MatterGen: a generative model for inorganic materials design

Claudio Zeni[1†], Robert Pinsler[1†], Daniel Zügner[1†],
Andrew Fowler[1†], Matthew Horton[1†], Xiang Fu[1],
Sasha Shysheya[1], Jonathan Crabbé[1], Lixin Sun[1], Jake Smith[1],
Ryota Tomioka[1*], Tian Xie[1*]

[1]Microsoft Research AI4Science.

*Corresponding author(s). E-mail(s): ryoto@microsoft.com;
tianxie@microsoft.com;
†Equal contribution; the authors are listed in random order.

Abstract
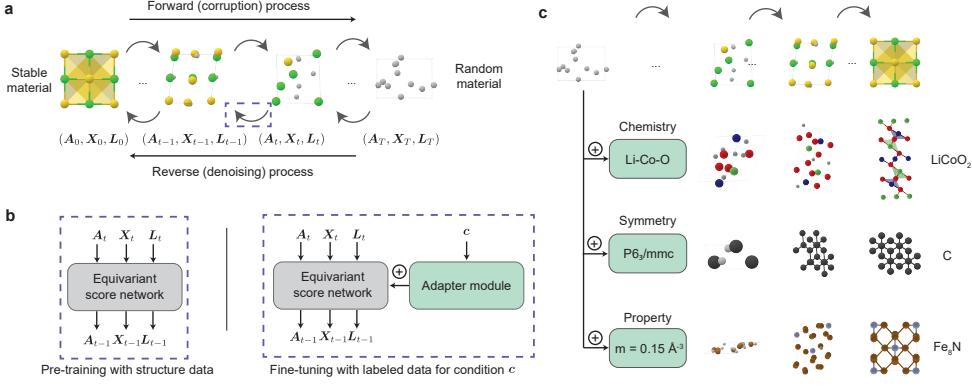
The design of functional materials with desired properties is essential in driving technological advances in areas like energy storage, catalysis, and carbon capture [1–3]. Generative models provide a new paradigm for materials design by directly generating entirely novel materials given desired property constraints. Despite recent progress, current generative models have low success rate in proposing stable crystals, or can only satisfy a very limited set of property constraints [4–13]. Here, we present MatterGen, a model that generates stable, diverse inorganic materials across the periodic table and can further be fine-tuned to steer the generation towards a broad range of property constraints. To enable this, we introduce a new diffusion-based generative process that produces crystalline structures by gradually refining atom types, coordinates, and the periodic lattice. We further introduce adapter modules to enable fine-tuning towards any given property constraints with a labeled dataset. Compared to prior generative models, structures produced by MatterGen are more than twice as likely to be novel and stable, and more than 15 times closer to the local energy minimum. After fine-tuning, MatterGen successfully generates stable, novel materials with desired chemistry, symmetry, as well as mechanical, electronic and magnetic properties. Finally, we demonstrate multi-property materials design capabilities by proposing structures that have both high magnetic density and a chemical composition with low supply-chain risk. We believe that the quality of generated materials and the breadth of MatterGen's capabilities represent a major advancement towards creating a universal generative model for materials design.

1

1   Introduction

The rate at which we can discover better materials has a major impact on the pace of technological innovation in areas such as carbon capture, semiconductor design, and energy storage. Traditionally, most novel materials have been found through experimentation and human intuition, which require long iteration cycles and are limited by the number of candidates that can be tested. Thanks to the advance of high throughput screening [14], open material databases [15–19], machine-learning-based property predictors [20, 21], and machine learning force fields (MLFFs) [22, 23], it has become increasingly common to screen hundreds of thousands of materials to identify promising candidates [24–27]. However, screening-based methods are still fundamentally limited by the number of known materials. The largest explorations of previously unknown crystalline materials are in the orders of 10^6–10^7 materials [23, 27–29], which is only a tiny fraction of the number of potential stable inorganic compounds (10^{10} quaternary compounds only considering stoichiometry [30]). Moreover, these methods cannot be efficiently steered towards finding materials with target properties.

Given these limitations, there has been an enormous interest in the inverse design of materials [31–34]. The aim of inverse design is to directly generate material structures that satisfy possibly rare or even conflicting target property constraints. Generative models are particularly promising since they have the potential to efficiently explore entirely new structures, yet they can also be flexibly adapted to different downstream tasks. Despite recent progress, current methods often fall short of producing stable materials according to density functional theory (DFT) calculations [4, 5, 35, 36], are constrained by a narrow subset of elements [8, 10, 11], and/or can only optimize a very limited set of properties, mainly formation energy [4, 5, 9, 13, 36].

In this study, we present MatterGen, a diffusion-based generative model for designing stable inorganic materials across the periodic table. MatterGen can further be fine-tuned via adapter modules to steer the generation towards materials with desired chemical composition, symmetry, and scalar property (e.g., band gap, bulk modulus, magnetic density) constraints. Compared to the previous state-of-the-art generative model for materials [4], MatterGen more than doubles the percentage of generated stable, unique, and novel (S.U.N.) materials, and generates structures that are more than 15 times closer to their ground-truth structures at the DFT local energy minimum. When fine-tuned, MatterGen often generates more S.U.N. materials in target chemical systems than well-established methods like substitution and random structure search (RSS), is capable of generating highly symmetric structures given the desired space groups, and directly generates S.U.N. materials that satisfy target mechanical, electronic, and magnetic property constraints. Finally, we showcase MatterGen's ability to design materials given multiple property constraints by generating promising materials that have both high magnetic density and a chemical composition with low supply-chain risk.

Fig. 1: **Inorganic materials design with MatterGen.** (a) MatterGen generates stable materials by reversing a corruption process through iteratively denoising an initially random structure. The forward diffusion process is designed to independently corrupt atom types $\boldsymbol{A}$, coordinates $\boldsymbol{X}$, and the lattice $\boldsymbol{L}$ to approach a physically motivated distribution of random materials. (b) An equivariant score network is pre-trained on a large dataset of stable material structures to jointly denoise atom types, coordinates, and the lattice. The score network is then fine-tuned with a labeled dataset through an adapter module that alters the model using the encoded property $\boldsymbol{c}$. (c) MatterGen can be fine-tuned to steer the generation towards materials with desired chemistry, symmetry, and scalar property constraints.

2 Results

2.1 Diffusion process for crystalline material generation

In MatterGen, we introduce a novel diffusion process tailored for crystalline materials (Fig. 1(a)). Diffusion models generate samples by learning a score network to reverse a fixed corruption process [37–39]. Corruption processes for images typically add Gaussian noisebut crystalline materials have unique periodic structures and symmetries which demand a customized diffusion process. A crystalline material can be defined by its repeating unit, i.e., its unit cell, which encodes the atom types $\boldsymbol{A}$ (i.e., chemical elements), coordinates $\boldsymbol{X}$, and periodic lattice $\boldsymbol{L}$ (Appendices A.1 and A.2). We define a corruption process for each component that suits their own geometry and has physically motivated limiting noise distributions. More concretely, the coordinate diffusion respects the periodic boundary by employing a wrapped Normal distribution and approaches a uniform distribution at the noisy limit (Appendix A.6). The lattice diffusion takes a symmetric form and approaches a distribution whose mean is a cubic lattice with average atomic density from the training data (Appendix A.7). The atom diffusion is defined in categorical space where individual atoms are corrupted into a masked state (Appendix A.5). Given the corrupted structure, we learn a score network that outputs equivariant scores for atom type, coordinate, and lattice, respectively,
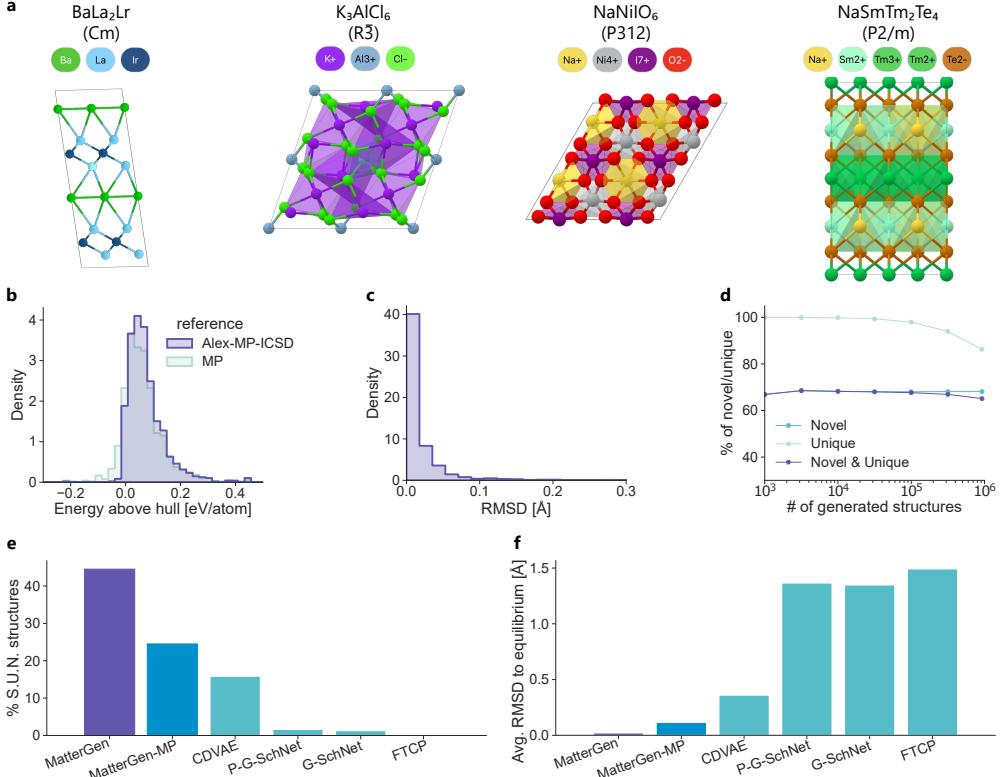
which removes the need to learn the symmetries from data (Appendix A.8). We refer to this network as the base model.

To generate materials with desired property constraints, we introduce adapter modules that can be used for fine-tuning the base model on an additional dataset with property labels (Fig. 1(b), more details in Appendix B). Fine-tuning is particularly appealing as it still works well if the labeled dataset is small compared to unlabeled structure datasets, which is often the case due to the high computational cost of calculating properties. The adapter modules are tunable components injected into each layer of the base model to alter its output depending on the given property label [40]. The resulting fine-tuned model is used in combination with classifier-free guidance [41] to steer the generation towards target property constraints. We apply this approach to multiple types of properties, producing a set of fine-tuned models that can generate materials with target chemical composition, symmetry, or scalar properties such as magnetic density (Fig. 1(c)).

2.2 Generating stable, diverse materials

We formulate learning a generative model for inverse materials design as a two-step process, where we first pre-train a general base model for generating stable, diverse crystals across the periodic table, and then we fine-tune this base model towards different downstream tasks. In this section, we focus on the ability of MatterGen's base model to generate stable, diverse materials, which we argue is a prerequisite for addressing any inverse materials design task. Since diversity is difficult to measure directly, we resort to quantifying MatterGen's ability to generate S.U.N. materials, and provide an additional analysis of the quality and diversity of generated structures. To train the base model, we curate a large, diverse dataset comprising 607,684 stable structures with up to 20 atoms recomputed from the Materials Project (MP) [15] and Alexandria [29, 42] datasets, which we refer to as Alex-MP-20. We consider a structure to be stable if its energy per atom after relaxation via DFT is below the 0.1 eV/atom threshold of a reference dataset comprising 1,081,850 unique structures recomputed from the MP, Alexandria, and Inorganic Crystal Structure Database (ICSD) datasets. We refer to this dataset as Alex-MP-ICSD. We consider a structure to be novel if it is not contained in Alex-MP-ICSD. We adopt these definitions throughout unless stated otherwise. More details are in Appendices C and D.3.1.

Fig. 2(a) shows several random samples generated by MatterGen, featuring typical coordination environments of inorganic materials; see Appendix D.3.2 for a more detailed analysis. To assess stability, we perform DFT calculations on 1024 generated structures. Fig. 2(b) shows that 78 % of generated structures fall below the 0.1 eV/atom threshold (13 % below 0.0 eV/atom) of MP's convex hull, while 75 % fall below the 0.1 eV/atom threshold (3 % below 0.0 eV/atom) of the combined Alex-MP-ICSD hull (Fig. 2(b)). Further, 95 % of generated structures have an RMSD w.r.t. their DFT-relaxed structures that is below 0.076 Å (Fig. 2(c)), which is almost one order of magnitude smaller than the atomic radius of the hydrogen atom (0.53 Å). These results indicate that the majority of structures generated by MatterGen are stable, and very close to the DFT local energy minimum. We further investigate whether

4

Fig. 2: **Generating stable, unique and novel inorganic materials. (a)** Visualization of four randomly selected crystals generated by MatterGen, with corresponding chemical formula and space group symbols. **(b)** Distribution of energy above the hull using MP and Alex-MP-ICSD dataset as energy references, respectively. **(c)** Distribution of root mean squared displacement (RMSD) between initial generated structures and DFT relaxed structures. **(d)** Percentage of unique, novel structures as a function of number of generated structures. Novelty is defined with respect to Alex-MP-ICSD. **(e-f)** Percentage of S.U.N. structures (e) and average RMSD between initial and DFT-relaxed structures (f) for MatterGen, MatterGen-MP, and several baseline models, including CDVAE [4], P-G-SchNet, G-SchNet [43], and FTCP [36].

MatterGen can generate a substantial amount of unique and novel materials. We showcase in Fig. 2(d) that the percentage of unique structures is 100 % when generating 1000 structures and only drops to 86 % after generating one million structures, while novelty remains stable around 68 %. This suggests that MatterGen is able to generate diverse structures without significant saturation even at a large scale, and that the majority of those structures are novel with respect to Alex-MP-ICSD.
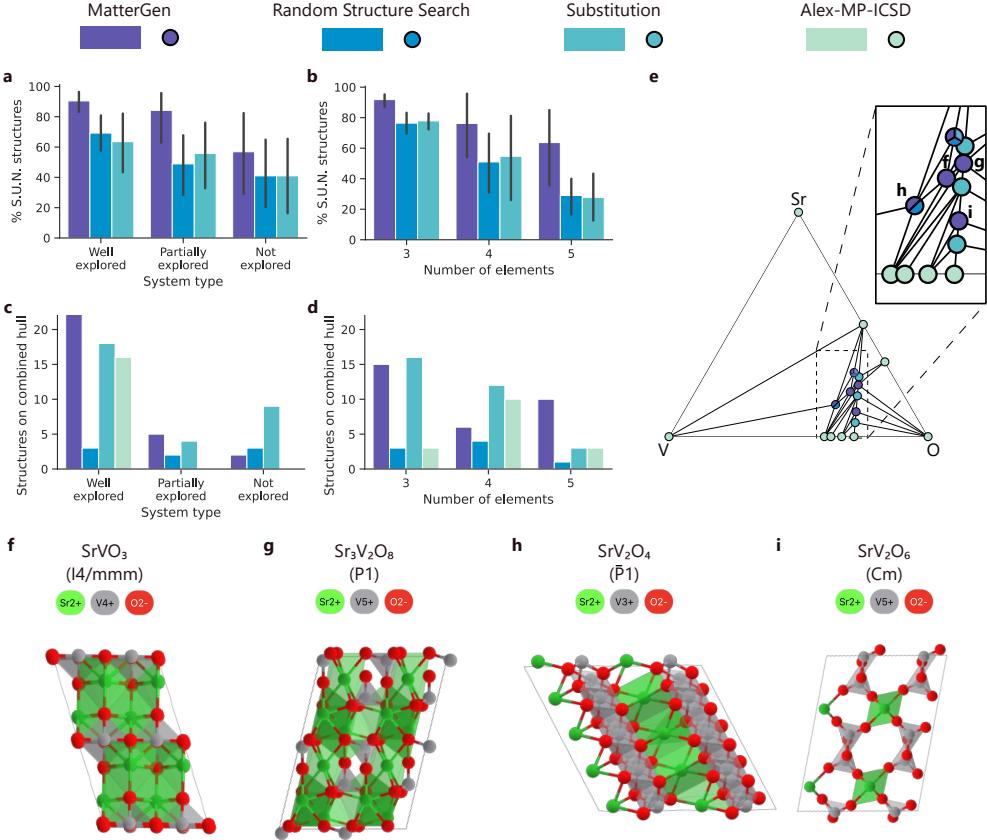
Moreover, we benchmark MatterGen against previous generative models for materials and show a significant improvement in performance. We focus on two key metrics: (1) the percentage of S.U.N. materials among generated samples, measuring the overall

success rate of generating promising candidates, and (2) the average RMSD between generated samples and their DFT-relaxed structures, measuring the distance to equilibrium. We also compare to MatterGen-MP, which is a MatterGen model trained only on MP-20, i.e., the same, smaller, dataset used by the other baselines. In Fig. 2(e-f), MatterGen-MP shows a 1.8 times increase in the percentage of S.U.N. structures and a 3.1 times decrease in average RMSD compared with the previous state-of-the-art CDVAE [4]. When comparing MatterGen with MatterGen-MP, we observe a further 1.6 times increase in the percentage of S.U.N. structures and a 5.5 times decrease in RMSD as a result of scaling up the training dataset.

In summary, we have shown that MatterGen is able to generate S.U.N. materials at a substantially higher rate compared to previous generative models while the generated structures are orders of magnitudes closer to their local energy minimum. Next, we fine-tune the pre-trained base model of MatterGen towards different downstream applications, including target chemistry (Section 2.3), symmetry (Section 2.4), and scalar property constraints (Sections 2.5 and 2.6).

2.3 Generating materials with target chemistry

Finding the most stable material structures in a target chemical system (e.g., Li-Co-O) is crucial to define the true convex hull required for assessing stability, and indeed is one of the major challenges in materials design [44]. The most comprehensive approach for this task is *ab initio* RSS [45], which has been used to discover many novel materials that were later experimentally synthesized [44]. The biggest drawback of RSS is its computational cost, as the thorough exploration of even a ternary compound can require hundreds of thousands of DFT relaxations. In recent years, the combination of random or substitution-guided generation with MLFFs has proven successful in exploring chemical systems [23, 27]. Here, we evaluate the ability of MatterGen to explore target chemical systems by comparing it with substitution [23] and RSS [45, 46]. We equip all methods with a MLFF based on the M3GNet architecture [23], which is used to pre-relax and filter the generated structures by their predicted stability before running more expensive DFT calculations. We fine-tune the MatterGen base model (Appendix B.1) and steer the generated structures towards different target chemical systems and an energy above hull of 0.0 eV/atom. We perform the benchmark evaluation for nine ternary, nine quaternary, and nine quinary chemical systems. For each of these three groups, we pick three chemical systems at random from the following categories: well explored, partially explored, and not explored. See Appendix D.4 for additional details. In Fig. 3(a-b) we see that MatterGen generates the highest percentage of S.U.N. structures for every system type and every chemical complexity. As highlighted in Fig. 3(c), MatterGen also finds the highest number of unique structures on the combined convex hull in (1) 'partially explored' systems, where existing known structures near the hull were provided during training, and in (2) 'well-explored systems', where structures near the hull are known but were not provided in training. While substitution offers a comparable or more efficient way to generate structures on the hull for ternary and quaternary systems, MatterGen achieves better performance on quinary systems, as shown in Fig. 3(d). Remarkably, the strong performance of MatterGen in quinary systems was achieved with only 10,240 generated samples, while
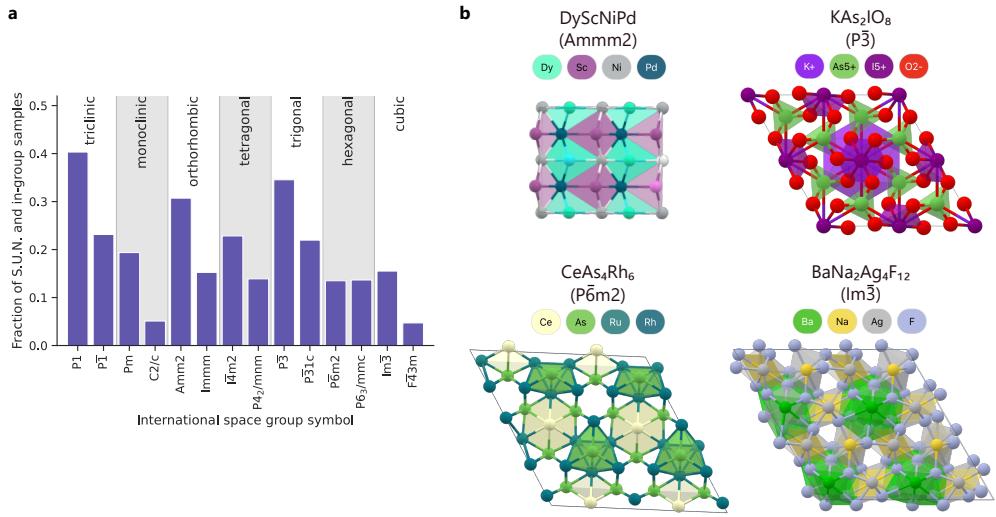
Fig. 3: **Generating materials in target chemical system. (a-b)** Mean percentage of S.U.N. structures generated by MatterGen and baselines for 27 chemical systems, grouped by system type (a) and number of elements (b). Vertical black lines indicate maximum and minimum values. **(c-d)** Number of structures on the combined convex hull found by each method and in the Alex-MP-ICSD dataset, grouped by system type (c) and number of elements (d). **(e)** Convex hull diagram for V-Sr-O, a well-explored ternary system. The dots represent structures on the hull, their coordinates represent the element ratio of their composition, and their color indicates by which method they were discovered. **(f-i)** Four of the five structures MatterGen discovered on the V-Sr-O hull depicted in (e), along with their composition and space group.

both substitution ($\sim$70,000 samples), and RSS (600,000) suffered from the combinatorial explosion of candidates (see Appendix D.4 for more details). This underscores the enormous efficiency gains that can be realized with generative models by proposing better initial candidates. Finally, in Fig. 3(e) we show that MatterGen finds five novel structures on the combined hull for V-Sr-O—an example of a well-explored
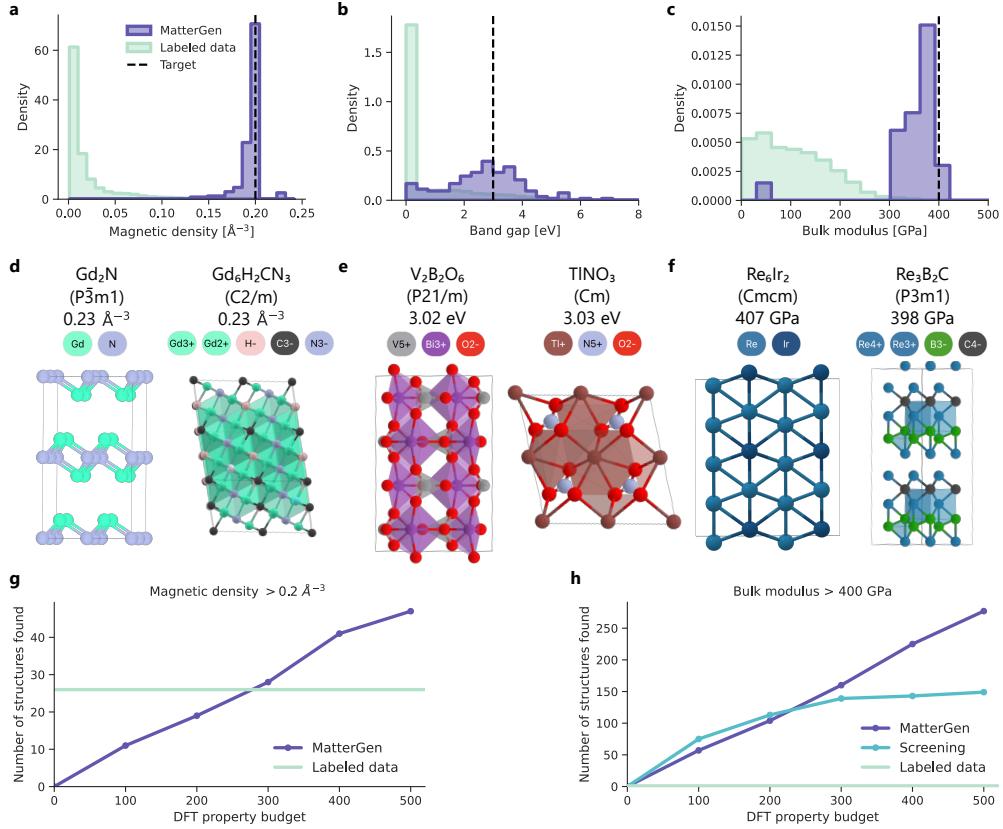
ternary system—while substitution finds four, and RSS only two. A few of the structures discovered by MatterGen are shown in Fig. 3(f-i), and are analyzed in-depth in Appendix D.4.2.

2.4 Designing materials with target symmetry

The symmetry of a material directly affects its electronic and vibrational properties, and is a determining factor for its topological [47] and ferroelectric [48] characteristics. The generation of S.U.N. materials with a given symmetry is a challenging task, as the symmetric arrangement of atoms in space is hard to enforce without resorting to explicit constraints based on already known materials. Here, we assess MatterGen's ability to generate S.U.N. materials with a target symmetry by fine-tuning it on space group labels. We choose 14 space groups at random from the subset of space groups that had at least 1000 entries in the training dataset, two for each of the seven crystal systems, and generate 256 structures per space group. The results are shown in Fig. 4(a). On average, the fraction of generated S.U.N. structures that belong to the target space group is 20%, and still surpassing 10% for some of the most highly symmetric space groups that were chosen, e.g., $P6_3/mmc$ and $Im\bar{3}$. This is a notable result given that most previous generative models struggled in generating highly symmetric crystals [4, 49]. In Fig. 4(b), we show four randomly generated S.U.N. structures from different space groups. Additional details and results are provided in Appendix D.5.


Fig. 4: **Generating materials with target symmetry. (a)** Fraction of generated S.U.N. structures that belong to the target space group for 14 randomly chosen space groups spanning the seven lattice types. **(b)** Four randomly selected S.U.N. structures generated by MatterGen, along with their chemical formula and space group.

Fig. 5: **Generating materials with target magnetic, electronic, and mechanical properties. (a-c)** Density of property values among (1) generated S.U.N. samples by MatterGen, and (2) structures in the labeled fine-tuning dataset for a magnetic, electronic, and mechanical property, respectively. The property target for MatterGen is shown as a black dashed line. Magnetic density values $< 10^{-3}$ Å^{-3} in (a) are excluded from the labeled data to improve readability. **(d-f)** Visualization of S.U.N. structures with the best property values generated by MatterGen for magnetic density (d), band gap (e), and bulk modulus (f). Alongside each structure, the chemical formula, space group and property value is shown. **(g-h)** Number of S.U.N. structures that satisfy target constraints found MatterGen compared to number of structures found by baselines across a range of DFT property calculation budgets.

2.5 Designing materials with target magnetic, electronic, and mechanical properties

There is an enormous need for new materials with improved properties across a wide range of real-world applications, e.g., for designing carbon capture technologies, solar cells, or semiconductors [24–26]. The classical screening-based approach starts from a set of candidates and selects the ones with the best properties. However, screening

methods are unable to explore structures beyond the set of known materials. Here, we demonstrate MatterGen's ability to directly generate S.U.N. materials with target constraints on three different single-property inverse design tasks. These feature a diverse set of properties—magnetic, electronic, and mechanical—with varying degrees of available labeled data for fine-tuning the model. In the first task, we aim to generate materials with high magnetic density, a prerequisite for permanent magnets. We fine-tune the model on 605,000 structures with DFT magnetic density labels (calculated assuming ferromagnetic ordering) and then generate structures with a target magnetic density value of 0.20 Å^{-3}. Second, we search for materials with a specific electronic property. We fine-tune the model on 42,000 structures with DFT band gap labels, then sample materials with a target calculated band gap value of 3.0 eV. Finally, we target structures with high bulk modulus—an important property for superhard materials. We fine-tune the model on only 5,000 labeled structures, and sample with a target value of 400 GPa. While the tasks above were chosen to evaluate the generality of the model, we note that additional follow-up investigations would be required to assess the suitability of these materials for specific applications, e.g., a superhard material needs to satisfy additional constraints such as a high shear modulus, or a permanent magnet needs a suitable magnetic order and critical temperature. See Appendix D.6 for more details.

In Fig. 5(a-c), we highlight the significant shift in the distribution of property values among S.U.N. samples generated by MatterGen towards the desired targets, even when the targets are at the tail of the data distribution. In particular, this still holds true for properties where the number of DFT labels available for fine-tuning the model is substantially smaller than the size of the unlabeled training data. In Fig. 5(d-f) we showcase the S.U.N. structures with the best property values generated by MatterGen for each task. See Appendix D.6.2 for additional analysis.

Moreover, we assess how many S.U.N. structures satisfying extreme property constraints can be found by MatterGen when given a limited budget for DFT property calculations. As a baseline, we count the number of materials in the labeled fine-tuning dataset that satisfy the constraint. We also compare with a screening approach, which scans previously unlabeled materials for promising candidates. In contrast to the previous experiment, we fine-tune MatterGen with labels predicted by a machine learning property predictor—the same used for the screening baseline—when the dataset is not fully labeled. As shown in Fig. 5(g), MatterGen is able to find up to 47 S.U.N. structures with magnetic density above 0.2 Å^{-3}, much more than the 26 materials with such high property values in the fine-tuning dataset. Since the dataset is fully labeled, there is no screening baseline available. In Fig. 5(h), we see that MatterGen finds substantially more S.U.N. materials with high bulk modulus than screening. While the number of structures found by screening saturates with increasing budget, MatterGen keeps discovering S.U.N. structures at an almost constant rate. Given a budget of 500 DFT property calculations, we find 277 S.U.N. structures, almost double the number found with a screening approach (149). In contrast, there are only two materials in the labeled fine-tuning dataset with such high bulk modulus values. We note however that although all structures in figure Fig. 5(h) are unique and stable according to the definition in Appendix D.3.1, both the MatterGen and screening sets of structures

contain, in many cases, multiple unique structures per composition. There are 126 and and 79 distinct compositions for MatterGen and screening, respectively.
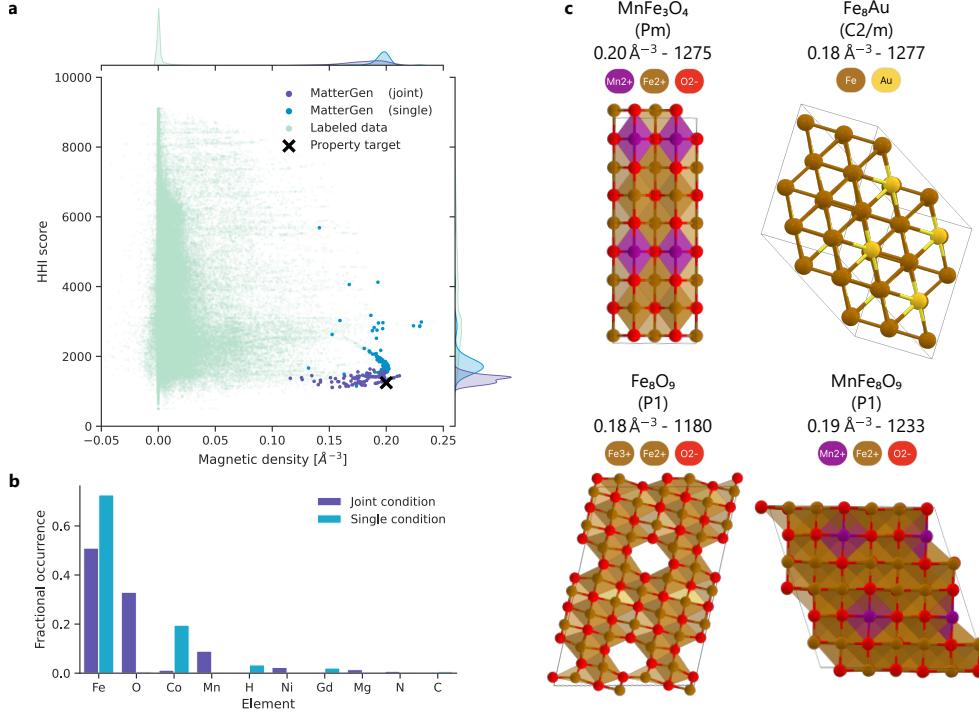
2.6 Designing low-supply-chain-risk magnets

Most materials design problems require finding structures satisfying multiple property constraints. MatterGen can be fine-tuned to generate materials given any combination of constraints. Here, we showcase its ability to tackle materials design problems with multiple constraints by searching for low-supply-chain-risk magnets. Since many existing high-performing permanent magnets contain rare earth elements that are subject to supply chain risks, there has been increasing interest in discovering rare-earth-free permanent magnets [50]. We simplify the problem of finding such a magnet to finding materials with high magnetic density and a low Herfindahl–Hirschman index (HHI). According to the U.S. Department of Justice and the Federal Trade Commission, a material with an HHI score below 1500 is considered to have low supply chain risk [51]. Thus, we ask the model to generate materials with a magnetic density of 0.2 Å^{-3} and an HHI score of 1250. We note that supply chain risk is not making a claim that a material is economically viable; an expensive element might have a low supply chain risk.

In Fig. 6(a), we observe that MatterGen generates S.U.N. structures that are narrowly distributed around the target values, despite the labeled fine-tuning data being extremely scarce in that region. Compared to a model that only targets high magnetic density values (single), targeting both properties (joint) shifts the distribution of HHI scores closer towards the desired target value while retaining high magnetic density values. Fig. 6(b) showcases the effect of jointly targeting both properties on the distribution of elements found in the generated materials. Due to the lower HHI scores, elements commonly employed for high-magnetic density materials that have supply chain issues, e.g., Cobalt (Co) and Gadolinium (Gd), are practically absent in the jointly generated structures. In contrast, these elements are still present in structures generated by the model that only targets materials with high magnetic density (single condition).

3 Discussion

Generative models are particularly promising for tackling inverse design tasks as they can potentially explore entirely *novel* structures with desired properties in an efficient way. However, generating the 3D structure of stable crystalline materials is challenging due to their periodicity and the interplay between atom types, coordinates, and lattice. MatterGen improves upon limitations of previous methods [4, 49] by introducing a joint diffusion process for atom types, coordinates, and lattice (Section 2.1 and Appendices A.5 to A.7), which—combined with the substantially larger training dataset Alex-MP-20—drastically increase the stability, uniqueness, and novelty of generated materials. Thanks to the introduction of the adapter modules (Appendix B.1), MatterGen can further be fine-tuned to generate S.U.N. structures satisfying target constraints across a wide range of properties, with performance improvements over

Fig. 6: **Designing low-supply-chain-risk magnets.** **(a)** Distribution of S.U.N. structures generated by MatterGen when fine-tuned on the HHI score (single) and on both HHI score and magnetic density (joint), as well as structures from the labeled fine-tuning dataset. The property target of MatterGen is denoted as a black cross. **(b)** Occurrence of most frequent elements in S.U.N. structures for the two fine-tuned MatterGen models. **(c)** S.U.N. structures on the Pareto front for the joint property optimization task, along with their chemical composition, space group, magnetic density, and HHI score.

widely-employed methods such as MLFF-assisted RSS and substitution (Section 2.3), as well as ML-assisted screening (Section 2.5).

Despite these advances, MatterGen could still be improved in several ways. For example, we observe that the model disproportionately generates structures with P1 symmetry compared to the training data, indicating a tendency for generating less symmetric structures, especially for larger crystals (see discussion in Appendix D.2). We hypothesize that further improvements on the denoising process, the backbone architecture, and the expansion of the training dataset could enable the model to overcome such issues. We also acknowledge that our extensive evaluations only cover some of the criteria required for real-world applications, with experimental validation and characterization being the ultimate test. We discuss the challenges in evaluating the quality of crystalline materials from generative models in Appendix D.2.

Overall, we believe that the breadth of MatterGen's capabilities and the quality of generated materials represent a major advancement towards creating a universal generative model for materials with high real-world impact. Given the transformative impact of generative models in domains like image generation [52] and protein design [53], we envision that generative models like MatterGen will have a major impact in materials design in the coming years. As such, we are excited about the many directions in which MatterGen could be further extended. For instance, MatterGen could be expanded to cover a broader class of materials ranging from catalyst surfaces to metal organic frameworks, enabling us to tackle challenging problems like nitrogen fixation [54] and carbon capture [55]. The property constraints can be extended to non-scalar quantities like the band structure or X-ray diffraction (XRD) spectrum, which would further enable applications ranging from band engineering to the prediction of atomic structures of experimentally-measured XRD spectra of unknown samples.

Supplementary information Additional explanations and details regarding the MatterGen architecture, fine-tuning approach, datasets, and results can be found in the supplementary information.

Declarations

Author contributions

AF, MH, RP, RT, TX, CZ and DZ conceived the study, implemented the methods, performed experiments, and wrote the manuscript. XF led the development of the adapter modules. JC, XF, JS, LS and SS helped with implementing the methods, conducting experiments and writing the manuscript. RT and TX led the research.

Appendix

Table of Contents

A   Diffusion model for periodic materials

This section contains additional model details, following the notation listed in Table A1.

General notation	
$n \in \mathbb{N}$	Number of atoms in a crystal
$\boldsymbol{M} = (\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{L})$	A crystal structure
$\boldsymbol{X} \in [0,1)^{3 \times n}$	Fractional atomic coordinates
$\tilde{\boldsymbol{X}} \in \mathbb{R}^{3 \times n}$	Cartesian atomic coordinates
$\boldsymbol{A} \in \mathbb{A}^n$	Atomic species in a crystal
$\boldsymbol{L} = (\boldsymbol{l}^1, \boldsymbol{l}^2, \boldsymbol{l}^3) \in \mathbb{R}^{3 \times 3}$	The unit cell lattice matrix
$\boldsymbol{l}^j \in \mathbb{R}^3,\ j \in \{1,2,3\}$	The j-th lattice vector
$\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \dots \boldsymbol{v}_N) \in \mathbb{R}^{d \times N}$	Concatenation of N d-dimensional column vectors into a matrix
$\mathcal{E} \subset \{1, 2, \dots, n\}^2 \times \mathbb{Z}^3$	Set of edges in a material
$i, j \in \{1, 2, \dots, n\}$	Index of an atom in a material
$d \in \mathbb{N}$	The number of hidden dimension in our GNN
$\boldsymbol{1}_n \in \mathbb{R}^n$	n-dimensional column vector containing ones
Diffusion notation	
$t \in 1, 2, \dots, T$	Diffusion timestep
$T \in \mathbb{N}$	Number of time discretization steps for the diffusion process
$q(\boldsymbol{x}_0)$	The data distribution
$q(\boldsymbol{x}_t \vert \boldsymbol{x}_{t-1})$	Single-step diffusion transition kernel
$q(\boldsymbol{x}_t \vert \boldsymbol{x}_0)$	One-shot diffusion kernel
$q(\boldsymbol{x}_T)$	Prior (noise) distribution
$\boldsymbol{s}_{\boldsymbol{\theta}}(\cdot, t)$	Score model
$\boldsymbol{s}_{\boldsymbol{X}, \boldsymbol{\theta}}(\cdot, t)$	Score model for atomic coordinates
$\log p_{\boldsymbol{\theta}}(\boldsymbol{A}_0 \vert \boldsymbol{X}_t, \boldsymbol{L}_t, \boldsymbol{A}_t, t)$	Predicted logits for atom types at $t = 0$.
$\boldsymbol{s}_{\boldsymbol{L}, \boldsymbol{\theta}}(\cdot, t)$	Score model for lattice
$\boldsymbol{z}$	Standard Gaussian noise $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

Table A1: Table of notations

A.1   Representation of periodic materials

Any crystal structure can be represented by some repeating unit (called the *unit cell*) that tiles the entire 3D space. The unit cell itself contains a number of atoms that are arranged inside of it. Thus, we use the following universal representation for a material $\boldsymbol{M}$:

$$\boldsymbol{M} = (\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{L}), \tag{A1}$$

where $\boldsymbol{A} = (a^1, a^2, \dots, a^n)^\top \in \mathbb{A}^n$ are the atomic species of the atoms inside the unit cell; $\boldsymbol{L} = (\boldsymbol{l}^1, \boldsymbol{l}^2, \boldsymbol{l}^3) \in \mathbb{R}^{3 \times 3}$ is the lattice, i.e., the shape of the repeating unit cell; and $\boldsymbol{X} = (\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^n) \in [0,1)^{3 \times n}$ are the *fractional* coordinates of the atoms inside the unit cell.

The lattice $\boldsymbol{L}$ is a parallelepiped defined by the three lattice vectors $\boldsymbol{l}^1, \boldsymbol{l}^2$, and $\boldsymbol{l}^3$. It can thus be compactly represented as a single 3×3 matrix with the three lattice vectors as its columns. The volume of a lattice is given by $\mathrm{Vol}(\boldsymbol{L}) = |\det \boldsymbol{L}|$. Any

physically sensible crystal must have a unit cell with nonzero volume, hence, we require any lattice matrix to be non-singular.

Fractional coordinates express the location of an atom using the lattice vectors as the basis vectors. For instance, an atom with fractional coordinates $\boldsymbol{x} = (0.2, 0.3, 0.5)^\top$ has Cartesian coordinates $\tilde{\boldsymbol{x}} = 0.2\boldsymbol{l}^1 + 0.3\boldsymbol{l}^2 + 0.5\boldsymbol{l}^3$. The periodicity in fractional coordinates is defined by the (flat) unit hypertorus, i.e., we have the equivalence relation $\boldsymbol{x} \sim \boldsymbol{x} + \boldsymbol{k}, \boldsymbol{k} \in \mathbb{Z}^3$. We can convert between fractional coordinates $\boldsymbol{X}$ and Cartesian coordinates $\tilde{\boldsymbol{X}}$ as follows:

$$\tilde{\boldsymbol{X}} = \boldsymbol{L}\boldsymbol{X}, \tag{A2}$$

$$\boldsymbol{X} = \boldsymbol{L}^{-1}\tilde{\boldsymbol{X}}. \tag{A3}$$

A.2 Invariance and equivariance in periodic materials

The energy per atom $\epsilon(\boldsymbol{M}) = E(\boldsymbol{M})/n$ of a periodic material $\boldsymbol{M} = (\boldsymbol{X}, \boldsymbol{L}, \boldsymbol{A})$ has several invariances.

- Permutation invariance: $\epsilon(\boldsymbol{X}, \boldsymbol{L}, \boldsymbol{A}) = \epsilon(\boldsymbol{P}(\boldsymbol{X}), \boldsymbol{L}, \boldsymbol{P}(\boldsymbol{A}))$ for every permutation matrix $\boldsymbol{P}$.
- Translation invariance: $\epsilon(\boldsymbol{X}, \boldsymbol{L}, \boldsymbol{A}) = \epsilon(\boldsymbol{X} + \boldsymbol{t}, \boldsymbol{L}, \boldsymbol{A})$ for every $\boldsymbol{t} \in \mathbb{R}^3$.
- Rotation invariance: $\epsilon(\boldsymbol{X}, \boldsymbol{L}, \boldsymbol{A}) = \epsilon(\boldsymbol{X}, \boldsymbol{R}(\boldsymbol{L}), \boldsymbol{A})$ for every rotation matrix $\boldsymbol{R} \in O(3)$.
- Periodic cell choice invariance: $\epsilon(\boldsymbol{X}, \boldsymbol{L}, \boldsymbol{A}) = \epsilon(\boldsymbol{C}^{-1}\boldsymbol{X}, \boldsymbol{L}\boldsymbol{C}, \boldsymbol{A})$, where $\boldsymbol{C}$ triangular with $\det \boldsymbol{C} = 1$ and $\boldsymbol{C} \in \mathbb{Z}^{3\times3}$. See Fig. A1 for an example.
- Supercell invariance: $\epsilon(\boldsymbol{X}, \boldsymbol{L}, \boldsymbol{A}) = \epsilon\left(\bigoplus_{i=0}^{\det(\boldsymbol{C})} \boldsymbol{C}^{-1}(\boldsymbol{X} + \boldsymbol{k}_i\boldsymbol{1}_n^\top), \boldsymbol{L}\boldsymbol{C}, \bigoplus_{i=0}^{\det(\boldsymbol{C})} \boldsymbol{A}\right)$, where $\boldsymbol{C}$ is a 3×3 diagonal matrix with positive integers on the diagonal, $\boldsymbol{k}_i \in \mathbb{N}^3$ indexes the cell repetitions in the three lattice components, and $\bigoplus$ indicates concatenation.

Forces are instead equivariant to permutation and rotation, while being invariant to translation and periodic cell choice. Stress tensors are similarly invariant to permutation, translation, supercell choice, and periodic cell choice; while being equivariant to rotation (see Appendix A.8.1 for additional details).

A.3 Diffusion model background and notation

Diffusion models [37, 38, 56, 57] are a class of generative models that learn to revert a diffusion process. The diffusion process (also called the *forward* process) gradually corrupts an input sample $\boldsymbol{x}_0$ via transition kernels $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$[1], defining a Markov chain $\boldsymbol{x}_0 \to \boldsymbol{x}_1 \to \cdots \to \boldsymbol{x}_T$, where $T \in \mathbb{N}$ is the number of diffusion steps and $1 \leq t \leq T$. Here, we cover the typical case where the data is continuous-valued and the transition kernels are Normal distributions. See Appendix A.5 for details on discrete diffusion models.

---

[1] We follow the convention in machine learning literature that the functional forms of (conditional) probability density functions depend on the variables that appear as arguments. For example, $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ could be written as $q_{\boldsymbol{X}_t|\boldsymbol{X}_{t-1}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ to make the dependence of the functional form on t explicit, but we avoid this to prevent clutter.

The transition kernels are of the general form $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(f(\boldsymbol{x}_{t-1}, t), \sigma_t^2 \boldsymbol{I})$, where $f(\boldsymbol{x}_{t-1}, t): \mathbb{R}^d \to \mathbb{R}^d$ is affine in $\boldsymbol{x}_{t-1}$. This implies that the one-shot transition kernel $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ is also Gaussian, and for popular choices $f(\cdot, t)$ the mean and variance are known in closed form. This enables us to efficiently obtain a noisy sample $\boldsymbol{x}_t$ at an arbitrary time step t during training.

Diffusion models are optimized to approximate the score function of the noise distributions $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ for any $1 \le t \le T$:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{t=1}^{T} \sigma_t^2 \mathbb{E}_{q(\boldsymbol{x}_0)} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[\| \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0) \|_2^2 \right], \qquad \text{(A4)}$$

where $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}, t): \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^d$ is called the *score model*. It is standard practice [37, 38] to parameterize the model to predict the *noise* $\epsilon_t = -\sigma_t \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ instead of the score, since the magnitude of $\epsilon_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ is independent of the diffusion time step t.

The forward diffusion process is designed such that $q(\boldsymbol{x}_T|\boldsymbol{x}_0) \approx q(\boldsymbol{x}_T)$, where $q(\boldsymbol{x}_T)$ is a prior distribution that is easy to sample from (e.g., Gaussian).

In this work we leverage two popular diffusion processes for continuous data, i.e., the variance-exploding diffusion [37, 58] and the variance-preserving diffusion [38, 56] process, which we briefly explain in the following.

Variance-exploding diffusion

This is the diffusion process used in denoising score matching (DSM) [37]. We define a sequence of exponentially increasing standard deviations $\sigma_{\min} = \sigma_1, \ldots, \sigma_T = \sigma_{\max}$ that define the transition kernels:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}\left(\boldsymbol{x}_t, \left(\sigma_t^2 - \sigma_{t-1}^2\right) \boldsymbol{I}\right), \qquad q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_0, \sigma_t^2 \boldsymbol{I}\right). \qquad \text{(A5)}$$

We can generate a sample using the learned model via annealed Langevin dynamics [37, 58] or ancestral sampling from the graphical model $\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ [39]:

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}_t + (\sigma_t^2 - \sigma_{t-1}^2)\boldsymbol{s}_{\boldsymbol{\theta}^*}(\boldsymbol{x}_t, t) + \boldsymbol{z}\sqrt{\sigma_t^2 - \sigma_{t-1}^2}, \qquad \text{(A6)}$$

where $\boldsymbol{x}_T \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_T^2 \boldsymbol{I}\right)$, and $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ is standard Gaussian noise. In Appendix A.6 we explain how we leverage variance-exploding diffusion in the diffusion process of the fractional coordinates.

Variance-preserving diffusion

This is the diffusion process used to train denoising diffusion probabilistic models (DDPMs) [38, 56]. In variance-preserving diffusion we define a sequence of positive noise scales $0 < \beta_1, \beta_2, \ldots \beta_T < 1$ to obtain transition kernels of the form

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}\left(\sqrt{1 - \beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}\right), \qquad q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1 - \bar{\alpha}_t) \boldsymbol{I}\right), \quad \text{(A7)}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} (1 - \beta_t)$. Sampling from a model trained to revert the variance-preserving diffusion process also works via *ancestral sampling* from the graphical model $\prod_{t=1}^{T} p_{\boldsymbol{\theta}} (\boldsymbol{x}_{t-1} | \boldsymbol{x}_t)$:

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{1 - \beta_t}} \left(\boldsymbol{x}_t + \beta_t \boldsymbol{s}_{\boldsymbol{\theta}^*}(\boldsymbol{x}_t, t) \right) + \sqrt{\beta_t} \boldsymbol{z}, \tag{A8}$$

starting from $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, where $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ is standard Gaussian noise. See Appendix A.7 for details about how we leverage variance-preserving diffusion in the diffusion process of the lattice.

A.4 Joint diffusion process

To apply the construction of a diffusion process described in Appendix A.3 to crystal structures described in Appendix A.1, we define the forward process through a Markov chain $\boldsymbol{M}_0 \to \boldsymbol{M}_1 \to \cdots \to \boldsymbol{M}_T$ via a transition kernel that diffuses the atom coordinates, atom types, and the lattice independently as follows:

$$\begin{aligned} q(\boldsymbol{A}_{t+1}, \boldsymbol{X}_{t+1}, \boldsymbol{L}_{t+1} | \boldsymbol{A}_t, \boldsymbol{X}_t, \boldsymbol{L}_t) \\ = q(\boldsymbol{A}_{t+1} | \boldsymbol{A}_t) q(\boldsymbol{X}_{t+1} | \boldsymbol{X}_t) q(\boldsymbol{L}_{t+1} | \boldsymbol{L}_t) \qquad (t = 0, 1, \dots, T-1). \end{aligned} \tag{A9}$$

In addition, the noise distributions of atom species $\boldsymbol{A}$ and the fractional coordinates $\boldsymbol{X}$ factorize into the diffusion of the individual atoms:

$$q(\boldsymbol{A}_{t+1} | \boldsymbol{A}_t) = \prod_{i=1}^{n} q(a_{t+1}^i | a_t^i), \qquad q(\boldsymbol{X}_{t+1} | \boldsymbol{X}_t) = \prod_{i=1}^{n} q(\boldsymbol{x}_{t+1}^i | \boldsymbol{x}_t^i).$$

Note that the factorization of the forward diffusion process does not imply that the reverse diffusion process factorizes in the same way. Details of the atom type diffusion, coordinate diffusion, and lattice diffusion are described in Appendix A.5, Appendix A.6, Appendix A.7, respectively. The architecture of the score network $s_\theta(\boldsymbol{M}_t, t)$ is described in Appendix A.8. The combined objective function is presented in Appendix A.9.

A.5 Atom type diffusion

For the diffusion of the (discrete) atom species $\boldsymbol{A}$, we use the discrete denoising diffusion probabilistic model (D3PM) approach [57], which is a generalization of DDPMs to discrete data problems. As in DDPM, the forward diffusion process is a Markov process that gradually corrupts an input sample a_0, which is a scalar discrete random variable with K categories (e.g., atomic species):

$$q(a_{1:T} | a_0) = \prod_{t=1}^{T} q(a_t | a_{t-1}), \tag{A10}$$

where $a_0 \sim q(a_0)$ is an atomic species sampled from the data distribution and $a_T \sim q(a_T)$, where $q(a_T)$ is a prior distribution that is easy to sample from.

Denoting the one-hot representation of a as a row vector $\boldsymbol{a}$, we can express the transitions as:

$$q(\boldsymbol{a}_t | \boldsymbol{a}_{t-1}) = \text{Cat}(\boldsymbol{a}_t; \boldsymbol{p} = \boldsymbol{a}_{t-1} \boldsymbol{Q}_t), \tag{A11}$$

where $[\boldsymbol{Q}_t]_{ij} = q(a_t = j | a_{t-1} = i)$ is the Markov transition matrix at time step t. $\text{Cat}(\boldsymbol{a}; \boldsymbol{p})$ is a categorical distribution over one-hot vectors whose probabilities are given by the row vector $\boldsymbol{p}$. Similar to DDPM, D3PM assumes that the forward diffusion factorizes over all discrete variables of a data point, i.e., all atomic species are diffused independently with the same transition matrices $\boldsymbol{Q}_t$. Hence, we only consider individual one-hot vectors in this section. D3PMs are trained by optimizing a variational lower bound:

$$L_{\text{vb}} = \mathbb{E}_{q(\boldsymbol{a}_0)} \Bigg[- \mathbb{E}_{q(\boldsymbol{a}_1 | \boldsymbol{a}_0)} \log p_{\boldsymbol{\theta}}(\boldsymbol{a}_0 | \boldsymbol{a}_1, 1) + D_{\text{KL}} \left[q(\boldsymbol{a}_T | \boldsymbol{a}_0) \, || \, q(\boldsymbol{a}_T) \right]$$

$$+ \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{a}_t | \boldsymbol{a}_0)} D_{\text{KL}} \left[q(\boldsymbol{a}_{t-1} | \boldsymbol{a}_t, \boldsymbol{a}_0) \, || \, p_{\boldsymbol{\theta}}(\boldsymbol{a}_{t-1} | \boldsymbol{a}_t) \right] \Bigg]. \tag{A12}$$

Moreover, Austin et al. [57] propose an additional cross-entropy loss on the model's prediction $p_{\boldsymbol{\theta}}(\boldsymbol{a}_0 | \boldsymbol{a}_t, t)$:

$$L_{\text{CE}} = -\mathbb{E}_{q(\boldsymbol{a}_0)} \left[\sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{a}_t | \boldsymbol{a}_0)} \log p_{\boldsymbol{\theta}}(\boldsymbol{a}_0 | \boldsymbol{a}_t, t) \right],$$

so that the overall loss becomes

$$L = L_{\text{vb}} + \lambda_{\text{CE}} L_{\text{CE}}. \tag{A13}$$

Three important characteristics of DDPM and DSM are that (1) given $\boldsymbol{x}_0$ we can sample noisy samples $\boldsymbol{x}_t$ for arbitrary t in constant time; (2) after sufficiently many diffusion steps, $\boldsymbol{x}_T$ follows a prior distribution that is easy to sample from; and (3) the posterior $q(\boldsymbol{x}_{t-1} | \boldsymbol{x}_t, \boldsymbol{x}_0)$ in Eq. (A12) is tractable and can be computed efficiently. D3PM also has these properties, as we briefly outline in the following:

(1) Fast sampling of $\boldsymbol{a}_t \sim q(\boldsymbol{a}_t | \boldsymbol{a}_0)$. Since the forward diffusion in D3PM is governed by discrete transition matrices $\{\boldsymbol{Q}_t\}_{t=1}^{T}$, we can write

$$q(\boldsymbol{a}_t | \boldsymbol{a}_0) = \text{Cat}(\boldsymbol{a}_t; \boldsymbol{p} = \boldsymbol{a}_{t-1} \bar{\boldsymbol{Q}}_t), \qquad \text{where } \bar{\boldsymbol{Q}}_t = \boldsymbol{Q}_1 \boldsymbol{Q}_2 \ldots \boldsymbol{Q}_t. \tag{A14}$$

The cumulative transition matrices $\bar{\boldsymbol{Q}}_t$ can be pre-computed and for many diffusion processes even have a closed form.

(2) Tractable prior distribution. Two of the proposed diffusion processes are the absorbing (which we employ in MatterGen) and uniform diffusion processes. Both

gradually diffuse the data towards a limit distribution, which are the one-hot distribution on the absorbing state and the uniform distribution over all categories, respectively. For more details, we refer to Appendix A of Austin et al. [57].

(3) Tractable posterior $q(\boldsymbol{a}_{t-1}|\boldsymbol{a}_t, \boldsymbol{a}_0)$. Using Bayes' rule and exploiting the Markov property $q(\boldsymbol{a}_t|\boldsymbol{a}_{t-1}, \boldsymbol{a}_0) = q(\boldsymbol{a}_t|\boldsymbol{a}_{t-1})$, we can write

$$q(\boldsymbol{a}_{t-1}|\boldsymbol{a}_t, \boldsymbol{a}_0) = \frac{q(\boldsymbol{a}_t|\boldsymbol{a}_{t-1})q(\boldsymbol{a}_{t-1}|\boldsymbol{a}_0)}{q(\boldsymbol{a}_t|\boldsymbol{a}_0)}. \tag{A15}$$

All terms in Eq. (A15) can be computed efficiently in closed form given the forward diffusion process.

Reverse sampling process.

We generate a sample $\boldsymbol{a}_0$ by first sampling $\boldsymbol{a}_T$ and then gradually updating it to obtain $p_{\boldsymbol{\theta}}(\boldsymbol{a}_{0:T}) = q(\boldsymbol{a}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{a}_{t-1}|\boldsymbol{a}_t)$. Austin et al. [57] propose to parameterize $p_{\boldsymbol{\theta}}(\boldsymbol{a}_{t-1}|\boldsymbol{a}_t)$ by predicting a distribution over $\boldsymbol{a}_0$ and then marginalizing it out:

$$p_{\boldsymbol{\theta}}(\boldsymbol{a}_{t-1}|\boldsymbol{a}_t) \propto \sum_{\boldsymbol{a}_0} q(\boldsymbol{a}_{t-1}, \boldsymbol{a}_t|\boldsymbol{a}_0) p_{\boldsymbol{\theta}}(\boldsymbol{a}_0|\boldsymbol{a}_t, t), \tag{A16}$$

where we can use our tractable posterior computation again. Since we have a discrete state space, marginalizing out $\boldsymbol{a}_0$ by explicit summation has complexity $\mathcal{O}(K)$. In the case of atomic species we have $K \simeq 100$; thus, this is relatively cheap. This parameterization has the advantage that potential sparsity in the diffusion process is efficiently enforced by using $q(\boldsymbol{a}_{t-1}, \boldsymbol{a}_t|\boldsymbol{a}_0)$ without having to be learned by the model.

Forward diffusion process.

As the specific flavor of D3PM forward diffusion we employ the masked diffusion process, which has shown best performance in the original study [57] as well as our initial experiments. Following Austin et al. [57], we introduce an extra atom species [MASK] at index $K - 1$, which is the absorbing or masked state. At each timestep t, the transition matrices have the particularly simple form

$$[\boldsymbol{Q}_t^{\text{absorbing}}]_{ij} = \begin{cases} 1 & \text{if } i = j = m, \\ 1 - \beta_t & \text{if } i = j \neq m, \\ \beta_t & \text{if } j = m \neq i, \\ 0 & \text{if } m \neq i \neq j \neq m, \end{cases} \tag{A17}$$

where m corresponds to the absorbing state. Intuitively, each species has probability $1 - \beta_t$ of staying unchanged, and probability β_t of transitioning to the absorbing state. Once a species is absorbed, it can never leave that state, and there are no transitions between different non-masked atomic species. Thus, the limit distribution of this diffusion process is a point mass on the absorbing state.

A.6 Coordinate diffusion

For our model we perform diffusion on the *fractional* coordinates and outline the approach in the following. See Appendix A.6.3 for a brief outline why we favor fractional coordinate diffusion over Cartesian. The fractional coordinates in a crystal structure live in a Riemannian manifold referred to as the flat torus $\mathbb{T}^3 = \mathbb{S}^1 \times \mathbb{S}^1 \times \mathbb{S}^1$, i.e., it is the quotient space $\mathbb{R}^3/\mathbb{Z}^3$ with equivalence relation:

$$\boldsymbol{x} + \boldsymbol{k} \sim \boldsymbol{x}, \qquad \boldsymbol{k} \in \mathbb{Z}^3. \tag{A18}$$

Thus, adding Gaussian noise to fractional coordinates naturally corresponds to sampling from a *wrapped* Normal distribution, whose probability density is

$$\mathcal{N}_{\mathrm{W}}\left(\bar{\boldsymbol{x}}; \boldsymbol{x}, \sigma^2 \boldsymbol{I}, \boldsymbol{I}\right) = \sum_{\boldsymbol{k} \in \mathbb{Z}^3} \mathcal{N}\left(\bar{\boldsymbol{x}}; \boldsymbol{x} - \boldsymbol{k}, \sigma^2 \boldsymbol{I}\right), \tag{A19}$$

where $\mathcal{N}_{\mathrm{W}}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{B}\right)$ denotes a wrapped Normal distribution with mean $\boldsymbol{\mu}$, covariance matrix $\boldsymbol{\Sigma}$, and periodic boundaries $\boldsymbol{B}$. If the periodic boundaries are $[0, 1)^3$, i.e., $\boldsymbol{B} = \boldsymbol{I}$, we write $\mathcal{N}_{\mathrm{W}}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ for brevity.

For the diffusion of the atom coordinates we use variance-exploding diffusion, i.e., the variance of the diffusion process increases exponentially with diffusion time. This has the advantage that the prior distribution $q(\boldsymbol{x}_T)$ is particularly simple, i.e., the uniform distribution in the range $[0, 1)^3$. Jing et al. [59] use this approach for torsional angles—which live in a 1D flat torus—in small molecule generation. The one-shot noising process of the fractional coordinates is therefore defined as

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}_{\mathrm{W}}\left(\boldsymbol{x}_t; \boldsymbol{x}_0, \sigma_t^2 \boldsymbol{I}\right). \tag{A20}$$

A.6.1 Variance adjustment for atomic density

One limitation of using a constant variance for the fractional coordinate diffusion is that the diffusion in Cartesian space will have difference variance depending on the size of the unit cell. This limitation becomes clear if we express the distribution of the Cartesian coordinates $\tilde{\boldsymbol{x}}_t$ using Eq. (A2) via linear transformation of a Gaussian random variable $\boldsymbol{x}_t$:

$$q(\tilde{\boldsymbol{x}}_t, |\boldsymbol{x}_0, \boldsymbol{L}_t) = \mathcal{N}_{\mathrm{W}}\left(\tilde{\boldsymbol{x}}_t; \boldsymbol{L}_t \boldsymbol{x}_0, \sigma_t^2 \boldsymbol{L}_t \boldsymbol{L}_t^\top, \boldsymbol{L}_t\right). \tag{A21}$$

Observe that the covariance matrix $\boldsymbol{\Sigma}_t = \sigma_t^2 \boldsymbol{L}_t \boldsymbol{L}_t^\top$ of the noisy Cartesian coordinates depends on the lattice. Thus, the (generalized) variance of the noise distribution also depends on the size of the unit cell, i.e., $|\det(\boldsymbol{\Sigma}_t)| = \left(\sigma_t^3 |\det \boldsymbol{L}_t|\right)^2$.

We mitigate this effect by scaling the variance in fractional coordinate diffusion based on the size of the unit cell. Assuming roughly constant atomic density $\mathrm{d}(\boldsymbol{L}_t) = \frac{n}{\mathrm{Vol}(\boldsymbol{L}_t)} \propto 1 \Leftrightarrow \mathrm{Vol}(\boldsymbol{L}_t) = |\det \boldsymbol{L}_t| \propto n$. We therefore choose to scale σ_t accordingly, i.e.,

$$\sigma_t(n) = \frac{\sigma_t}{\sqrt[3]{n}}, \tag{A22}$$

such that $|\det(\boldsymbol{\Sigma}_t)| = \left(\frac{\sigma_t^3}{n}|\det \boldsymbol{L}_t|\right)^2$ is no longer proportional to n.

A.6.2   Score computation for fractional coordinates

Recall from Eq. (A4) that training diffusion models requires computing the score function for the one-shot transition kernel. However, for the wrapped Normal distribution in Eq. (A19), (log-)likelihood and score computation are intractable because of the infinite sum. Given the thin tails of the Normal distribution, both can be approximated reasonably well with a truncated sum. More specifically, the score function of the isotropic wrapped Normal distribution can be expressed as

$$\nabla_{\bar{\boldsymbol{x}}} \log q_\sigma(\bar{\boldsymbol{x}}|\boldsymbol{x}) = \sum_{\boldsymbol{k} \in \mathbb{Z}^3} w_{\boldsymbol{k}} \frac{\bar{\boldsymbol{x}} - \boldsymbol{x} + \boldsymbol{k}}{\sigma^2}, \tag{A23}$$

where

$$w_{\boldsymbol{k}} = \frac{1}{Z} \exp\left(-\frac{\|\bar{\boldsymbol{x}} - \boldsymbol{x} + \boldsymbol{k}\|^2}{2\sigma^2}\right), \qquad Z = \sum_{\boldsymbol{k}' \in \mathbb{Z}^3} \exp\left(-\frac{\|\bar{\boldsymbol{x}} - \boldsymbol{x} + \boldsymbol{k}'\|^2}{2\sigma^2}\right). \tag{A24}$$

A.6.3   Fractional vs Cartesian coordinate diffusion

Instead of diffusing fractional coordinates as in MatterGen, one could diffuse Cartesian coordinates, e.g., as done in CDVAE [4] (and in generative methods for molecules [60]).

However, this approach is not suitable for our framework. To see this, note that while in CDVAE the lattice $\boldsymbol{L}$ is fixed during the diffusion of the atom coordinates, we diffuse the lattice simultaneously to the atom coordinates (and atomic species). This makes diffusion of Cartesian coordinates dependent on the lattice diffusion because the wrapped Normal's covariance matrix and periodic boundaries at diffusion timestep t depend on knowing the lattice matrix $\boldsymbol{L}_t$. Consequently, our diffusion process from Eq. (A9) no longer factorizes into lattice and coordinates and needs to be adapted:

$$\begin{aligned}
&q(\tilde{\boldsymbol{X}}_{t+1}, \boldsymbol{L}_{t+1}, \boldsymbol{A}_{t+1} | \tilde{\boldsymbol{X}}_t, \boldsymbol{L}_t, \boldsymbol{A}_t) \\
&= q(\tilde{\boldsymbol{X}}_{t+1} | \tilde{\boldsymbol{X}}_t, \boldsymbol{L}_{t+1}, \boldsymbol{L}_t) q(\boldsymbol{L}_{t+1} | \boldsymbol{L}_t) q(\boldsymbol{A}_{t+1} | \boldsymbol{A}_t).
\end{aligned} \tag{A25}$$

Here, we need to condition $q(\tilde{\boldsymbol{X}}_{t+1})$ on $\boldsymbol{L}_{t+1}$ and $\boldsymbol{L}_t$ because in order to convert the Cartesian coordinates at time step t to time step $t+1$ we first need to convert $\tilde{\boldsymbol{x}}_t$ to fractional coordinates using $\boldsymbol{L}_t^{-1}$, and then to Cartesian coordinates at $t+1$ using $\boldsymbol{L}_{t+1}$.

The one-shot distribution of noisy Cartesian coordinates (similar to Eq. (A20) for the fractional case) becomes:

$$q(\tilde{\boldsymbol{x}}_t | \tilde{\boldsymbol{x}}_0, \{\boldsymbol{L}_{t'}\}_{t'=1}^t) = \mathcal{N}_{\mathrm{W}}\left(\tilde{\boldsymbol{x}}_t; \boldsymbol{L}_t \boldsymbol{L}_0^{-1} \tilde{\boldsymbol{x}}_0, \boldsymbol{L}_t \left(\sum_{t'=1}^t \sigma_{t'}^2 \boldsymbol{L}_{t'}^{-1} (\boldsymbol{L}_{t'}^\top)^{-1}\right) \boldsymbol{L}_t^\top, \boldsymbol{L}_t\right). \tag{A26}$$

Observe that we require the entire trajectory of noisy lattices $\boldsymbol{L}_1, \ldots, \boldsymbol{L}_t$ in order to express the noise distribution of the Cartesian atomic coordinates. This means that we first need to sample the *entire* diffusion trajectory of the lattice, which is slow. Further, we have found computing the one-shot covariance matrix for the Cartesian coordinates to be numerically unstable for long diffusion trajectories. We therefore use the diffusion process of fractional coordinates described in the previous section.

A.7   Lattice diffusion

In addition to the diffusion of the atom types and coordinates described above, we also diffuse and denoise the lattice $\boldsymbol{L}$ in our approach. We use variance-preserving diffusion, as variance-exploding diffusion would lead to extremely large unit cells in the noisy limit. Those are challenging to handle for a graph neural network (GNN) with a fixed edge cutoff and would require the model to learn scores over a wide range of different length scales.

A.7.1   Fixed-rotation lattice diffusion

As the distribution of materials is invariant to global rotation, we can either choose a rotation-invariant prior distribution over unit cells, or decide on a canonical rotational alignment that we use throughout diffusion and denoising. We opt for the latter, as it gives us some more flexibility designing the diffusion process. Here, we choose to represent the lattice as a symmetric matrix. We can do so via the polar decomposition based on the singular value decomposition:

$$\boldsymbol{L} = \boldsymbol{U}\tilde{\boldsymbol{L}}, \qquad \boldsymbol{U} = \boldsymbol{W}\boldsymbol{V}^\top, \qquad \tilde{\boldsymbol{L}} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^\top, \tag{A27}$$

where $\boldsymbol{W}$ and $\boldsymbol{V}$ are the left and right singular vectors of $\boldsymbol{L}$, respectively, and $\boldsymbol{\Sigma}$ is the diagonal matrix of singular values. $\boldsymbol{U}$ is a rotation matrix and $\tilde{\boldsymbol{L}}$ is a symmetric positive-definite matrix.

We restrict our entire forward lattice diffusion to symmetric matrices by enforcing the noise $\boldsymbol{z} \in \mathbb{R}^{3\times3}$ on the lattice to be symmetric, e.g., by only modeling the upper-triangular part of the matrix and mirroring it to the lower triangular part. Notice that this effectively fixes the rotation, i.e., we only have six degrees of freedom left. Going forward, we only consider symmetric lattices and lattice noise.

A.7.2   Lattice diffusion with custom limit mean and variance

Naively using variance-preserving diffusion independently on the lattice vectors leads to the following forward diffusion distribution:

$$q(\boldsymbol{L}_t | \boldsymbol{L}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{L}_0, (1 - \bar{\alpha}_t)\boldsymbol{I}\right). \tag{A28}$$

However, for large t we observed that the majority of the resulting unit cells have very small volume and steep angles, which means that the atoms are extremely densely

packed inside the noisy cells. We therefore modify the diffusion process as follows:

$$q(\boldsymbol{L}_t|\boldsymbol{L}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{L}_0 + (1 - \sqrt{\bar{\alpha}_t})\mu(n)\boldsymbol{I}, (1 - \bar{\alpha}_t)\sigma_t^2(n)\boldsymbol{I}\right), \tag{A29}$$

which yields the following limit distribution for $T \to \infty$:

$$q(\boldsymbol{L}_T) = \mathcal{N}\left(\mu(n)\boldsymbol{I}, \sigma_T^2(n)\boldsymbol{I}\right). \tag{A30}$$

Thus, in the limit distribution we have a tendency towards cubic lattices $\boldsymbol{L} \propto \boldsymbol{I}$, which often occur in nature and have a relatively narrow range of volumes. Further, the lattice vector angles when sampling from the prior are mostly concentrated between 60° and 120°. This aligns both with the range of angles of Niggli-reduced cells as well as the initialization range of cell vector angles in RSS [61], suggesting that this is a good starting point for the generation process.

To understand the choice of the mean in the limit distribution, recall that the volume of a parallelepiped $\boldsymbol{L}$ is $|\det \boldsymbol{L}|$. We introduce a scalar coefficient $\mu(n)$ that depends on the number of atoms in the cell to make the atomic density of the mean noisy lattice roughly constant for differently sized systems. Setting $\mu(n) = \sqrt[3]{nc}$, the volume of the prior mean becomes $\mathrm{Vol}(\sqrt[3]{nc}\boldsymbol{I}) = nc$. Thus, the atomic density of the prior mean becomes $\mathrm{d}(\mu(n)\boldsymbol{I}) = \frac{n}{\mathrm{Vol}(\mu(n)\boldsymbol{I})} = \frac{1}{c}$. We can set c to the inverse average density of the dataset as a reasonable prior.

Similarly, we adjust the variance in the limit distribution to be proportional to the volume, such that the signal-to-noise ratio of the noisy lattices is constant across the numbers of nodes. To this end, we set the limit standard deviation as $\sigma(n) = \sqrt[3]{n\nu}$. Thus, for a diagonal entry of the lattice matrix the signal-to-noise-ratio in the limit is $\lim_{t \to \infty} \frac{|\mu(n)|}{\sigma(n)} = \frac{\sqrt[3]{nc}}{\sqrt[3]{n\nu}} = \left(\frac{c}{\nu}\right)^{1/3}$ and therefore independent of the number of atoms.

A.8 Architecture of the score network

We employ an SE(3)-equivariant GNN to predict scores for the lattice, atom positions, and atom types in the denoising process. In particular, we adapt the GemNet architecture by Gasteiger et al. [62], originally developed to be a universal MLFF. GemNet is a symmetric message-passing GNN that uses directional information to achieve SO(3)-equivariance, and incorporates two- and three-body information in the first layer for efficiency. Since we do not predict energies, we adapt the direct (i.e., non-conservative) force prediction variant of the model, named GemNet-dT, which has been shown to be more computationally efficient and accurate in these scenarios [62]. We employ four message-passing layers, a cutoff radius of 7 Å for the neighbor list construction, and set the dimension of hidden representations for nodes and edges to 512.

We train the model to predict Cartesian coordinate scores $\boldsymbol{s}_{\boldsymbol{X},\boldsymbol{\theta}}(\boldsymbol{X}_t, \boldsymbol{L}_t, \boldsymbol{A}_t, t)$ as if they were non-conservative forces. These scores are equivariant to rotation and permutation, and invariant to translation. We then transform the Cartesian scores into fractional scores following Eq. (A2).

For the atom-type reverse diffusion, recall from Eq. (A16) that we predict the atom types $\boldsymbol{A}_0$ given the atom types $\boldsymbol{A}_t$ at time t. For materials, we additionally condition

on lattice $\boldsymbol{L}_t$ and coordinates $\boldsymbol{X}_t$; more precisely, the (unnormalized) log-probabilities $\log p_{\boldsymbol{\theta}}(\boldsymbol{A}_0|\boldsymbol{X}_t, \boldsymbol{L}_t, \boldsymbol{A}_t, t)$ of the atomic species at $t = 0$ are computed as:

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{A}_0|\boldsymbol{X}_t, \boldsymbol{L}_t, \boldsymbol{A}_t, t) = \boldsymbol{H}^{(L)}\boldsymbol{W}, \tag{A31}$$

where $\boldsymbol{H}^{(L)} \in \mathbb{R}^{n \times d}$ are the hidden representations of atoms at the last message-passing layer L, and $\boldsymbol{W} \in \mathbb{R}^{d \times K}$ are the weights of a fully-connected linear layer, with K being the number of atom types (including the masked null state).

A.8.1 Computation of the predicted lattice scores

To predict the lattice scores $\boldsymbol{s}_{L,\boldsymbol{\theta}}(\boldsymbol{X}_t, \boldsymbol{L}_t, \boldsymbol{A}_t, t)$, we utilize the model's hidden representations of the edges. For layer l, we denote the edge representation of the edge $(ij\boldsymbol{k}) \in \mathcal{E}$ between atoms i and j as $\boldsymbol{m}_{ij\boldsymbol{k}}^l \in \mathbb{R}^d$, where i is inside the unit cell and j is $\boldsymbol{k} \in \mathbb{Z}^3$ unit cells displaced from the center unit cell. We use a multi-layer perceptron (MLP) $\phi^l : \mathbb{R}^d \to \mathbb{R}$ to predict a scalar score per edge. We treat this as a prediction by the model indicating by how much an edge's length should increase or decrease, and translate this into a predicted transformation of the lattice via chain rule derivation:

$$\begin{aligned}
\frac{\partial \tilde{d}_{ij\boldsymbol{k}}}{\partial \boldsymbol{L}_t} &= \frac{\partial}{\partial \boldsymbol{L}_t} \left\| \boldsymbol{L}_t \left(\boldsymbol{x}_t^j - \boldsymbol{x}_t^i + \boldsymbol{k} \right) \right\|_2 \\
&= \frac{1}{\tilde{d}_{ij\boldsymbol{k}}} \boldsymbol{L}_t \left(\boldsymbol{x}_t^j - \boldsymbol{x}_t^i + \boldsymbol{k} \right) \cdot \left(\boldsymbol{x}_t^j - \boldsymbol{x}_t^i + \boldsymbol{k} \right)^\top \\
&= \frac{1}{\tilde{d}_{ij\boldsymbol{k}}} \tilde{\boldsymbol{d}}_{ij\boldsymbol{k}}(\boldsymbol{d}_{ij\boldsymbol{k}})^\top, \tag{A32}
\end{aligned}$$

where $\tilde{d}_{ij\boldsymbol{k}} = \|\tilde{\boldsymbol{d}}_{ij\boldsymbol{k}}\|_2$ and $\tilde{\boldsymbol{d}}_{ij\boldsymbol{k}} = \boldsymbol{L}_t \left(\boldsymbol{x}_t^j - \boldsymbol{x}_t^i + \boldsymbol{k} \right)$ are the edge length and edge displacement in Cartesian coordinates, respectively, and $\boldsymbol{d}_{ij\boldsymbol{k}} = \boldsymbol{x}_t^j - \boldsymbol{x}_t^i + \boldsymbol{k}$ is the edge displacement in fractional coordinates. The predicted lattice score *per edge* is then $\phi^l(\boldsymbol{m}_{ij\boldsymbol{k}}^l) \cdot \frac{\partial \tilde{d}_{ij\boldsymbol{k}}}{\partial \boldsymbol{L}_t}$. These predicted scores are averaged over all edges $(ij\boldsymbol{k}) \in \mathcal{E}$ to get the predicted lattice score for layer l:

$$\hat{\boldsymbol{s}}_{L,\boldsymbol{\theta}}^l(\boldsymbol{X}_t, \boldsymbol{L}_t, \boldsymbol{A}_t, t) = \frac{1}{|\mathcal{E}|} \sum_{(ij\boldsymbol{k}) \in \mathcal{E}} \phi^l(\boldsymbol{m}_{ij\boldsymbol{k}}^l) \cdot \frac{1}{\tilde{d}_{ij\boldsymbol{k}}} \tilde{\boldsymbol{d}}_{ij\boldsymbol{k}}(\boldsymbol{d}_{ij\boldsymbol{k}})^\top. \tag{A33}$$

Stacking the model's predictions into a diagonal matrix $\boldsymbol{\Phi}^l \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|} = \mathrm{diag}\left(\frac{\phi^l(\boldsymbol{m}_{ij\boldsymbol{k}}^l)}{|\mathcal{E}| \cdot \tilde{d}_{ij\boldsymbol{k}}} \right)$, we can write more concisely

$$\hat{\boldsymbol{s}}_{L,\boldsymbol{\theta}}^l(\boldsymbol{X}_t, \boldsymbol{L}_t, \boldsymbol{A}_t, t) = \tilde{\boldsymbol{D}}\boldsymbol{\Phi}^l\boldsymbol{D}^\top = \boldsymbol{L}_t\boldsymbol{D}\boldsymbol{\Phi}^l\boldsymbol{D}^\top, \tag{A34}$$

where $\tilde{\boldsymbol{D}}, \boldsymbol{D} \in \mathbb{R}^{3 \times |\mathcal{E}|}$ are the stacked matrices of Cartesian and fractional distance vectors, respectively, with $\tilde{\boldsymbol{D}} = \boldsymbol{L}_i\boldsymbol{D}$ for structure i. This form reveals that these predicted lattice scores have a key shortcoming. To see this, recall from Appendix A.7.1

25

that we perform lattice diffusion on the subspace of *symmetric* lattice matrices. However, the lattice scores from $\hat{s}^l_{L,\theta}(X_t, L_t, A_t, t)$ are generally not symmetric matrices. We address this issue with the following modification:

$$
\begin{aligned}
s^l_{L,\theta}(X_t, L_t, A_t, t) &= \tilde{s}^l_{L,\theta}(X_t, L_t, A_t, t) L_t^\top \\
&= L_t D \tilde{\Phi}^l D^\top L_t^\top = \tilde{D} \tilde{\Phi}^l \tilde{D}^\top,
\end{aligned} \tag{A35}
$$

where $\tilde{\Phi}^l = \mathrm{diag}\left(\frac{\phi^l(m^l_{ijk})}{|\mathcal{E}| \cdot d^2_{ijk}}\right)$. Finally, the predicted layer-wise lattice scores are summed to obtain the predicted lattice score:

$$
s_{L,\theta}(X_t, L_t, A_t, t) = \sum_{l=1}^{L} s^l_{L,\theta}(X_t, L_t, A_t, t), \tag{A36}
$$

which is scale-invariant and equivariant under rotation. The equivariance derives from the way it is composed with the Cartesian coordinate matrix, and the scale invariance is due to the normalization happening inside $\tilde{\Phi}^l$. In particular, the diagonal entries of $\tilde{\Phi}^l$ related to the edges are normalized three times: they are divided by the total number of edges, and then multiplied twice by the inverse of the norm of the edge vectors. Given these properties, $\hat{s}_\theta$ behaves like a symmetric stress tensor σ, since the stress tensor is scale-invariant and equivariant under the rotation operator R:

$$
\sigma'(\lambda M) = \sigma(M), \tag{A37}
$$
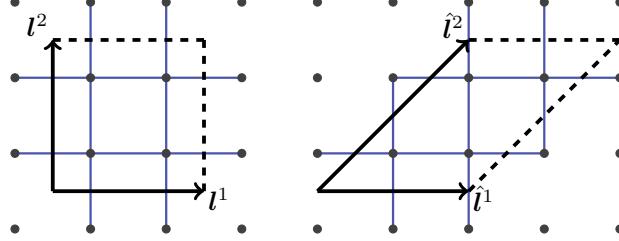
$$
\sigma'(R M) = R \sigma(R M) R^\top, \tag{A38}
$$

where we use λ to indicate the supercell replication operation.

A.8.2 Augmenting the input with lattice information

The chain-rule-based lattice score predictions from Eq. (A36) have shown to lack expressiveness for modeling the score of our Gaussian forward diffusion in our early experiments. We hypothesize that this is because our periodic GNN model is invariant to the particular choice of unit cell. For instance, it cannot distinguish the two structures in Fig. A1. To address this, we drop the invariance of the GNN w.r.t. equivalent choices of the unit cell by injecting information about the lattice angles into the internal representations. This means that the generative distribution is no longer invariant to the concrete choice of unit cell. We nonetheless note that any lattice can be *uniquely* transformed into its so-called Niggli-reduced cell [63]. We apply this transformation to all training data points and, consequently, side-step the loss of cell choice equivariance we introduce. Concretely, we concatenate the roto-translation invariant input edge representations m^{inp} with the cosines of the angles of the edge vectors w.r.t. the lattice cell vectors:

$$
\hat{m}^{\mathrm{inp}}_{ijk} = \left(m^{\mathrm{inp}}_{ijk}, \ \cos(d_{ijk}, l^1), \ \cos(d_{ijk}, l^2), \ \cos(d_{ijk}, l^3) \right). \tag{A39}
$$

Fig. A1: Diagram showing two equivalent lattice choices $\boldsymbol{L}$ and $\hat{\boldsymbol{L}} = \boldsymbol{LC}$ that lead to the same periodic structure. The dots represent atoms in the 2D periodic structures, and the blue lines indicate edges of atoms inside the unit cell to their four nearest neighbors, inside and outside the unit cell. Note that both choices of unit cell lead to indistinguishable structures, as indicated by the identical placement of atoms and equivalent edges. Here, $\boldsymbol{C} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$.

This additional information allows the model to distinguish the two cases in Fig. A1, while the internal representations remain invariant to rotation and translation.

A.9   Training loss

Our model is trained to minimize the following loss, which is a sum of the score matching loss (see Eq. (A4)) for the coordinates and cell, respectively, and the atom type loss (compare with D3PM objective in Eq. (A13)):

$$L = \lambda_{\text{coord}} L_{\text{coord}} + \lambda_{\text{cell}} L_{\text{cell}} + \lambda_{\text{types}} L_{\text{types}}, \tag{A40}$$

where

$$L_{\text{coord}} = \sum_{t=1}^{T} \sigma_t(n)^2 \mathbb{E}_{q(\boldsymbol{x}_0)} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[\|\boldsymbol{s}_{\boldsymbol{x},\boldsymbol{\theta}}(\boldsymbol{X}_t, \boldsymbol{L}_t, \boldsymbol{A}_t, t) - \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2 \right], \tag{A41}$$

$$L_{\text{cell}} = \sum_{t=1}^{T} (1 - \bar{\alpha}_t) \sigma_t(n)^2 \mathbb{E}_{q(\boldsymbol{L}_0)} \mathbb{E}_{q(\boldsymbol{L}_t|\boldsymbol{L}_0)} \left[\|\boldsymbol{s}_{\boldsymbol{L},\boldsymbol{\theta}}(\boldsymbol{X}_t, \boldsymbol{L}_t, \boldsymbol{A}_t, t) - \nabla_{\boldsymbol{L}_t} \log q(\boldsymbol{L}_t|\boldsymbol{L}_0)\|_2^2 \right] \tag{A42}$$

$$L_{\text{types}} = \mathbb{E}_{q(\boldsymbol{a}_0)} \left[\sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{a}_t|\boldsymbol{a}_0)} [D_{\text{KL}} \left[q(\boldsymbol{a}_{t-1}|\boldsymbol{a}_t, \boldsymbol{a}_0) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{a}_{t-1}|\boldsymbol{X}_t, \boldsymbol{L}_t, \boldsymbol{A}_t)] \right.$$

$$- \lambda_{\mathrm{CE}} \log p_{\boldsymbol{\theta}}(\boldsymbol{a}_0|\boldsymbol{X}_t, \boldsymbol{L}_t, \boldsymbol{A}_t, t)] - \mathbb{E}_{q(\boldsymbol{a}_1|\boldsymbol{a}_0)} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{a}_0|\boldsymbol{X}_1, \boldsymbol{L}_1, \boldsymbol{A}_1, 1)] \Bigg].$$

(A43)

For simplicity, Eqs. (A41) and (A43) show the loss only for a single atom's coordinates and specie, respectively; the overall losses for coordinates and atom types sum over all atoms in a structure.

B Fine-tuning the score network for property-guided generation

Here we discuss our fine-tuning procedure of the score model to enable *property-guided* generation via classifier-free guidance [41].

B.1 Fine-tuning the score network with adapter modules

Leveraging the large-scale unlabeled Alex-MP-20 dataset enables MatterGen to generate a broad distribution of stable material structures via reverse diffusion, driven by unconditional scores. To facilitate conditional generation with classifier-free guidance, the property-conditioned scores need to be learned through a labeled dataset. However, labeled datasets, often limited in size and diversity, present challenges in learning the conditional scores from scratch.

To enable rapid learning of the conditional scores in the sparsely-labeled data regime, we propose to fine-tune the unconditional score network with additional trainable adapter modules. Each adapter layer is a combination of an MLP layer and a zero-initialized mix-in layer [40], so MatterGen still outputs the learned unconditional scores at initialization. This is desired because the unconditional scores have been optimized to generate stable materials during pre-training, which is a prerequisite for modeling the property-conditional distribution of materials.

The additional adapter modules consist of an embedding layer f_{embed} for the property label that outputs a property embedding $\boldsymbol{g}$, and a series of adapter layers, one before each message-passing layer (four in total). The adapter layer augments the atom embedding of the original GemNet score network to incorporate property information. Concretely, at the L-th interaction layer, given the property embedding $\boldsymbol{g}$ and the intermediate node hidden representation $\{\boldsymbol{H}_j^{(L)}\}_{j=1}^n$, the property-augmented node hidden representation $\{\boldsymbol{H}_j^{\prime(L)}\}_{j=1}^n$ is given by:

$$\boldsymbol{H}_j^{\prime(L)} = \boldsymbol{H}_j^{(L)} + f_{\text{mixin}}^{(L)}\left(f_{\text{adapter}}^{(L)}\left(\boldsymbol{g}\right)\right) \cdot \mathbb{I}(\text{property is not null}), \qquad \text{(B44)}$$

where $f_{\text{mixin}}^{(L)}$ is the L-th mix-in layer, which is a zero-initialized linear layer without bias weights. $f_{\text{adapter}}^{(L)}$ is the L-th adapter layer, which is a two-layer MLP model. The indicator function $\mathbb{I}(\text{property is not null})$ ensures the model outputs the unconditional score when no conditional label is given. The adapter modules add additional weights of f_{embed}, $f_{\text{adapter}}^{(L)}$, and $f_{\text{mixin}}^{(L)}$ to each layer of the model.

The fine-tuning process uses the same training objective as the unconditional pre-training stage with conditional property labels incorporated. When fine-tuning finishes, the score network is able to predict both conditional and unconditional scores. The fine-tuned model enables us to generate structures satisfying the property condition without a major sacrifice in terms of stability and novelty. With the unconditional score network as a strong initialization, the fine-tuning procedure is more computation- and sample-efficient than re-training from scratch if the labeled dataset is only sparsely labeled.

B.2 Classifier-free guidance

To generate samples conditioned on a specific value c of a property, we adopt classifier-free diffusion guidance [41] throughout this work. In classifier-free guidance, a guidance factor γ is applied to the conditional distribution $p(\boldsymbol{M}_t|c)$, such that

$$
\begin{aligned}
p_\gamma(\boldsymbol{M}_t|c) &\propto p(c|\boldsymbol{M}_t)^\gamma p(\boldsymbol{M}_t) \\
&\propto \left(\frac{p(\boldsymbol{M}_t|c)}{p(\boldsymbol{M}_t)} \right)^\gamma p(\boldsymbol{M}_t) \\
&\propto p(\boldsymbol{M}_t|c)^\gamma p(\boldsymbol{M}_t)^{1-\gamma}
\end{aligned}
\tag{B45}
$$

is used instead of $p(\boldsymbol{M}_t)$ when evaluating the model score during the reverse process in the conditional setting. We adopt a value of $\gamma = 2$ in all conditional generation experiments.

B.2.1 Continuous case

The conditional score follows from Eq. (B45) by taking gradients of the logarithm w.r.t. continuous variables in $\boldsymbol{M}_t$. For example, for fractional coordinates $\boldsymbol{X}_t$ we have

$$
\nabla_{\boldsymbol{X}_t} \ln p_\gamma(\boldsymbol{X}_t|c) = \gamma \nabla_{\boldsymbol{X}_t} \ln q(\boldsymbol{X}_t|c) + (1-\gamma)\nabla_{\boldsymbol{X}_t} \ln q(\boldsymbol{X}_t).
\tag{B46}
$$

Practically, learning a conditional score $\nabla_{\boldsymbol{X}_t} \ln q(\boldsymbol{X}_t|c)$ equates to concatenating a latent embedding $\boldsymbol{g}_c \in \mathbb{R}^d$ of the condition c to the score model $\boldsymbol{s_\theta}(\boldsymbol{M}_t, \boldsymbol{g}_c, t)$ during score matching. The unconditional score $\nabla_{\boldsymbol{X}_t} \ln p(\boldsymbol{X}_t)$ is obtained by providing a null embedding for the condition, i.e., using $\boldsymbol{s_\theta}(\boldsymbol{M}_t, \boldsymbol{g}_c = \text{null}, t)$. When we condition on multiple properties, the conditional score for N properties with embeddings $\boldsymbol{g}_{c_i}$ is obtained by $\boldsymbol{s_\theta}(\boldsymbol{M}_t, \boldsymbol{g}_{c_1}, \boldsymbol{g}_{c_2}, \ldots, \boldsymbol{g}_{c_N}, t)$.

B.2.2 Discrete case

The model's task in denoising discrete atom types $\boldsymbol{a}$ is to fit and predict $\tilde{q}(\boldsymbol{a}_{t-1}|\boldsymbol{a}_t, c)$. Following Eq. (4) in [57], we can rewrite this as

$$
\tilde{q}(\boldsymbol{a}_{t-1}|\boldsymbol{a}_t, c) \propto \sum_{\boldsymbol{a}_0} q(\boldsymbol{a}_{t-1}, \boldsymbol{a}_t|\boldsymbol{a}_0) \cdot \tilde{q}(\boldsymbol{a}_0|\boldsymbol{a}_t, c).
$$

Thus, the predictive task is to approximate $p_\theta(\boldsymbol{a}_0|\boldsymbol{a}_t, c) \approx \tilde{q}(\boldsymbol{a}_0|\boldsymbol{a}_t, c)$. For this distribution we can perform classifier-free guidance as follows:

$$
\begin{aligned}
\tilde{q}_\gamma(\boldsymbol{a}_0|\boldsymbol{a}_t, c) &\propto \tilde{q}(c|\boldsymbol{a}_0, \boldsymbol{a}_t)^\gamma \cdot \tilde{q}(\boldsymbol{a}_0|\boldsymbol{a}_t) \\
&= \left(\frac{\tilde{q}(\boldsymbol{a}_0|c, \boldsymbol{a}_t) \cdot \tilde{q}(c|\boldsymbol{a}_t)}{\tilde{q}(\boldsymbol{a}_0|\boldsymbol{a}_t)} \right)^\gamma \cdot \tilde{q}(\boldsymbol{a}_0|\boldsymbol{a}_t) \\
&\propto \tilde{q}(\boldsymbol{a}_0|c, \boldsymbol{a}_t)^\gamma \cdot \tilde{q}(\boldsymbol{a}_0|\boldsymbol{a}_t)^{1-\gamma}.
\end{aligned}
$$

We can approximate this guided distribution accordingly with an unconditional and a conditional prediction model, i.e., $p_\theta(\boldsymbol{a}_0|c, \boldsymbol{a}_t, t)^\gamma \cdot p_\theta(\boldsymbol{a}_0|\boldsymbol{a}_t, t)^{1-\gamma} \approx \tilde{q}(\boldsymbol{a}_0|c, \boldsymbol{a}_t)^\gamma \cdot \tilde{q}(\boldsymbol{a}_0|\boldsymbol{a}_t)^{1-\gamma}$. Taking the logarithm, we obtain

$$\log\left(p_\theta(\boldsymbol{a}_0|c, \boldsymbol{a}_t, t)^\gamma \cdot p_\theta(\boldsymbol{a}_0|\boldsymbol{a}_t, t)^{1-\gamma}\right) = \gamma \log p_\theta(\boldsymbol{a}_0|c, \boldsymbol{a}_t, t) + (1-\gamma) \log p_\theta(\boldsymbol{a}_0|\boldsymbol{a}_t, t).$$

C   Dataset generation

Here we provide details about the training dataset Alex-MP-20 and the reference dataset Alex-MP-ICSD used throughout this work.

C.1   Data sources

We obtained crystal structures via three sources:
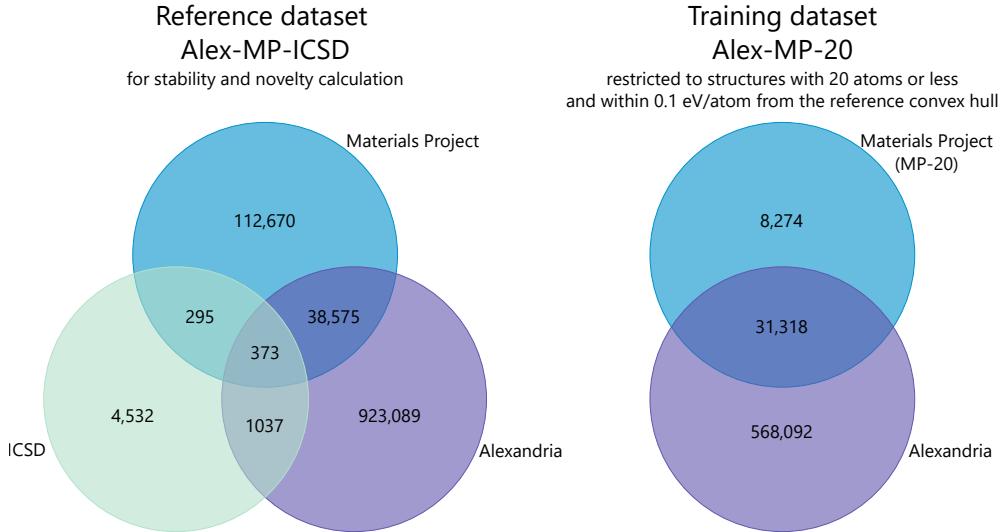
- MP (v2022.10.28, Creative Commons Attribution 4.0 International License) [15], an open-access resource containing DFT-relaxed crystal structures obtained from a variety of sources, but largely based upon experimentally-known crystals.
- The Alexandria dataset [64–66] (Creative Commons Attribution 4.0 International License), an open-access resource containing DFT-relaxed crystal structures from a variety of sources, including a large quantity of hypothetical crystal structures generated by ML methods or other algorithmic means.
- ICSD (release 2023.1) [67], a proprietary database containing crystal structures refined from experiment. For the purposes of dataset generation, we queried ICSD only for experimental crystal structures that were not tagged as already included in MP, and that were directly calculable by DFT (i.e., ordered crystals).

For crystal structures from MP, we retrieved existing calculations via the MP API. For other data sources, we performed new calculations using MP settings to guarantee consistency of data (see Appendix C.2). We then followed MP's data analysis approach as implemented in `emmet` [68], which includes the following steps:

1. Validation of each individual DFT calculation to ensure required minimum quality criteria are met.
2. Grouping of calculations of equivalent crystal structures, which de-duplicates crystal structures when the same crystal is present in multiple data sources. See Fig. C2 for an overview of the resulting statistics.
3. Application of an empirical correction scheme [69] to address known systematic errors from the Perdew–Burke–Ernzerhof (PBE) functional.
4. Construction of convex hull phase diagrams for each chemical system.

This process resulted in a reference dataset of 1,081,850 unique structures with associated energy above hull values calculated using DFT. We refer to this as the Alex-MP-ICSD dataset. This dataset was then used to derive the Alex-MP-20 dataset, whose element distribution is shown in Fig. C3. The Alex-MP-ICSD dataset is used as a reference for the computation of stability (i.e., energy above hull) and uniqueness of generated structures. To train MatterGen, we employ a subset of the Alex-MP-ICSD dataset, selecting only structures with up to 20 atoms and whose energy above hull

31

is below 0.1 eV/atom; we refer to this as the Alex-MP-20 dataset. Furthermore, we manually exclude from the Alex-MP-20 dataset all structures belonging to the "well-explored" chemical systems, as defined in Appendix D.4. Additionally, we reserve structures present only in ICSD for testing purposes, and therefore exclude them from the Alex-MP-20 dataset. We report in Fig. C2 the structure provenance and quantity for the reference (Alex-MP-ICSD) and training (Alex-MP-20) datasets. Finally, the dataset employed to train the MatterGen-MP model contains structures from the MP-20 dataset (containing structures with up to 20 atoms) whose energy above hull is below 0.1 eV/atom from the reference convex hull. This is also highlighted in Fig. C2 (right panel, blue circle).

Reference dataset
Alex-MP-ICSD
for stability and novelty calculation

Training dataset
Alex-MP-20
restricted to structures with 20 atoms or less
and within 0.1 eV/atom from the reference convex hull
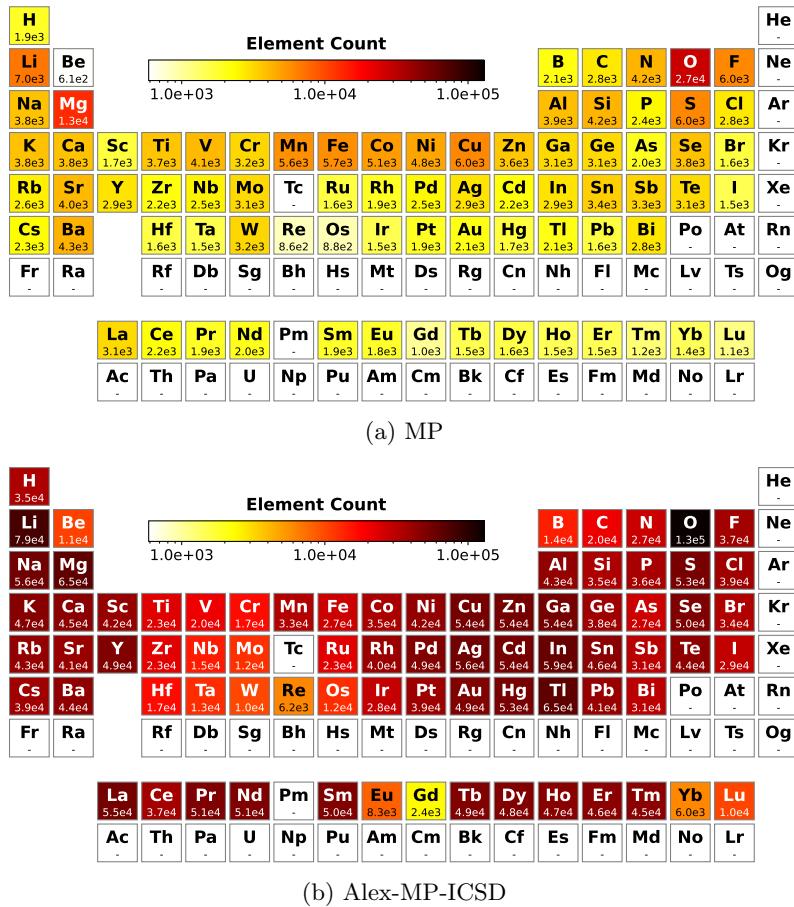
Fig. C2: Venn diagrams (not to scale) showing the overlap of crystal structures from the three data sources used in this work, for the reference Alex-MP-ICSD dataset (left) and the Alex-MP-20 training dataset (right). Crystal structures were de-duplicated after calculation and therefore the overlap in this diagram shows cases where the same crystal structure was present in multiple data sources. Note that the statistics for ICSD include only the structures sourced from the ICSD in this study, and not the full ICSD database.

C.2   DFT details

All DFT calculations were performed using the Vienna ab initio simulation package (VASP) [70] within the projector augmented-wave formalism via `atomate2`, `pymatgen` and `custodian` [71]. All parameters of the calculations were chosen to be consistent with MP [15], including use of the PBE functional [72] and Hubbard U corrections. Specifically, the following workflows were used for the calculation of associated properties:

- Structural relaxations and static calculations for total energy were calculated with the `MPRelaxSet` settings and via the `DoubleRelaxMaker` and `StaticMaker` classes.
- Band gaps were calculated as above, but via the `BandStructureMaker` class.
- Elastic tensors were calculated with the default `ElasticMaker` class with a stencil including -0.01 % and 0.01 % deformations, as specified by `wf_elastic_constant_minimal` in the `atomate` code. The "minimal" preset is used for reasons of computational efficiency.

All settings were previously benchmarked or in use by MP, and every effort was made to ensure consistent settings were used in the current work.

(a) MP — Element Count (1.0e+03, 1.0e+04, 1.0e+05)

Element	Count	Element	Count	Element	Count	Element	Count	Element	Count	Element	Count	Element	Count	Element	Count
H	1.9e3													He	-
Li	7.0e3	Be	6.1e2	B	2.1e3	C	2.8e3	N	4.2e3	O	2.7e4	F	6.0e3	Ne	-
Na	3.8e3	Mg	1.3e4	Al	3.9e3	Si	4.2e3	P	2.4e3	S	6.0e3	Cl	2.8e3	Ar	-

K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr
3.8e3	3.8e3	1.7e3	3.7e3	4.1e3	3.2e3	5.6e3	5.7e3	5.1e3	4.8e3	6.0e3	3.6e3	3.1e3	3.1e3	2.0e3	3.8e3	1.6e3	-

Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe
2.6e3	4.0e3	2.9e3	2.2e3	2.5e3	3.1e3	-	1.6e3	1.9e3	2.5e3	2.9e3	2.2e3	2.9e3	3.4e3	3.3e3	3.1e3	1.5e3	-

Cs	Ba	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn
2.3e3	4.3e3	1.6e3	1.5e3	3.2e3	8.6e2	8.8e2	1.5e3	1.9e3	2.1e3	1.7e3	2.1e3	1.6e3	2.8e3	-	-	-

Fr -, Ra -, Rf -, Db -, Sg -, Bh -, Hs -, Mt -, Ds -, Rg -, Cn -, Nh -, Fl -, Mc -, Lv -, Ts -, Og -

La	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu
3.1e3	2.2e3	1.9e3	2.0e3	-	1.9e3	1.8e3	1.0e3	1.5e3	1.6e3	1.5e3	1.5e3	1.2e3	1.4e3	1.1e3

Ac -, Th -, Pa -, U -, Np -, Pu -, Am -, Cm -, Bk -, Cf -, Es -, Fm -, Md -, No -, Lr -

(a) MP

(b) Alex-MP-ICSD — Element Count (1.0e+03, 1.0e+04, 1.0e+05)

Element	Count	Element	Count	Element	Count	Element	Count	Element	Count	Element	Count	Element	Count	Element	Count
H	3.5e4													He	-
Li	7.9e4	Be	1.1e4	B	1.4e4	C	2.0e4	N	2.7e4	O	1.3e5	F	3.7e4	Ne	-
Na	5.6e4	Mg	6.5e4	Al	4.3e4	Si	3.5e4	P	3.6e4	S	5.3e4	Cl	3.9e4	Ar	-

K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr
4.7e4	4.5e4	4.2e4	2.3e4	2.0e4	1.7e4	3.3e4	2.7e4	3.5e4	4.2e4	5.4e4	5.4e4	5.4e4	3.8e4	2.7e4	5.0e4	3.4e4	-

Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe
4.3e4	4.1e4	4.9e4	2.3e4	1.5e4	1.2e4	-	2.3e4	4.0e4	4.9e4	5.6e4	5.4e4	5.9e4	3.1e4	3.1e4	4.4e4	2.9e4	-

Cs	Ba	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn
3.9e4	4.4e4	1.7e4	1.3e4	1.0e4	6.2e3	1.2e4	2.8e4	3.9e4	4.9e4	5.3e4	6.5e4	4.1e4	3.1e4	-	-	-

Fr -, Ra -, Rf -, Db -, Sg -, Bh -, Hs -, Mt -, Ds -, Rg -, Cn -, Nh -, Fl -, Mc -, Lv -, Ts -, Og -

La	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu
5.5e4	3.7e4	5.1e4	5.1e4	-	5.0e4	8.3e3	2.4e3	4.9e4	4.8e4	4.7e4	4.6e4	4.5e4	6.0e3	1.0e4

Ac -, Th -, Pa -, U -, Np -, Pu -, Am -, Cm -, Bk -, Cf -, Es -, Fm -, Md -, No -, Lr -

(b) Alex-MP-ICSD

Fig. C3: Distribution of elements in MP (top) and the combined Alex-MP-ICSD (bottom) datasets.

D Results

This section contains supplementary information for the results detailed in Section 2.

D.1 Common experimental details

In this section we provide information about settings used across different experiments, including training and sampling details of MatterGen, and additional information about the MLFF we employ. Details specific to certain experiments are deferred to Appendices D.3 to D.7.

D.1.1 Hyperparameters for training base model

The base unconditional model was trained for 1.74 million steps with a batch size of 64 per GPU over eight A100 GPUs using the Adam optimizer [73]. The learning rate was initialized at 0.0001 and was decayed using the ReduceLROnPlateau scheduler with decay factor 0.6, patience 100 and minimum learning rate 10^{-6}. For the training loss we use $\lambda_{\text{coord}} = 0.1$ and $\lambda_{\text{cell}} = \lambda_{\text{types}} = 1$, as well as the recommended value $\lambda_{\text{CE}} = 0.01$ for the D3PM cross-entropy loss.

D.1.2 Hyperparameters for fine-tuning models

For fine-tuning the base model towards different properties, we used a global batch size of 128 and the Adam optimizer. Gradient clipping was applied by value at 0.5. The learning rate was initialized at 6×10^{-5} and we used the same learning rate scheduler as that for the base model. The training was stopped when the validation loss stopped improving for 100 epochs, which resulted in 32,000 - 1.1 million steps depending on the dataset.

D.1.3 MatterGen sampling parameters

For both unconditional and conditional generation, we discretized the reverse diffusion process over the continuous time interval $[0, 1]$ into $T = 1000$ steps. For each time step, we use ancestral sampling (according to Eqs. (A6), (A8) and (A16)) to sample $(\boldsymbol{X}_{t-1}, \boldsymbol{L}_{t-1}, \boldsymbol{A}_{t-1})$ given $(\boldsymbol{X}_t, \boldsymbol{L}_t, \boldsymbol{A}_t)$ using the score model described in Appendix A.8. After each predictor step, one corrector step was applied. We used the Langevin corrector (see Algorithms 4 and 5 in [39]) for the coordinates $\boldsymbol{X}_t$ and the lattice $\boldsymbol{L}_t$ with signal-to-noise ratio parameters 0.4 and 0.2, respectively.

D.1.4 Machine learning force field (M3GNet) details

We used an MLFF trained on 1.08 million crystalline structures which employed the M3GNet [23] architecture with three graph convolution layers and had in total 890,000 parameters. To compute the energy above hull, we used the energy correction scheme compatible with MP (i.e., `MaterialsProject2020Compatibility` from `pymatgen` [71]). Further details on the MLFF will be provided in a separate publication.

D.2 Qualitative analysis of generated structures

Assessing the quality of generated crystals is difficult. In this work, we have used computational metrics to assess the quality of generated materials. However, additional human review of generated materials can be useful to identify failure modes not captured by these metrics. We perform this human analysis of generated crystals in Appendices D.3.2, D.6.2 and D.7.2.

Ultimately, it must at least be possible to create the material in a laboratory setting for any hypothetical material to be of practical use, i.e., the material must be synthesizable. Although it is not practically possible to conclusively determine that a material is synthesizable using theoretical and computational evidence alone, we can use computation as a guide. Throughout this work we use the energy above the convex hull, calculated at 0 K under 0 GPa applied stress, as a signal [74] that a material might be stable at ambient temperature and pressure. We acknowledge that there are additional calculations we could perform (such as computing the phonon spectra) that would improve upon the approximations we make in this work, but robustly assessing synthesizability of a hypothetical crystal structure is still an open research question.

Energy above hull alone is not sufficient, since many metastable materials with finite energy above hulls ("off-hull") are routinely synthesized, while many on-hull materials have not been, despite best efforts. While some attempt [75] has been made to suggest reasonable threshold values for energy above hull, this is very dependent on chemical system: while some materials (such as carbides, nitrides) are able to tolerate very high energies above convex hull (0.1–1 eV), other materials (such as intermetallics) are only able to tolerate a very small degree of metastability (meV). It is also known [76] that traditional methods of simulation using DFT will give inaccurate energies for some materials, even after empirical corrections [69] are applied which can correct some of the better understood systematic errors. Furthermore, the energy above hull is a measure that is only meaningful if the particular chemical system has been well-explored: for unexplored or partially-explored chemical systems, the energy above hull might be inaccurate simply due to more energetically-favourable phases being unknown.

Using empirical priors to assess the synthesizability of structures is traditionally done by hand by domain experts and is therefore highly dependent on the chemical intuition and background knowledge of a particular system by the expert. It becomes more difficult for a scientist to evaluate a crystal structure picked "at random", especially as the number of elements increases, as they are less likely to have encountered this material in their prior work. Some materials are hard to analyze in this way because the prior is simply not known or may be biased; for example, any distribution of a specific property calculated from crystalline materials that have already been synthesized will include bias by not simply including the properties of materials that have not yet been synthesized. These biases can be because certain elements are more abundant, cheaper, or easier to process, or because certain materials have garnered more technological interest, rather than because of an *a priori* physical reason why those materials could not be made. Simply put, we do not yet know what the distribution of "possible" crystal structures looks like, even within certain constraints (e.g., maximum primitive cell size or number of elements). Therefore, assessing synthesizability

35

using empirical priors is difficult, but might still provide insights. When examining a generated material, in addition to performing a literature review, additional factors can be considered, including:

1. Symmetry; crystals are defined by their symmetry, and nature prefers symmetrical crystals unless there is a specific mechanism by which symmetry is broken. While lower symmetry would be expected as the number of elements in a system increases, in general generated crystals are expected to be symmetrical. P1 crystals are rare in nature, and when they are reported in databases this is often either because they have not been refined or even because of mis-identification [77].
2. The presence of defects. Defects could include structural distortions, with a crystal containing the "correct" atoms but in distorted geometry, or could include point defects, such as vacancies or interstitial atoms. The presence of defects is not necessarily bad; for example, many "off-stochiometric" materials such as NiO_x are routinely synthesized which can contain large concentrations of vacancies. Sometimes vacancies might be required so that a structure might charge balance; a classic example might be the bixbyite crystal structure derived from a fluorite structure with an ordered array of structural vacancies. However, the presence of defects could also be a concern, especially in the case of unexplored chemical systems where a material might be erroneously calculated to be "on hull" due to incomplete knowledge of that chemical system.
3. Local atomic environments should be reasonable, meaning that the material contains reasonable bond lengths and co-ordination polyhedra, where other materials with those combinations of elements are known.
4. The material should charge balance if highly ionic, meaning that the sum of the formal valence of its constituent atoms is zero. If a material does not charge balance, and is ionic, it will likely have a very low defect formation energy if synthesizable. However, the importance of charge balance should be taken with caution, since the proportion of new materials that have been discovered that are nominally charge-balanced has decreased over time [78], with only around 40% newly-discovered materials being charge-balanced compared to over 80% of materials discovered a hundred years ago.

The definition of a material used in this work (Section 2) also allows for many potential failure modes including but not limited to: non 3D periodic materials such as 2D materials that contain a vacuum in one dimension, amorphous materials, or other random arrangements of "atoms-in-a-box". Effort can be made to avoid these classes of materials by altering the training data of the generative model. Some efforts have been made to algorithmically categorize crystals [79], which could then be used to this effect, however these tools are not yet well-developed, and some spurious structures in the training set should be expected. Finally, the definition of a material used in this work also explicitly assumes an ordered material, whereas many real materials exist as alloys or solid solutions: in traditional DFT, only fully ordered materials can be calculated (meaning, a material that contains whole atoms, with exactly one atom on a given atomic site). This is also the constraint placed by MatterGen on its generated

crystals. However, real materials are often disordered, with fractional atomic occupancy (on *average*, a given atomic site might contain, say, 50% of one atom, and 50% of another atom). A disordered material has many "ordered approximations"–small unit cells with the correct overall composition–that can represent the parent disordered material. As such, MatterGen might generate several ordered approximations of the same parent disordered material. In these cases, novelty will not be assessed correctly, and properties such as energy above hull might be misleading.

Given these factors, we acknowledge that there are limitations in materials discovery efforts that still require methods advancements to overcome. We restrict our confidence that structures we generate are synthesizable to the level of theory and computation we use in this work, in addition to the finite reference we use for the convex hull. We have attempted human-assisted evaluation of predicted structures using empirical priors to gain additional knowledge for how our model performs at generating synthesizable materials.

D.2.1 Visualization

Manual analysis of crystal structures can be influenced by how they are represented visually, for example the specific bonds that are drawn. In this work, crystal structures are visualized using Crystal Toolkit [80] with the CrystalNN [81] bonding algorithm, since this is known to give good results in most cases. A uniform atomic radius was used since a wide variety of chemical bonding is expected to be present, and no one type of atomic radii (covalent, ionic, etc.) can be assumed. When valences are indicated, these are formal valences assigned using heuristic methods in `pymatgen` [71]. With the exception of Fig. 1, all visualizations are of a $2 \times 2 \times 2$ supercell to ensure at least one full repeat of a crystal and its periodic images are shown. All visualizations are of crystal structures as-calculated, meaning they are not necessarily in their conventional setting, and therefore axes are not labelled.

D.3 Generating stable and diverse materials

This section provides supplementary information to the results in Section 2.2.

D.3.1 RMSD, stability, uniqueness and novelty

To evaluate the performance of a generative model on the task of unconditional generation, we look at two keys metrics. First, we use the RMSD between the generated and the DFT-relaxed structure to measure local stability. Second, we use the fraction of S.U.N. structures to capture global stability and, to some extent, diversity. The RMSD metric is defined as

$$RMSD = \sqrt{\min_{\boldsymbol{P}} \frac{1}{N} \sum_{n}^{N} \left| \tilde{\boldsymbol{x}}_{\boldsymbol{P}(n)}^{gen} - \tilde{\boldsymbol{x}}_{n}^{DFT} \right|^2}, \qquad (D47)$$

where $\tilde{\boldsymbol{x}}_n$ indicates the Cartesian coordinates of atom n, and $\boldsymbol{P}$ is the element-aware permutation operator on the atoms of the generated structure. A lower RMSD indicates that generated structures are closer to their DFT-relaxed counterpart. This in turn reduces the computational time for the DFT relaxation, which is typically the most costly part of crystal structure generation. Novelty and uniqueness are computed in all model evaluations by comparing the atomic arrangement of every pair of structures that have the same reduced formula and space group via the `StructureMatcher` utility from the `pymatgen` Python package [71], with the following default parameters: `ltol=0.2`, `stol=0.3`, `angle_tol=5`. This definition of novelty is not able to determine whether an ordered structure might be an ordered approximation of a disordered structure, and so some structures might be falsely determined to be novel in this scenario.

The plots displayed in Fig. 2(b,c,e,f), and the structures showed in Fig. 2(1) refer to samples of 1024 generated structures which have also been relaxed via DFT.

D.3.2 Additional qualitative analysis of structures

Within the 1024 structures generated unconditionally for Fig. 2(a), a total of 43 unique, on-hull crystal structures were found: 11 binaries, 22 ternaries, and 10 quaternaries. These are summarized in Table D2. Of these, 3 had P1 symmetry, and 3 contained molecules or were molecular crystals. Prototype assignment limited to prototypes available in the `robocrystallographer` [82] tool, which will assign the "closest" matching prototype subject to tolerances. The ability for a composition to charge balance assessed by `pymatgen` and known common oxidation states for each element. As previously discussed, a generated crystal can still be reasonable even if it does not charge balance, and not all materials are ionic.

Four randomly-selected examples were highlighted in Fig. 2 in the main text. These were $BaLa_2Ir$, K_3AlCl_6, $NaNiIO_6$, and $NaSmTm_2Te_4$. For $BaLa_2Ir$, it is well-known that La_2Ir forms an intermetallic with a Laves structure, and that Ba can often substitute for La since both can exist in a $+2$ oxidation state with a 6s2 outer shell. However, in this example, we see octahedrally co-ordinated Ir bonded to La, with Ba inserted as a single plane of atoms in a close-packed configuration, as it would exist in elemental Ba. It is unclear if this material could exist. The K_3AlCl_6 has low space group symmetry ($P\bar{1}$), but consists of Al in an ideal octahedral co-ordination, with K in a mixed bonding environment. This structure charge balances under the assumption of Al^{3+}, K^+ and Cl^-, and could be thought of as derived from a defected rocksalt. The $NaNiIO_6$ structure exists in the training set and is well-known belonging to a family of periodate structures $AMIO_6$ (where A is an alkali metal and M is another metal atom). This material is therefore an example of a material incorrectly classified by our novelty filter; a material might be classified as novel prior to DFT relaxation, and not after relaxation. $NaSmTm_2Te_4$ is a rocksalt structure with Te in the anion site and a mix of Na, Sm and Tm in the cation sites.

D.4 Generating materials with target chemistry

This section provides supplementary information to the results in Section 2.3.

Formula	Symmetry	# Elements	Charge balances	Prototype	Classification
Ba_4Au	C2/m	2	-	-	Crystal with 1D chains
SbAu	$P6_3/mmc$	2	Yes	Molybdenum Carbide MAX Phase	Bulk crystal
ReF_6	$Im\bar{3}m$	2	Yes	Tungsten	Molecular crystal
Tb_2Zn_{17}	$R\bar{3}m$	2	-	-	Bulk crystal
SeS	P1	2	Yes	red selenium	Crystal with 1D chains
V_3Cl_8	C2/m	2	-	-	Layered/2D crystal
KCl_8	P1	2	-	-	Hybrid
AlI_7	P1	2	-	-	Crystal with 1D chains
VBr_5	Cm	2	Yes	Silicon tetrafluoride	Molecular crystal
Li_4Hg	I4/m	2	-	-	Bulk crystal
$DyIn_3$	$Pm\bar{3}m$	2	-	Uranium Silicide	Bulk crystal
HfAlAu	$P\bar{6}2m$	3	-	-	Bulk crystal
Lu_2AgOs	P4/mmm	3	-	Heusler	Bulk crystal
GdScBi	P4/nmm	3	-	Matlockite	Bulk crystal
$Sm(FeC)_2$	Fddd	3	Yes	-	Bulk crystal
$Li(AlPd)_2$	P4/mbm	3	-	-	Bulk crystal
$YbNiSn_2$	Cmcm	3	-	-	Bulk crystal
$Nd(GaPt)_2$	$P4/nmm$	3	-	-	Bulk crystal
LiPrAs	$P\bar{6}m2$	3	-	-	Bulk crystal
$Eu(AgSe)_2$	$P\bar{3}m1$	3	Yes	-	Bulk crystal
Tl_4IrO_6	C2/m	3	Yes	-	Bulk crystal
$CsTe_2Pd$	C2/m	3	Yes	-	Bulk crystal
Al_2PdI_8	C2/m	3	Yes	Indium	Molecular crystal
$CaTbCd_2$	P4/mmm	3	-	Heusler	Bulk crystal
Hf_2ZnMo	P4/mmm	3	-	-	Bulk crystal
Sb_5PPb_2	Amm2	3	Yes	-	Bulk crystal
La_6SbTe_5	C2/m	3	Yes	Caswellsilverite	Bulk crystal
CeTeAs	Pnma	3	Yes	-	Bulk crystal
NaH_3Pd	$Pm\bar{3}m$	3	Yes	(Cubic) Perovskite	Bulk crystal
Er_2ZnNi_2	Immm	3	-	-	Bulk crystal
YBeSi	$P6_3/mmc$	3	-	-	Bulk crystal
$TePb_5Cl_8$	C2/m	3	Yes	-	Bulk crystal
$FeCoH_2$	P4/mmm	3	-	Caswellsilverite	Bulk crystal
Sc_4GaCu_2Rh	$R\bar{3}m$	4	-	Heusler	Bulk crystal
$La_8Os_3PdBr_4$	$R\bar{3}m$	4	-	Caswellsilverite	Bulk crystal
$TbCe(Ho_2Te_3)_2$	Cm	4	-	alpha Po	Bulk crystal
$Ba_8Sb_3PBr_4$	$R\bar{3}m$	4	Yes	alpha Po	Bulk crystal
Cs_2KZnF_6	$Fm\bar{3}m$	4	-	(Cubic) Perovskite	Bulk crystal
Zn_2Ni_6BH	$Fm\bar{3}m$	4	-	(Cubic) Perovskite	Bulk crystal
Cs_2AuIBr_6	I4/mmm	4	Yes	-	Bulk crystal
$LiTb_3(DySe_3)_2$	C2/m	4	-	alpha Po	Bulk crystal
Ba_8Pd_2RhAu	P2/m	4	-	beta Vanadium nitride	Bulk crystal
Te_2MoWSe_2	Cm	4	Yes	Molybdenite	Layered/2D crystal

Table D2: Summary information on 43 crystal structures that were calculated as being thermodynamically stable from a batch of 1024 crystal structures from an unconditional generation task.

D.4.1 Additional experimental details

We explore the capability of MatterGen to find novel stable crystals across the 27 chemical systems listed in Table D3. We group the systems in terms of how many elements they contain (ternary, quaternary, and quinary), and in terms of how many structures near the the convex hull were present in the reference Alex-MP-ICSD dataset ('well explored', 'partially explored', 'not explored'). The latter classes are defined as follows:

- 'well explored': systems with the highest numbers of structures near the convex hull. We removed all structures belonging to 'well explored' systems from the training data set to assess the capability of our model to recover existing stable structures without having seen them during training.

- 'partially explored': systems that lie between the 30th and the 90th percentile of the distribution of chemical systems based on the number of structures they have near the convex hull. This class was designed to assess the capability of our model to expand known convex hulls. Therefore, we did not remove the existing data belonging to such systems from the training set.
- 'not explored': systems with no data near the convex hull. This class was designed to test our model in chemical systems where no structures on the hull are present in the reference dataset.

Here, we define 'near the convex hull' structures as structures whose energy per atom is between 0.0 and 0.1 eV/atom above the convex hull. For all three groups, we randomly chose nine ternary, nine quaternary and nine quinary chemical systems (see Table D3). Moreover, we replaced those chemical systems that had an overlap of more than two elements with another system to promote chemical diversity. The replacement was chosen randomly as well.

For this task, we fine-tune our base model on two properties: chemical system and energy above hull. We encode the latent embedding for the energy above hull and the chemical system as detailed in Appendix B.2.1 and Appendix B.2.2, respectively. Both properties are available for all structures in the training set of the base model. Therefore, the training set is used in full for fine-tuning. At sampling time, we condition on both an energy above the convex hull of 0.0 eV/atom, and on the chemical system we want to sample.

To compare the performance of MatterGen against substitution and RSS, we employ an MLFF (M3GNet, see Appendix D.1.4) to relax the generated structures, and then perform *ab initio* relaxation and static calculations via DFT (see Appendix C.2 for details). In particular, we perform the following steps: (1) generate structures, (2) relax structures using the MLFF, (3) filter structures for uniqueness, (4) select the 100 structures with lowest predicted energy above hull according to the MLFF, (5) run DFT on these structures. We report metrics only with respect to those structures. To allow for a fair comparison between our generative model and non-generative approaches, we employ the MLFF relaxation on a greater number of samples for the latter. For RSS, we sample 600,000 structures per chemical system according to the protocol described in Appendix D.4.1. For substitution, we enumerate every possible structure according to the algorithm detailed in Appendix D.4.1, which yields between 15,000 and 70,000 structures per chemical system. For MatterGen, we generate 10,240 structures per chemical system.

Random structure search details

For every chemical system, we performed two rounds of RSS using the `airss` [45] package. In each round, we generated 300,000 structures by sampling 100,000 structures across three different ranges of number of atoms per unit cell. We used the following non-overlapping intervals: 3-9, 10-15, and 16-20 for ternary systems; 4-10, 11-15, and 16-20 for quaternary systems; 5-11, 12-16, and 17-20 for quinary systems. For the first round, we used `airss` to propose structures without structural relaxation using `MINSEP = 0.7-3` (minimum separation between atoms in Å) and `SYMMOPS = 2-4` (number of symmetry operations). All proposed structures were relaxed using an

	Ternary	Quaternary	Quinary
Well explored	O-Sr-V	Bi-Cu-Pb-S	C-H-N-O-S
	Mn-O-Se	As-Cl-O-Pb	Ba-Ca-Cu-O-Tl
	La-Mo-O	Fe-Na-O-P	Eu-F-K-O-Si
Partially explored	Li-Pr-Te	Cl-Cu-Dy-Rb	Cu-Gd-O-Ru-Sr
	C-Pr-Ru	Na-Te-Tm-Zr	La-Na-O-Sb-Sc
	C-Mg-Sc	F-Mg-Rb-Sn	Cs-F-O-Tl-Zr
Not explored	Br-Pb-Rh	Cr-Ga-Mg-P	Al-C-H-Sb-Zr
	As-Cu-Sr	C-Cl-Ho-Ru	As-Br-Cr-I-Pt
	Cl-Er-In	Al-Au-Co-S	K-Mo-O-P-Sr

Table D3: Categorization of the 27 chemical systems used to benchmark model capabilities on chemical system exploration

MLFF (M3GNet, see Appendix D.1.4), and the resulting 300,000 MLFF relaxation trajectories were used in the second round of RSS to automatically tune the `MINSEP` parameter. Again, `airss` was run without structural relaxation followed by a MLFF relaxation. Finally, we combined the 600,000 MLFF-relaxed structures from both rounds and ran DFT structural relaxation and static calculation on the 100 unique structures with the lowest predicted energy above hull according to the MLFF.

Substitution details

A total of 5,143 ordered crystal structures (2,695 ternary, 1,875 quaternary, and 573 quinary) with less than 100 atoms in a unit cell from the ICSD [83] were used as prototypes. For each chemical system in Table D3, we computed all possible unique substitutions of the prototypes, relaxed all structures using a MLFF (M3GNet, see Appendix D.1.4), and selected the 100 unique structures with the lowest predicted energy above the hull according to the MLFF. Finally, we ran DFT structural relaxation and static calculation on the selected structures.

D.4.2 Additional qualitative analysis of structures

The V-Sr-O chemical system example provided in Fig. 3 produced four new on-hull crystal structures: SrV_2O_6 (V^{5+}), $SrVO_3$ (V^{4+}), $Sr_3V_2O_8$ (V^{5+}) and SrV_2O_4 (V^{4+}). This chemical system has been well-studied in literature, with $SrVO_3$ being a well-known perovskite [84], Sr_2VO_4 expected to crystallize into a K_2NiF-like crystal structure, and $Sr_3(VO_4)_2$ synthesized in a cation-deficient variant of the $SrVO_3$ crystal structure [85].

Vanadates are known to by synthesizable in a variety of frameworks, with the expected co-ordination of the VO_4 sub-unit varying with oxidation state [86] from the ideal tetrahedron in V^{5+} to a variety of other co-ordination environments. All generated structures have plausible atomic environments with VO_4 sub-units, either ideal or distorted, and oxygen co-ordinated Sr atoms.

One SrV_2O_4 structure, having $P\bar{1}$ symmetry, consists of a layers of ideal VO_4 edge-sharing tetrahedra separated by a Sr in a triangular prismatic bonding configuration, resulting in a 1D channel of voids in the Sr layer.

D.5 Designing materials with target symmetry

This section provides supplementary information to the results in Section 2.4.

D.5.1 Additional experimental details

For generating structures belonging to a target space group, we fine-tune our base model on the whole training set, and represent the latent embedding of the space group of a crystal via one-hot encoding of the space group.

We assess the capability of our model to correctly generate structures belonging to any space group via two tasks. For the first task (Fig. 4), we sample two space groups for each of the seven lattice systems, and choose to sample only from space groups that contain at least 1000 structures in the training set. We then compute the fraction of S.U.N. structures our fine-tuned model generates when conditioned on these space groups that are classified as belonging to that space group according to the `SpaceGroupAnalyzer` module of `pymatgen` [71, 87]. This metric is computed for 256 generated structures per space group after DFT relaxation has been performed. For the second task (Fig. D4), we generate 10,000 structures conditioned on space groups sampled randomly from the data distribution of the training set, and check whether our model is able to reproduce the distribution of space groups from the training data. For both of the above tasks, the number of atoms in the systems are sampled from the distribution of number of atoms for that space group in the training set. This way, we avoid 'impossible' space group constraints, where the space group we condition on cannot be satisfied given the number of atoms we set.

D.5.2 Additional Analysis of Structures
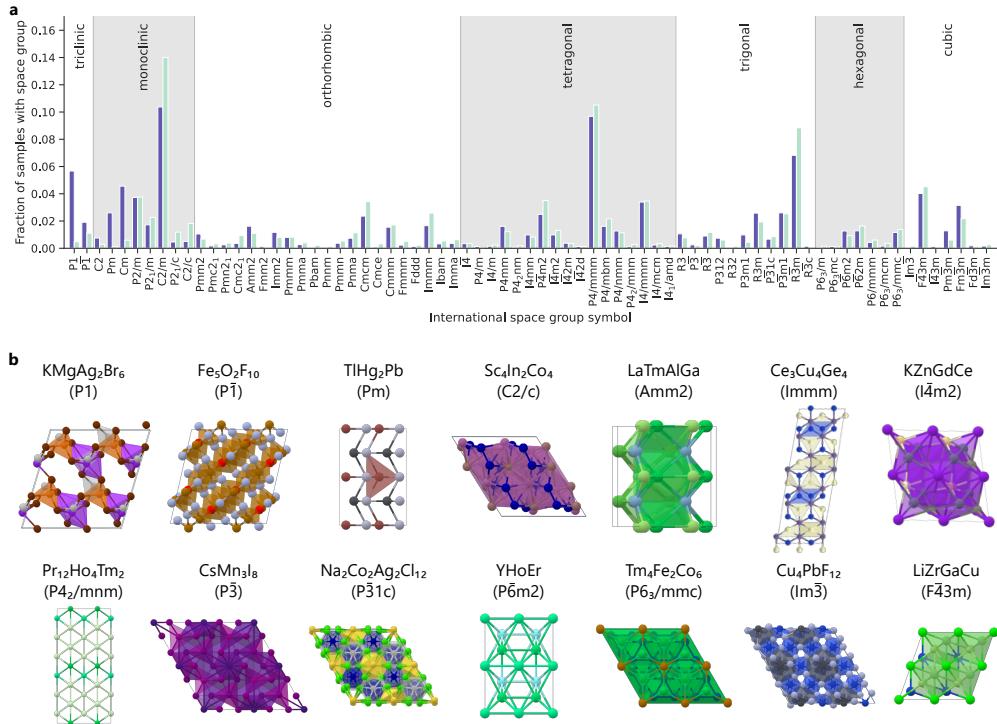
The highlighted examples in Fig. 4 show a high degree of novelty, with few matching known prototypes. The combination of elements in generated structures is also not common, e.g., DyScNiPd, CeAsRh. It is therefore difficult to reference to known structures in literature. Half of the examples could be assigned formal valences, and the majority were robust for their target symmetry even when symmetry was calculated with a higher symmetry-finding tolerance.

D.6 Designing materials with target magnetic, electronic and mechanical properties

This section provides supplementary information to the results in Section 2.5.

D.6.1 Additional experimental details

To generate structures conditioned on a target property, we fine-tune our base model on magnetic density ($N \approx 605,000$ DFT labels), band gap ($N \approx 42,000$) and bulk modulus ($N \approx 5,000$), respectively. See Appendix B.1 for more details on the fine-tuning scheme, and Appendix D.1.2 for hyperparameter settings. We represent the latent embedding of scalar properties via a sinusoidal encoding from the Transformer architecture [88].

Fig. D4: **Generating materials with target symmetry.** (a) Fraction of generated structures belonging to space groups generated by MatterGen fine-tuned on symmetry groups, sampled unconditionally (violet), and structures present in the reference dataset (green). (b) Fourteen generated S.U.N. structures, one for each space group reported in Fig. 4; composition and space group are reported.

For each property in Fig. 5(a-c), we generate 512 samples with our fine-tuned model by conditioning on a value of 0.2 Å^{-3} for magnetic density, 3.0 eV for band gap, and 400 GPa for bulk modulus. We relax those structures using the MLFF and filter the relaxed structures by stability and uniqueness. We then relax the remaining structures with DFT and filter by stability, uniqueness, and novelty. Finally, we compute the desired property of the remaining structures using DFT, and filter out structures where we consider the computed property to be an outlier, i.e., high bulk modulus values with more than 600 GPa and magnetic density values with more than 0.3 Å^{-3}. For more details on the DFT calculations, see Appendix C.2. After all filters and DFT computations, we obtain 251 S.U.N. structures for magnetic density, 142 for band gap, and 22 for bulk modulus.

For Fig. 5(g), we use MatterGen to generate 15,360 samples conditioned on a magnetic density value of 0.2 Å^{-3}, relax those structures using the MLFF and filter the relaxed structures by stability and uniqueness. This results in 5,365 candidate structures. Next, we randomly sub-sample 600 structures and relax them via DFT, with

43

540 of them being DFT stable. We then randomly sub-sample k structures according to the given DFT property calculation budget, and compute their magnetic density via DFT. As a baseline, we count the number of structures in the labeled training dataset that satisfy the target property constraint. For Fig. 5(h), we train a separate property predictor for bulk modulus (see details below), which we use for both MatterGen and the screening baseline. In particular, we use predicted bulk modulus values to fine-tune the base MatterGen model. We generate 8,192 samples conditioned on a bulk modulus value of 400 GPa, relax those structures using the MLFF and filter the relaxed structures by stability and uniqueness. This results in 801 candidate structures. We relax those via DFT, with 736 of them being DFT stable. We then randomly sub-sample k structures according to the given DFT property calculation budget, and compute their bulk modulus via DFT. For the screening baseline, we use the bulk modulus property predictor to predict the bulk modulus values of all structures in the training dataset for which we do not have an existing DFT label, rank those structures by their predicted bulk modulus values, and choose the top k structures according the DFT property calculation budget. We then verify their actual bulk modulus values via DFT.

Property predictor details

The bulk modulus property predictor used in Fig. 5(h) consists of a GemNet-dT [62] encoder that provides atom and edge embeddings, followed by a mean readout layer. We employ three message-passing layers, a cutoff radius of 10 Å for the neighbor list construction, and set the dimension of nodes and edges hidden representations to 128.

To train the property predictor, we use all materials with DFT Voigt-Reuss-Hill average bulk modulus values from MP [15] (including structures with more than 20 atoms), which are 7,108 structures in total. We allocate 80% of the data for the training set, 10% for validation, and 10% for testing. We follow the MatBench benchmark [89] and predict the $\log_{10}$ bulk modulus. At the end of training, the model achieves a mean absolute error (MAE) of 9.5 GPa. The model was trained using the Adam optimizer. Gradient clipping was applied by value at 0.5. The learning rate was initialized at 5×10^{-4} and decayed using the ReduceLROnPlateau scheduler with decay factor 0.8, patience 10 and minimum learning rate 10^{-8}. The training was stopped when the validation loss stopped improving for 150 epochs.

D.6.2 Additional qualitative analysis of structures

Magnetic density conditional generation

For the high magnetic density generation task, a manual review of ten generated crystals chosen at random was performed, as well as the two representative structures with high magnetization density that were highlighted in Fig. 5. Of the random selection, eight of the ten generated structures were ordered approximations of a Fe-Co alloy: α-Fe with a partial Co substitution from 10% Co to 40% Co. The Fe-Co system is a well-known and versatile soft magnetic material system with a wide region where a α-Fe$_x$Co$_{1-x}$ phase is stable, with a mixture of γ-Fe and α-Co expected at lower Co content. While these generated structures could be considered "good", in the sense that they are physically plausible and indeed would be useful magnetic materials, they

are likely only considered novel since these specific ordered unit cells do not currently exist in the reference databases. Also, as an alloy, predicted quantities such as energy above hull based on a single ordered approximation will be misleading. Two of the ten generated structures contained hydrogen, either as Fe_9H_2 or Fe_4Co_2H, with hydrogen in an ideal octahedral environment. The generation of these structures can be rationalized owing to extensive study of iron hydrides, with the generated structures having local atomic environments similar to that in a double hexagonal close-packed iron hydride structure that can exist at high pressure. The realization of these specific structures is unlikely with phase segregation expected to occur. The two structures highlighted in Fig. 5(d) contain Gd, an element with a large magnetic moment due to its 7 unpaired electrons when in its Gd^{3+} oxidation state and in its elemental state. Therefore, it is unsurprising that the model would preferentially generate materials containing Gd in this instance: as in the randomly selected sample, this task necessarily shows a strong compositional bias in the materials generated. The Gd_2N example is a layered material. Although the Gd_2N molecule has been studied [90], it is unknown if and how it might crystallize. The generated $Gd_6H_2CN_3$ structure is rocksalt-derived, with Gd on one site, and a mix of C, H and N on another site, with all atoms in almost ideal octahedral environments.

Band gap conditional generation

For the target band gap generation task, a manual review of ten generated crystals chosen at random was performed, as well as the two representative structures with desired target band gap that were highlighted in Fig. 5. Unlike the other single property optimization tasks, the generated crystals did not show a strong compositional dependence, with a wide range of elements presented in generated materials. All generated materials also could nominally charge balance, unlike the materials analyzed from other single property optimization tasks. This seems reasonable for a task designed to generate insulating systems; the target band gap of 3 eV (calculated with the PBE functional, and thus underestimating the true electronic band gap) should generate insulating materials, and thus more ionic solids. Of the random sample, we could only find $NaNO_3$ and the molecular crystal BI_3 as a compositions that had previously been synthesized. Both were in incorrect symmetry compared to their experimentally known structures, with the generated $NaNO_3$ in C2/c compared to the experimental R$\bar{3}$c, and BI_3 in P1 compared to the experimental $P6_3$/m. However, both had correct local bonding and similar calculated band gaps to the band gaps calculated for their experimental structures. This indicates one instance whereby the model might still generate useful results even if the generated crystal structure is different from the experimental ground state, if it can guide a scientist towards investigation of a specific system.

The examples highlighted in Fig. 5(e) were $VBiO_4$ and $TlNO_3$. Of these, the local bonding in $VBiO_4$ was very similar to that in the experimentally-known bismuth vanadate which crystallizes in the $I4_1$/amd [91] space group or $I4_1$/a space group [92], unlike the generated $P2_1$/m space group, and so is another example of reasonable local environment but with a seemingly incorrect space group. Nevertheless, this crystal structure was calculated to be thermodynamically stable with respect to these experimentally-known crystal structures; this could be a limitation with respect to the

DFT methods used, or a sign of under-convergence for several very similar-in-energy polymorphs. As such, it should not be seen as a failure of the model. Likewise, $TlNO_3$ thallous nitrate is experimentally-known in the Pnma space group [93] with a calculated (PBE) band gap of 2.8 eV [94], however the generated crystal appears quite different to the experimental structure due to a change in Tl co-ordination around NO_3^- anions.

Bulk modulus conditional generation

For the high bulk modulus generation task, a manual review of ten generated crystals chosen at random was performed, as well as the two representative structures with high bulk modulus that were highlighted in Fig. 5(f). All structures show a strong compositional bias, containing a mix of refactory elements Re, W, Mo and Ir and frequently also B and C; this is consistent with literature on superhard materials (which includes materials with high bulk modulus). When the composition contains only refactory elements, the generated structure typically seems like an ordered approximation of an alloy of that composition, while those that also contain B or C typically take a very anisotropic, layered structure. As an example, Re_3Ir highlighted in Fig. 5 can be interpreted as an ordered approximation of a Ir_xRe_{1-x} alloy, which has been previously synthesized and is known to exist in solid solution [95]. The Re_3B_2C example is more unusual, with layers of all Re, B, Re, C, Re, ..., and nominally can charge balance. We note that the bulk modulus calculated is an averaged quantity over the full elastic tensor, and we have not examined the directional bulk moduli in these highly anisotropic systems.

D.7 Designing low-supply-chain risk magnets

This section provides supplementary information to the results in Section 2.6.

D.7.1 Additional experimental details

To generate structures conditioned on magnetic density and HHI score, we fine-tune our base model on these two properties, encoded as in Appendix D.6. To evaluate the performance of our model, we proceed as detailed in Appendix D.6, and generate 512 samples with our fine-tuned model, by conditioning on a magnetic density value of 0.2 Å^{-3} and an HHI score of 1200. Of those, 130 samples remain after filtering by stability and uniqueness following the DFT relaxation. Finally, a total of 112 structures pass the novelty check w.r.t. the reference dataset and are reported in Fig. 6(a).

D.7.2 Additional qualitative analysis of structures

Targeting a low HHI index in addition to high magnetic density steers MatterGen away from generating structures with Co, which is associated with poor HHI scores. Example structures include $Fe_x Mn_{1-x}O$ rocksalt alloys ($MnFe_3O_4$, $MnFe_8O_9$); however, they only exhibit a high magnetization density in a hypothetical ferromagnetic state, and not in their actual antiferromagnetic ground state. Other similar example outputs include a defected FeO containing vacancies (Fe_8O_9), and a body-centered-cubic Fe_8Au system, both of which are well-known experimentally. While the joint optimization task

could further be extended to produce more reasonable candidates—e.g., by penalizing expensive elements, and prefering metallic systems more likely to be ferromagnetic—the overall performance of the model with respect to the labels used for training is reasonable. It is possible that a better treatment of alloy systems would be required to improve performance of the high magnetic density generation task, and to ensure that the generated structures are truly novel.

References

[1] Zhao, Q., Stalin, S., Zhao, C.-Z., Archer, L.A.: Designing solid-state electrolytes for safe, energy-dense batteries. Nature Reviews Materials **5**(3), 229–252 (2020)

[2] Zhao, Z.-J., Liu, S., Zha, S., Cheng, D., Studt, F., Henkelman, G., Gong, J.: Theory-guided design of catalytic materials using scaling relationships and reactivity descriptors. Nature Reviews Materials **4**(12), 792–804 (2019)

[3] Osman, A.I., Hefny, M., Abdel Maksoud, M., Elgarahy, A.M., Rooney, D.W.: Recent advances in carbon capture storage and utilisation technologies: a review. Environmental Chemistry Letters **19**(2), 797–849 (2021)

[4] Xie, T., Fu, X., Ganea, O.-E., Barzilay, R., Jaakkola, T.S.: Crystal diffusion variational autoencoder for periodic material generation. In: International Conference on Learning Representations (2022)

[5] Zhao, Y., Siriwardane, E.M.D., Wu, Z., Fu, N., Al-Fahdi, M., Hu, M., Hu, J.: Physics guided deep learning for generative design of crystal materials with symmetry constraints. npj Computational Materials **9**(1), 38 (2023)

[6] Kim, S., Noh, J., Gu, G.H., Aspuru-Guzik, A., Jung, Y.: Generative adversarial networks for crystal structure prediction. ACS central science **6**(8), 1412–1420 (2020)

[7] Long, T., Fortunato, N.M., Opahle, I., Zhang, Y., Samathrakis, I., Shen, C., Gutfleisch, O., Zhang, H.: Constrained crystals deep convolutional generative adversarial network for the inverse design of crystal structures. npj Computational Materials **7**(1), 66 (2021)

[8] Zheng, S., He, J., Liu, C., Shi, Y., Lu, Z., Feng, W., Ju, F., Wang, J., Zhu, J., Min, Y., et al.: Towards predicting equilibrium distributions for molecular systems with deep learning. arXiv preprint arXiv:2306.05445 (2023)

[9] Yang, M., Cho, K., Merchant, A., Abbeel, P., Schuurmans, D., Mordatch, I., Cubuk, E.D.: Scalable diffusion for materials generation. arXiv preprint arXiv:2311.09235 (2023)

[10] Noh, J., Kim, J., Stein, H.S., Sanchez-Lengeling, B., Gregoire, J.M., Aspuru-Guzik, A., Jung, Y.: Inverse design of solid-state materials via a continuous representation. Matter **1**(5), 1370–1384 (2019)

[11] Court, C.J., Yildirim, B., Jain, A., Cole, J.M.: 3-d inorganic crystal structure generation and property prediction via representation learning. Journal of Chemical Information and Modeling **60**(10), 4518–4535 (2020)

[12] Antunes, L.M., Butler, K.T., Grau-Crespo, R.: Crystal structure generation with

autoregressive large language modeling. arXiv preprint arXiv:2307.04340 (2023)

[13] Mila AI4Science, Hernandez-Garcia, A., Duval, A., Volokhova, A., Bengio, Y., Sharma, D., Carrier, P.L., Koziarski, M., Schmidt, V.: Crystal-GFN: sampling crystals with desirable properties and constraints. arXiv preprint arXiv:2310.04925 (2023)

[14] Curtarolo, S., Hart, G.L., Nardelli, M.B., Mingo, N., Sanvito, S., Levy, O.: The high-throughput highway to computational materials design. Nature materials **12**(3), 191–201 (2013)

[15] 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.: Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. APL materials **1**(1), 011002 (2013)

[16] Curtarolo, S., Setyawan, W., Hart, G.L., Jahnatek, M., Chepulskii, R.V., Taylor, R.H., Wang, S., Xue, J., Yang, K., Levy, O., *et al.*: Aflow: An automatic framework for high-throughput materials discovery. Computational Materials Science **58**, 218–226 (2012)

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

[18] Choudhary, K., Garrity, K.F., Reid, A.C., DeCost, B., Biacchi, A.J., Hight Walker, A.R., Trautt, Z., Hattrick-Simpers, J., Kusne, A.G., Centrone, A., *et al.*: The joint automated repository for various integrated simulations (jarvis) for data-driven materials design. npj computational materials **6**(1), 173 (2020)

[19] Talirz, L., Kumbhar, S., Passaro, E., Yakutovich, A.V., Granata, V., Gargiulo, F., Borelli, M., Uhrin, M., Huber, S.P., Zoupanos, S., *et al.*: Materials cloud, a platform for open computational science. Scientific data **7**(1), 299 (2020)

[20] Xie, T., Grossman, J.C.: Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. Physical review letters **120**(14), 145301 (2018)

[21] Chen, C., Ye, W., Zuo, Y., Zheng, C., Ong, S.P.: Graph networks as a universal machine learning framework for molecules and crystals. Chemistry of Materials **31**(9), 3564–3572 (2019)

[22] Unke, O.T., Chmiela, S., Sauceda, H.E., Gastegger, M., Poltavsky, I., Schütt, K.T., Tkatchenko, A., Müller, K.-R.: Machine learning force fields. Chemical Reviews **121**(16), 10142–10186 (2021)

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

[24] Jun, K., Sun, Y., Xiao, Y., Zeng, Y., Kim, R., Kim, H., Miara, L.J., Im, D., Wang, Y., Ceder, G.: Lithium superionic conductors with corner-sharing frameworks. Nature materials **21**(8), 924–931 (2022)

[25] Rosen, A.S., Fung, V., Huck, P., O'Donnell, C.T., Horton, M.K., Truhlar, D.G., Persson, K.A., Notestein, J.M., Snurr, R.Q.: High-throughput predictions of metal–organic framework electronic properties: theoretical challenges, graph neural networks, and data exploration. npj Computational Materials **8**(1), 112 (2022)

[26] Zhong, M., Tran, K., Min, Y., Wang, C., Wang, Z., Dinh, C.-T., De Luna, P., Yu, Z., Rasouli, A.S., Brodersen, P., *et al.*: Accelerated discovery of CO2 electrocatalysts using active machine learning. Nature **581**(7807), 178–183 (2020)

[27] Merchant, A., Batzner, S., Schoenholz, S.S., Aykol, M., Cheon, G., Cubuk, E.D.: Scaling deep learning for materials discovery. Nature (2023)

[28] Shen, J., Griesemer, S.D., Gopakumar, A., Baldassarri, B., Saal, J.E., Aykol, M., Hegde, V.I., Wolverton, C.: Reflections on one million compounds in the open quantum materials database (OQMD). Journal of Physics: Materials **5**(3), 031001 (2022)

[29] Schmidt, J., Hoffmann, N., Wang, H.-C., Borlido, P., Carriço, P.J., Cerqueira, T.F., Botti, S., Marques, M.A.: Large-scale machine-learning-assisted exploration of the whole materials space. arXiv preprint arXiv:2210.00579 (2022)

[30] Davies, D.W., Butler, K.T., Jackson, A.J., Morris, A., Frost, J.M., Skelton, J.M., Walsh, A.: Computational screening of all stoichiometric inorganic materials. Chem **1**(4), 617–627 (2016)

[31] Zunger, A.: Inverse design in search of materials with target functionalities. Nature Reviews Chemistry **2**(4), 0121 (2018)

[32] Sanchez-Lengeling, B., Aspuru-Guzik, A.: Inverse molecular design using machine learning: Generative models for matter engineering. Science **361**(6400), 360–365 (2018)

[33] Schmidt, J., Marques, M.R., Botti, S., Marques, M.A.: Recent advances and applications of machine learning in solid-state materials science. npj Computational Materials **5**(1), 83 (2019)

[34] Weiss, T., Mayo Yanes, E., Chakraborty, S., Cosmo, L., Bronstein, A.M., Gershoni-Poranne, R.: Guided diffusion for inverse molecular design. Nature Computational Science, 1–10 (2023)

[35] Nouira, A., Sokolovska, N., Crivello, J.-C.: CrystalGAN: learning to discover crystallographic structures with generative adversarial networks. arXiv preprint arXiv:1810.11203 (2018)

[36] Ren, Z., Tian, S.I.P., Noh, J., Oviedo, F., Xing, G., Li, J., Liang, Q., Zhu, R., Aberle, A.G., Sun, S., *et al.*: An invertible crystallographic representation for general inverse design of inorganic crystals with targeted properties. Matter **5**(1), 314–335 (2022)

[37] Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. Advances in Neural Information Processing Systems **32** (2019)

[38] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)

[39] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: International Conference on Learning Representations (2021)

[40] Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3836–3847 (2023)

[41] Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)

[42] Schmidt, J., Wang, H.-C., Cerqueira, T.F., Botti, S., Marques, M.A.: A dataset of 175k stable and metastable materials calculated with the PBEsol and SCAN functionals. Scientific Data **9**(1), 64 (2022)

[43] Gebauer, N., Gastegger, M., Schütt, K.: Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. Advances in Neural Information Processing Systems **32** (2019)

[44] Oganov, A.R., Pickard, C.J., Zhu, Q., Needs, R.J.: Structure prediction drives materials discovery. Nature Reviews Materials **4**(5), 331–348 (2019)

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

[46] Zhu, B., Lu, Z., Pickard, C.J., Scanlon, D.O.: Accelerating cathode material discovery through ab initio random structure searching. APL Materials **9**(12) (2021)

[47] Tang, F., Po, H.C., Vishwanath, A., Wan, X.: Comprehensive search for topological materials using symmetry indicators. Nature **566**(7745), 486–489 (2019)

51

[48] Smidt, T.E., Mack, S.A., Reyes-Lillo, S.E., Jain, A., Neaton, J.B.: An automatically curated first-principles database of ferroelectrics. Scientific data **7**(1), 72 (2020)

[49] Jiao, R., Huang, W., Lin, P., Han, J., Chen, P., Lu, Y., Liu, Y.: Crystal structure prediction by joint equivariant diffusion. In: Thirty-seventh Conference on Neural Information Processing Systems (2023). https://openreview.net/forum?id=DNdN26m2Jk

[50] Cui, J., Kramer, M., Zhou, L., Liu, F., Gabay, A., Hadjipanayis, G., Balasubramanian, B., Sellmyer, D.: Current progress and future challenges in rare-earth-free permanent magnets. Acta Materialia **158**, 118–137 (2018)

[51] Gaultois, M.W., Sparks, T.D., Borg, C.K., Seshadri, R., Bonificio, W.D., Clarke, D.R.: Data-driven review of thermoelectric materials: performance and resource considerations. Chemistry of Materials **25**(15), 2911–2920 (2013)

[52] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with CLIP latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)

[53] Watson, J.L., Juergens, D., Bennett, N.R., Trippe, B.L., Yim, J., Eisenach, H.E., Ahern, W., Borst, A.J., Ragotte, R.J., Milles, L.F., *et al.*: De novo design of protein structure and function with RFdiffusion. Nature **620**(7976), 1089–1100 (2023)

[54] Guo, W., Zhang, K., Liang, Z., Zou, R., Xu, Q.: Electrochemical nitrogen fixation and utilization: theories, advanced catalyst materials and system design. Chemical Society Reviews **48**(24), 5658–5716 (2019)

[55] Sumida, K., Rogow, D.L., Mason, J.A., McDonald, T.M., Bloch, E.D., Herm, Z.R., Bae, T.-H., Long, J.R.: Carbon dioxide capture in metal–organic frameworks. Chemical reviews **112**(2), 724–781 (2012)

[56] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning, pp. 2256–2265 (2015). PMLR

[57] Austin, J., Johnson, D.D., Ho, J., Tarlow, D., Berg, R.: Structured denoising diffusion models in discrete state-spaces. Advances in Neural Information Processing Systems **34**, 17981–17993 (2021)

[58] Song, Y., Ermon, S.: Improved techniques for training score-based generative models. Advances in Neural Information Processing Systems **33**, 12438–12448 (2020)

[59] Jing, B., Corso, G., Chang, J., Barzilay, R., Jaakkola, T.: Torsional diffusion

for molecular conformer generation. Advances in Neural Information Processing Systems **35**, 24240–24253 (2022)

[60] Hoogeboom, E., Satorras, V.G., Vignac, C., Welling, M.: Equivariant diffusion for molecule generation in 3d. In: International Conference on Machine Learning, pp. 8867–8887 (2022). PMLR

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

[62] Gasteiger, J., Becker, F., Günnemann, S.: Gemnet: Universal directional graph neural networks for molecules. Advances in Neural Information Processing Systems **34**, 6790–6802 (2021)

[63] Niggli, P., Wien, W.: Kristallographische und Strukturtheoretische Grundbegriffe - 7/1. Handbuch der Experimentalphysik, pp. 108–176. Akademische Verlagsgesellschaft, Leipzig (1928)

[64] Schmidt, Jonathan, Hoffmann, Noah, Wang, Hai-Chen, Borlido, Pedro, M.A. Carriço, Pedro J., F. T. Cerqueira, Tiago, Botti, Silvana, L. Marques, Miguel A.: Large-scale machine-learning-assisted exploration of the whole materials space. Materials Cloud (2022)

[65] Schmidt, J., Pettersson, L., Verdozzi, C., Botti, S., Marques, M.A.L.: Crystal graph attention networks for the prediction of stable materials. Science Advances **7**(49) (2021)

[66] Schmidt, J., Hoffmann, N., Wang, H., Borlido, P., Carriço, P.J.M.A., Cerqueira, T.F.T., Botti, S., Marques, M.A.L.: Machine-learning-assisted determination of the global zero-temperature phase diagram of materials. Advanced Materials **35**(22) (2023)

[67] Zagorac, D., Müller, H., Ruehl, S., Zagorac, J., Rehme, S.: Recent developments in the inorganic crystal structure database: theoretical crystal structure data and related features. Journal of Applied Crystallography **52**(5), 918–925 (2019)

[68] Materials Project: GitHub - materialsproject/emmet. https://github.com/materialsproject/emmet

[69] Wang, A., Kingsbury, R., McDermott, M., Horton, M., Jain, A., Ong, S.P., Dwaraknath, S., Persson, K.A.: A framework for quantifying uncertainty in DFT energy corrections. Scientific reports **11**(1), 15496 (2021)

[70] Kresse, G., Furthmüller, J.: Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. Physical review B **54**(16), 11169 (1996)

[71] Ong, S.P., Richards, W.D., Jain, A., Hautier, G., Kocher, M., Cholia, S., Gunter,

D., Chevrier, V.L., Persson, K.A., Ceder, G.: Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. Computational Materials Science **68**, 314–319 (2013)

[72] Perdew, J.P., Burke, K., Ernzerhof, M.: Generalized gradient approximation made simple. Physical review letters **77**(18), 3865 (1996)

[73] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

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

[75] Aykol, M., Dwaraknath, S.S., Sun, W., Persson, K.A.: Thermodynamic limit for synthesis of metastable inorganic materials. Science advances **4**(4), 0148 (2018)

[76] Isaacs, E.B., Wolverton, C.: Performance of the strongly constrained and appropriately normed density functional for solid-state materials. Physical Review Materials **2**(6), 063801 (2018)

[77] Marsh, R.E.: P1 or P$\bar{1}$? Or something else? Acta Crystallographica Section B: Structural Science **55**(6), 931–936 (1999)

[78] Antoniuk, E.R., Cheon, G., Wang, G., Bernstein, D., Cai, W., Reed, E.J.: Predicting the synthesizability of crystalline inorganic materials from the data of known material compositions. npj Computational Materials **9**(1), 155 (2023)

[79] Himanen, L., Rinke, P., Foster, A.S.: Materials structure genealogy and high-throughput topological classification of surfaces and 2D materials. npj Computational Materials **4**(1), 52 (2018)

[80] Horton, M., Shen, J.-X., Burns, J., Cohen, O., Chabbey, F., Ganose, A.M., Guha, R., Huck, P., Li, H.H., McDermott, M., Montoya, J., Moore, G., Munro, J., O'Donnell, C., Ophus, C., Petretto, G., Riebesell, J., Wetizner, S., Wander, B., Winston, D., Yang, R., Zeltmann, S., Jain, A., Persson, K.A.: Crystal Toolkit: A Web App Framework to Improve Usability and Accessibility of Materials Science Research Algorithms (2023)

[81] Pan, H., Ganose, A.M., Horton, M., Aykol, M., Persson, K.A., Zimmermann, N.E., Jain, A.: Benchmarking coordination number prediction algorithms on inorganic crystal structures. Inorganic chemistry **60**(3), 1590–1603 (2021)

[82] Ganose, A.M., Jain, A.: Robocrystallographer: automated crystal structure text descriptions and analysis. MRS Communications **9**(3), 874–881 (2019)

[83] Zagorac, D., Müller, H., Ruehl, S., Zagorac, J., Rehme, S.: Recent developments

in the inorganic crystal structure database: theoretical crystal structure data and related features. Journal of applied crystallography **52**(5), 918–925 (2019)

[84] Nekrasov, I.A., Keller, G., Kondakov, D., Kozhevnikov, A., Pruschke, T., Held, K., Vollhardt, D., Anisimov, V.: Comparative study of correlation effects in CaVO3 and SrVO3. Physical Review B **72**(15), 155106 (2005)

[85] Pati, B., Choudhary, R., Das, P.R.: Phase transition and electrical properties of strontium orthovanadate. Journal of alloys and compounds **579**, 218–226 (2013)

[86] Zavalij, P.Y., Whittingham, M.S.: Structural chemistry of vanadium oxides with open frameworks. Acta Crystallographica Section B: Structural Science **55**(5), 627–663 (1999)

[87] Grosse-Kunstleve, R., Adams, P.: Algorithms for deriving crystallographic space-group information. II. Treatment of special positions. Acta crystallographica. Section A, Foundations of crystallography **58**, 60–5 (2002) https://doi.org/10.1107/S0108767301016658

[88] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems **30** (2017)

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

[90] Willson, S.P., Andrews, L.: Characterization of the reaction products of laser-ablated early lanthanide metal atoms with dinitrogen. infrared spectra of LnN, LnN2, (LnN) 2, and Ln(NN)x molecules. The Journal of Physical Chemistry A **102**(50), 10238–10249 (1998)

[91] Dreyer, E. G.; Tillmanns: Dreyerit: Ein natürliches, tetragonales Wismutvandat von Hirschhorn/Pfalz. Neues Jahrb. Mineral., Monatsh. (4), 151–154 (1981)

[92] Sleight, A., Chen, H.-Y., Ferretti, A., Cox, D.: Crystal growth and structure of BiVO4. Materials Research Bulletin **14**(12), 1571–1581 (1979)

[93] Fraser, W., Kennedy, S., Snow, M.: The nitrate positions in phase III of thallous nitrate. Acta Crystallographica Section B: Structural Crystallography and Crystal Chemistry **31**(2), 365–370 (1975)

[94] Materials Project: mp-5915: TlNO3 (Orthorhombic, Pnma, 62). https://doi.org/10.17188/1277176 (2023). https://doi.org/10.17188/1277176

[95] Gromilov, S.A., Dyachkova, T.V., Bykova, E.A., Tarakina, N.V., Zaynulin, Y.G., Yusenko, K.V.: Synthesis of Ir1-xRex ($0.15 \leq x \leq 0.40$) solid solutions under

high-pressure and high-temperature. International Journal of Materials Research **104**(5), 476–482 (2013)