Chapter 13. AI and Materials Science

Zecheng GAN; Jiaying WU; Jin-Guo LIU; Guo-Yi ZHU; Ping GAO

Thrust of Advanced Materials, Function Hub, HKUST(GZ)

13.1 A Question to Start With

Suppose your phone battery could charge safely in ten minutes, your window could quietly turn sunlight into electricity, and your jacket could stay warm without becoming bulky. None of these dreams begins with an app. They begin with materials: the solid electrolyte inside a battery, the thin film in a solar cell, the fibers and pores inside a fabric.

Materials science asks a deceptively simple question: *What should matter be made of, and how should it be arranged, so that it does something useful?* Steel, glass, silicon chips, lithium batteries, catalysts, ceramics, polymers, and biological materials all belong to this story. A material is not just a substance. It is a substance whose internal structure gives it a function. Figure 13.1 gives the visual map for this chapter: AI helps us search a vast materials space, but every promising route still has to return to experiment, measurement, and judgment.

Figure 13.1: A visual summary of AI-assisted materials discovery.

That structure exists at many scales. At the atomic scale, atoms sit in patterns. At the microscopic scale, grains, pores, cracks, and defects appear. At the human scale, the

material becomes a bridge cable, a phone screen, a hip implant, a turbine blade, or a solar panel. A tiny change at one scale can change the behavior at another. Add a little carbon to iron and steel is born. Rearrange carbon atoms and the difference between graphite and diamond appears.

This is why materials discovery has always been hard. The periodic table looks small, but the number of possible combinations is enormous. A battery material might contain lithium, oxygen, phosphorus, sulfur, nickel, manganese, cobalt, or many other elements. The same chemical formula may also appear in different crystal structures. The same material may behave differently depending on how it was heated, cooled, pressed, stretched, or mixed.

For much of history, materials discovery felt like exploration with a lantern. Scientists combined knowledge, experience, and patience. Sometimes they made a planned discovery. Sometimes they got lucky. Penicillin, conducting polymers, and graphene are famous reminders that chance observations can matter. But modern society is asking for too much, too quickly, to rely only on lucky accidents: safer batteries, low-carbon cement, better catalysts, recyclable plastics, efficient solar cells, and materials for quantum technologies.

So the starting question of this chapter is not "Can AI replace materials scientists?" That question is too shallow. A better question is:

> If the possible materials are too many to test one by one, can AI help us choose where to look next?

The answer is yes, but with a condition. AI can help us search, predict, compare, and plan. It cannot excuse us from understanding chemistry, physics, processing, measurement, or evidence. In materials science, an AI result is not a final answer. It is a well-organized guess that still has to survive reality.

13.2 Core Exploration: Why AI + Materials

AI entered materials research not because traditional materials science had failed, but because the field had become too large, too expensive, and too data-rich for old habits alone. Three changes made the bridge possible.

First, materials research produces expensive data. A quantum-mechanical calculation can take hours or days. A synthesis experiment can take a week. A microscopy dataset can contain thousands of images. If each attempt is slow, then asking "What should we try next?" becomes a serious scientific question. AI is useful because it can learn from past attempts and rank future ones.

Second, materials data became more open and more organized. Databases such as the Materials Project collect computed properties for many inorganic materials (Jain et al., 2013). Tools such as pymatgen and matminer help researchers turn chemical formulas

and crystal structures into data that a model can use (Ong et al., 2013; Ward et al., 2018). Benchmarks such as Matbench make it easier to compare models fairly instead of celebrating isolated success stories (Dunn et al., 2020).

Third, AI methods became better at reading the natural language of matter. A molecule or crystal is not a sentence, but it does have structure. Atoms are connected to neighbors. Bonds have lengths and angles. Images have shapes and textures. Recipes have steps. Modern AI can learn from tables, graphs, images, spectra, and text, which makes it much more useful than a simple spreadsheet model.

Figure 13.2 shows the basic loop. The crucial point is the loop, not the model. AI-assisted materials research usually begins with data, proposes candidates or experiments, checks them by physics or measurement, and feeds the result back into the next round.

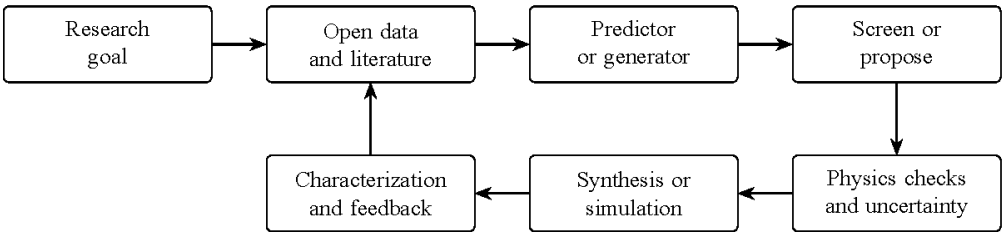

Figure 13.2: The design-make-test-learn loop in AI-assisted materials research.

A useful bridge metaphor is a map. Traditional materials science gives us deep local knowledge: this alloy family is strong, this oxide is stable, this polymer is flexible. AI helps sketch a larger map and highlight routes worth walking. But a highlighted route is not the same as arrival. A model may predict that a crystal is stable, but it may still be difficult to make. A model may suggest a catalyst, but it may degrade in the real reaction environment. A model may fit a benchmark, but fail on a new chemical family.

Consequently, the healthiest way for one to understand AI + materials: not magic, not replacement, but a new search discipline. It changes how scientists decide what to try next.

13.3 Cross-Border Bridge: How AI is Transforming Materials Research

13.3.1 Learning from Data: From Recipes to Predictions

The simplest AI task in materials science is property prediction. We give the model examples such as "this composition has this band gap" or "this crystal has this formation energy," and the model learns a pattern. Later, when it sees a new material, it makes a prediction.

This sounds ordinary, but it changes the scale of the work. If a reliable model can estimate a property in seconds, researchers can screen thousands or millions of candidates before choosing a smaller number for expensive calculations or experiments. This does not remove the need for physics. It moves physics to the most promising places.

Early and still useful models often rely on descriptors: numerical summaries of a material. For example, a model might use average atomic mass, electronegativity differences, density, or structural features. The basic steps include: collect data, build features, train a model, test it, and ask whether it beats a simple baseline. MODNet is a good example of this spirit for limited materials datasets (De Breuck et al., 2021).

The warning is equally important. If a model performs well only because the test set is too similar to the training set, it has not learned discovery; it has learned to recognize cousins. Good benchmarks and careful splits are therefore part of the science, not paperwork.

A useful rule of thumb is that the data regime should choose the method, not the other way around. If a dataset has only a few hundred reliable examples, a careful feature-based model may beat a fashionable deep network. If the task depends on atomic geometry, graph models deserve attention. If every experiment takes weeks, choosing the next experiment wisely may matter more than making the predictor slightly more accurate.

13.3.2 Representing Materials: Crystals as Graphs

Many modern materials models use graph neural networks. The idea is friendly once we remove the jargon. In a crystal, atoms have neighbors. A graph is simply a way to record "who is near whom." Atoms become nodes; neighbor relationships become edges; the model passes messages along those edges until it builds a picture of the whole structure, which is also illustrated in Figure 13.3. MEGNet showed that graph networks could serve as a general framework for molecules and crystals (Chen et al., 2019). ALIGNN then added bond-angle information, which matters because materials are not only about which atoms touch, but also about geometry (Choudhary and DeCost, 2021).

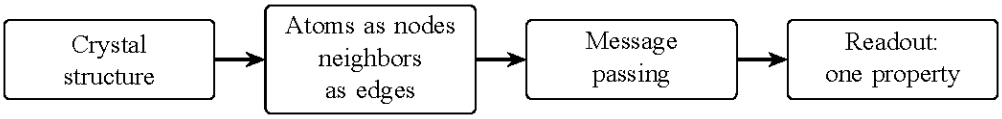

Figure 13.3: A simplified graph view of a crystal. The model learns from local atomic neighborhoods and then predicts a material-level property.

Here, the key lesson is not the mathematics of neural networks. It is the representation problem. If we represent a material poorly, even a powerful model may struggle. If we represent it in a way that respects its structure, the model has a better chance.

13.3.3 From Prediction and Screening to Design

Prediction asks: "What property does this material have?" Design asks the harder question: "What material should have the property we want?"

This is called inverse design. Instead of starting with a material and calculating its behavior, researchers start with a target: a solid electrolyte, a solar-cell layer, a corrosion-resistant alloy, or a catalyst. The model then proposes candidates. Generative models such as MatterGen push in this direction by proposing inorganic crystal structures under property constraints (Zeni et al., 2025). A more device-level example comes from perovskite solar cells. Wu and co-workers used a closed-loop inverse-design workflow to search for hole-transport materials, starting from a relatively small molecular dataset and optimizing toward solar-cell performance rather than a single abstract property (Wu et al., 2024).

Large-scale screening is a related route. GNoME, developed by Google DeepMind, used graph networks and active learning to search a huge space of inorganic crystals. The study reported 2.2 million candidate structures stable with respect to previous work, with 381,000 entries on an updated convex hull (Merchant et al., 2023). That is impressive, but one should notice the wording. These are computational candidates. Some may become useful materials; many may not. A predicted stable crystal is not automatically a battery, a chip, or a product.

13.3.4 From Design to Making Materials

The hardest step is often not prediction but making. Materials scientists know this painfully well. A candidate may be stable on a computer but unreachable in the lab. It may require a temperature, pressure, precursor, or processing route that is impractical. It may form a different phase. It may look right in one measurement and wrong in another.

Active learning gives a concrete example of AI as an experimental strategist. In high-entropy Invar alloy discovery, Rao and co-workers faced an enormous composition space, but the workflow tested only 17 new alloys and identified two alloys with unusually low thermal expansion near room temperature (Rao et al., 2022). The lesson is not that AI "invented" the alloy by itself. The lesson is sharper: AI helped allocate scarce experimental effort.

This is why synthesis planning and autonomous laboratories matter. He and co-workers trained a precursor recommendation system using 29,900 solid-state synthesis recipes mined from the literature and reported strong performance on unseen targets (He et al., 2023). DiffSyn later treated synthesis planning itself as a generative task for zeolites, using more than 23,000 recipes from decades of literature (Pan et al., 2026). These studies show a new direction: AI is learning not only *what* to make, but also *how* people have made related materials before.

Autonomous laboratories go one step further. The A-Lab connected computation, synthesis planning, robotic powder handling, heating, characterization, and active learning in one workflow (Szymanski et al., 2023). That is the dream version of Figure 13.2: a lab that can keep trying, measuring, and improving.

But this is exactly where we must be most skeptical. A 2026 author correction to the A-Lab paper clarified novelty language and reported that 4 of 40 originally reported successes were inconclusive from X-ray diffraction alone (Szymanski et al., 2026). This does not make autonomous labs unimportant. It makes the deeper lesson sharper: when AI speeds up science, validation must speed up too.

Table 13.1 summarizes these roles across the research cycle.

Table 13.1: Major AI roles in materials research

AI role	Detailed Explanation	Example in materials research
Property prediction	Estimate a property before doing a costly calculation or experiment.	Predict band gap, stability, elasticity, or ionic conductivity from composition or structure.
Screening	Rank many candidates and keep only the most promising ones.	Search large crystal databases for possible battery or solar-cell materials.
Generative design	Propose new candidates that might meet a target.	MatterGen-style crystal generation under property constraints.
Active learning	Choose the next experiment that teaches the model the most.	Explore alloy compositions without testing every possible mixture.
Synthesis planning	Suggest precursors, temperatures, or routes for making a target.	Literature-trained models recommend solid-state synthesis recipes.
Characterization analysis	Read images, spectra, or diffraction patterns faster and more consistently.	Segment microscopy images or help interpret X-ray diffraction data.

The methods above explain the core workflow. The following cases show how the same ideas extend from crystal generation to solar-cell formulation and programmable quantum matter.

13.3.5 Case Study: Generating Crystals with Symmetry-Aware AI

A concrete way to see generative design in action is to ask how a model can create an entire crystal. A crystal is not a random clump of atoms; it is a pattern that repeats in three dimensions, much like wallpaper repeats on a wall. The smallest repeating block

is called the unit cell: a small tilted box containing a handful of atoms. Repeat that box edge to edge in every direction, and the whole crystal follows. Describing a crystal therefore means describing what sits inside one unit cell, together with the size and shape of that box.

For an ideal periodic crystal, the unit cell is rarely arbitrary. It usually has internal symmetry: rotations, reflections, or combinations of the two that leave the pattern unchanged. Crystallographers have catalogued 230 possible three-dimensional space groups. For AI training, a space-group number can be treated as a useful label for one symmetry pattern.

Symmetry gives a generative model a shortcut. Many atoms in a unit cell are copies of one another, so the model needs to place only the symmetry-inequivalent atoms at special locations called Wyckoff positions. Figure 13.4 shows calcite, CaCO3, in space group #167. Once one carbon, one calcium, and one oxygen are placed at the appropriate positions, symmetry generates the remaining atoms in the 30-atom cell automatically.

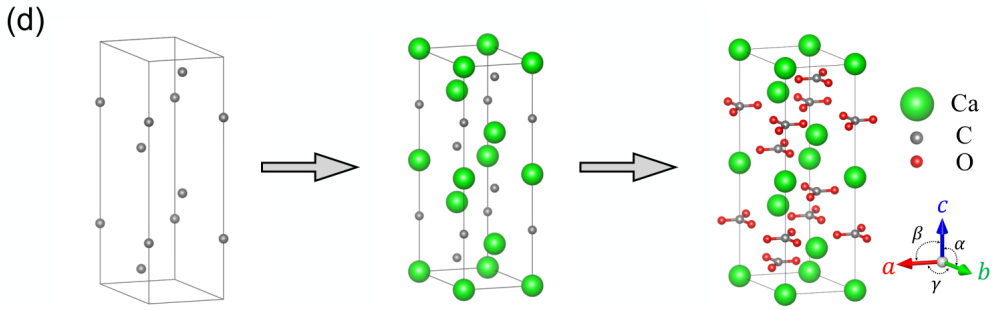

Figure 13.4: Building the calcite (CaCO3, space group #167) crystal one symmetry-inequivalent atom at a time. One carbon atom is placed at a Wyckoff position and symmetry generates its copies (left); calcium follows (middle), then oxygen (right). Thirty atoms in the unit cell follow from three decisions. Adapted from Cao et al. (2025a), Figure 1(d).

CrystalFormer uses this idea directly (Cao et al., 2025a). It is an autoregressive transformer - a model family also used in large language models - but it predicts symmetry-inequivalent atomic sites rather than words. The model receives a space-group number and then proposes the sites one at a time, letting symmetry fill in the rest of the crystal.

Other generators, such as CDVAE and DiffCSP, take a different route: they write every atom in a cell directly and do not start by fixing a space group (Xie et al., 2021; Jiao et al., 2023). CrystalFormer has since been extended with reinforcement fine-tuning for target properties (Cao and Wang, 2025) and with a hybrid workflow that pairs fast generation with a slower physics-based relaxation step (Cao et al., 2025b). This is the familiar pattern of a fast guess followed by a slower check.

Generating a plausible structure is not the same as discovering a new material. A generator can appear more original than it really is if it repeats known structures, makes only small changes to them, or is judged by weak evaluation rules (Martirossyan et al., 2025). Mini-Project III in Section 13.4.3 asks students to act as independent referees: they generate structures with a released model and use separate analysis tools to check whether the results really match rock salt or diamond.

13.3.6 Case Study: Data-Efficient Formulation Discovery in Organic Solar Cells

Many materials problems are not about choosing one substance; they are about choosing a recipe made from several components. Trial-and-error formulation can be slow because every possible combination needs preparation and testing. Machine learning can rank promising recipes, but it faces a familiar problem: in many important materials settings, the available dataset is too small for a purely data-driven model to generalize reliably.

Organic solar cells provide a useful example. A conventional device commonly blends an electron-donor material with an electron-acceptor material so that light can be converted into electrical charge. Device engineers can sometimes improve performance by 5-10% by adding further photoactive components, moving from binary blends to ternary or even quaternary formulations. For one newly discovered solar-cell material, characterizing feasible binary and ternary devices can conservatively require about 100 binary recipes and 90 ternary recipes - roughly 130 weeks of work for an experienced researcher or engineer.

Literature data can help predict which recipes are worth trying, but the field may contain only a few hundred reported binary and ternary systems. A boosting model combines many decision trees, with later trees learning from the errors of earlier ones. It can be accurate and interpretable, yet a small dataset makes it prone to overfitting: it may describe the old examples beautifully while failing on the next experiment.

The practical response is to combine data with domain knowledge. In this example, expert knowledge guides the learning algorithm toward physically sensible multi-component formulations. With a training dataset of just over one hundred samples, the approach can generate new predictions within minutes, while keeping the predictions interpretable and open to experimental verification. The broader lesson is not that the model replaces the experimentalist. It is that a well-chosen model can focus scarce laboratory effort on a much smaller set of candidate recipes.

13.3.7 Case Study: AI-Guided Design and Control of Programmable Quantum Matter

The same search-and-control logic also appears in programmable quantum matter: carefully controlled collections of atoms, trapped ions, or superconducting circuits. Their useful behavior comes not simply from chemical composition, but from the collective quantum state shared by their components. Such systems are important physical platforms for quantum computing and sensing. They also create an enormous search space of possible configurations and control sequences, while each experiment is delicate and expensive.

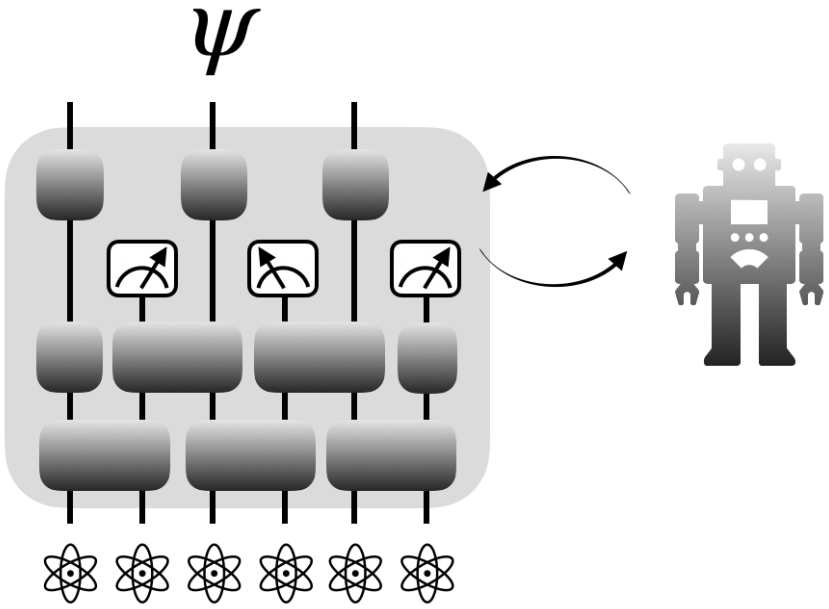

Figure 13.5: A simplified view of feedback-controlled programmable quantum matter. Measurements provide information about the system; a classical controller uses that information to choose the next action.

A key challenge is protecting fragile quantum information from noise. A qubit is the quantum counterpart of a bit, but it can be disturbed by its environment or by imperfect control. Quantum error correction distributes one logical qubit across many physical qubits so that the collective state can be more robust than any single component. Recent experiments have demonstrated programmable logical processors and below-threshold surface-code memories (Bluvstein et al., 2024; Google Quantum AI, 2025). The toric code is a classic theoretical model for this type of topological protection (Kitaev, 2003), and related ideas connect error thresholds with phase-transition-like behavior in quantum matter (Anderson, 1973; Chen et al., 2025).

AI can help in two complementary ways. It can search for useful control sequences and many-body states, and it can interpret a stream of measurements while an experiment is running. In a feedback-controlled circuit, a classical decoder receives outcomes such as 00110110000, infers what likely happened to the remaining quantum state, and recommends a corrective action. This is a natural pattern-recognition problem. Machine-learning decoders have been demonstrated on quantum

processors (Bausch et al., 2024), while other experiments have used measurement and feedback to prepare exotic topological quantum states (Iqbal et al., 2024). Here too, AI does not replace the physics. It helps scientists navigate a search-and-control problem that is too large for intuition alone.

13.3.8 Case Study: AI-Guided Enzyme Engineering for Plastic Recycling

Proteins are molecular machines. Enzymes are proteins that speed up chemical reactions, and some can help break the chemical bonds in polyethylene terephthalate (PET), the plastic used in many bottles and textiles. This is promising for recycling, but natural PET-degrading enzymes can be slow or lose activity under useful processing conditions.

Lu and co-workers used a structure-based machine-learning system to guide improvements to an existing PETase enzyme. The model examined the local three-dimensional environments around amino acids and ranked substitutions that might make the enzyme more active or robust. This was not a case of inventing a completely new protein from scratch: the AI proposed changes to a known enzyme scaffold, and the proposed changes still had to be made and tested in the laboratory.

One resulting enzyme, FAST-PETase, combined five mutations. In the reported study, it degraded samples of postconsumer PET and the recovered monomers were used to make PET again, illustrating a possible closed-loop route from plastic waste back to useful material (Lu et al., 2022).

This case extends the design-make-test-learn loop to biological materials. AI can focus experimental effort on promising enzyme variants, but a useful recycling process must still be judged by activity, stability, real feedstocks, energy use, cost, and the quality of the recycled product. A successful prediction is the beginning of the evidence chain, not the end.

13.4 Mini-Projects

The best way for one to learn AI + materials is not to begin with a giant neural network. It is to run a small project where every step is visible. The four mini-projects below are designed for a general education course. They are modest on purpose. A simple experiment that one can audit is better than a glamorous result they cannot question.

13.4.1 Mini-Project I: A First Property Predictor

Context and problem. Imagine you are choosing a material for a solar cell or a light sensor. One useful property is the band gap, which roughly tells us what range of light

a material can absorb. Here we do not need the quantum mechanical derivations. We only need the practical idea: different materials interact with light differently.

Objective. Build a simple model that predicts a material property from composition-based features, then compare it with a naive baseline.

Data. Use a small open dataset from Matbench or matminer. Suitable targets include band gap, formation energy, or a simple classification such as metal versus non-metal (Ward et al., 2018; Dunn et al., 2020).

Method. Start with transparent models such as linear regression, random forest, or gradient boosting. Use composition descriptors rather than crystal structures since we do not require you to understand crystallography.

Implementation steps.

1. Load the dataset and remove entries with missing target values.
2. Split the data into training and test sets before looking at performance.
3. Convert each formula into numerical descriptors.
4. Train one simple model and one baseline model.
5. Report an error metric and inspect several wrong predictions.

Expected learning. The point is not to get a publishable model. The point is to see that a material formula can become data, that the test set matters, and that errors are often more instructive than successes. A good report should include at least three failed predictions and a discussion of why they might have failed. AI-assisted coding is encouraged.

13.4.2 Mini-Project II: A Toy Self-Driving Materials Lab

Context and problem. A real autonomous laboratory can be expensive. But a classroom version can be simple: let us consider designing an insulating sleeve for a paper cup. The material choices might include paper, cotton, aluminum foil, foam, cardboard, and air gaps. The performance target is easy to measure: how slowly does warm water cool?

Objective. Use a small active-learning loop to choose the next sleeve design instead of testing every possible combination.

Data. Each team records the sleeve materials, thickness, number of layers, cost, and temperature drop after a fixed time. The class pools the data into one spreadsheet.

Method. Train a simple model after the first round of experiments. Ask it to recommend two types of next tests: one design it thinks will perform well, and one design where it is uncertain. This tells the difference between exploitation and exploration.

Implementation steps.

1. Round 1: each group tests three sleeve designs chosen by common sense.
2. The class combines the results and trains a simple model.
3. Round 2: each group tests one model-recommended design and one uncertainty-driven design.
4. Compare the best human-chosen design with the best AI-guided design.
5. Discuss whether the model learned physics, shortcuts, or measurement noise.

Expected learning. This mini-project makes the design-make-test-learn loop physical. We learn that AI is not only a predictor on a laptop. It can be part of an experimental strategy. We also learn an uncomfortable lesson: if the measurements are sloppy, the AI becomes confidently sloppy.

13.4.3 Mini-Project III: Can the AI Rediscover Diamond?

Context and problem. In 1913, William Henry Bragg and his son William Lawrence Bragg solved the first crystal structures in history - rock salt and diamond - by shining X-rays through them. The work won the 1915 Nobel Prize in Physics; Lawrence was 25, and remains the youngest science Nobel laureate ever. CrystalFormer (Cao et al., 2025a) claims it can predict a crystal structure from a chemical formula in seconds, so let us hand it the two puzzles the Braggs solved. But the question "can the AI rediscover diamond?" hides two different questions: does the model know what diamond looks like, and does it bet on diamond when all you give it is carbon? This project separates the two - and the answers may not be the same.

Objective. Measure, with your own referee code, two rediscovery rates for each material: one where CrystalFormer freely chooses the space group from the formula alone, and one where you pin the space group to the right answer (#225 for rock salt, #227 for diamond). The gap between the two rates is the finding.

Data. None to collect. Use the openly released "multitask" checkpoint (13.8 million parameters) linked from the CrystalFormer README, trained on Alex-20s, a large public database of computed crystal structures. This project checks a model; it does not train one.

Method. Install CrystalFormer following its README, with one correction: as of this writing, the released checkpoint only loads with an older JAX (pip install "jax[cpu]==0.4.31"; CPU-only is fine, and the dependency warnings this triggers concern training, not sampling). Confirm the example sampling command runs, then run the two arms. Free-choice arm: about 50 structures each with --formula NaCl and --formula C, letting the model pick among its top-ranked space groups (the --K 40 setting in the README's structure-prediction example). Pinned arm: about 20 structures each, adding --spacegroup 225 or --spacegroup 227. One warning: the provided conversion script (awl2struct.py) writes a CSV of structure records, not .cif files, and silently drops samples whose composition came out wrong - turning every

sample into a real CIF file is your first small coding task (pymatgen can rebuild a structure from each record and save it). The answer keys are free: rock salt and diamond can each be built in about three lines of pymatgen, with no database access. Your referee script then applies the same checks to every structure in both arms. First, a sanity check: if the smallest distance between any two atoms is below about 0.5 Å, atoms are sitting on top of each other and the structure is nonsense. Second, the match check: does pymatgen's StructureMatcher say the structure is the reference rock salt or diamond? For the misses, ask pymatgen's SpacegroupAnalyzer what the model made instead - for carbon, watch for graphite (space group #194): the same atoms as diamond, arranged differently, and suddenly you have pencil lead instead of the hardest natural material. Symmetry detection has a tolerance setting (symprec); run it at the default and at one looser value, and note whether the answers differ.

Implementation steps.

1. Install JAX (CPU version, pinned as above) and CrystalFormer; confirm the example sampling command runs on your laptop.
2. Free-choice arm: generate ~50 structures each with --formula NaCl and --formula C; convert every sample to a CIF file.
3. Pinned arm: generate ~20 structures each with the space group forced to #225 (NaCl) and #227 (carbon).
4. Referee every structure in both arms: minimum interatomic distance, StructureMatcher match against the reference, and - for the misses - the detected space group at two tolerance settings, keeping a tally of graphite.
5. Report the free-choice and pinned rediscovery rates side by side, and include a picture of one rediscovered diamond (from VESTA, a free viewer, or rendered in code with ASE).

Expected learning. The point is to referee a model's claim with your own code - and to see how easily an evaluation can mislead. If you ran only the free-choice arm, you might conclude the model cannot make diamond. The pinned arm tests whether that is true, or whether the model knows the structure perfectly well and simply does not rank it highly for so simple a formula. "The model cannot do it" and "the model did not choose to" look identical in a single number; separating them is exactly the kind of judgment Section 13.5.1 is about. The diamond-versus-graphite tally adds the deeper materials lesson: the same atoms, arranged differently, are a different material. AI-assisted coding is strongly encouraged throughout; deciding what the numbers mean is up to you.

13.4.4 Mini-Project IV: Can an AI See Molecular Handedness?

Context and problem. A molecule can have a left- and right-handed form. The two forms have the same atoms and the same connections, but their three-dimensional arrangements are mirror images. The familiar claim that orange and lemon smell different because of two limonene mirror images is too simple: citrus oils are mixtures of many volatile molecules (Hong et al., 2017). The more useful question is therefore narrower: if two molecules are mirror images, does an AI model even receive that difference as part of its input?

Objective. Compare molecular representations that ignore chirality with representations that preserve it. Decide what the model can see, and keep that question separate from the much harder question of predicting what a person will smell.

Data. Use the companion file chiral_odorant_pairs.csv. It contains ten odorant pairs tested in a human discrimination study, including limonene, carvone, and alpha-pinene. Each row gives two isomeric SMILES strings, PubChem identifiers, and the group-level result from the study. Humans distinguished some pairs but not others (Laska et al., 1999).

Method. If RDKit is available, read each pair and generate two kinds of molecular representation. First turn chirality off, then turn it on. Compare canonical SMILES and Morgan fingerprints in the two settings. This is not a smell-prediction exercise: a small list of molecules cannot establish a general rule for human perception.

Implementation steps.

1. Open chiral_odorant_pairs.csv and draw the two members of the limonene and carvone pairs.
2. Export SMILES once with isomericSmiles=False and once with isomericSmiles=True. Record which pairs become identical when stereochemistry is omitted.
3. Generate Morgan fingerprints with useChirality=False and useChirality=True. For each setting, calculate the similarity of the two members of every pair.
4. Make one summary table: molecular pair, representation result, and the human group result reported by Laska et al. (1999).
5. Write a short reflection: why does preserving molecular handedness not automatically let a model predict a real-world smell?

Expected learning. A model can only use distinctions that its representation retains. Chirality can matter for an odorant, but it does not always do so; concentration, mixtures, receptors, and personal experience also matter. The project involves no smelling experiment. Its lesson is not that an AI can identify orange or lemon from one molecule, but that choosing a representation is already a scientific decision.

Table 13.2: Four classroom mini-projects for a first-year AI + materials module

Mini-project	What we practice	Main risk to discuss
Property predictor from formulas	Data cleaning, descriptors, train/test split, baseline comparison, error analysis.	Mistaking a high test score for true discovery ability.
Toy self-driving insulation lab	Experimental design, active learning, pooled classroom data, uncertainty, measurement discipline.	Feeding noisy or biased measurements into the model.
Rediscovering rock salt and diamond with a crystal generator	Running a pretrained generative model; independent verification with pymatgen.	Concluding that the model cannot do something when its sampler simply did not choose to do it.
Molecular handedness and odor perception	Stereochemistry-aware molecular representations and evidence audit.	Confusing a molecular representation with a prediction of real-world smell.

All four projects should end with a short reflection. The reflection question is simple: What did the AI help us see, and what did it hide from us?

For anyone who wants to go one step further, the most sensible learning path is not to jump straight into the largest model. Start with Matbench, matminer, and scikit-learn; then move to pymatgen and structure-aware graph tools such as MatGL; only after that should one explore catalyst datasets, transfer learning, or generative models. This order matters because it forces you to first learn baselines, data cleaning, and error analysis before you are impressed by scale. Mini-Project III follows this ladder: rather than train a large generative model, students run a released one and audit it with pymatgen.

13.5 3D Speculation: What Does AI + Materials Really Mean?

13.5.1 Scientific Meaning: AI Turns Search into a Strategy

The scientific meaning of AI + materials is not that discovery becomes automatic. It is that search becomes more strategic. Instead of testing candidates one by one, scientists can use data to decide which candidates are worth attention. Instead of staring at thousands of microscopy images, they can use computer vision to find patterns. Instead of treating failed experiments as waste, they can feed them back into the next decision.

But the hierarchy of evidence does not disappear. A model prediction is weaker than a simulation checked by physics. A simulation is weaker than a careful experiment. One experiment is weaker than a reproducible result. AI changes the route to evidence; it does not lower the standard for evidence.

A benchmark should therefore be treated like an exam, not like reality itself. If the training set and test set contain many close chemical relatives, a model can receive a high score by recognizing family resemblance. Real discovery is harsher: it asks the model to face unfamiliar chemistry, unusual structures, and conditions outside its comfort zone. This is why uncertainty estimates, failed predictions, and out-of-distribution tests are not technical decorations; they are part of scientific honesty.

The most reliable workflow therefore combines complementary strengths. A model can search broadly, physical knowledge can rule out impossible ideas, and experiments can reveal what a model missed. When independent groups can reproduce a result, the evidence becomes stronger. AI does not replace this chain; it changes where scientists spend their time within it.

13.5.2 Ethical Meaning: Speed Makes Responsibility More Urgent

AI can accelerate good science, but it can also accelerate bad habits. If a dataset is biased toward well-studied elements, the model may keep recommending familiar chemistry. If a paper reports predicted materials as if they were experimentally verified, readers may overestimate what has been proven. If an autonomous lab produces more data than humans can audit, weak validation can hide behind impressive scale.

There is also a resource question. Large models and large computations consume energy, money, and attention. A society that wants sustainable materials should not ignore the sustainability of its discovery tools. The ethical standard should be practical: record data provenance, share code and assumptions when possible, report uncertainty, and keep a human scientist responsible for interpretation.

Responsibility begins before a model is trained. Researchers choose which data to collect, which property to optimize, and which failures to record. These choices shape what the AI can notice and what it may ignore. A fast system aimed at the wrong question can make poor decisions more efficiently, so technical power must be matched by clear goals and careful oversight.

13.5.3 Social Meaning: Who Gets to Discover?

AI could democratize materials discovery. A young college student with open databases and a laptop can now reproduce workflows that once required a large

specialized group. Public resources such as the Materials Project, matminer, pymatgen, JARVIS, and Matbench lower the entry barrier.

But AI could also centralize discovery. The most powerful models may require proprietary data, large compute budgets, robotic labs, and corporate platforms. If only a few institutions can run the full pipeline, then AI + materials may widen the gap between well-funded and under-funded researchers.

This leads to the final question of the chapter:

> When AI suggests a material, who deserves credit: the model builder, the database creators, the experimentalist, the institution that owns the robot, or the human who asked the right question?

This tension is not decided by software alone. Shared datasets, open methods, transparent benchmarks, and meaningful access to computing and experiments can make participation broader. Without them, AI may simply move scientific advantage from one scarce resource to another.

There is no simple answer. AI + materials is a story about atoms and algorithms, but also about judgment. The future materials scientist will need to know how to train models. More importantly, they will need to know when not to trust them.

13.5.4 Future Outlook: More Capable Tools, Better Questions

The next generation of AI for materials will probably connect more kinds of evidence at once: chemical formulas and crystal structures, microscope images and spectra, laboratory notes and simulation results. Automated instruments may shorten the time between a prediction and a measurement. Open data and shared tools could let more students and researchers take part in this work.

Yet a more capable tool does not automatically produce a wiser science. The important questions will remain human ones: What problem is worth solving? What evidence is strong enough? Who benefits, and who bears the cost when a solution fails? The future of AI + materials should therefore be judged not only by how quickly it finds candidates, but also by whether it helps us ask better questions and build more reliable knowledge.

References

1. Anderson P. W., 1973. Resonating valence bonds: A new kind of insulator?[J]. Materials Research Bulletin, 8(2): 153-160. DOI: 10.1016/0025-5408(73)90167-0.

2. Bausch J., Senior A. W., Heras F. J. H., Edlich T., Davies A., Newman M., Jones C., Satzinger K., Niu M. Y., Blackwell S., Holland G., Kafri D., Atalaya J., Gidney C.,

Hassabis D., Boixo S., Neven H., Kohli P., 2024. Learning high-accuracy error decoding for quantum processors[J]. Nature, 635: 834-840. DOI: 10.1038/s41586-024-08148-8.

3. Bluvstein D., Evered S. J., Geim A. A., Li S. H., Zhou H., Manovitz T., Ebadi S., Cain M., Kalinowski M., Hangleiter D., Bonilla Ataides J. P., Maskara N., Cong I., Gao X., Sales Rodriguez P., Karolyshyn T., Semeghini G., Gullans M. J., Greiner M., Vuletić V., Lukin M. D., 2024. Logical quantum processor based on reconfigurable atom arrays[J]. Nature, 626: 58-65. DOI: 10.1038/s41586-023-06927-3.

4. Cao Z., Luo X., Lv J., Wang L., 2025a. Space group informed transformer for crystalline materials generation[J]. Science Bulletin, 70(21): 3522-3533. DOI: 10.1016/j.scib.2025.09.035.

5. Cao Z., Ou S., Wang L., 2025b. CrystalFormer-CSP: Thinking Fast and Slow for Crystal Structure Prediction. arXiv:2512.18251.

6. Cao Z., Wang L., 2025. CrystalFormer-RL: Reinforcement Fine-Tuning for Materials Design. arXiv:2504.02367.

7. Chen C., Ye W., Zuo Y., Zheng C., Ong S. P., 2019. Graph networks as a universal machine learning framework for molecules and crystals[J]. *Chemistry of Materials*, 31(9): 3564-3572. DOI: 10.1021/acs.chemmater.9b01294.

8. Chen E. H., Zhu G.-Y., Verresen R., Seif A., Bäumer E., Layden D., Tantivasadakarn N., Zhu G., Sheldon S., Vishwanath A., Trebst S., Kandala A., 2025. Nishimori transition across the error threshold for constant-depth quantum circuits[J]. Nature Physics, 21: 161-167. DOI: 10.1038/s41567-024-02696-6.

9. Choudhary K., DeCost B., 2021. Atomistic Line Graph Neural Network for improved materials property predictions[J]. *npj Computational Materials*, 7: 185. DOI: 10.1038/s41524-021-00650-1.

10. De Breuck P.-P., Hautier G., Rignanese G.-M., 2021. Materials property prediction for limited datasets enabled by feature selection and joint learning with MODNet[J]. *npj Computational Materials*, 7: 83. DOI: 10.1038/s41524-021-00552-2.

11. Dunn A., Wang Q., Ganose A., Dopp D., Jain A., 2020. Benchmarking materials property prediction methods: the Matbench test set and Automatminer reference algorithm[J]. *npj Computational Materials*, 6: 138. DOI: 10.1038/s41524-020-00406-3.

12. Goodall R. E. A., Lee A. A., 2020. Predicting materials properties without crystal structure: deep representation learning from stoichiometry[J]. *Nature Communications*, 11: 6280. DOI: 10.1038/s41467-020-19964-7.

13. Google Quantum AI and Collaborators, 2025. Quantum error correction below the surface code threshold[J]. Nature, 638: 920-926. DOI: 10.1038/s41586-024-08449-y.

14. He T., Huo H., Bartel C. J., Wang Z., Cruse K., Ceder G., 2023. Precursor recommendation for inorganic synthesis by machine learning materials similarity from scientific literature[J]. *Science Advances*, 9(23): eadg8180. DOI: 10.1126/sciadv.adg8180.

15. Hong J. H., Jang H. W., Son J. H., 2017. Determination of Volatile Flavour Profiles of Citrus spp. Fruits by SDE-GC-MS and Enantiomeric Composition of Chiral Compounds by MDGC-MS. Phytochemical Analysis, 28(5): 397-406. PMID: 28444796.

16. Iqbal M., Tantivasadakarn N., Verresen R., Campbell S. L., Dreiling J. M., Figgatt C., Gaebler J. P., Johansen J., Mills M., Moses S. A., Pino J. M., et al., 2024. Non-Abelian topological order and anyons on a trapped-ion processor[J]. Nature, 626: 505-511. DOI: 10.1038/s41586-023-06934-4.

17. Jiao R., Huang W., Lin P., Han J., Chen P., Lu Y., Liu Y., 2023. Crystal Structure Prediction by Joint Equivariant Diffusion[C]. Advances in Neural Information Processing Systems 36. arXiv:2309.04475.

18. Kitaev A. Yu., 2003. Fault-tolerant quantum computation by anyons[J]. Annals of Physics, 303(1): 2-30. DOI: 10.1016/S0003-4916(02)00018-0.

19. Lu H., Diaz D. J., Czarnecki N. J., et al., 2022. Machine learning-aided engineering of hydrolases for PET depolymerization[J]. Nature, 604: 662-667. DOI: 10.1038/s41586-022-04599-z.

20. Laska M., Liesen A., Teubner P., 1999. Enantioselectivity of odor perception in squirrel monkeys and humans. American Journal of Physiology-Regulatory, Integrative and Comparative Physiology, 277(4): R1098-R1103. DOI: 10.1152/ajpregu.1999.277.4.R1098.

21. Martirossyan M. M., Egg T., Hoellmer P., Karypis G., Transtrum M., Roitberg A., Liu M., Hennig R. G., Tadmor E., Martiniani S., 2025. All that structure matches does not glitter[C]. Advances in Neural Information Processing Systems 38, Datasets and Benchmarks Track. arXiv:2509.12178.

22. Jain A., Ong S. P., Hautier G., Chen W., Richards W. D., Dacek S., Cholia S., Gunter D., Skinner D., Ceder G., Persson K. A., 2013. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation[J]. *APL Materials*, 1: 011002. DOI: 10.1063/1.4812323.

23. Merchant A., Batzner S., Schoenholz S. S., Aykol M., Cheon G., Cubuk E. D., 2023. Scaling deep learning for materials discovery[J]. *Nature*, 624: 80-85. DOI: 10.1038/s41586-023-06735-9.

24. Ong S. P., Richards W. D., Jain A., Hautier G., Kocher M., Cholia S., Gunter D., Chevrier V. L., Persson K. A., Ceder G., 2013. Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis[J].

Computational Materials Science, 68: 314-319. DOI: 10.1016/j.commatsci.2012.10.028.

25. Pan E., Kwon S., Liu S., Xie M., Hoffman A. J., Duan Y., Prein T., Sheriff K., Roman-Leshkov Y., Moliner M., Gomez-Bombarelli R., Olivetti E. A., 2026. DiffSyn: a generative diffusion approach to materials synthesis planning[J]. *Nature Computational Science*, 6: 404-416. DOI: 10.1038/s43588-025-00949-9.

26. Rao Z., Tung P.-Y., Xie R., Wei Y., Zhang H., Ferrari A., Klaver T. P. C., Kwiatkowski da Silva A., Chen Y., Sukumar P. T., Korte-Kerzel S., Ponge D., Neugebauer J., Körmann F., Raabe D., 2022. Machine learning-enabled high-entropy alloy discovery[J]. *Science*, 378: 78-85. DOI: 10.1126/science.abo4940.

27. Roth K., Bauer H., Sjursnes B. J., Kvittingen L., Schmid R., 2025. The Persistent Myth of Limonene's Smell. ChemBioChem, 26(10): e202401085. DOI: 10.1002/cbic.202401085.

28. Szymanski N. J., Rendy B., Fei Y., Kumar R. E., He T., Milsted D., McDermott M. J., Gallant M., Cubuk E. D., Merchant A., Kim H., Jain A., Bartel C. J., Persson K., Zeng Y., Ceder G., 2023. An autonomous laboratory for the accelerated synthesis of inorganic materials[J]. *Nature*, 624: 86-91. DOI: 10.1038/s41586-023-06734-w.

29. Szymanski N. J., Rendy B., Fei Y. et al., 2026. Author Correction: An autonomous laboratory for the accelerated synthesis of inorganic materials[J]. *Nature*, 650: E1. DOI: 10.1038/s41586-025-09992-y.

30. Ward L., Dunn A., Faghaninia A., Zimmermann N. E. R., Bajaj S., Wang Q., Montoya J., Chen J., Bystrom K., Dylla M., Chard K., Asta M., Persson K. A., Snyder G. J., Foster I., Jain A., 2018. Matminer: An open source toolkit for materials data mining[J]. *Computational Materials Science*, 152: 60-69. DOI: 10.1016/j.commatsci.2018.05.018.

31. Wu J., Torresi L., Hu M., Reiser P., Zhang J. et al., 2024. Inverse design workflow discovers hole-transport materials tailored for perovskite solar cells[J]. *Science*, 386(6727): 1256-1264. DOI: 10.1126/science.ads0901.

32. Xie T., Fu X., Ganea O., Barzilay R., Jaakkola T., 2021. Crystal Diffusion Variational Autoencoder for Periodic Material Generation[C]. International Conference on Learning Representations. arXiv:2110.06197.

33. Zeni C., Pinsler R., Zügner D., Fowler A., Horton M., Fu X., Wang Z., Shysheya A., Crabbé J., Ueda S., Sordillo R., Sun L., Smith J., Nguyen B., Schulz H., Lewis S., Huang C.-W., Lu Z., Zhou Y., Yang H., Hao H., Li J., Yang C., Li W., Tomioka R., Xie T., 2025. A generative model for inorganic materials design[J]. *Nature*, 639: 624-632. DOI: 10.1038/s41586-025-08628-5.