Auto Research for Materials: Auditable AI-Scientist Workflows with Held-Out Transfer

Jingjie Ning [*1], Xiaochuan Li [1], Shanshan Zhong [1], Ji Zeng [1], and Guolin Ke [2]

[1]School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
[2]DP Technology, Beijing, China

July 21, 2026

Abstract

An AI research agent can improve the score it sees without finding a modelling change that works on new materials. We ask a stricter question. After repeated experiments, does the selected change survive on data that never entered the loop, and can its code be reused? We separate the search into changes to features, models, representations, and training data. Seven searches produce 701 evaluated changes across ten Matbench endpoints. Agents receive only the mean over five inner folds, reducing reliance on any single development split. We then freeze the selected code and evaluate it once on an untouched holdout. Nine of ten choices remain the best tested single intervention. The surviving changes reveal two materials modelling regimes. With composition alone, feature, model, and representation changes provide comparable routes to improvement. They include held-out MAE reductions of 17.4% for band gap and 18.6% for steel strength, as well as gains on both classification endpoints, while screened external data adds little. For structure tasks, richer geometry descriptors and model or calibration changes lower mean held-out MAE by 14.6% and 7.1% and lead on different property families, whereas composition embeddings do not transfer. Combining separately found feature and model changes yields a 26.3% mean held-out improvement. These results show in materials prediction that closed-loop agents can produce decisions that survive unseen evidence and code changes that can be reused across tasks and combined. More broadly, they provide an evaluation design for testing executable discoveries beyond the feedback loop.

Keywords: Auto Research; AI scientist; materials informatics; transferable machine learning; closed-loop experimentation; adaptive data analysis.

1 Introduction

A closed-loop research agent can produce an impressive sequence of improving scores without producing a modelling change that survives outside the loop. The useful output is not the search trajectory itself. It is an attributable change that works on materials the agent never evaluated. We use *Auto Research* for a loop in which a language-model agent proposes a hypothesis, edits source code, runs an experiment, reads the result, and decides what to try next.[1] Every idea must become executable and face empirical feedback.

*Corresponding authors: jening@cs.cmu.edu, kegl@dp.tech.

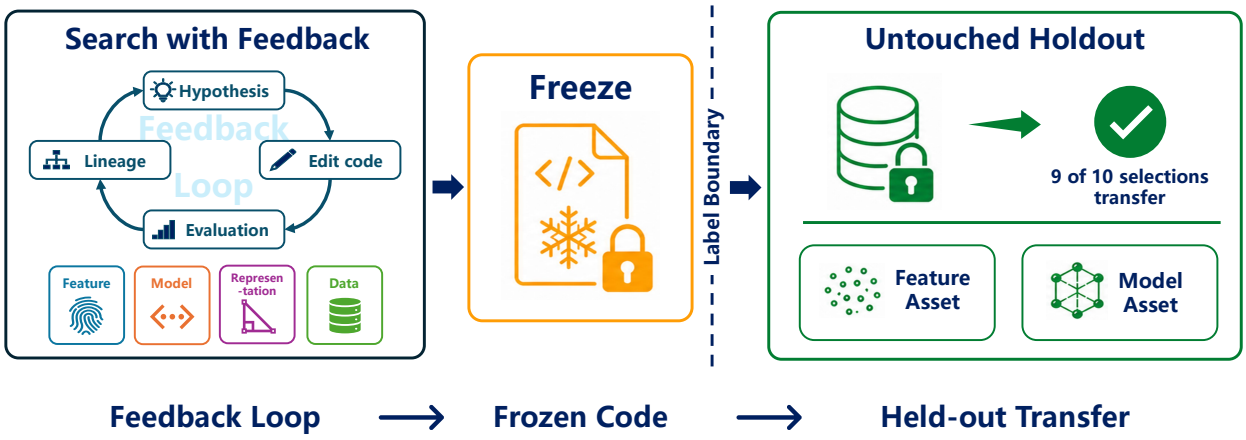

Figure 1: **Auto Research separates search from final evidence.** The agent searches feature, model, representation, and data interventions using the mean over five inner training folds. This feedback is not controlled by one development split. The selected code is frozen before evaluation on inaccessible holdout labels. Nine of ten single-intervention choices remain best on the holdout, and the searches produce reusable feature and model changes.

Repeated feedback creates two problems. A pipeline may improve without revealing whether the gain came from its features, model, representation, or data. Hundreds of choices made against the same development score can also overfit that score.[2–4] We address both problems with a simple research design. Each search changes only one intervention class and receives the mean score across five folds inside the training set. Averaging across these folds reduces the influence of any one development split on the next experiment. The selected code is then frozen before the holdout is evaluated. Figure 1 shows this separation.

We test this design through seven searches that produce 701 evaluated changes on ten Matbench endpoints. The experiments ask whether a search choice remains best on unseen data and whether the frozen code remains useful across endpoints or in combination. Nine of ten choices pass the first test. The choices that transfer reveal two regimes. Composition-only prediction supports several routes to improvement, while structure-based prediction separates the roles of geometry descriptors and predictive models. The same feature and model code changes remain useful across property families and can be combined into a stronger predictor. The study therefore evaluates Auto Research through the executable changes it produces, rather than through agent activity or the highest score observed inside the loop.

2 Related work

Materials property prediction. Matbench provides standard tasks and evaluation splits for inorganic property prediction.[5] Magpie composition statistics,[6] available through matminer,[7] remain a strong and interpretable basis for tabular models. MODNet combines feature selection with representation learning and is an important reference on smaller Matbench tasks.[8] Formula-based networks such as Roost and CrabNet learn from stoichiometry,[9,10] while CGCNN, MEGNet, and ALIGNN use atomic coordinates and bonding geometry.[11–13] We use these representation families to ask which type of modelling change remains useful after closed-loop selection.

Benchmarks and reliable selection. JARVIS-Leaderboard and Matbench Discovery broaden materials benchmarking toward reproducible workflows and stability screening.[14,15] Materials evaluation studies also stress data quality, construct validity, and transparent measurement choices.[16] Repeated model selection can bias a development estimate, especially when the next experiment depends on earlier results.[2–4] Our methodology places the separation inside the research loop by reserving one inaccessible holdout and treating official Matbench cross-validation as later benchmark context.

Autonomous research. The AI Scientist and related code-search methods generate ideas, edit programs, and evaluate their outputs.[1,17–20] Executable research loops have also been used to evolve training recipes and molecular predictors.[21,22] In materials science, LLMatDesign, SparksMatter, and MASTER connect language-model reasoning to design or simulation tools.[23–25] A-Lab and Coscientist close related loops through physical experiments.[26,27] Large-scale generators such as GNoME and MatterGen address candidate generation and stability.[28,29] Prior work rarely tests whether an agent-selected code change survives unseen evidence or remains useful beyond the task that selected it.

Contribution. Prior work shows that agents can search, code, and experiment. We evaluate the frozen executable intervention that remains after this activity. Four intervention classes make the source of improvement attributable, while the inner five-fold mean provides feedback that is less dependent on one development split. An inaccessible holdout then tests whether the choice survives search. Applying this design reveals that the available materials input shapes which interventions remain useful. Separately found code changes can also remain useful across endpoints and combine into a stronger model. The result is both a materials modelling study and an evaluation design for closed-loop AI science.

3 Methodology

The object we evaluate is a frozen code change, not the amount of agent activity or its highest in-loop score. The analysis asks whether the search-selected intervention survives untouched labels and whether the same code remains useful across endpoints or when combined with another separately found change.

3.1 Four intervention classes

We divide a materials prediction pipeline into four intervention classes. A *feature* intervention changes hand-crafted descriptors. A *model* intervention changes the estimator or calibration. A *representation* intervention changes learned or embedding-based inputs. A *data* intervention changes the external training rows admitted by the pipeline. Each search explores only one class. All other source files are restored from the baseline before evaluation and verified by hash. This design assigns each measured gain to a specific kind of research change.

3.2 Closed-loop research

Each trial follows the same loop. The agent proposes one hypothesis, edits the permitted code, runs the fixed evaluator, and reads the score. The complete record of earlier hypotheses, code changes, results, and failures informs the next experiment.

3.3 Search signal and code freeze

The agent sees one *search signal*. It is the mean of five-fold cross-validation with shuffle seed 42, computed only inside the Matbench fold-0 training partition. Each candidate is therefore evaluated across five development partitions before the agent chooses its next experiment. Averaging these scores reduces dependence on any one partition and gives adaptive search a less split-specific ranking signal. The fold-0 test labels are never loaded into an agent process. At the end of each search, we freeze the code version with the highest mean normalised improvement over the tasks evaluated in that search. For each endpoint, the search signal then chooses among the baseline and the frozen class-specific versions. The holdout is used in neither step. We propose training-only multi-fold feedback as a practical default for Auto Research when repeated fitting is affordable.

3.4 Held-out evaluation

The frozen code is fitted on the complete fold-0 training partition and evaluated once on the untouched fold-0 test. We call a choice *selection transfer* when the single intervention chosen by the search signal is also the best tested single intervention on this holdout. We separately run the frozen configurations through official Matbench five-fold cross-validation to compare with published references. This *benchmark replay* is reported for context because folds 1–4 contain examples used during the fold-0 search.

After the primary single-intervention analysis, we combine separately found changes. These combinations are formed only after the individual results are known. They test whether frozen discoveries work together and do not enter the nine-of-ten selection-transfer count.

3.5 Label isolation and audit record

Agent-written code receives training labels and unlabeled evaluation inputs, while only the fixed evaluator can access test labels and compute metrics. External rows are screened against training and test compositions by reduced formula, and same-source duplicates are removed before sampling. Each trial stores its hypothesis, code difference, score, runtime, status, parent trial, and source hash. This record makes both accepted and rejected experiments replayable.

3.6 Materials tasks and experimental details

The composition study uses four Matbench endpoints.[5] Experimental band gap and steel yield strength use MAE. Metallicity and glass-forming ability use ROC-AUC. The baseline combines 132 Magpie composition descriptors[6,7] with CatBoost.[30] Four searches cover the feature, model, representation, and data classes separately. Representation candidates include fraction-weighted mat2vec, MEGNet-derived, and one-hot elemental embeddings.[12,31]

The structure study uses six MAE endpoints. They cover phonons, bulk and shear moduli, refractive index in the Matbench dielectric endpoint, two-dimensional exfoliation energy, and perovskite formation energy. The baseline adds a light density and symmetry block to Magpie, giving 140 inputs. Three searches cover richer structure descriptors, model and calibration choices, and composition embeddings. We study external data on composition tasks, where overlap can be screened exactly by reduced formula. A controlled representation comparison also holds CatBoost fixed while adding composition embeddings or nine light structure descriptors.

For task i with baseline score b_i and candidate score s_i, normalised improvement is $(b_i - s_i)/b_i$ for MAE and $(s_i - b_i)/b_i$ for ROC-AUC. We report the unweighted mean across the tasks evaluated in each study. Each intervention class is searched once with the same seed and trial budget. Experiments

ran under Python 3.12 on a 32-vCPU AWS c7a.8xlarge instance. Structure featurisation and model fitting used CPUs. Agent sessions allowed at most 200 turns and an 8000-token reasoning budget.

3.7 Agent and author roles

DeepSeek-v4-pro generated hypotheses and code changes between 24 June and 8 July 2026. It received only evaluator-provided search signals. OpenAI Codex later assisted code review and manuscript revision and did not generate experimental measurements. The authors recomputed the reported aggregates, checked the archived code and records, and remain responsible for the study.

4 Results

4.1 Search choices usually survive unseen data

Figure 2 gives the primary result. On the four composition tasks, the search signal selects model changes for band gap and glass formation, a representation change for steel strength, and a feature change for metallicity. Each remains the best tested single intervention on the holdout. On the six structure tasks, the signal selects richer features for phonons, two-dimensional exfoliation energy, and perovskite formation energy. It selects the model change for both elastic moduli and refractive index. Five of these six choices remain best on the holdout.

The one disagreement is the two-dimensional exfoliation endpoint, `matbench_jdft2d`. Feature search improves the signal MAE from 42.85 to 38.79, while the 128-case holdout favours the baseline at 25.53 MAE rather than the selected feature code at 26.11. Across all ten endpoints, the training-set choice transfers in nine.

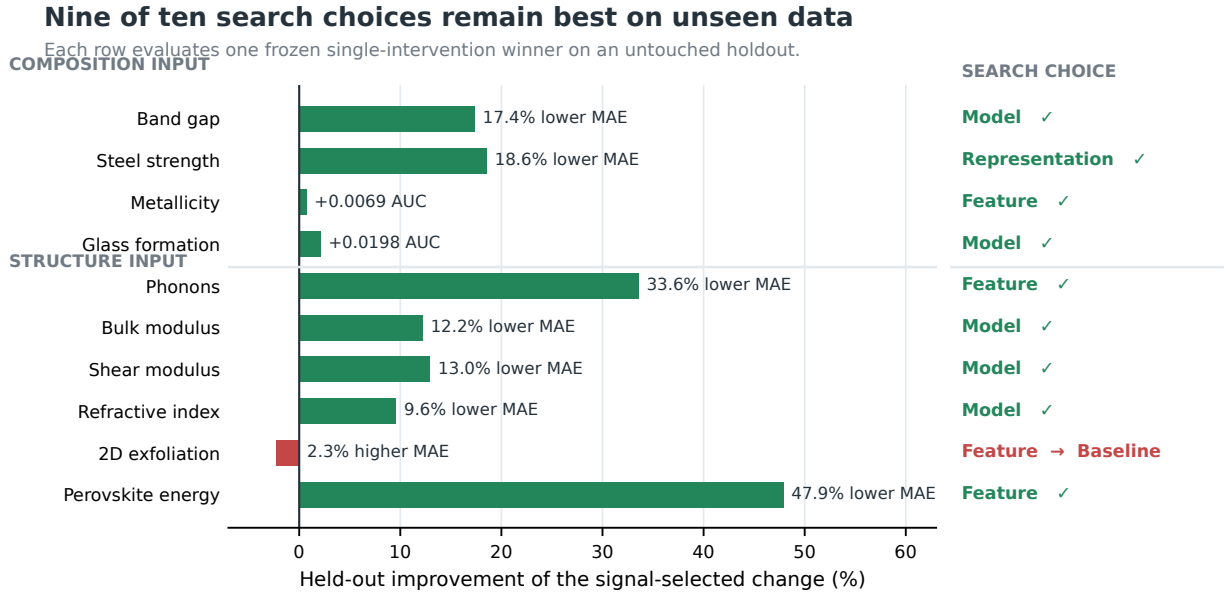

Figure 2: **Search choices usually remain useful on unseen data.** Bars show held-out effects of choices made by inner five-fold feedback. Labels give relative MAE reduction or absolute AUC change, and the right column marks whether each choice remains best.

Official Matbench replay also ranks the search-selected class first on all ten tasks. Because

folds 1–4 reuse samples from the search pool, this agreement is benchmark context rather than independent evidence. The untouched fold-0 test supplies the independent result in Figure 2.

4.2 Composition offers several routes to improvement

The transfer result answers whether the choices survive. We next ask what they teach us about materials prediction. The four composition tasks do not point to one universal recipe. Table 1 shows that different endpoints favour model, feature, or representation changes. The selected changes lower held-out MAE by 17.4% for experimental band gap and 18.6% for steel strength. They raise ROC-AUC by 0.0069 for metallicity and 0.0198 for glass formation.

Table 1: **Composition tasks support different successful interventions. Bold** marks the class chosen from the search signal. It is also the best tested single intervention on each holdout. SIGNAL is inner five-fold CV, HELD-OUT is the untouched test, and OFFICIAL-CV is the official five-fold replay. The embedding column is the representation intervention. MODNet is a published official-fold reference.[8]

task	evaluation	baseline	feature	embedding	model	data	MODNet
band gap (MAE↓)	SIGNAL	0.4645	0.4143	0.4640	**0.3973**	0.4572	n/a
	HELD-OUT	0.4332	0.3691	0.4317	**0.3579**	0.4269	n/a
	OFFICIAL-CV	0.450	0.395	0.448	**0.379**	0.445	0.333
steel strength (MAE↓)	SIGNAL	97.46	86.71	**79.07**	94.05	96.14	n/a
	HELD-OUT	115.6	103.7	**94.16**	110.9	116.5	n/a
	OFFICIAL-CV	98.7	86.6	**80.1**	95.2	99.8	87.76
metallicity (AUC↑)	SIGNAL	0.9662	**0.9761**	0.9662	0.9709	0.9667	n/a
	HELD-OUT	0.9735	**0.9804**	0.9735	0.9782	0.9738	n/a
	OFFICIAL-CV	0.969	**0.978**	0.969	0.974	0.969	0.916
glass formation (AUC↑)	SIGNAL	0.9274	0.9321	0.9326	**0.9419**	0.9279	n/a
	HELD-OUT	0.9476	0.9534	0.9494	**0.9673**	0.9483	n/a
	OFFICIAL-CV	0.936	0.942	0.941	**0.952**	0.937	0.960

4.3 The available input changes what works

Figure 3 compares the mean effect of each intervention class and reveals two regimes. Composition inputs support several routes to improvement. Feature, model, and representation changes each produce about 5–7% mean improvement across search, holdout, and benchmark replay. The screened data changes remain near zero. Structure inputs produce a sharper separation. Richer descriptors lead, model changes provide a second transferable route, and added composition embeddings fail to improve the holdout.

Screened external data. The data search evaluates 53 external-data interventions. Formula and source screening admits 98 rows for band gap, 18 for steels, 12 for glass, and 7 for metallicity. The selected data change improves the mean search signal by 0.76%, the holdout by 0.20%, and benchmark replay by 0.03%. The screened additions therefore contribute much less than changes to features, models, or representations in these tasks.

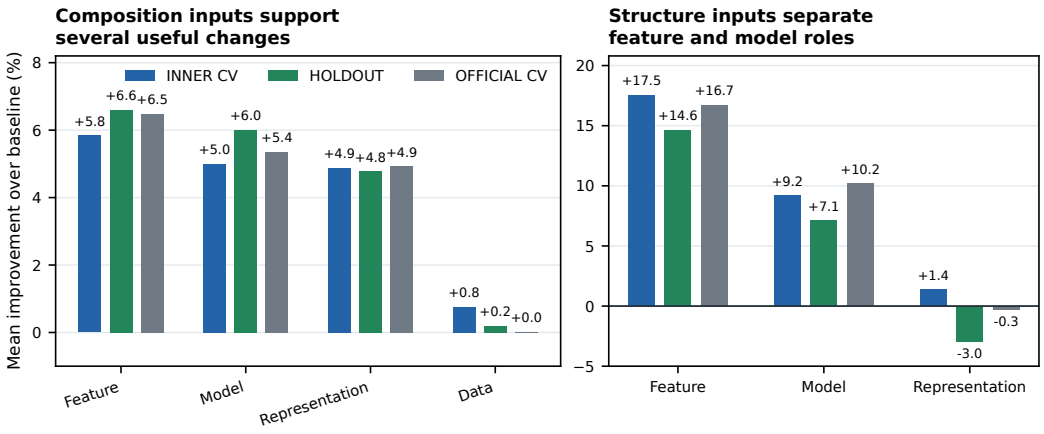

Figure 3: **The useful intervention depends on the available input.** Composition tasks benefit from feature, model, and representation changes. With structure inputs, richer descriptors and model changes transfer, while composition embeddings reverse from a small search gain to a held-out loss.

4.4 Crystal structure changes the intervention hierarchy

We first compare a formula-only Magpie and CatBoost baseline with selected published Matbench references. Its MAE is 2.30 times the MEGNet reference on phonons, 1.70 times the coNGN reference on bulk modulus, and 1.22 times the MODNet reference on refractive index.[8,12,32,33] The first two reference models use crystal structures. The refractive-index reference uses composition-based MODNet.

We then hold CatBoost fixed and change only its input. Adding mat2vec and MEGNet-derived composition embeddings changes phonon MAE from 66.05 to 79.26 and bulk-modulus MAE from 0.0837 to 0.0847. Adding nine light structure descriptors instead lowers the two errors to 54.98 and 0.0655. These correspond to reductions of 17% and 22% relative to the formula baseline.

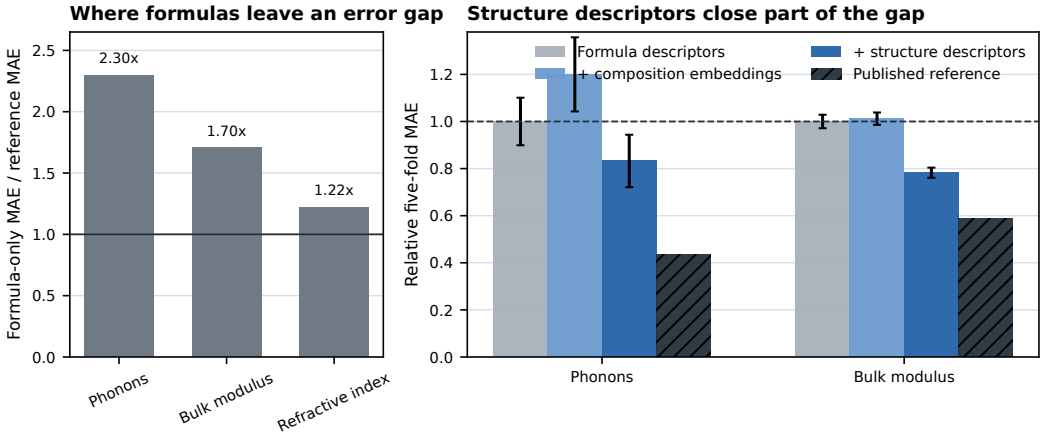

Figure 4: **Structure information reduces errors that richer formula embeddings do not.** The left panel compares the formula-only baseline with selected published references. The right panel holds CatBoost fixed. Added composition embeddings do not reduce error, while nine light structure descriptors improve both tasks. The plotted values use official five-fold evaluation.

4.5 Features and models specialize by property family

The closed-loop searches start from a baseline that already contains light structure information. Table 2 shows the next layer of improvement. Richer descriptors are the strongest choice for phonons and perovskite formation energy. The model and calibration code is the strongest choice for both elastic moduli and refractive index. The same feature code and model code are reused across these property families.

Table 2: **Feature and model changes lead on different structure tasks.** All tasks use MAE. **Bold** marks the best tested value in each row. SIGNAL is inner five-fold CV, HELD-OUT is the untouched test, and OFFICIAL-CV is the official five-fold replay. The embedding column is the representation intervention. Search and holdout agree on five tasks. The two-dimensional exfoliation holdout favours the baseline.

task	evaluation	baseline	feature	embedding	model
phonons	SIGNAL	67.42	**42.79**	66.66	60.60
	HELD-OUT	72.07	**47.87**	77.45	66.64
	OFFICIAL-CV	63.83	**41.93**	65.94	55.21
bulk modulus	SIGNAL	0.0688	0.0661	0.0683	**0.0619**
	HELD-OUT	0.0668	0.0638	0.0657	**0.0586**
	OFFICIAL-CV	0.0668	0.0643	0.0664	**0.0593**
shear modulus	SIGNAL	0.0916	0.0867	0.0913	**0.0815**
	HELD-OUT	0.0904	0.0868	0.0898	**0.0787**
	OFFICIAL-CV	0.0898	0.0854	0.0895	**0.0786**
refractive index	SIGNAL	0.3360	0.3309	0.3343	**0.3148**
	HELD-OUT	0.2026	0.2025	0.2054	**0.1832**
	OFFICIAL-CV	0.3073	0.3024	0.3033	**0.2822**
2D exfoliation	SIGNAL	42.85	**38.79**	41.32	40.62
	HELD-OUT	**25.53**	26.11	29.06	29.31
	OFFICIAL-CV	38.63	**35.82**	39.52	37.90
perovskite energy	SIGNAL	0.2580	**0.1335**	0.2529	0.2265
	HELD-OUT	0.2562	**0.1335**	0.2498	0.2170
	OFFICIAL-CV	0.2527	**0.1299**	0.2483	0.2179

4.6 Separate discoveries can be combined

The frozen outputs are not only useful individually. Because they change separate parts of the pipeline, their code can also be combined. Combining the separately found composition changes reaches a 28.9% held-out MAE reduction for band gap and AUC gains of 0.0103 and 0.0232 for metallicity and glass. Under benchmark replay, this combination reaches or exceeds the published MODNet value on three of four tasks.

The structure feature and model searches also modify separate files, so their frozen code can be combined directly. Figure 5 compares the combined code with the feature class chosen by the mean search signal before the holdout. It also shows the best individual class for each task within each displayed evaluation. The feature plus model code reaches 26.3% mean improvement on the fold-0 holdout and 27.0% in benchmark replay. It improves on both constituent changes for every structure task in these two evaluations. The combination was assembled after the individual results were available, so it tests whether discoveries compose and does not enter the nine-of-ten count.

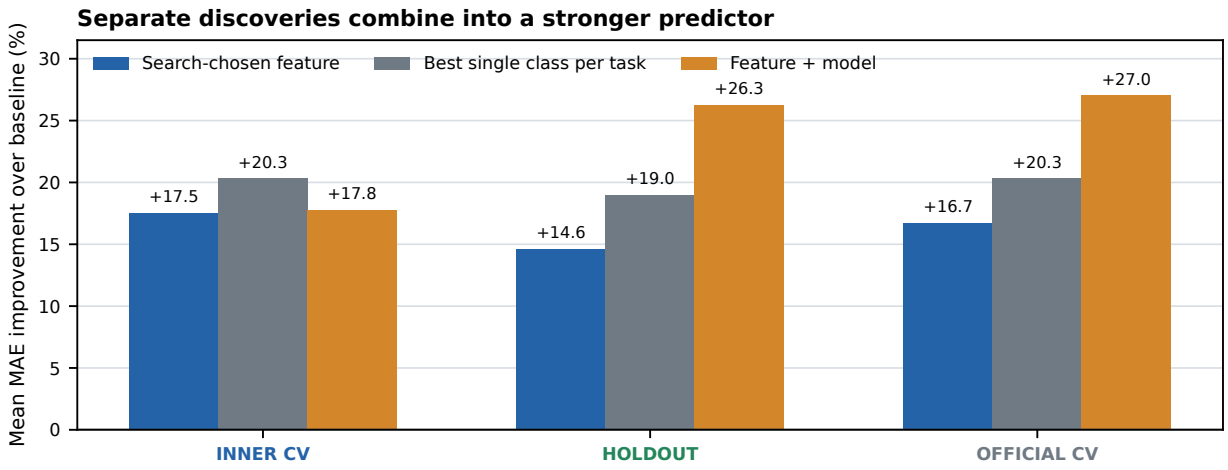

Figure 5: **Separate feature and model discoveries combine into a stronger predictor.** Blue shows the feature class chosen by its mean search signal before the holdout. Grey shows the best individual class for each task within each displayed evaluation. Orange combines the frozen feature and model code after their separate searches. The combination gives the largest mean result on the holdout and benchmark replay.

5 Discussion

Across the ten tasks, Auto Research reveals two modelling regimes rather than one winning recipe. Formula-based prediction improves through descriptors, learned representations, or models. When explicit structures are available, geometry descriptors and model changes lead on different property families. Feature changes lead on phonons and perovskite energy, while model and calibration changes lead on elastic and refractive-index tasks. In the composition tasks, the small screened external sets contribute much less than changes to the existing pipeline. The composition and structure assemblies further show that separate discoveries can reinforce one another.

These materials findings matter because they survive a search process with repeated adaptive feedback. Separating feature, model, representation, and data changes reveals where an improvement comes from. Freezing code before the holdout then tests whether that decision survives outside the loop. Nine choices do so, while the two-dimensional exfoliation disagreement shows why final unseen evidence still matters.

The feedback signal is also part of this research design. Averaging performance across inner folds reduces the influence of one development partition on a long adaptive search. The untouched holdout still has a separate role because it tests the final frozen choice only after search has ended. Together, multi-fold feedback during search and once-only holdout evaluation provide a practical pattern for future Auto Research workflows.

The broader contribution is an evaluation principle for closed-loop AI science. The object being evaluated should be the frozen executable change, not the amount of agent activity or the best score seen during search. Searching each intervention class separately identifies the source of improvement, and the holdout measures whether the choice remains useful. Reuse across endpoints and combination with other frozen changes then test whether the output carries technical value beyond one score. This design can be applied in other digitally evaluated research settings where an agent changes executable code and the domain evaluator can reserve final evidence.

6 Conclusion

This study demonstrates two forms of transfer from Auto Research. Search decisions survive unseen evidence on nine of ten endpoints, and the resulting feature and model code remains useful across property families and in combined predictors. The transferred changes reveal that composition inputs admit several improvement routes, while structure inputs separate the roles of geometry descriptors and predictive models. More broadly, inner-fold averaging, separate intervention searches, frozen code, and unseen evaluation provide a practical way to test whether a closed-loop research agent has produced a reusable executable discovery.

Author contributions

Jingjie Ning: Conceptualization, Methodology, Software, Investigation, Project administration, Writing – original draft. Xiaochuan Li: Writing – review and editing. Shanshan Zhong: Writing – review and editing. Ji Zeng: Writing – review and editing. Guolin Ke: Supervision.

Conflicts of interest

There are no conflicts to declare.

Data availability

The public study archive contains all 701 trial records, the seven frozen winners, per-task certification scores, the small candidate data files and embedding tables used by selected interventions, and scripts that recalculate the headline results. The main evaluation entry points are `holdout_certify.py`, `frontier_probe.py`, and `bridge_eval.py`. The archive also records one implementation discrepancy in the two-dimensional exfoliation model search. Calibration state set inside the campaign process was not restored in a separate-process replay, changing its score from 40.62 to 60.47. This affects neither the feature intervention selected in the primary comparison nor the nine-of-ten count. The archive and one-command verification guide are available at `https://github.com/cxcscmu/Auto-Research-AI-Scientist/tree/main/paper_artifact`. Matbench data remain available through the upstream benchmark package and are not redistributed. A DOI-backed snapshot will be deposited before publication.

References

[1] C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune and D. Ha, *The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery*, 2024.

[2] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold and A. Roth, *Science*, 2015, **349**, 636–638.

[3] S. Varma and R. Simon, *BMC Bioinformatics*, 2006, **7**, 91.

[4] G. C. Cawley and N. L. C. Talbot, *Journal of Machine Learning Research*, 2010, **11**, 2079–2107.

[5] A. Dunn, Q. Wang, A. Ganose, D. Dopp and A. Jain, *npj Computational Materials*, 2020, **6**, 138.

[6] L. Ward, A. Agrawal, A. Choudhary and C. Wolverton, *npj Computational Materials*, 2016, **2**, 16028.

[7] L. Ward *et al.*, *Computational Materials Science*, 2018, **152**, 60–69.

[8] P.-P. De Breuck, G. Hautier and G.-M. Rignanese, *npj Computational Materials*, 2021, **7**, 83.

[9] R. E. A. Goodall and A. A. Lee, *Nature Communications*, 2020, **11**, 6280.

[10] A. Y.-T. Wang, S. K. Kauwe, R. J. Murdock and T. D. Sparks, *npj Computational Materials*, 2021, **7**, 77.

[11] T. Xie and J. C. Grossman, *Physical Review Letters*, 2018, **120**, 145301.

[12] C. Chen, W. Ye, Y. Zuo, C. Zheng and S. P. Ong, *Chemistry of Materials*, 2019, **31**, 3564–3572.

[13] K. Choudhary and B. DeCost, *npj Computational Materials*, 2021, **7**, 185.

[14] K. Choudhary *et al.*, *npj Computational Materials*, 2024, **10**, 93.

[15] J. Riebesell *et al.*, *Matbench Discovery – A framework to evaluate machine learning crystal stability predictions*, 2023.

[16] N. Alampara, M. Schilling-Wilhelmi and K. M. Jablonka, *Computational Materials Science*, 2025, **259**, 114041.

[17] Y. Yamada, R. T. Lange, C. Lu, S. Hu, C. Lu, J. Foerster, J. Clune and D. Ha, *The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search*, 2025.

[18] Z. Jiang, D. Schmidt, D. Srikanth, D. Xu, I. Kaplan, D. Jacenko and Y. Wu, *AIDE: AI-Driven Exploration in the Space of Code*, 2025.

[19] A. Novikov, N. Vu, M. Eisenberger, E. Dupont, P.-S. Huang, A. Z. Wagner, S. Shirobokov, B. Kozlovskii, F. J. R. Ruiz, A. Mehrabian, M. P. Kumar, A. See, S. Chaudhuri, G. Holland, A. Davies, S. Nowozin, P. Kohli and M. Balog, *AlphaEvolve: A coding agent for scientific and algorithmic discovery*, 2025.

[20] E. Real, C. Liang, D. R. So and Q. V. Le, Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 8007–8019.

[21] J. Ning, X. Li, J. Zeng, H. Kang and C. Xiong, *Auto Research with Specialist Agents Develops Effective and Non-Trivial Training Recipes*, arXiv:2605.05724, 2026.

[22] J. Ning, X. Li, J. Zeng, C. Xiong and G. Ke, *Closed-loop Auto Research for Molecular Property Prediction: Discovering and Certifying Generalizable Improvements*, arXiv:2606.22731, 2026.

[23] S. Jia, C. Zhang and V. Fung, *LLMatDesign: Autonomous Materials Discovery with Large Language Models*, 2024.

[24] A. Ghafarollahi and M. J. Buehler, *Autonomous Inorganic Materials Discovery via Multi-Agent Physics-Aware Scientific Reasoning*, 2025.

[25] S. Rothfarb, M. C. Davis, I. Matanovic, B. Li, E. F. Holby and W. J. M. Kort-Kamp, *Hierarchical Multi-agent Large Language Model Reasoning for Autonomous Functional Materials Discovery*, 2025.

[26] N. J. Szymanski *et al.*, *Nature*, 2023, **624**, 86–91.

[27] D. A. Boiko, R. MacKnight, B. Kline and G. Gomes, *Nature*, 2023, **624**, 570–578.

[28] A. Merchant, S. Batzner, S. S. Schoenholz, M. Aykol, G. Cheon and E. D. Cubuk, *Nature*, 2023, **624**, 80–85.

[29] C. Zeni *et al.*, *MatterGen: a generative model for inorganic materials design*, 2023.

[30] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush and A. Gulin, *Advances in Neural Information Processing Systems*, 2018, **31**, 6638–6648.

[31] V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder and A. Jain, *Nature*, 2019, **571**, 95–98.

[32] Materials Project, *Official Matbench v0.1 benchmark submissions and leaderboard*, Official Matbench repository, benchmarks directory, 2024, Revision 936176d, dated 20 January 2024; accessed 10 July 2026.

[33] R. Ruff, P. Reiser, J. Stühmer and P. Friederich, *Digital Discovery*, 2024, **3**, 594–601.