on predictive tasks such as formation energy and band-gap regression. Recent efforts like Matbench Discovery [37] have shifted slightly toward discovery-oriented tasks, yet remain limited to predictive tasks on fixed datasets [43]. Generative models such as MatterGen [45], GNoME [25], and Chemeleon [32] efficiently generate structures, but their evaluation is typically a single batch filtration process, where many promising candidates are generated and then filtered for DFT or experimental evaluation. MADE seeks to bridge this gap by benchmarking agentic pipelines that simulate the discovery workflow end-to-end.

Agent Benchmarks Recent scientific benchmarks for agentic methods—such as DREAMS [44] and ChemBench [26]—primarily evaluate static tasks or predefined tool use, lacking the ability to measure iterative adaptation. While agentic discovery methods (e.g., AI-driven scientists [24, 30]) have been proposed, their evaluations remain use-case-specific without standardized environments. Conversely, iterative benchmarking frameworks from classical reinforcement learning gym environments [41],

to LLM environments [27, 17] demonstrate the value of dynamic interaction and feedback-driven evaluation.

Active learning and Bayesian Optimization Active learning, Bayesian optimization (BO), and related experimental design methods provide a principled framework for iterative decision-making under uncertainty, with the goal of improving search and optimization efficiency [40, 12, 35]. These methods combine surrogate models with acquisition functions that balance exploration and exploitation, and have been widely applied in materials science [23, 19, 38, 42, 29] to optimize properties in low-dimensional design spaces such as mapping phase diagrams. Unlike classical black-box optimization, which targets a single global optimum, materials discovery is inherently multi-modal, seeking a diverse set of local minima corresponding to stable or metastable compounds for experimental verification. MADE enables integration and evaluation of BO strategies within broader discovery pipelines.

3 MADE: A Dynamic Benchmark for Materials Discovery

We frame materials discovery as an interactive process between an agent and an environment, with the objective of identifying new crystal structures that are thermodynamically stable [5]. While here we focus on stability, the framework naturally extends to multi-objective settings. Rather than a conventional black-box optimization task targeting a single global optimum, we formulate discovery as an exploratory search over a structured chemical landscape, aiming to uncover a diverse set of local minima corresponding to distinct stable or metastable compounds under a limited oracle budget. The search space is discrete, sparse, and chemically constrained, requiring strategies that integrate prior knowledge and adapt dynamically to feedback. Figure 1b provides a graphical overview, and Algorithm 1 provides an algorithmic description of a rollout of an episode in the benchmark.

3.1 Problem Definition

Let S denote the chemical search space, where each candidate $s \in S$ is defined by its chemical composition and crystal structure. We assume access to an oracle $O : S \to \mathbb{R}$ which returns the predicted formation energy per atom E_s for a given structure s. Let $B \in \mathbb{N}$ denote the oracle query budget, and define $H_0 \subset S$ as the initial set of known reference materials.

Algorithm 1: MADE episode rollout

Input: Chemical search space S, initial materials H_0, policy π, oracle $O(\cdot)$, budget B, threshold ϵ

Initialize known materials $H \leftarrow H_0$
Evaluate energies $E_s = O(s),\ s \in H$
Construct convex hull $CH(H)$
for $t = 1, \ldots, B$ **do**
  $s_t \leftarrow \pi(\{(s, E_s) : s \in H\})$
  $E_t \leftarrow O(s_t)$
  $H \leftarrow H \cup \{s_t\}$
  Update $CH(H)$ and stable set S_{stable}
end
return S_{stable}

An agent is defined by its discovery policy π that depends on the history of observed (structure, energy) pairs, $\pi : \{(s_i, E_i)\}_{i=1}^{t-1} \to S$. At each iteration $t \leq B$, the agent selects the next candidate structure $s_t \sim \pi$, the oracle evaluates its energy, $E_t = O(s_t)$, and the candidate is added to the set of known materials. After updating $H_t = H_{t-1} \cup \{s_t\}$, the convex hull $CH(H_t)$ is recomputed. For each candidate $s \in H_t$, we calculate its energy above the convex hull as: $\Delta_{\text{hull}}(s, H_t) \in \mathbb{R}$. A material is considered thermodynamically stable if its energy lies on or below the convex hull, $S_{\text{stable},t} = \{s \in H_t \mid \Delta_{\text{hull}}(s, H_t) \leq \epsilon\}$, where ϵ is a small stability threshold [5].

The sequence of proposed materials is defined as: $Q_\pi = \{s_1, s_2, \ldots, s_B\} \subset S$. The objective is to design a policy π that maximizes the total number of new stable materials discovered after B queries: $\max_\pi |Q_\pi \cap S_{\text{stable},B}|$.

Evaluation Discovery policies are evaluated on both efficiency and diversity. The primary metric is the **number of novel stable compounds** discovered within a query budget B, defined by the convex hull criterion. To prevent data leakage, we propose constructing hold-out systems by searching the MLIP landscape for stable structures absent from MP. This is effective but increasingly difficult in high-dimensional spaces. Novelty is further enforced with structural matching (`pymatgen StructureMatcher` [31]), which avoids reward hacking via trivial perturbations of known phases.

For held-out chemical systems, we can additionally report precision and recall relative to the known stable set.

3.2 Example Implementations

Our framework is designed to be flexible enough to allow for varying levels of assumptions and complexity. To define an instantiation of the benchmark environment, one needs to define the chemical space S to explore, its initialization with known compounds (H_0) and an oracle O. Then one can evaluate different agent policies on the task. We provide pseudocode for each class in Appendix A, which mirrors a gym [41] environment for ease of implementation and extension.

Chemical System The chemical space for exploration S is defined by the constituent elements that make up the material. This allows for adjustable difficulty by varying system complexity (e.g. easy: binary metal oxides, medium: ternary intermetallic compounds, hard: quaternary and beyond). Initial known structures (H_0) can be retrieved from datasets (e.g. from Materials Project [14]). The user can decide whether to retain all known structures as H_0, simulating a real discovery campaign, or just a subset (e.g. end-member stable structures) for testing simpler algorithms.

Oracles For fast evaluation of policies, we can employ MLIP energy oracles. We note that MLIPs are relatively cheap to evaluate, and hence is why they are often used explicitly as a batch filtering step in modern discovery pipelines. This gives a good testbed for evaluating methods quickly, with the hope that methods developed here translate to when evaluation is expensive and thus decision making under uncertainty is crucial. Extensions to higher-fidelity evaluations (e.g., DFT, experimental validation) are readily accommodated in our framework to simulate a real discovery campaign.

Agents An agent's policy can be viewed as a combination of a *structure generator* (proposing candidates) and an *acquisition function* (selecting which to evaluate). In practice, these may be intertwined, particularly in the context of LLM agents orchestrating tools, but this separation is useful to enumerate possibilities and mirrors existing "generate-then-filter" discovery workflows (Figure 1a). **Structure generators** range from random sampling and classical search methods such as AIRSS [33] to modern generative models (e.g. MatterGen [45], GNoMe [25], Chemeleon [32]) or LLM-driven proposals [16]. **Acquisition strategies** include random selection, LLM-guided heuristics, diversity- or uncertainty-based search [3, 29], prototype substitution [31, 43]. With higher-fidelity oracles (e.g., DFT, experiments), lower-fidelity models (e.g., MLIPs) can also guide acquisition, as well as filters for chemical validity and synthesizability, mirroring real pipelines. This list is intended as illustrative rather than exhaustive; our goal is not to fix an implementation but to enable systematic comparison of diverse agent designs within a unified closed-loop benchmark.

4 Results

4.1 Experimental Setup

As a demonstration, we implement two generator baselines: *Random* and *MatterGen* [45]. Random structures were generated by first sampling a composition (up to a total of 12 atoms) randomly assigned to constituent elements, and placing atoms uniformly at random in the unit cell, with lattice parameters a, b, c sampled from the uniform distribution $U(3, 15)$ Å, and angles α, β, γ from $U(60, 120)$ degrees. MatterGen structures were sampled using the pretrained model conditioned on chemical system using a diffusion guidance parameter of 2.0.

We pair these generators with two simple acquisition functions: *Random* and *Diversity*. Random acquisition involves selecting structures uniformly at random from the generations. Diversity acquisition involves selecting structures furthest in composition space from those already observed.

We test these policies on two systems: a binary metal oxide (Na–O) and a ternary intermetallic (Co–Nb–Sn), using an ORB MLIP oracle [36] and a query budget of 20 structures. Further details on implementation are provided in Appendix B.1. Our code is available here.

4.2 Baseline Results

Table 1 shows the number of novel stable compounds found for each strategy, chemical system, and H_0. Further results and acquisition plots are presented in Appendix B.2.

Table 1: Average number of stable compounds discovered over 20 queries, with standard error over 5 episodes in parentheses. Results are reported for each chemical system under different initialization settings: using only elemental end-members (H_0: El.) or including all known stable Materials Project structures (H_0: MP).

Structure Generator	**Acquisition Function**	**Na-O** (H_0: El.)	**Na-O** (H_0: MP)	**Co-Nb-Sn** (H_0: El.)	**Co-Nb-Sn** (H_0: MP)
Random	Random	3.0 (0.5)	0.0 (0.0)	1.2 (0.8)	0.0 (0.0)
Random	Diversity	4.0 (0.9)	0.0 (0.0)	1.6 (0.6)	0.0 (0.0)
MatterGen	Random	4.0 (0.6)	0.8 (0.5)	3.2 (0.4)	0.2 (0.2)
MatterGen	Diversity	4.4 (0.7)	0.8 (0.4)	5.8 (0.4)	0.2 (0.2)

Generative models accelerate discovery We find that pretrained generative model policies sample a greater number of new stable structures than random generation for the same budget, particularly in higher-dimensional search spaces such as the ternary system. Since MatterGen was trained on Materials Project data, generating stable structures with respect to elemental end-members is expected. Discovering new stable structures beyond the Materials Project convex hull remains challenging. Naïve random search proves ineffective, but MatterGen occasionally identifies new minima in the MLIP energy landscape, illustrating the potential of generative priors for accelerating discovery.

Active acquisition strategies accelerate discovery We find that simple active acquisition strategies, such as the diversity-based criterion, already improve discovery efficiency across datasets and generative models. This highlights the potential of more sophisticated adaptive experiment design approaches such as those leveraging prior knowledge, uncertainty and information gain to further enhance sample efficiency and guide exploration.

5 Discussion

Limitations and Ongoing Work We note that these are preliminary results aimed to show a minimal example of the benchmark proposal in action. We are actively extending these experiments to more chemical systems and policies, including LLM-based agents. We are also looking to include better metrics for evaluating uniqueness and novelty beyond `StructureMatcher` which better account for structural diversity [28], as well as leveraging ideas from Bayesian optimization to measure the magnitude of improvement of a discovery pipeline over a baseline policy [38, 21, 1].

Extensions This benchmark is intentionally dynamic and extensible, designed to evolve over time to allow assumptions to be progressively relaxed. We envisage several natural next steps. First, integrating **higher-fidelity oracles** such as DFT, free-energy calculations, and experimental validations will allow more realistic evaluation of agents, enabling lower-fidelity models (e.g., MLIPs) to guide decision-making. Second, extending the benchmark to **multi-objective optimization** tasks, considering additional target properties alongside stability, will better reflect realistic discovery campaigns. Third, implementing **batch-mode evaluation**, allowing simultaneous queries, would align the benchmark more closely with experimental workflows. Fourth, we foresee it being possible to train agents using **reinforcement learning** using this environment.

Outlook MADE enables the study of key challenges in computational materials discovery. Searching discrete, structurally complex spaces is difficult, as methods like Bayesian optimization struggle with combinatorial growth and structural validity. LLM systems may be able to better handle the exploration exploitation trade off. Reliable uncertainty estimation is also essential, as, by definition, new discoveries will be out of training distributions. By shifting focus from static prediction tasks to full discovery workflows, we hope MADE encourages the community to adopt dynamic benchmarks that better measure and accelerate autonomous materials discovery.

References

[1] A. D. Adesiji, J. Wang, C.-S. Kuo, and K. A. Brown. Benchmarking self-driving labs. *arXiv preprint arXiv:2508.06642*, 2025.

[2] N. Alampara, M. Schilling-Wilhelmi, M. Ríos-García, I. Mandal, P. Khetarpal, H. S. Grover, N. A. Krishnan, and K. M. Jablonka. Probing the limitations of multimodal language models for chemistry and materials research. *Nature Computational Science*, pages 1–10, 2025.

[3] A. Anelli, E. A. Engel, C. J. Pickard, and M. Ceriotti. Generalized convex hull construction for materials discovery. *Physical Review Materials*, 2(10):103804, 2018.

[4] S. Badrinarayanan, R. Magar, A. Antony, R. S. Meda, and A. B. Farimani. Mofgpt: Generative design of metal-organic frameworks using language models. *arXiv preprint arXiv:2506.00198*, 2025.

[5] C. J. Bartel. Review of computational approaches to predict the thermodynamic stability of inorganic solids. *Journal of Materials Science*, 57(23):10475–10498, 2022.

[6] I. Batatia, P. Benner, Y. Chiang, A. M. Elena, D. P. Kovács, J. Riebesell, X. R. Advincula, M. Asta, M. Avaylon, W. J. Baldwin, et al. A foundation model for atomistic materials chemistry. *arXiv preprint arXiv:2401.00096*, 2023.

[7] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch. Structural relaxation made simple. *Physical review letters*, 97(17):170201, 2006.

[8] C. Chen and S. P. Ong. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, 2022.

[9] B. Deng, P. Zhong, K. Jun, J. Riebesell, K. Han, C. J. Bartel, and G. Ceder. Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, 5(9):1031–1041, 2023.

[10] A. Dunn, Q. Wang, A. Ganose, D. Dopp, and A. Jain. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Computational Materials*, 6(1):138, 2020.

[11] X. Fu, B. M. Wood, L. Barroso-Luque, D. S. Levine, M. Gao, M. Dzamba, and C. L. Zitnick. Learning smooth and expressive interatomic potentials for physical property prediction. *arXiv preprint arXiv:2502.12147*, 2025.

[12] R. Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

[13] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[14] M. K. Horton, P. Huck, R. X. Yang, J. M. Munro, S. Dwaraknath, A. M. Ganose, R. S. Kingsbury, M. Wen, J. X. Shen, T. S. Mathis, et al. Accelerated data-driven materials science with the materials project. *Nature Materials*, pages 1–11, 2025.

[15] T. J. Inizan, S. Yang, A. Kaplan, Y.-h. Lin, J. Yin, S. Mirzaei, M. Abdelgaid, A. H. Alawadhi, K. Cho, Z. Zheng, et al. System of agentic ai for the discovery of metal-organic frameworks. *arXiv preprint arXiv:2504.14110*, 2025.

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

[17] C. E. Jimenez, J. Yang, A. Wettig, S. Yao, K. Pei, O. Press, and K. Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.

[18] S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl, and C. Wolverton. The open quantum materials database (oqmd): assessing the accuracy of dft formation energies. *npj Computational Materials*, 1(1):1–15, 2015.

[19] A. G. Kusne, H. Yu, C. Wu, H. Zhang, J. Hattrick-Simpers, B. DeCost, S. Sarker, C. Oses, C. Toher, S. Curtarolo, et al. On-the-fly closed-loop materials discovery via bayesian active learning. *Nature communications*, 11(1):5966, 2020.

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

[21] Q. Liang, A. E. Gongora, Z. Ren, A. Tiihonen, Z. Liu, S. Sun, J. R. Deneault, D. Bash, F. Mekki-Berrada, S. A. Khan, et al. Benchmarking the performance of bayesian optimization across multiple experimental materials science domains. *npj Computational Materials*, 7(1):188, 2021.

[22] X. Liu, H. Yu, H. Zhang, Y. Xu, X. Lei, H. Lai, Y. Gu, H. Ding, K. Men, K. Yang, et al. Agentbench: Evaluating llms as agents. In *ICLR*, 2024.

[23] T. Lookman, P. V. Balachandran, D. Xue, and R. Yuan. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Computational Materials*, 5(1):21, 2019.

[24] C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

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

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

[27] D. Nathani, L. Madaan, N. Roberts, N. Bashlykov, A. Menon, V. Moens, A. Budhiraja, D. Magka, V. Vorotilov, G. Chaurasia, et al. MLGym: A new framework and benchmark for advancing ai research agents. *arXiv preprint arXiv:2502.14499*, 2025.

[28] M. Negishi, H. Park, K. O. Mastej, and A. Walsh. Continuous uniqueness and novelty metrics for generative modeling of inorganic crystals. *arXiv preprint arXiv:2510.12405*, 2025.

[29] A. Novick, D. Cai, Q. Nguyen, R. Garnett, R. Adams, and E. Toberer. Probabilistic prediction of material stability: integrating convex hulls into active learning. *Materials Horizons*, 11(21):5381–5393, 2024.

[30] A. Novikov, N. Vũ, M. Eisenberger, E. Dupont, P.-S. Huang, A. Z. Wagner, S. Shirobokov, B. Kozlovskii, F. J. Ruiz, A. Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

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

[32] H. Park, A. Onwuli, and A. Walsh. Exploration of crystal chemical space using text-guided generative artificial intelligence. *Nature Communications*, 16(1):4379, 2025.

[33] C. J. Pickard and R. Needs. Ab initio random structure searching. *Journal of Physics: Condensed Matter*, 23(5):053201, 2011.

[34] K. Popper. *The logic of scientific discovery*. Routledge, 2005.

[35] T. Rainforth, A. Foster, D. R. Ivanova, and F. Bickford Smith. Modern bayesian experimental design. *Statistical Science*, 39(1):100–114, 2024.

[36] B. Rhodes, S. Vandenhaute, V. Šimkus, J. Gin, J. Godwin, T. Duignan, and M. Neumann. Orb-v3: atomistic simulation at scale. *arXiv preprint arXiv:2504.06231*, 2025.

[37] J. Riebesell, R. E. Goodall, P. Benner, Y. Chiang, B. Deng, G. Ceder, M. Asta, A. A. Lee, A. Jain, and K. A. Persson. A framework to evaluate machine learning crystal stability predictions. *Nature Machine Intelligence*, 7(6):836–847, 2025.

[38] B. Rohr, H. S. Stein, D. Guevarra, Y. Wang, J. A. Haber, M. Aykol, S. K. Suram, and J. M. Gregoire. Benchmarking the acceleration of materials discovery by sequential learning. *Chemical science*, 11(10):2696–2706, 2020.

[39] A. N. Rubungo, K. Li, J. Hattrick-Simpers, and A. B. Dieng. Llm4mat-bench: benchmarking large language models for materials property prediction. *Machine Learning: Science and Technology*, 6(2):020501, 2025.

[40] B. Settles. From theories to queries: Active learning in practice. In *Active learning and experimental design workshop in conjunction with AISTATS 2010*, pages 1–18. JMLR Workshop and Conference Proceedings, 2011.

[41] M. Towers, A. Kwiatkowski, J. Terry, J. U. Balis, G. De Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

[42] A. Wang, H. Liang, A. McDannald, I. Takeuchi, and A. G. Kusne. Benchmarking active learning strategies for materials optimization and discovery. *Oxford Open Materials Science*, 2(1):itac006, 2022.

[43] H.-C. Wang, S. Botti, and M. A. Marques. Predicting stable crystalline compounds using chemical similarity. *npj Computational Materials*, 7(1):12, 2021.

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

[45] C. Zeni, R. Pinsler, D. Zügner, A. Fowler, M. Horton, X. Fu, Z. Wang, A. Shysheya, J. Crabbé, S. Ueda, et al. A generative model for inorganic materials design. *Nature*, 639(8055):624–632, 2025.

A Implementation Details

A.1 MADE Pseudocode

Listing 1 contains pseudocode for the base classes of the MADE benchmark. The Oracle, Environment and Agent base classes are flexible to allow for different methods to be implemented. We make use of `pymatgen` [31] classes to use phase diagrams as the environment state and convex hull computations.

Listing 1: Pseudocode for key classes in MADE

```
class Oracle:
    def __init__(self, model):
        self.model = model  # e.g., MLIP, DFT, experimental oracle

    def predict_energy(self, structure):
        energy = self.model.predict(structure)
        return energy


class Environment:
    def __init__(self, oracle, initial_known_structures,
        chemical_system):
        self.oracle = oracle
        self.chemical_system = chemical_system
        self.known_structures = initial_known_structures
        self.energies = {s: self.oracle.predict_energy(s)
                         for s in initial_known_structures}
        self.update_convex_hull()

    def step(self, structure):
        energy = self.oracle.predict_energy(structure)
        self.known_structures.append(structure)
        self.energies[structure] = energy
        self.update_convex_hull()
        return energy

    def update_convex_hull(self):
        self.convex_hull = ConvexHull(self.known_structures, self.
            energies)

    def reset(self):
        self.known_structures.clear()
        self.energies.clear()
        self.update_convex_hull()


class Agent:
    def __init__(self, chemical_system):
        self.chemical_system

        self.policy = Policy(chemical_system)

    def predict_next_structure(self, known_structures, energies):
        next_structure = self.policy(known_structures, energies)
        return next_structure


oracle = Oracle(model)
env = Environment(oracle, initial_known_structures, chemical_system)
agent = Agent(chemical_system)

for t in range(query_budget):
    structure = agent.predict_next_structure(env.known_structures,
                                             env.energies)
    energy = env.step(structure)
```

B Further Information on Baseline Experiment

In Section 4 we show results for two chemical system datasets. We present implementation details for this experiment in Section B.1 and show more detailed results in Section B.2.

B.1 Implementation Details

B.1.1 Environment

Oracle We use `orb-v3-conservative-inf-omat` from the ORB MLIP family [36] as our formation energy oracle. All structures were relaxed (including unit cell parameters) using the FIRE optimizer [7] for a maximum of 100 steps or a maximum total projected force on the atoms (fmax) of 0.01 in `ase` [20] before evaluating the potential energy.

Chemical System and Initial Structures Ground truth structures for elemental end-member structures and other stable structures were retrieved from mixed GGA and GGA+U functionals using the Materials Project API [14]. The energies of these structures were recomputed using the Oracle and saved as the initial environment state. A stability tolerance $\epsilon = 0.01$ eV/atom was used to classify newly proposed structures as stable with respect to the convex hull. `pymatgen StructureMatcher` was used to check whether proposed structures were novel.

B.1.2 Structure Generators

For both generators, we sample 8 candidates before using the acquisition function to choose which one to evaluate.

Random Structures were generated by first sampling a composition. First, the total number of atoms in the composition was sampled up to a maximum of 12, then these were randomly distributed to the constituent elements. Then these atoms were placed uniformly at random in the unit cell in fractional coordinates. The lattice parameters a, b, c were sampled from the uniform distribution $U(3, 15)$ Å, and angles α, β, γ from $U(60, 120)$ degrees.

MatterGen We use the default MatterGen command line interface from the GitHub repository to generate structures[2]. We use the pretrained `chemical_system` model to condition on the chemical system in question, and use a diffusion guidance parameter of 2.0.

B.1.3 Acquisition Functions

Random We choose from the candidates generated uniformly at random.

Diversity For each candidate, we compute the Euclidean distance in (normalized) composition space to all of the previously observed structures (stable or previously tried). We then choose the candidate with the maximum minimum distance to a previously seen structure. This biases sampling in chemical compositions not previously explored.

B.2 Additional Results

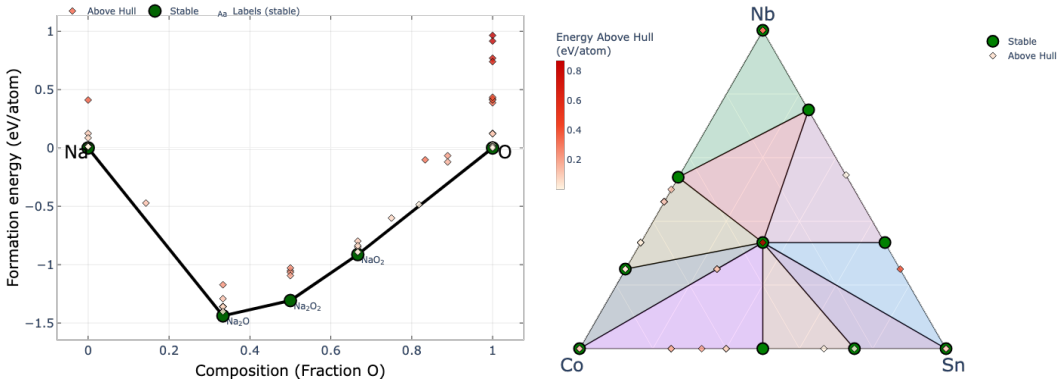
Number of Unique Proposed Structures Table 2 shows the number of unique compounds proposed by each policy (using `pymatgen StructureMatcher` [31]). Naturally, we find that using MatterGen as a structure generator sometimes leads to the same structures being proposed, as it has been trained on structures in this family. This can be mitigated by using an acquisition function that accounts for the structures seen already, such as *Diversity*.

Ground Truth Phase Diagrams Figure 2 shows the ground truth phase diagrams for the chemical systems in the experiments from Materials Project [14].

[2] https://github.com/microsoft/mattergen/releases/tag/v1.0.3

Table 2: Baseline results showing the average number of unique compounds proposed over 20 queries, with standard deviation over 5 episodes in parentheses.

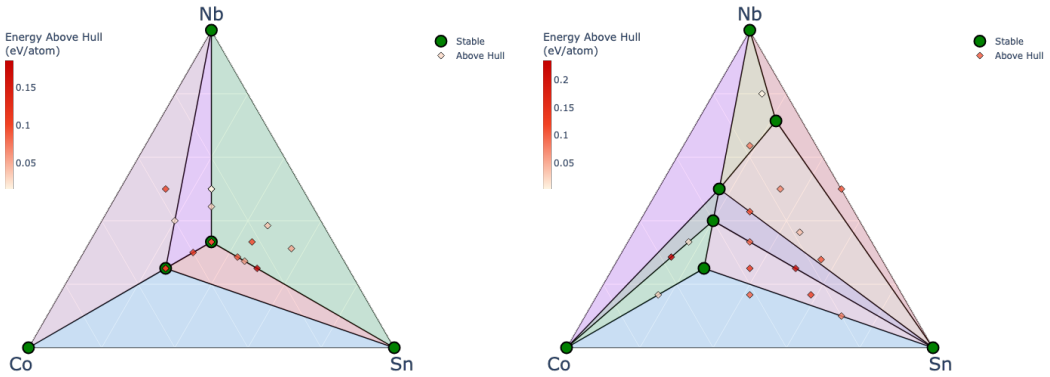
Structure Generator	**Acquisition Function**	**Na-O** (H_0: El.)	**Na-O** (H_0: MP)	**Co-Nb-Sn** (H_0: El.)	**Co-Nb-Sn** (H_0: MP)
Random	Random	19.8 (0.4)	19.8 (0.4)	19.8 (0.4)	20.0 (0.0)
Random	Diversity	20.0 (0.0	20.0 (0.0)	20.0 (0.0)	20.0 (0.0)
MatterGen	Random	18.0 (2.2)	17.8 (1.3)	19.6 (0.5)	19.0 (1.3)
MatterGen	Diversity	19.8 (0.4)	19.2 (0.7)	20.0 (0.0)	20.0 (0.0)



(a) Na-O. (b) Co-Nb-Sn.

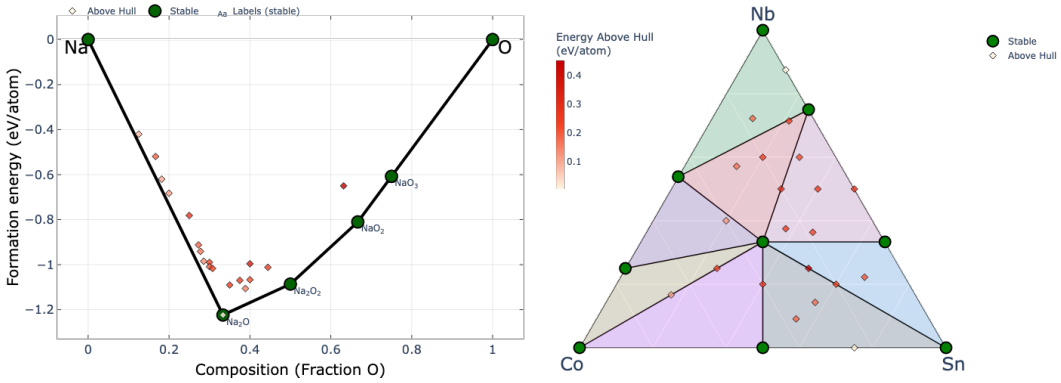
Figure 2: Ground truth phase diagrams from Materials Project [10].

Example Observed Phase Diagrams Here we show example phase diagrams generated from the structures sequentially proposed by the baseline policies. We see that the diversity acquisition strategy samples the space more effectively, often leading to more stable structures found in Figure 3, and even finds a few new stable structures (within tolerance) on the Materials Project convex hull (Figure 4). Random structure generation does not perform as well, often producing unstable structures (Figure 5), highlighting the need for intelligent agents.



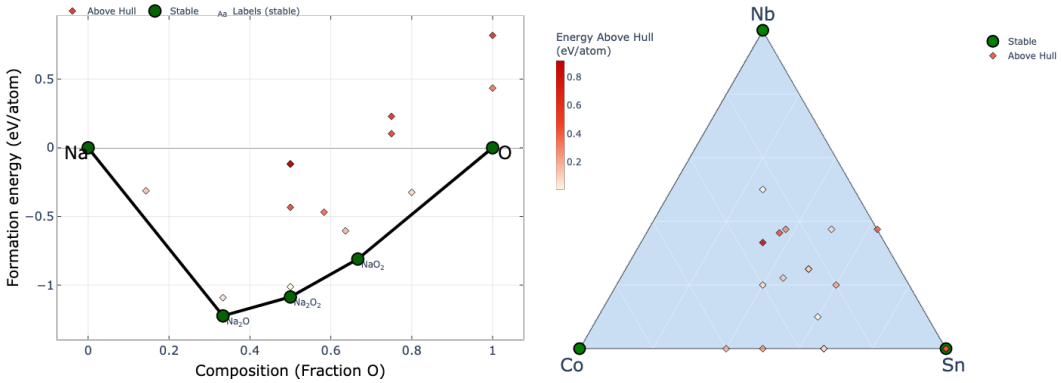
(a) Random acquisition. (b) Diversity acquisition.

Figure 3: MatterGen acquired phase diagrams starting from elemental end-members in Co-Nb-Sn. Diversity acquisition samples the space more effectively.



(a) Na-O. (b) Co-Nb-Sn.

Figure 4: MatterGen with Diversity finds new (within tolerance) stable structures starting from Materials Project convex hull (Figure 2).



(a) Na-O, starting from Materials Project convex hull. (b) Co-Nb-Sn, starting from elemental end-members.

Figure 5: Random structure generation does not find new stable structures.