Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Check for updates

Cite this: *Digital Discovery*, 2024, **3**, 1889

Dismai-Bench: benchmarking and designing generative models using disordered materials and interfaces†

Adrian Xiao Bin Yong, *[ab] Tianyu Su [ab] and Elif Ertekin *[bc]

Received 10th April 2024
Accepted 14th August 2024

DOI: 10.1039/d4dd00100a

rsc.li/digitaldiscovery

Generative models have received significant attention in recent years for materials science applications, particularly in the area of inverse design for materials discovery. However, these models are usually assessed based on newly generated, unverified materials, using heuristic metrics such as charge neutrality, which provide a narrow evaluation of a model's performance. Also, current efforts for inorganic materials have predominantly focused on small, periodic crystals ($\leq$20 atoms), even though the capability to generate large, more intricate and disordered structures would expand the applicability of generative modeling to a broader spectrum of materials. In this work, we present the Disordered Materials & Interfaces Benchmark (Dismai-Bench), a generative model benchmark that uses datasets of disordered alloys, interfaces, and amorphous silicon (256–264 atoms per structure). Models are trained on each dataset independently, and evaluated through direct structural comparisons between training and generated structures. Such comparisons are only possible because the material system of each training dataset is fixed. Benchmarking was performed on two graph diffusion models and two (coordinate-based) U-Net diffusion models. The graph models were found to significantly outperform the U-Net models due to the higher expressive power of graphs. While noise in the less expressive models can assist in discovering materials by facilitating exploration beyond the training distribution, these models face significant challenges when confronted with more complex structures. To further demonstrate the benefits of this benchmarking in the development process of a generative model, we considered the case of developing a point-cloud-based generative adversarial network (GAN) to generate low-energy disordered interfaces. We tested different GAN architectures and identified reasons for good/poor performance. We show that the best performing architecture, CryinGAN, outperforms the U-Net models, and is competitive against the graph models despite its lack of invariances and weaker expressive power. This work provides a new framework and insights to guide the development of future generative models, whether for ordered or disordered materials.

1 Introduction

Generative modeling has emerged as a powerful tool for tackling problems in materials science.[1,2] Initially limited to simpler molecules[3,4] and proteins,[5] generative modeling has since advanced to include inorganic materials[6–8] as well. The primary interest in generative modeling has been its promise for inverse materials design,[9–11] where the primary objective is to create new materials tailored to specific properties rather than screening known materials for desired characteristics. Generative models are distinguished from discriminative models, as the latter learns the conditional probability $p(y|x)$ of observing a property (y) given a material representation (x). Instead, a generative model learns the joint probability distribution $p(x, y)$ of the data that it was trained on, and samples from the distribution of structures.

While generative modeling efforts for inorganic materials[9,11,12] have primarily centered around simpler bulk crystals, there has been comparatively less emphasis on disordered systems, despite their relevance across a wide spectrum of applications.[13–15] Disordered systems usually have complex and irregular structures, necessitating large atomic representations and requiring more powerful generative models than those developed for simple crystals. They include structures that completely lack crystal lattices such as amorphous materials, as well as structures beyond bulk materials such as surfaces and interfaces. In direct physical modeling, disordered materials are

[a] *Department of Materials Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA. E-mail: axyong2@illinois.edu*

[b] *Materials Research Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA*

[c] *Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA. E-mail: ertekin@illinois.edu*

† Electronic supplementary information (ESI) available. See DOI: https://doi.org/10.1039/d4dd00100a

most typically represented by large so-called "supercells" that (spuriously) introduce periodicity at larger length scales. Databases of disordered materials are growing,[16–18] offering compelling prospects for the inverse design of metal–organic frameworks, porous amorphous materials, amorphous battery materials, and more. Beyond materials discovery, generative modeling can be used to generate amorphous structures of arbitrarily large sizes upon training on smaller samples that capture material correlation lengths.[19] This capability enables more thorough investigations into properties that are influenced by size effects, such as thermal conductivity and mechanical properties. Generative modeling can also be used to refine atomic structures to align with experimental observations,[20] typically focusing on the refinement of disordered structures.[21,22] Yet, the application of generative modeling to disordered systems remains limited, such as generating 2D morphology rather than precise atomic structures.[19] Moreover, generative models have been reported to fail when applied to large systems.[11,23] To reap the benefits of generative modeling for disordered systems, better generative models need to be developed and evaluated on disordered systems.

When building a generative model, two major design decisions are the type of generative model and the material representation (*i.e.*, the input used to describe the material). The compatibility of these two choices is important as well. The types of generative models that have been used for materials include variational autoencoders (VAEs),[6,7,9,11] generative adversarial networks (GANs),[8,10,24,25] diffusion models,[26–30] and language models.[31–33] Generative models were initially developed with two main types of material representations: (1) voxels[6,7,11,12] and (2) point clouds.[8–10,25] Voxel representation is memory intensive, resulting in limitations in the voxelization resolution and thus number of atoms (*e.g.*, Court *et al.*[11] restricted the number of atoms to ≤40 atoms per cell). Reconstruction issues were also reported[11] for non-cubic cells. On the other hand, point clouds directly represent structures using their atomic coordinates and lattice parameters, making them highly scalable with the number of atoms. However, the design of point cloud architectures that are symmetry-invariant is not trivial, as the commonly used PointNet architecture[34] does not include the desired symmetry invariances. More recently, other representations such as graphs,[26,30] coordinate-based representations (*e.g.*, UniMat,[29] CrysTens[28]), and text-based representations[31–33] have also been explored. Graphs are particularly attractive due to their symmetry invariances and strong expressive power, capturing both geometrical features and neighbor information. However, graph convolutions become computationally and memory intensive as the number of atoms increases.

To compare different generative models and make design choices, it is necessary to sufficiently evaluate the generated structures for their validity. Training generative models for materials discovery inherently makes the evaluation of the models' performance difficult. In more conventional problems such as image generation or speech synthesis, it is relatively easy to discern if the model has learned to generate realistic images or speech from the training data. However, it is much more difficult to determine if a newly generated material is realistic, or if the model is simply generating noise. Recent generative models[26,27,29,33] have relied on limited and heuristic metrics (*e.g.*, charge neutrality, material space coverage) to evaluate and compare between models, making it difficult to meaningfully assess model performance. One approach to circumventing the issues of evaluating on new, unknown materials is to instead train the model on a fixed set of materials (*e.g.*, perovskites). Restricting the material space allows for easier determination of whether the correct structure is being generated, and direct comparison of the properties between the generated and training structures can be performed. In this scenario, however, the materials on which the model is trained should be sufficiently challenging to provide meaningful evaluation of the model's performance. In this regard, disordered materials are good candidates for the task, given that generative models can fail to generate even a single type of disordered material (*e.g.*, amorphous silicon[23]).

In this work, we present the Disordered Materials & Interfaces Benchmark (Dismai-Bench), a generative model benchmark that uses datasets of an $Fe_{60}Ni_{20}Cr_{20}$ austenitic stainless steel, a disordered $Li_3ScCl_6(100)$–$LiCoO_2(110)$ battery interface, and amorphous silicon. Dismai-Bench evaluates generative models on a wide range of material disorder ranging from structural to configurational (see Fig. 1). Structural disorder increases from left to right, and configurational disorder increases from right to left, in Fig. 1. The composition of each dataset is fixed, and each structure has 256–264 atoms. We selected four recent diffusion models to be benchmarked on Dismai-Bench, including two models that use graph representations (CDVAE[26] & DiffCSP[27]) and two models that use coordinate-based representations (CrysTens[28] & UniMat[29]). The models were trained on one dataset at a time, and the generated structures were compared with the training structures to obtain structural similarity metrics. These metrics quantify the model's ability to learn complex structural patterns found in disordered materials. We show that the graph models outperform the coordinate-based models due to the higher expressive power of graphs. The success of the less expressive models in materials discovery[29,33] suggests that noisy models are better for discovering small crystals, but face challenges when tasked with generating larger, more complex structures.

To demonstrate the application of Dismai-Bench in the development of a generative model, we further considered the design of a GAN to generate low-interface-energy $Li_3ScCl_6(100)$–$LiCoO_2(110)$ interface structures. We chose the simple point cloud representation, and tested multiple different GAN architectures for which we included bond distance information explicitly in the GANs, instead of just atomic coordinates. Direct comparison between the generated and training structures identified the architecture that best achieved the intended goal, along with explanations for why the other architectures were less successful. We demonstrate that the best architecture, Crystal Interface Generative Adversarial Network (CryinGAN), can generate the disordered interfaces with low interface energy, and similar structural features to the training structures. Despite its design simplicity, CryinGAN outperforms the

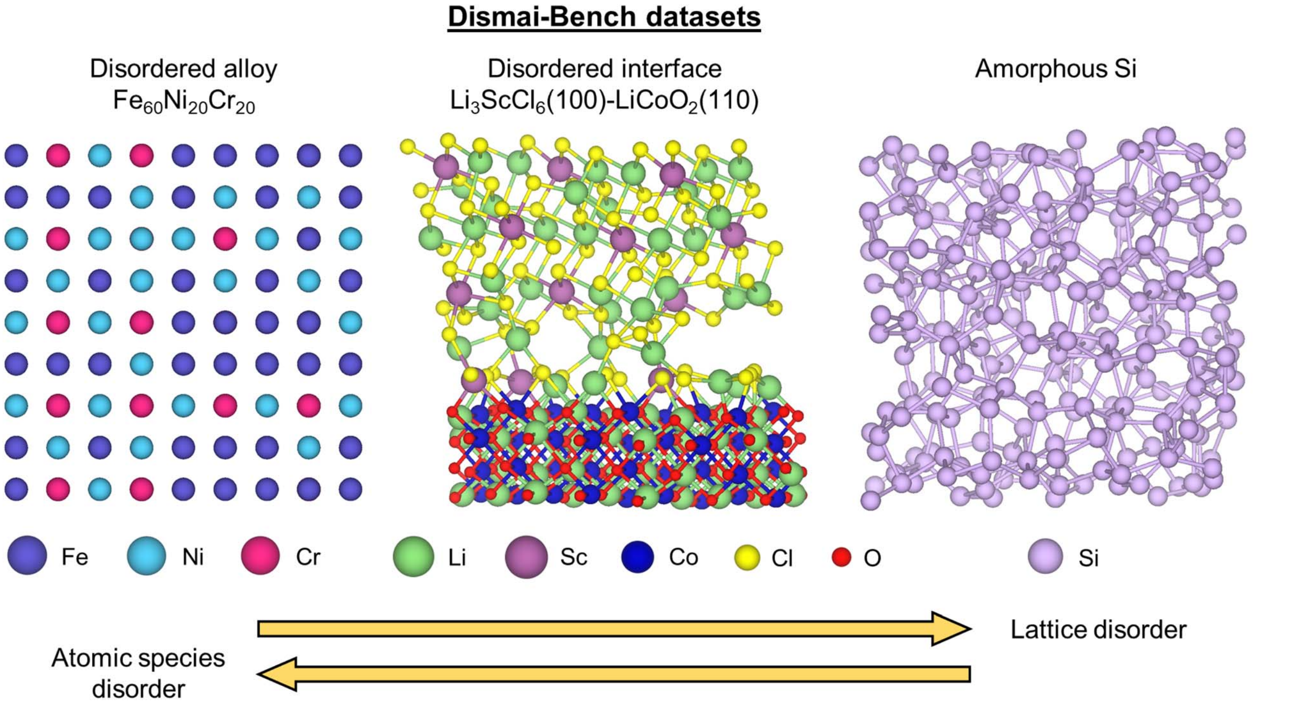

Fig. 1 Datasets used in Dismai-Bench, consisting of a disordered $Fe_{60}Ni_{20}Cr_{20}$ austenitic stainless steel system, a disordered Li_3ScCl_6(100)–$LiCoO_2$(110) battery interface system, and an amorphous silicon system.

more recent coordinate-based diffusion models on Dismai-Bench. It does not outperform the graph diffusion models across all datasets, however, possibly as a result of its weaker expressive power and lack of invariances. Through this work, we present a novel framework for conducting meaningful comparisons between models, providing valuable insights into model weaknesses and failures to inform the design of future generative models.

2 Results and discussion

2.1 Datasets and interatomic potentials

An overview of the dataset curation is outlined here; further details are found in the Methods section. A total of six datasets are used in Dismai-Bench, consisting of four alloy datasets, one interface dataset, and one amorphous silicon dataset. Each dataset contains a total of 1500 structures, split into 80% training and 20% validation data. Test sets are not needed since model performance is measured using the benchmark metrics.

2.1.1 $Fe_{60}Ni_{20}Cr_{20}$ austenitic stainless steel. The stainless steel datasets consist of face-centered cubic (FCC) crystals that are structurally simple, but configurationally complex (refer to Fig. 1). Atoms of various species occupy the lattice sites with different ordering tendencies. The generative models are challenged with generating structures that not only have well-defined FCC lattices, but also the correct degrees of ordering. The stainless steel datasets were created using a cluster expansion Monte Carlo (CEMC) approach.[35,36] The cluster expansion (CE) model was adapted from ref. 36, where pair interactions up to the 7th neighbor shell were included. The composition of the structures is $Fe_{60}Ni_{20}Cr_{20}$, and each structure contains 256 atoms. Monte Carlo (MC) simulations were carried out in the canonical ensemble at 300 K and 1500 K, such as to obtain datasets with different degrees of short-range order (SRO). The SRO is quantified using the Warren-Cowley SRO parameter,[37]

$$\alpha_l^{AB} = 1 - \frac{P_l^{AB}}{C_A C_B} = 1 - \frac{p_{l,A}^{B}}{C_B}, \quad (1)$$

where P_l^{AB} is the probability of finding AB pairs in the l-th neighbor shell, and $p_{l,A}^{B} = P_l^{AB}/C_A$ is the conditional probability of finding atom B in the l-th coordination shell of atom A. C_A and C_B are the concentration of A and B atoms respectively. SRO parameter $\alpha = 0$ indicates zero correlation between atoms (as in a random solution), while $\alpha < 0$ indicates an attractive interaction and $\alpha > 0$ indicates a repulsive interaction. Note that the A and B atoms can be of the same atomic species.

The SRO distributions of the 300 K and 1500 K alloy training datasets are shown in ESI Fig. S1† for the 1st and 2nd nearest neighbor interactions. The 300 K dataset shows more prominent SRO than the 1500 K dataset. The SRO parameter tends to distribute away from zero at 300 K, and consistently distribute near zero at 1500 K. We also created two additional datasets by filtering the CEMC-generated structures such that the SRO parameters have narrow distributions within ±0.1 of the average values (see ESI Fig. S2†). Sufficiently large number of structures were generated to obtain 1500 structures for each dataset. We refer to the unfiltered and filtered datasets as the wide SRO and narrow SRO datasets respectively. This SRO filtering was performed to enable comparison of generative model performance when trained on structures with more noisy SRO distribution (wide SRO dataset) and structures with less noisy SRO distribution (narrow SRO dataset).

2.1.2 Amorphous silicon. Amorphous silicon can be thought of as the polar opposite of the FCC alloys. Amorphous silicon consists of a single atomic species only, but completely lacks ordering in the form of a crystalline lattice. The generative models are not assessed on any ability to learn ordering relationships between different atomic species. Instead, they are assessed on their abilities to learn the complex structural patterns found in amorphous silicon, such as near-tetrahedral local environments and pair distribution functions. The amorphous silicon dataset was adapted from ref. 38. The original data consists of a 100 000-atom amorphous silicon structure generated through melt-quench molecular dynamics simulation.[38] The structure was sliced into smaller blocks with lattice parameters corresponding to 256-atom amorphous silicon structures. The blocks were sliced at different locations to obtain a total of 1500 blocks. Blocks with <256 atoms had atoms added at random to low density regions, and blocks with >256 atoms had atoms removed at random from high density regions, so that all blocks have 256 atoms. The 1500 structures were relaxed using a pre-trained SOAP-GAP[39] machine learning interatomic potential for Si. The resulting structures have a higher concentration of defects compared to the original 100 000-atom structure (refer to ESI Fig. S3†) due to the slicing and atom addition/removal, but the Si coordination geometry remains predominantly tetrahedral.

2.1.3 Li_3ScCl_6(100)–$LiCoO_2$(110) battery interface. The disordered interface dataset assesses the generative models on structures that exhibit a mixture of structural and configurational disorder (refer to Fig. 1). Atoms in the disordered interface region are not arranged in well-defined lattices, and coordinate with each other in a range of motifs. The models have to learn to generate these complex heterogeneous interfaces, where they need to construct the crystalline slabs and disordered interface region correctly. The interface considered in this work is a solid-state battery interface between the $LiCoO_2$ (LCO) cathode and the Li_3ScCl_6 (LSC) solid electrolyte. LCO is one of the most commonly used cathode materials in commercial Li-ion batteries.[40] LSC[41] belongs to the class of halide solid electrolytes, which can achieve both high ionic conductivity and high-voltage stability, enabling the use of high-voltage cathodes in all-solid-state batteries.[42,43] In previous work,[44] a similar disordered interface structure was observed between LCO and a different halide solid electrolyte, Li_3YCl_6, despite the different compositions and crystal structures of the halide solid electrolytes. Disordered interfaces have been observed experimentally across diverse material systems. They arise for a variety of reasons including elemental segregation to the interface,[45,46] intermixing across the interface,[47] and ion irradiation.[48] For the oxide-chloride interface system, we found that chlorides have an innate tendency to form disordered interfaces with oxides; further details are documented in ESI Supplementary Note 1.†

The Li_3ScCl_6(100)–$LiCoO_2$(110) interface dataset was created by generating random interface structures and relaxing them. Each structure was first constructed by randomly generating 3 formula units of LSC atoms in the interface region between the LSC and LCO slabs. For each structure, the thickness of the interface region was randomly selected between 4 and 6 Å, and a random lateral displacement was applied to the LSC slab (translation allowed along the full range of both lateral directions). The randomly generated structures were relaxed using density functional theory (DFT) calculations.

To perform relaxations faster, we trained from scratch a machine learning interatomic potential, M3GNet,[49] for the LSC–LCO interfaces. A total of 15 484 training structures consisting of optimized structures and intermediate ionic steps of the DFT relaxations were used to train the M3GNet interatomic potential. The M3GNet model achieved low test set mean absolute errors (MAEs) of 2.70 meV per atom, 20.9 meV $Å^{-1}$, and 0.0146 GPa for energy, force, and stress respectively. Machine learning interatomic potentials with similar (or higher) MAEs showed good performance in relaxations and molecular dynamics simulations when applied to other Li-ion conductors.[49,50] We then relaxed randomly generated interface structures using the M3GNet interatomic potential. The M3GNet-relaxed structures were found to be near DFT convergence (refer to Table 5 in the Methods section). The interface energies of the relaxed structures were distributed across a wide range (approximately 1.4 J m^{-2}) as shown in ESI Fig. S4.† We define structures with interface energies no higher than 0.4 J m^{-2} relative to the lowest energy structure to be low-interface-energy structures. For reference, the observation frequency of a grain boundary in aluminum metal decreases by 95% when the grain boundary energy increases by around 0.35 J m^{-2} (ref. 51). We assembled 1500 low-interface-energy structures (all relaxed by M3GNet only) as the disordered interface dataset.

2.2 Generative models

Five generative models were benchmarked on Dismai-Bench. Four of these (CDVAE,[26] DiffCSP,[27] CrysTens,[28] UniMat[29]) are existing models, and one (CryinGAN) was developed as part of this work to demonstrate the application of Dismai-Bench in model development. An overview of the first four models is outlined here, whereas CryinGAN will be presented in detail in Section 2.4.

CDVAE[26] is a VAE that uses equivariant graph neural networks for its encoder and decoder. The encoder is a graph convolutional network that encodes material structures into latent representations (z). Three multilayer perceptrons (MLPs) are used to predict the composition, lattice parameters, and number of atoms from z. These predictions are used to initialize structures (corresponding to the sampled z), where the atoms are initialized at random positions. The decoder is a graph diffusion model that denoises both the atomic coordinates and atomic species of the atoms.

In our early tests of training CDVAE on Dismai-Bench datasets, CDVAE was found to fail in generating structures with the correct compositions (see ESI Fig. S5†). Although CDVAE was able to predict the compositions correctly from z, the compositions became incorrect after the atomic species of the atoms were denoised. Therefore, we modified CDVAE such that the atomic species denoising becomes an optional feature, and all CDVAE benchmarking was performed without atomic species denoising. We also tested the effect of atomic species denoising when CDVAE is trained on the MP-20 dataset,[26] which includes structures from the Materials Project[52] of various compositions. Here, denoising the atomic species was found to increase the composition accuracy of reconstructed structures from around 24% to 54%. However, denoising the atomic species appears to be detrimental for larger structures (such as those in the Dismai-Bench datasets), and fails even when all structures have the same composition.

DiffCSP[27] is another graph diffusion model. The main feature introduced in DiffCSP is the ability to jointly denoise the atomic coordinates and lattice parameters. In contrast, CDVAE predicts the lattice parameters first, and they remain fixed throughout the diffusion steps. Note that DiffCSP does not denoise the atomic species. To better capture periodicity, DiffCSP also uses fractional instead of Cartesian coordinates (as in CDVAE), and uses periodic translation invariant Fourier transformations in its message passing. However, DiffCSP does not include bond angle information in its graphs, whereas CDVAE does.

When DiffCSP was trained on the disordered interface dataset allowing joint atomic coordinate and lattice parameter diffusion, the generated structures were found to be of poor quality (refer to ESI Fig. S6a†). We modified DiffCSP to allow teacher forcing of lattice parameters during the initial training epochs, where the ground truth lattice parameters are used as input, and lattice cost is not used to update the model (only coordinate cost is used). The quality of the generated structures improved when trained with teacher forcing (see ESI Fig. S6b†). However, structures with the best quality were still obtained when DiffCSP was trained without any lattice diffusion, using the ground truth lattice parameters as input (see ESI Fig. S6c†). Therefore, all benchmarking of DiffCSP presented in Section 2.3 was performed without lattice denoising. This choice also provides a consistent comparison between DiffCSP and CDVAE, since CDVAE also does not perform lattice denoising. For reference, the Dismai-Bench metrics of DiffCSP trained with lattice denoising and teacher forcing are listed in ESI Table S1.† DiffCSP performance drops with lattice denoising across the metrics (see Section 2.3 for benchmarking details).

CrysTens[28] is an image-like representation for materials. The pixel values of a CrysTens image are filled with information of the structure such as lattice parameters, fractional coordinates, and atomic number. Each CrysTens image has four channels, analogous to the RGB color channels of an image. The first channel includes a pairwise distance matrix between all atom pairs, and the remaining three channels include the pairwise Δx, Δy, and Δz matrices respectively. CrysTens is used with a 2D U-Net diffusion model for image generation, and the generated CrysTens images are reconstructed back into material structures. We made no major modifications to the baseline implementation. The dimensions of the CrysTens images were increased to fit Dismai-Bench structures (the original implementation only allowed up to 52 atoms). Refer to the Methods section for more details.

UniMat[29] is a video-like representation for materials. Each frame of a UniMat video represents a single atom in the structure, and has three channels corresponding to the x, y, and z coordinates. The atomic species of the atom is indicated by the pixel location in the frame (*e.g.*, top left pixel is H), where those pixel values correspond to the coordinates of the atom, and all other pixel values are set to -1. UniMat is used with a 3D U-Net diffusion model for video generation, and the generated UniMat videos are reconstructed back into materials. Despite the lack of any geometrical information beyond atomic coordinates, UniMat was shown to outperform CDVAE in discovering materials with lower formation energy (upon DFT relaxation of the generated structures). As the UniMat code is currently not openly available, we used an open-access implementation[53] of the 3D U-Net model[54] that the UniMat model was repurposed from. Besides this change, no major modifications were made to UniMat. The dimensions of the UniMat frames were decreased, since each Dismai–Bench structure has only five atomic species at most.

2.3 Dismai-Bench

Benchmarking was performed by training all generative models on each dataset separately from scratch, such that the models only generate one type of structure at a time. 1000 structures were generated using each model, and post-processed as described in the Methods section. The disordered interface structures were relaxed using the M3GNet[49] interatomic potential, and the amorphous Si structures were relaxed using the SOAP-GAP[39] interatomic potential. No relaxations were performed on the alloy structures. These final structures were used to calculate the benchmark metrics. For each generative model architecture, three separate models were trained for each dataset, and the metrics were averaged across the three models.

Although the benchmark metrics for all five generative model architectures are listed in this section, only the results for CDVAE, DiffCSP, CrysTens, and UniMat will be discussed here, since the CryinGAN architecture has not been presented yet (see Section 2.4). The benchmark results for CryinGAN will be discussed in Section 2.6, along with an overall comparison of the general performance of all architectures.

2.3.1 Disordered LSC–LCO interface. Examples of interface structures generated by the models are shown in Fig. 2. The structures shown are as-generated without any post-processing or relaxation. Visually, the structures generated by CryinGAN, CDVAE, and DiffCSP are similar to the training structures. The coordinate-based U-Net diffusion models generated the most noisy structures.

The generated structures were relaxed, and the benchmark metrics were calculated by analyzing the local coordination environment (motifs) of the atoms. The CrystalNNFingerprint,[55] which contains the coordination likelihoods and local structure order parameters of a given atom, was averaged across the structures generated by each model, and separately obtained for the training dataset as well. We calculated the fingerprints of

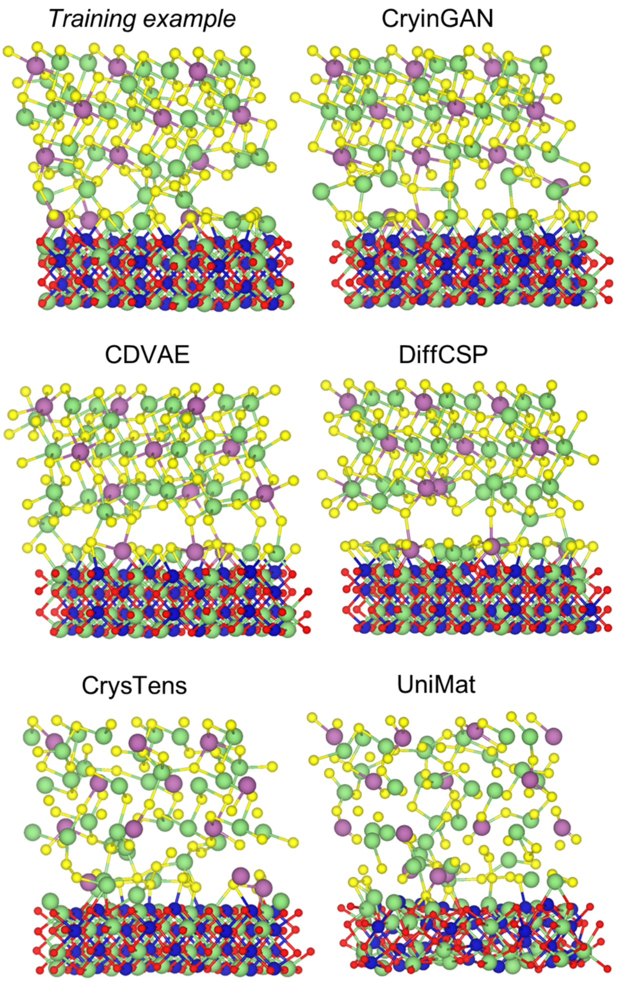

Fig. 2 Example disordered LSC–LCO interface structures generated by the generative models. The structures shown are as-generated by the models (*i.e.*, no post-processing or relaxation). An example structure from the training dataset is also included for reference.

the cations (Li, Co, and Sc) coordinated to the anions (Cl and O). We also appended the fraction of Cl and O neighbors in each motif to the fingerprints, so that the fingerprints contain both chemical and coordination information. For each model, the Euclidean distance between the average fingerprint of the training structures and the generated structures was calculated.

The benchmark metrics for the disordered interfaces are shown in Table 1. Parameters d_{Li}, d_{Co}, d_{Sc}, and d_{all} represent the fingerprint Euclidean distance of Li, Co, Sc, and all cations respectively. The percentages of structures that failed to be post-processed or relaxed are also listed. Notably, CDVAE is the only model that generates structures close to being fully relaxed, requiring only an average of 10 relaxation steps per structure (see ESI Table S2†). All other models required at least 68 relaxation steps on average, and the U-Net models have >50% failed structures. UniMat exhibited the highest % failed structures and d_{all}, likely because it uses a less expressive material representation (atomic coordinates only). CrysTens, despite the inclusion of bond distances, also has high % failed structures. Although its Euclidean distance metrics are small, the low values are only achieved after relaxation (refer to Fig. 2 and Table S2† for unrelaxed structures). On the other hand, the graph diffusion models perform significantly better than the U-Net diffusion models due to the higher expressive power of graphs. Comparing between CDVAE and DiffCSP, CDVAE generates interface structures closer to convergence than DiffCSP, but upon relaxation, DiffCSP achieves lower distance metrics than CDVAE.

2.3.2 Amorphous silicon. Examples of amorphous Si structures generated by the models are shown in Fig. 3. Only the graph diffusion models (CDVAE & DiffCSP) were able to generate the structures successfully. The other models generated random-looking structures with large voids and little bonding between atoms. Therefore, further detailed benchmarking of amorphous Si was only performed for CDVAE and DiffCSP. Similar to the disordered interfaces, we determined the coordination motif fingerprints of Si, and the Euclidean distance between the average fingerprint of the training and generated structures. We also determined the radial distribution functions (RDFs) and bond angle distributions of the structures, and compared them to that of the training set.

The benchmark metrics for amorphous Si are shown in Table 2. Parameters d_{motif}, d_{rdf}, and d_{angle} represent the Euclidean distance for motif fingerprint, RDF, and bond angle distribution respectively. Both CDVAE and DiffCSP had 0% failed structures, but CDVAE-generated structures were significantly closer to being fully relaxed than DiffCSP-generated structures. CDVAE only required an average of 23 relaxation steps per structure, whereas DiffCSP required an average of 140 steps per structure (see ESI Table S3†). Similarly, all distance metrics of CDVAE are lower than DiffCSP. The lack of bond angle information in DiffCSP's graphs is likely one main contributor to these trends. The RDFs and bond angle distributions of CDVAE and DiffCSP are shown in ESI Fig. S7.† DiffCSP and CDVAE encode distance information in their

Table 1 Dismai-Bench metrics for the disordered LSC–LCO interfaces. Each metric is represented by the average value over 3 separately trained models. The minimum and maximum values are shown in brackets. For all metrics, lower is better

Model	d_{Li} (min, max)	d_{Co} (min, max)	d_{Sc} (min, max)	d_{all} (min, max)	% Struc. failed (min, max)
CDVAE	0.0664 (0.0601, 0.0697)	0.0427 (0.0372, 0.0504)	0.201 (0.186, 0.214)	0.0547 (0.0508, 0.0589)	0.00 (0.00, 0.00)
DiffCSP	0.0439 (0.0414, 0.0474)	0.0272 (0.0240, 0.0298)	0.0954 (0.0909, 0.102)	0.0370 (0.0353, 0.0388)	7.17 (6.00, 8.90)
CrysTens	0.0202 (0.0160, 0.0244)	0.0141 (0.00707, 0.0239)	0.0972 (0.0888, 0.106)	0.0213 (0.0164, 0.0257)	55.3 (50.6, 61.7)
UniMat	0.131 (0.0966, 0.154)	0.235 (0.165, 0.292)	0.101 (0.0891, 0.111)	0.151 (0.111, 0.181)	64.6 (60.6, 68.3)
CryinGAN	0.0538 (0.0466, 0.0600)	0.0274 (0.0210, 0.0359)	0.0888 (0.0838, 0.0963)	0.0426 (0.0379, 0.0451)	9.00 (7.20, 10.2)

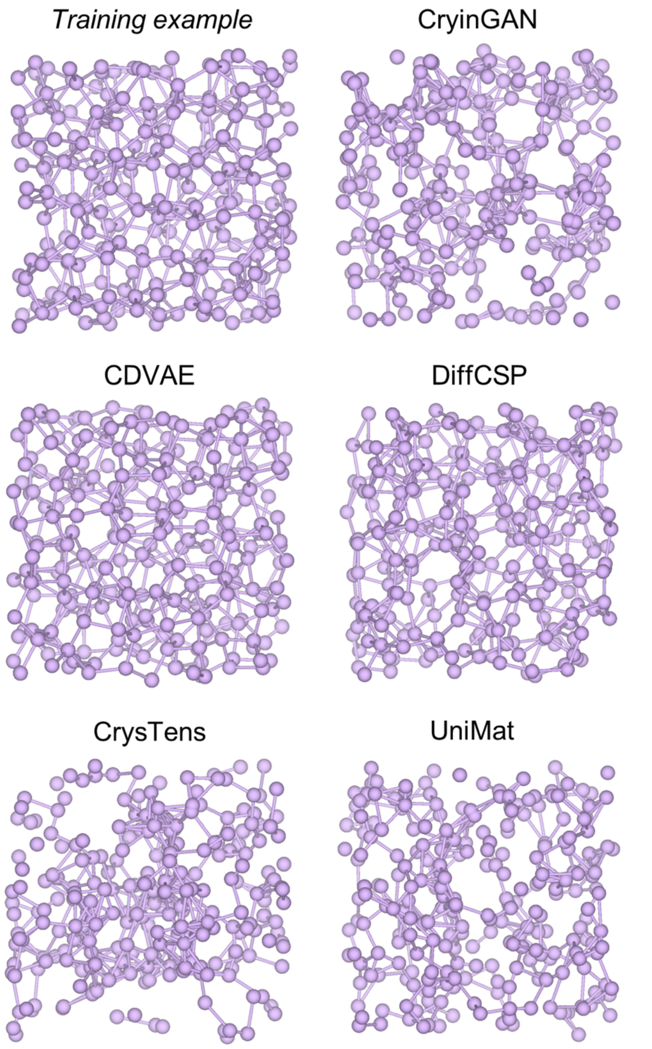

Fig. 3 Example amorphous Si structures generated by the generative models. The structures shown are as-generated by the models (*i.e.*, no post-processing or relaxation). An example structure from the training dataset is also included for reference.

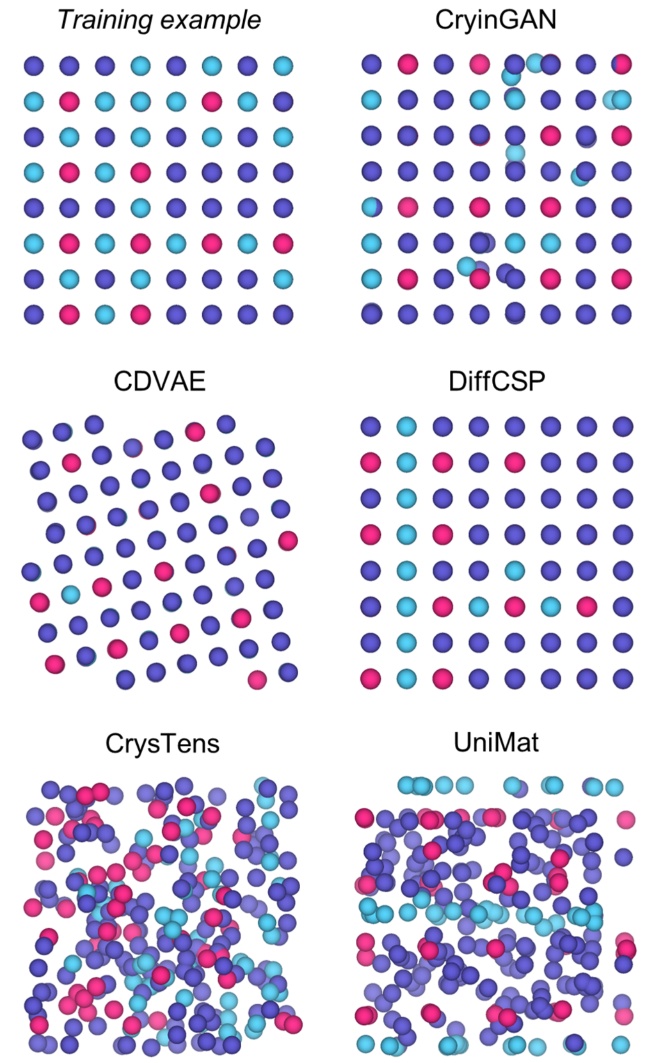

Fig. 4 Example alloy structures (300 K, narrow SRO) generated by the generative models. The structures shown are as-generated by the models (*i.e.*, no post-processing). An example structure from the training dataset is also included for reference.

graphs, and both show similar RDFs to the training dataset. However, there is a clear difference when comparing the bond angle distributions, where DiffCSP shows a larger discrepancy with the training dataset than CDVAE. These results indicate that bond angle information is particularly beneficial for helping generative models learn amorphous structures, since amorphous structures have more complicated arrangements of atoms that do not locate on lattice sites.

2.3.3 Disordered stainless steel alloy.

Examples of disordered alloy structures generated by the models are shown in Fig. 4. CryinGAN, CDVAE, and DiffCSP were able to generate the FCC alloy structures successfully, whereas CrysTens and UniMat were unable to reproduce the underlying FCC lattice. Therefore, detailed alloy benchmarking was only performed for CryinGAN, CDVAE, and DiffCSP. Interestingly, CDVAE was prone to generating rotated structures (since graphs are rotationally invariant), but all DiffCSP-generated structures were unrotated. Some CDVAE-generated structures had noisy lattices, where atoms exhibited relatively large deviations from

Table 2 Dismai-Bench metrics for amorphous Si. Each metric is represented by the average value over 3 separately trained models. The minimum and maximum values are shown in brackets. For all metrics, lower is better

Model	d_{motif} (min, max)	d_{rdf} (min, max)	d_{angle} (min, max)	% Struc. failed (min, max)
CDVAE	0.0402 (0.0396, 0.0412)	0.392 (0.381, 0.408)	0.00332 (0.00312, 0.00353)	0.00 (0.00, 0.00)
DiffCSP	0.0647 (0.0462, 0.0908)	1.39 (1.03, 1.69)	0.0103 (0.00797, 0.0133)	0.00 (0.00, 0.00)

pristine FCC lattice sites (see ESI Fig. S8a†), whereas all DiffCSP-generated structures had well-defined lattices. A fraction of CDVAE-generated structures had slightly shorter lattice spacing, creating additional sites that were vacant since the total number of atoms was fixed (see ESI Fig. S8b†). DiffCSP's stronger ability to learn the FCC lattice is likely due to the use of fractional coordinates and Fourier transformations in its message passing to capture periodicity.[27]

The generated structures were post-processed to remove atoms not on lattice sites. Any structure with >50 atoms removed (~20% of all atoms) was considered a failed structure and rejected. Then, the fingerprint of each structure was calculated using a vector of conditional probabilities of observing each cluster (monomer/dimer) in the structure, up to the 7th neighbor shell. The Euclidean distance between the average cluster fingerprint of the training and generated structures, d_{cluster}, was calculated for each model.

Benchmark metrics for the disordered alloys are shown in Table 3. The percentage of failed structures is 0% for DiffCSP and CryinGAN, while CDVAE exhibits a tiny percentage of failed structures. However, the percentage of structures with site vacancies is 90–100% for DiffCSP and CryinGAN. In comparison, CDVAE has significantly lower percentages of structures with vacancies, around roughly 40% for 300 K structures and 60% for 1500 K structures. Although DiffCSP-generated structures had more well-defined lattices than CDVAE-generated structures, almost all structures had overlapping atoms on lattice sites, resulting in site vacancies in the structures.

For the 300 K structures with narrow SRO, the d_{cluster} values of CDVAE and DiffCSP are similar. However, for the 300 K structures with wide SRO, CDVAE has lower d_{cluster} than DiffCSP. The wide SRO structures have less consistent SRO distributions, so DiffCSP had more difficulty in learning the wide SRO structures than the narrow SRO structures. On the other hand, d_{cluster} only increased slightly for CDVAE between the wide and narrow SRO structures, indicating CDVAE's stronger ability to learn different degrees of SRO. For the 1500 K structures, which more resemble random solid solutions, there is little difference in d_{cluster} between the narrow and wide SRO structures. Here, d_{cluster} is lower for DiffCSP than CDVAE. Some factors contributing to DiffCSP's better performance in generating random solid solutions may be its stronger ability in learning the FCC lattice, weaker ability in learning SRO patterns, and low number of site vacancies (~1.5 vacancies per structure). The Warren–Cowley SRO parameter distributions for the 1st and 2nd nearest neighbor interactions are shown in ESI Fig. S9–11† for reference. Overall, the models were able to generate structures with similar SRO distributions to the training structures.

2.4 CryinGAN development

We present here a case study of developing a generative model with the help of Dismai-Bench, demonstrating how meaningful feedback about model performance is obtained through direct comparisons between generated and training structures. We considered the case of a point-cloud-based GAN to generate low-interface-energy LSC–LCO interface structures. We chose to focus on the interface structures during development for simplicity, and subsequently benchmarked CryinGAN on other datasets to evaluate its generalizability. The disordered interfaces were also the only structures where all four diffusion models were able to generate successfully, hence providing the most informative model comparison. We show that, given the correct architecture, a coordinate-based representation can still perform well on Dismai-Bench (unlike CrysTens and UniMat), and that more complicated architectures do not necessarily outperform simpler architectures.

Table 3 Dismai-Bench metrics for the disordered stainless steel alloy. Each metric is represented by the average value over 3 separately trained models. The minimum and maximum values are shown in brackets. For all metrics, lower is better

Model	d_{cluster} (min, max)	% Struc. w/vac (min, max)	% Struc. failed (min, max)
300 K, narrow SRO			
CDVAE	0.0604 (0.0540, 0.0675)	35.8 (32.0, 41.2)	0.13 (0.00, 0.40)
DiffCSP	0.0645 (0.0566, 0.0792)	94.3 (91.8, 96.5)	0.00 (0.00, 0.00)
CryinGAN	0.117 (0.106, 0.126)	100 (100, 100)	0.00 (0.00, 0.00)
300 K, wide SRO			
CDVAE	0.0658 (0.0598, 0.0742)	37.4 (34.0, 43.4)	0.07 (0.00, 0.10)
DiffCSP	0.103 (0.0831, 0.121)	95.8 (95.4, 96.2)	0.00 (0.00, 0.00)
CryinGAN	0.125 (0.119, 0.129)	100 (100, 100)	0.00 (0.00, 0.00)
1500 K, narrow SRO			
CDVAE	0.0621 (0.0589, 0.0665)	54.8 (51.9, 56.8)	0.37 (0.20, 0.50)
DiffCSP	0.0308 (0.0304, 0.0313)	91.7 (91.5, 91.8)	0.00 (0.00, 0.00)
CryinGAN	0.0643 (0.0629, 0.0659)	100 (100, 100)	0.00 (0.00, 0.00)
1500 K, wide SRO			
CDVAE	0.0618 (0.0549, 0.0694)	61.3 (55.9 68.2)	0.27 (0.20, 0.30)
DiffCSP	0.0338 (0.0332, 0.0342)	92.3 (89.2, 94.4)	0.00 (0.00, 0.00)
CryinGAN	0.0650 (0.0609, 0.0691)	100 (100, 100)	0.00 (0.00, 0.00)

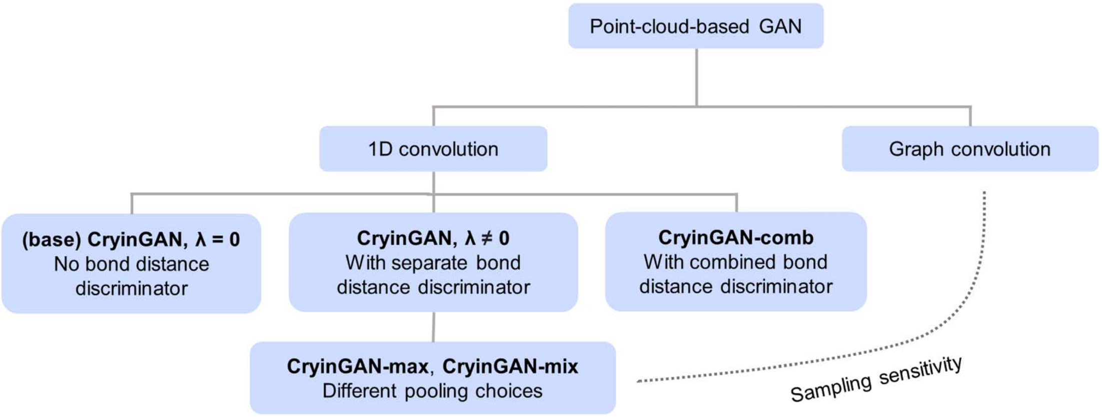
Point-cloud-based GAN
1D convolution
Graph convolution
(base) CryinGAN, λ = 0
No bond distance discriminator
CryinGAN, λ ≠ 0
With separate bond distance discriminator
CryinGAN-comb
With combined bond distance discriminator
CryinGAN-max, CryinGAN-mix
Different pooling choices
Sampling sensitivity

Fig. 5 Schematic of GAN architectures tested. The discriminator either uses 1D convolutions (PointNet) or graph convolutions (CGCNN). For the 1D-convolution-based discriminators, the primary CryinGAN design consists of a fractional coordinate discriminator and a separate bond distance discriminator, where the output of the latter is weighted by λ. Alternate pooling choices were tested (CryinGAN-max and CryinGAN-mix). Graph convolution and different types of pooling affect the sampling sensitivity of the discriminator. CryinGAN-comb combines the two discriminators into a single discriminator.

A typical GAN consists of two neural networks: a generator and a discriminator. The role of the generator is to generate material structures from input noise, whereas the role of the discriminator is to distinguish between the real (training) structures and the fake (generated) structures. The generator and discriminator compete with each other during training to progressively improve the quality of the generated structures. The point cloud representation was used for the GANs, and since all of the structures in the dataset share the same lattice parameters, each structure was represented by the fractional coordinates of its atoms only. We tested a couple of different GAN architectures as summarized in Fig. 5.

The base GAN model (CryinGAN) was adapted from the Composition-Conditioned Crystal GAN (CCCGAN) presented by Kim *et al.*,[25] which was used to generate Mg–Mn–O ternary materials. One-dimensional (1D) convolutions were used in the discriminator to extract the latent features of structures, an inspiration taken from PointNet,[34] a 3D object classification and segmentation network. These 1D convolutions have been similarly implemented in other point-cloud-based crystal generative models such as FTCP-VAE[9] and CubicGAN.[10] We simplified and generalized CCCGAN to be used for any periodic system with fixed lattice, composition, and number of atoms. We did not include any conditional generation capability in CryinGAN for simplicity. The discriminator of the original CCCGAN relied solely on atomic coordinates and lattice parameters to distinguish between real and fake structures. To further provide explicit bond distance information to the discriminator, we added a second discriminator to the CryinGAN model that has the same architecture, but accepts bond distances as input instead of coordinates.

The CryinGAN model architecture is shown in Fig. 6. The generator accepts random gaussian noise as input, and produces fractional coordinates of structures as output. Of the two discriminators, one accepts fractional coordinates as input, and the other accepts bond distances (6 nearest-neighbors of each atom) as input. In the first convolutional layer of the discriminators, the fractional coordinates/bond distances are convoluted along each row separately (*i.e.*, separate 1D convolutions for each atom). Note that CryinGAN is permutationally invariant to atom ordering within each atomic species block (*e.g.*, order of the 72 Li atoms does not matter). CryinGAN implements the Wasserstein loss, which was shown to provide more stable training by preventing the vanishing gradients of the traditional GAN.[56,57] The discriminator loss function with a gradient penalty term for improved stability[57] is:

$$L_{\text{disc}} = \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)] + \mu \mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}\left[\left(\|\nabla_{\hat{x}}D(\hat{x})\|_2 - 1\right)^2\right], \quad (2)$$

where D is the discriminator output, $\tilde{x}$ and x are the inputs for generated and real structures respectively. The distribution $\mathbb{P}_{\hat{x}}$ is taken over interpolated samples between the distribution of real structures $\mathbb{P}_r$, and the distribution of generated structures $\mathbb{P}_g$. In the code implementation, $\hat{x}$ is obtained by interpolating between the fractional coordinates/bond distances of training structures and generated structures. The interpolation point between any two data points is chosen randomly. The parameter μ is the gradient penalty coefficient set to 10 similar to past Wasserstein GANs.[10,25,57] The total discriminator loss, to be minimized, is a weighted sum of the losses from both discriminators

$$L_{\text{disc, total}} = L_{\text{disc, coord}} + \lambda L_{\text{disc, bond}}. \quad (3)$$

Here, $L_{\text{disc, coord}}$ and $L_{\text{disc, bond}}$ are the losses from the fractional coordinate discriminator and bond distance discriminator respectively, and λ is the weight of the bond distance discriminator loss. The total generator loss, to be maximized, is computed similarly according to:

$$L_{\text{gen}} = \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})], \quad (4)$$

$$L_{\text{gen, total}} = L_{\text{gen, coord}} + \lambda L_{\text{gen, bond}}. \quad (5)$$

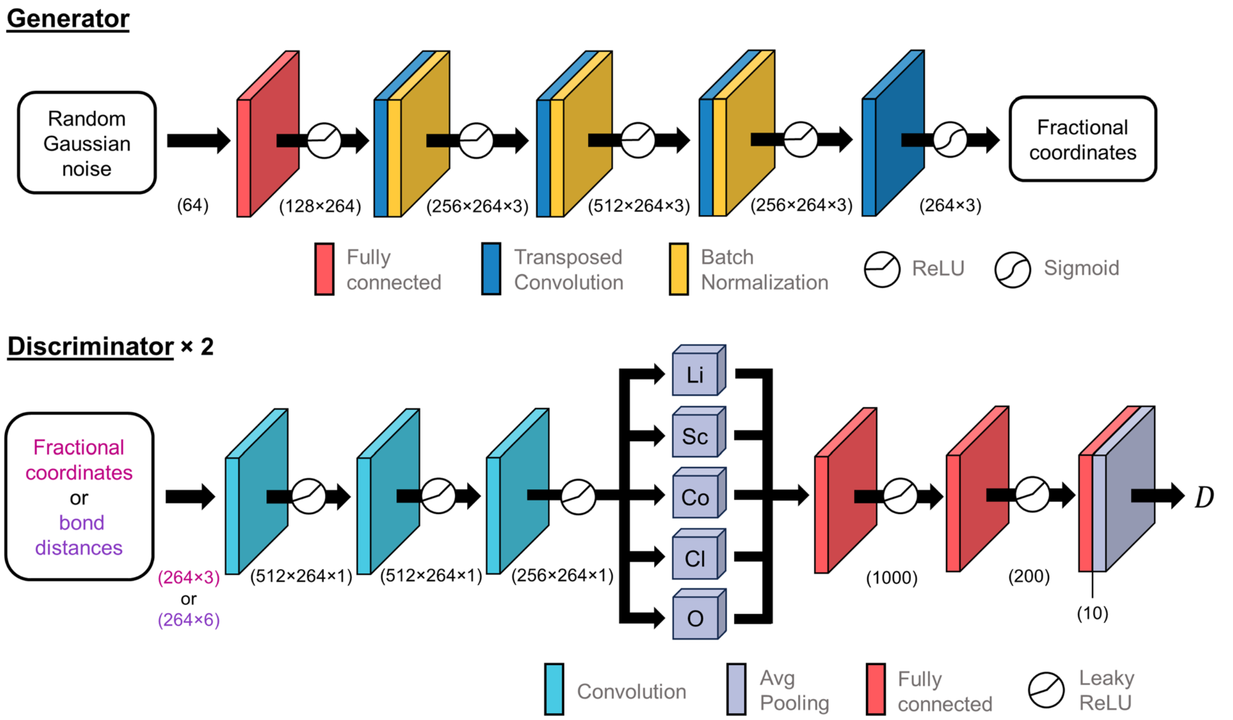

Fig. 6 CryinGAN architecture. The generator takes in random Gaussian noise as input and produces fractional coordinates as output. There are two discriminators with the same architecture, but with different inputs (fractional coordinates or bond distances). *D* is the discriminator output used to calculate the losses. The numbers in brackets represent the dimensions of tensors before/after a layer, where the batch dimension is omitted. Note that 264 corresponds to the number of atoms in each interface structure. The size of the layers shown in the figure do not reflect the tensor dimensions.

Note that the base CryinGAN model uses only the fractional coordinate discriminator ($\lambda = 0$), which we use as the baseline reference model.

We developed the GAN models with the intention of evaluating the best performing model by comparing between DFT-relaxed training and generated structures. The training dataset was curated from structures relaxed with the M3GNet interatomic potential followed by DFT calculations (refer to Methods section). Note that the CryinGAN Dismai-Bench metrics shown in Section 2.3 were calculated using the same procedure as described previously. To study the effect of λ on CryinGAN model performance, we trained CryinGAN models on the (DFT-relaxed) dataset of interface structures with varying λ. A visualization of the training process is shown in ESI Movie S1,† where CryinGAN progressively learns to generate low interface energy structures over training epochs. Due to noise in the models, the generated structures often have a small fraction of atoms that are too close to each other. As λ increases, we found that the number of pairs of atoms generated too close together quickly decreases and then levels off around $\lambda = 0.05$–0.1 (see ESI Fig. S12†). This observation is an indicator that including the bond distance discriminator is helpful for training the generator to create structures with more reasonable atom–atom distances. The generated structures were then relaxed using M3GNet, refer to ESI Fig. S13† for examples of structures before and after relaxation.

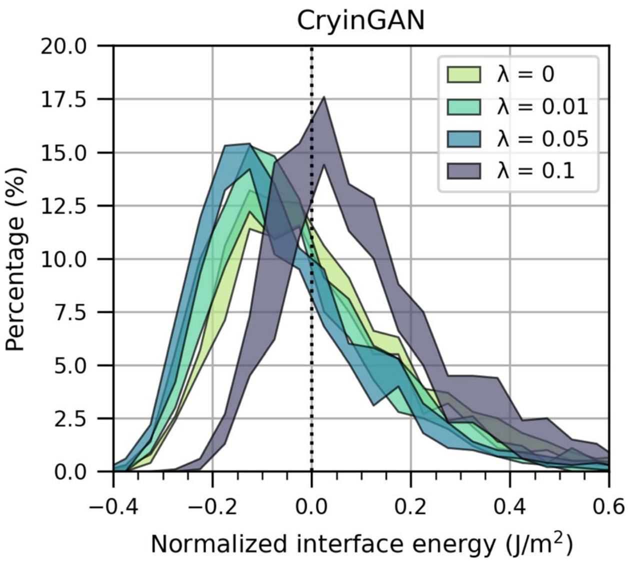

Fig. 7 Normalized interface energy distributions of structures generated using CryinGAN trained with different λ values (0, 0.01, 0.05, and 0.1). The generated structures were relaxed using M3GNet, and the interface energies shown are based on M3GNet-calculated energies. For each λ value, three separate models were trained, and the spread of the interface energy distributions is indicated by the shading. As λ increases, the interface energy distribution initially shifts to lower energies, then shifts to higher energies.

The objective was to obtain a model with the lowest interface energy distribution. Fig. 7 shows the interface energy distribution for the models trained with different λ. For each λ, three separate models were trained and the spread of the interface energy distributions is indicated by the shading (the distribution of all trained models without shading is provided in ESI Fig. S14†). The results indicate that as λ increases from 0 to 0.05, the interface energy distribution shifts to lower energies. However, as λ further increases to 0.1, the interface energy distribution shifts to higher energies. The coordination motif fingerprint distance between the generated and training structures also shows the same trend (see ESI Fig. S15†), where it decreases between $\lambda = 0$ and $\lambda = 0.05$, then increases when $\lambda > 0.05$. These observations show that optimal values of λ improve model performance, but excessive weight on the bond distance discriminator causes the generator to prioritize the bond distances too much over positioning the atoms correctly. The use of a second discriminator does slow down training compared to using only a single discriminator, requiring around twice as long to train the model for the same number of epochs. However, we found that two discriminators still outperform a single discriminator given the same amount of training time (see ESI Fig. S16†), justifying the benefits of the bond distance discriminator. Whereas around 34% of the generated structures failed to converge during M3GNet relaxation for $\lambda = 0$, only around 11% did not converge for $\lambda = 0.05$. These results show that with an appropriately tuned λ, the bond distance discriminator improves model performance.

We further considered a couple of alternative GAN architectures (refer to Fig. 5). CryinGAN-comb combines both discriminators into a single discriminator, circumventing the need to tune λ. CryinGAN-max uses max pooling in the discriminator, instead of average pooling as in CryinGAN. CryinGAN-mix uses the mix pooling operation proposed by Wang *et al.*,[58] where both max and average pooling operations are used together. The type of pooling operation affects the sampling sensitivity of the discriminator and the overall performance of the GAN. The sampling sensitivity describes how sensitive the discriminator is to changes in point density or the sampling pattern of the input point cloud. Max pooling was reported to cause lower sampling sensitivity than average pooling.[58] Overall, CryinGAN was found to outperform all of these alternative architectures, see ESI Supplementary Note 2† for further details.

The superior performance of CryinGAN over CryinGAN-max and CryinGAN-mix shows that higher sampling sensitivity is beneficial for learning atomic configurations. The sampling sensitivity can be further increased through the use of graph convolutions, where each atom is convoluted with its surrounding atoms and bonds. We attempted a graph convolutional discriminator by adapting the commonly used Crystal Graph Convolutional Neural Networks (CGCNN).[59] However, the training losses diverged and it was not possible to train a GAN that could generate useful structures (see ESI Fig. S17†). This result is consistent with the findings of Wang *et al.*,[58] whose graph convolutional GAN also failed to produce point clouds of 3D objects. Graph convolutions are highly sensitive to sampling, making them prone to overfocus on the sampling pattern of a point cloud instead of the overall structure. This overfocus bears similarity to the behavior observed here for CryinGAN at high λ values. For graph convolutions to be implemented in point-cloud-based GANs, we expect that a carefully designed architecture will be required to take advantage of its sampling sensitivity without destabilizing training.

2.5 Detailed evaluation of CryinGAN-generated interfaces

Here, we further evaluate CryinGAN to demonstrate that the generated structures are energetically and structurally similar to the training structures. We trained a CryinGAN model ($\lambda = 0.05$) for a higher number of epochs with a shorter interval between generator trainings (see Methods section for more details). Fig. 8 shows the interface energy distribution of the relaxed CryinGAN structures, compared to randomly generated structures (also relaxed). The shift of the interface energy distribution to the left shows that CryinGAN has learnt to generate low-interface-energy structures. With random generation, only around 48% of the structures had low interface energy, whereas with CryinGAN, the percentage is around 85%.

We filtered CryinGAN structures with low interface energy (≤ 0 J m^{-2}) for structural comparison with the training structures. We also compared the CryinGAN structures to a dataset of high-interface-energy structures (>0 J m^{-2}) that were randomly generated and relaxed. We analyzed the coordination motif fingerprints of the cations (Li and Sc) in the interface region coordinated to the anions (Cl and O), to determine the structural similarities/differences. The comparison for Li motifs showed smaller differences (see ESI Supplementary Note 3† for more details), so we focus the discussion here on Sc motifs.

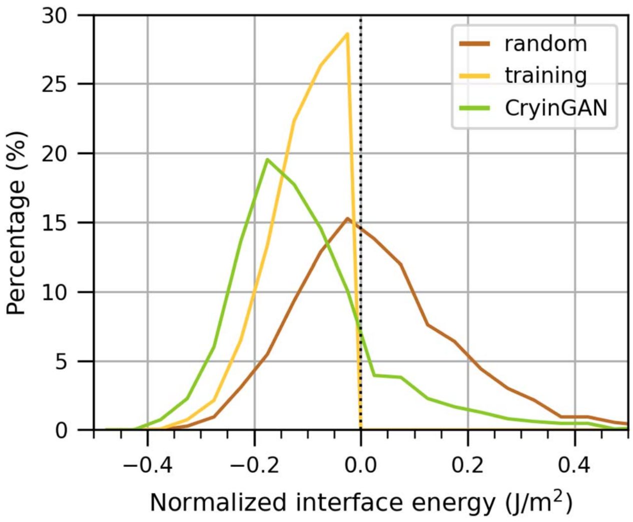

Fig. 8 Normalized interface energy distribution of structures generated using CryinGAN (green), compared against structures that were randomly generated (brown). All structures shown were relaxed using M3GNet followed by DFT calculations, and the energies shown are with respect to DFT-calculated energies. Randomly generated structures with (relaxed) normalized interface energy ≤ 0 J m^{-2} were used as the CryinGAN training dataset (yellow).

Table 4 Euclidean distance and cosine similarity between the average interface Sc site fingerprint of the training structures and the CryinGAN/high-interface-energy structures. The 95% bootstrap confidence intervals are shown in brackets

Dataset	Euclidean distance (95% CI)	Cosine similarity (95% CI)
CryinGAN	0.2791 (0.2500 to 0.3077)	0.9905(0.9886 to 0.9926)
High energy	0.7074 (0.6734 to 0.7429)	0.9316 (0.9250 to 0.9380)

Table 4 shows the Euclidean distance and cosine similarity between the average interface Sc fingerprint of the CryinGAN/high energy dataset and the training dataset. The CryinGAN dataset has a lower Euclidean distance and higher cosine similarity than the high energy dataset. These results indicate that the interface Sc atoms in the CryinGAN structures are more similar to the low-interface-energy training structures, than the high-interface-energy structures.

Fig. 9 shows the distribution of selected Sc coordination motifs, focusing on the motifs that show the largest differences across the datasets. Each motif is further subdivided based on the number of O atoms in the coordination shell of Sc (the distribution of all coordination motifs is provided in ESI Fig. S18b† for reference). Compared to the training dataset, the high energy dataset shows significantly higher percentages of motifs with no O atoms present, and lower percentages of motifs with O atoms. This finding indicates that the lower frequency of bonding between O (in LCO) and Sc leads to weaker interface binding and higher interface energy. In contrast, the CryinGAN dataset shows a higher frequency of Sc–O bonding than the training dataset, with higher percentages of motifs with 2 O atoms (trigonal bipyramidal and pentagonal pyramidal), and lower percentages of motifs with no O atoms (trigonal bipyramidal and octahedral). The average O bond count per Sc atom provided in ESI Fig. S19b† also shows that the high energy structures have fewer Sc–O bonds, whereas the CryinGAN structures have higher number of Sc–O bonds. This analysis shows that although the Sc coordination environments of the CryinGAN structures differ from the training structures due to a higher frequency of Sc–O bonding, the Sc–O bonds are still a feature of the low-interface-energy structures. We also analyzed the RDFs of the interface Li and Sc atoms. The RDFs confirm the trends revealed by the coordination motif analysis, in which the CryinGAN dataset shows higher similarity to the training dataset than the high energy dataset (see ESI Supplementary Note 4† for more details).

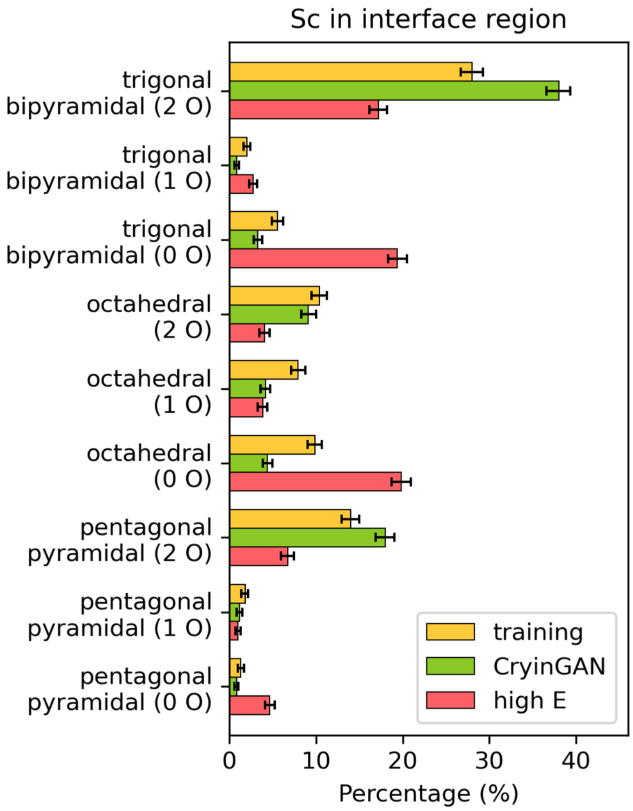

Fig. 9 Percentages of selected coordination motifs for Sc in the interface region. The motifs are subdivided based on the number of O atoms in the motif as indicated in brackets in the *y*-labels. The percentages of coordination motifs not shown are similar across the three datasets. Error bars represent 95% bootstrap confidence intervals.

2.6 Overall generative model comparisons and insights

The Dismai-Bench metrics of CryinGAN across all datasets are listed in Section 2.3. For the disordered interfaces, CryinGAN outperforms both U-Net diffusion models, despite that all three models use coordinate-based representations, and diffusion models are often reported to outperform GANs in image synthesis.[28,60,61] CryinGAN performs competitively with the graph diffusion models for the disordered interfaces, which is unexpected considering the lack of invariances and graph convolutions in CryinGAN. Nonetheless, the limited expressive power of point clouds does introduce challenges when CryinGAN is tasked with generating amorphous Si structures. However, CryinGAN is still able to generate the disordered alloy structures and reproduce the SRO distributions, while the other coordinate-based diffusion models struggled with this task. Apart from the 300 K dataset with narrow SRO, CryinGAN demonstrated similar metrics to the graph diffusion models (refer to Table 3). The results of this simple point-cloud-based GAN highlight the benefits and importance of robust generative model evaluation.

An overall ranking of all generative models based on their general performance on each Dismai-Bench dataset is visualized using a spider chart as shown in Fig. 10. The U-Net diffusion models struggled with most tasks in Dismai-Bench, while performing the best on the disordered interface dataset. Between the two, CrysTens performed marginally better on the disordered interfaces, probably as a result of the inclusion of distance matrices (albeit at a memory cost that scales with O(N^2), N being the number of atoms). Also, both UniMat and CrysTens require data augmentation to compensate for their lack of all invariances, further increasing their memory requirements by orders of magnitude. We expect that the reason for the better performance on the disordered interface dataset is

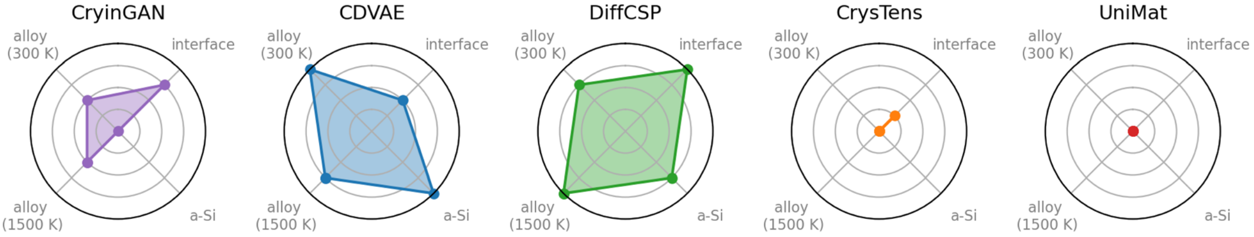

Fig. 10 Spider chart of generative model ranking based the models' general performance on each dataset. The models are ranked 1–5, where the outermost ring corresponds to rank 1 (best), and the center of the chart corresponds to rank 5 (worst).

that the interface structures have sorted atom orderings that make learning easier for these models with image-/video-like representation. For example, the first 72 atoms are always Li, with the first 9 Li atoms always in the interface region, the next 9 Li atoms in the top layer of LCO, *etc.* In contrast, the other datasets do not have such well-sorted atom orderings. When CrysTens and UniMat were trained on disordered interface structures with randomized atom orderings, they were unable to generate the interface structures (see ESI Fig. S20†). On the other hand, CryinGAN is permutationally invariant, so it does not rely on the sorted atom orderings to perform well on the interface dataset, and was also able to generate the crystalline alloy structures.

The graph diffusion models demonstrated the best performance overall. Comparing the two models, CDVAE performed better on the amorphous Si and the 300 K alloy datasets, whereas DiffCSP performed better on the 1500 K alloy and interface datasets. CDVAE stands out as the only model that is able to generate structures that are nearly fully relaxed. In use cases where relaxation/post-processing is not feasible/practical (*e.g.*, high-throughput materials discovery), CDVAE may show some advantages here. Although the graph models were successful in performing diffusion on atomic coordinates alone, they suffered when performing joint diffusion with atomic species or lattice parameters. When using atomic species diffusion, CDVAE was not able to generate the compositions correctly even though all Dismai-Bench datasets have fixed compositions. When using lattice diffusion, DiffCSP generated structures of lower quality. Such issues may be less severe when training on datasets with smaller number of atoms or lattice lengths, but future models can still improve upon their joint diffusion performance. To this end, Dismai-Bench provides a means for testing and evaluating different features/architectures (*e.g.*, training DiffCSP with teacher forcing of the lattice parameters; see ESI Fig. S6†).

Despite the superior performance of graph models on Dismai-Bench, there are also reports that less expressive generative models (*e.g.*, UniMat,[29] language models[31–33]) can show comparable or even better performance in discovering novel materials than graph models. These seemingly conflicting observations suggest a need to reconsider material discovery strategies. Materials discovery can be formulated as an exploration-exploitation problem. Graph models are superior at learning structural patterns, and should likely learn and exploit more of the training distribution during generation. On the other hand, less expressive models learn more noisy distributions, and can readily explore outside of the training distribution during generation. Considering the relatively small size of training datasets (~10^4 for the MP-20 dataset[26]) *vs.* the material space of possible compounds (~10^{12} possible combinations of quaternary compounds[62]), it is arguable that exploration is more important than exploitation in materials discovery. However, good performance for materials discovery should not be confused with the ability to learn well from training structures. While less expressive models may do well for the discovery of small structures, they may struggle when the size/complexity of the structures increases, as suggested by this work.

As the number of atoms and/or structural complexity of training samples increase(s), the importance of symmetry invariances and the expressive capability of generative models also grows. Graphs serve as one type of invariant representation among alternatives like smooth overlap of atomic positions (SOAP) vectors[63] and atom-centered symmetry functions (ACSFs).[64] Exploring these invariant representations further is crucial, even if reconstruction back to atomic coordinates is necessary. Importantly, reconstruction does not necessarily pose a fundamental barrier, as demonstrated by Fung *et al.*,[65] who achieved it through gradient-based optimization using automatic differentiation. The Dismai-Bench findings underscore the need for higher expressive power to model more complex structures, despite potential increases in computational and memory requirements. As generative modeling advances towards larger datasets and more complex structures, the development of models capable of parallelization will be a critical research direction.

In this work, we introduced Dismai-Bench as a novel method for assessing generative models and obtaining valuable insights into their performance. The development of generative models, and machine learning models in general, is iterative and often counterintuitive, as evidenced by the development of CryinGAN. Surprising findings include instances where an older generation GAN architecture has outperformed newer diffusion architectures, and that using two separate discriminators has yielded better results than using a single discriminator. Dismai-Bench proves effective in evaluating a generative model's capability to learn intricate structural patterns, particularly those present in disordered materials. Consequently, Dismai-Bench may contribute to improving generative models not only for ordered materials but also for disordered ones. However, Dismai-Bench's scope is limited in that it primarily assesses the accuracy of atomic positions, and does not evaluate a model's proficiency in learning compositions or lattice parameters, because Dismai-Bench operates by fixing the composition and lattice while varying atomic positions. Moreover, generative models incorporating symmetry constraints[66,67] may not leverage Dismai-Bench for evaluation unless these constraints can be disabled. Dismai-Bench marks a significant initial stride towards robustly evaluating generative models, and we anticipate that new complementary benchmarks and datasets will emerge in the future.

3 Conclusions

We developed a new benchmark for generative models, Dismai-Bench, which evaluates models on datasets of large, disordered materials exhibiting different degrees of structural and configurational disorder. Instead of training across different compositions and space groups, models are trained on datasets with fixed composition and structure type. By fixing the material system, Dismai-Bench circumvents the challenges of evaluating models based on newly generated materials that cannot be verified. Graph diffusion models (CDVAE & DiffCSP) were found to outperform coordinate-based diffusion models (CrysTens & UniMat) on Dismai-Bench, due to the invariant nature and higher expressive power of graphs. Additionally, we introduced CryinGAN, a novel GAN based on point clouds, that was developed by evaluating candidate architectures through direct comparisons between training and generated structures. Despite its simple architecture without symmetry invariances or complex components, CryinGAN outperformed the coordinate-based diffusion models and demonstrated competitiveness with the graph diffusion models. Dismai-Bench provides meaningful evaluation for comparing between architectures, understanding model strengths and weaknesses, and ultimately informing design choices. Building the next generation of generative models will rely on not only developing better architectures and representations, but also adopting better evaluation methods. We hope that this work will help advance future generative models for both ordered and disordered materials, and inspire the development of other new innovative benchmarks.

4 Methods

4.1 Dismai-Bench datasets

Dismai-Bench uses a total of six datasets. Each dataset contains a total of 1500 structures, split into 80% training and 20% validation data.

4.1.1 $Fe_{60}Ni_{20}Cr_{20}$ austenitic stainless steel. A cluster expansion Monte Carlo approach[35,36] was used to generate the datasets of FCC $Fe_{60}Ni_{20}Cr_{20}$ austenitic stainless steels. The CE model operates as a generalized Ising model[68] to describe the formation energy as a function of configuration. We adapted the CE model from ref. 36. A CE model with seven chemical dimers and three spin dimers was used to fit a dataset of FCC Fe–Ni–Cr alloys that were generated from spin-polarized DFT calculations. The CE model was fit using the least absolute shrinkage and selection operator (LASSO) and 10-fold cross-validation (CV) to determine the effective cluster interactions (ECI) values of the clusters. A comparison between the CE-calculated and DFT-calculated formation energies is shown in ESI Fig. S21a,† and the ECI values obtained by the LASSO CV fit are shown in ESI Fig. S21b.†

To generate the Dismai-Bench alloy structures, only the chemical terms of the CE model were used to obtain structures in a non-magnetic state. A 4 × 4 × 4 conventional FCC supercell, containing 256 atoms (154 Fe atoms, 51 Ni atoms, 51 Cr atoms), was used. We performed MC simulations in the canonical ensemble based on the Metropolis-Hastings algorithm.[69] Kawasaki dynamics[70] for atom swaps was applied to ensure that the composition of the system remained fixed. The alloy structures were sampled at temperatures of 300 K and 1500 K, where 10 independent MC simulations were initialized for each temperature. For every MC simulation, 50 000 passes were performed, and a MC snapshot was saved every 10 passes. For each temperature, 1500 structures were assembled (without any restriction on SRO distribution) as the wide SRO dataset. Another 1500 structures with SRO distribution within ±0.1 of the average SRO values were filtered as the narrow SRO dataset. Comparisons of the SRO between the $Fe_{60}Ni_{20}Cr_{20}$ alloy modelled in Dismai-Bench and the original $Fe_{56}Cr_{21}Ni_{23}$ alloy modelled in ref. 36 is shown in ESI Fig. S21c.† The Dismai-Bench alloy shows qualitatively similar SRO trends to the original alloy that was modelled with magnetism and a larger set of clusters.

4.1.2 Amorphous silicon. We adapted the amorphous silicon dataset from ref. 38. The original data consists of a 100 000-atom amorphous silicon structure generated through melt-quench molecular dynamics simulation.[38] We sliced the structure into smaller blocks with lattice parameters corresponding to 256-atom amorphous silicon structures. The lattice lengths of the blocks were calculated by linearly scaling the lattice lengths of the 100 000-atom structure to 256 atoms. The blocks were sliced at different locations to obtain a total of 1500 blocks. Blocks with <256 atoms had atoms added at random to low density regions, and blocks with >256 atoms had atoms removed at random from high density regions, until all blocks had 256 atoms. Atoms were added to or removed from the boundary of the blocks only (where they were sliced). The density was calculated by dividing each face of a block into 2 × 2 regions and counting the number of atoms in each region. The 1500 blocks were relaxed using a pre-trained SOAP-GAP[39] machine learning interatomic potential for Si. The structures were optimized using a conjugate gradient algorithm[71] through the Atomic Simulation Environment (ASE) package.[72] Only the atomic positions were allowed to relax, and the relaxations were stopped when the force on each atom was below 0.05 eV Å^{-1}.

4.1.3 $Li_3ScCl_6(100)$–$LiCoO_2(110)$ battery interface. To construct the interface structures, we chose the orientations of LCO(110), a Li fast-diffusing plane,[73] and LSC(100), a representative plane. The surfaces of the LSC(100) slab are polar, so half of the Cl atoms were moved from one surface to the other to neutralize the polarity (resulting in 'Tasker Type 2b' surfaces[74,75]). Lattice matching between the two slabs was carried out using the MPInterfaces package,[76] which implements the lattice matching algorithm proposed by Zur *et al.*[77] The configuration of the lattice-matched interface is given in ESI Table S4,† where the average lattice mismatch is 2.17%.

In a preliminary test calculation, the interface was constructed by simply placing the LCO(110) and LSC(100) slabs in contact with each other, and we obtained a DFT-relaxed structure similar to that depicted in Fig. 1. An interface region with disordered LSC atoms formed with a thickness of around 5 Å. Using this structure as reference, we then generated random interface structures using the CALYPSO package.[78–80] We used LCO(110) and LSC(100) slabs with 4 and 7 layers respectively. For each structure, an interface region thickness was randomly chosen between 4 and 6 Å, then the region was randomly populated with 3 formula units of LSC (9 Li, 3 Sc, and 18 Cl atoms). A random lateral displacement (in-plane direction parallel to the interface) was also applied to the LSC slab. A vacuum spacing of 14 Å was included in all the interface structures. Each interface structure has a total of 264 atoms.

We relaxed the randomly generated interface structures with the M3GNet interatomic potential. Only the atom positions were allowed to relax, where the lateral lattice vectors were fixed to the optimized values of the LCO slab (the elastic moduli of LCO[81] is significantly larger than LSC[82]). Similar to the amorphous Si dataset, the structures were optimized using a conjugate gradient algorithm through the ASE package, and the relaxations were stopped when the force on each atom was below 0.05 eV Å^{-1}. The normalized interface energies, $\tilde{\gamma}_{\text{int}}$, of the relaxed structures were calculated using the following equation:

$$\tilde{\gamma}_{\text{int}} = \frac{E_{\text{int}} - N(-4.78 \text{ eV per atom})}{A} \times \left(1.60218 \times 10^{-19} \text{ J eV}^{-1}\right) \tag{6}$$

where E_{int} is the total energy of the interface in eV, $N = 264$ is the total number of atoms, and A is the interface area. The interface energies are normalized such that structures with interface energy ≤ 0 J m^{-2} are considered to be low-interface-energy structures. 1500 low-interface-energy structures were filtered from the relaxed structures to form the dataset.

4.2 Generative models

4.2.1 CDVAE. We modified CDVAE such that atomic species denoising becomes an optional feature, since atomic species denoising caused CDVAE to generate incorrect compositions when trained on Dismai-Bench datasets. All CDVAE models were trained without atomic species denoising. The same hyperparameters, optimizer, and learning rate scheduler as those used for the MP-20 dataset[26] were applied. All models were trained using a batch size of 8 for 1000 epochs. Disordered interface and alloy structures were generated by running Langevin dynamics using 100 steps per noise level. For the amorphous Si structures, we were able to match the energy distribution of generated structures to the training structures (refer to example in ESI Fig. S22†), so we used 6–7 steps per noise level based on whichever setting gave the best match.

4.2.2 DiffCSP. We used 4 layers and 256 hidden states for all DiffCSP models. The exponential noise scheduler parameter, σ_T,[27] was set to 0.05 for amorphous Si, and 0.1 for disordered interfaces and alloys. The weight of the lattice cost was set to 0 for all models (*i.e.*, no lattice diffusion). All other hyperparameters were set to the default values, and we used the same optimizer and learning rate scheduler as the original implementation.[27] All models were trained using a batch size of 4 for 1000 epochs. All structures were generated using step size $\gamma = 1 \times 10^{-5}$ and 1000 time steps. Although we added the option to train DiffCSP with teacher forcing of the lattice parameters, this feature was not used for the DiffCSP models benchmarked in Section 2.3, which did not use any lattice diffusion.

4.2.3 CrysTens. CrysTens uses the Imagen[83] model, a cascaded diffusion model consisting of a base U-Net that generates a lower resolution image, followed by a super-resolution U-Net that upsamples the lower resolution image to a higher resolution. We used 64 base channels for both U-Nets, where the first U-Net generates 64 × 64 images and the second U-Net upsamples them to the full-size CrysTens images. The optimizer and all other hyperparameters were set to be the same as the original CrysTens implementation.[28] Each U-Net was trained using a batch size of 8 for 150 000 training steps. Structures were generated using 100 time steps, where the energy distribution of the generated structures had already converged at this setting (refer to ESI Fig. S23†). The CrysTens images were reconstructed into atomic structures using the ground truth lattice parameters. Althought the generated lattice lengths and angles from the CrysTens images were not used, their MAEs were small, around 0.02 Å and 0.02° respectively. The composition accuracy of the generated structures was 100%. The atomic coordinates of each atom was determined by constructing directional graphs using the coordinate pixels and the pairwise Δx, Δy, and Δz pixels, as described in ref. 28. Then, the coordinate predictions from the directional graphs were averaged. However, unlike the original implementation, we did not perform *k*-means clustering on the averaged atomic coordinates (and atomic species), since this was a post-processing step intended to manually "denoise" the reconstructed structures using an arbitrary choice of *k*.

For CrysTens, we augmented the disordered interface dataset by applying random permutations (shuffling) to the atoms in the interface region. CrysTens was found to perform poorly when data augmentation was performed by permutating all atoms (see ESI Fig. S20†). Each structure in the interface dataset was constructed with the same LSC and LCO slabs, as well as three formula units of LSC randomly generated in the interface region. As a result, the ordering of the atoms in the slabs is consistent across all structures, making it easier for CrysTens to learn these structures. Therefore, we only shuffled the ordering of the atoms in the interface region, where each structure was augmented 49 times. For amorphous Si and the alloys, which have no consistent atom orderings in their datasets, CrysTens was unable to generate meaningful structures even with data augmentation (see Fig. 3 and 4).

4.2.4 UniMat. The original UniMat[29] code is unfortunately not openly available, so we used an open-access implementation[53] of the 3D U-Net model[54] that the UniMat model was repurposed from. We used the hyperparameters as listed in ESI Table S5† for the U-Net. 4×4 video frames were used for the UniMat representation, which is the smallest frame size compatible with the U-Net model configuration. The Adam optimizer[84] was used with learning rate $= 1 \times 10^{-4}$, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. The U-Nets were trained using a batch size of 8 for 150 000 training steps. Structures were generated using 100 time steps, where the energy distribution of the generated structures had already converged at this setting (refer to ESI Fig. S24†). We did not include lattice parameters in the UniMat representation, and simply used the ground truth lattice parameters to reconstruct the structures (*i.e.*, no lattice diffusion). The average composition accuracy of the generated structures was 99.5%. Structures with incorrect compositions were considered failed structures and rejected. Similar to CrysTens, we augmented the disordered interface dataset by shuffling the atoms in the interface region, augmenting 49 times per structure. For amorphous Si and the alloys, UniMat was unable to generate meaningful structures even with data augmentation (see Fig. 3 and 4).

4.2.5 **CryinGAN.** The CryinGAN architecture is as shown in Fig. 6. CryinGAN was developed using a different interface dataset from the Dismai-Bench interface datatset, consisting of 1500 low-interface-energy structures relaxed using M3GNet followed by DFT calculations. This is to facilitate evaluation of the best performing model through comparisons between DFT-relaxed training and generated structures. Benchmarking of CryinGAN was still carried out by training models on the Dismai-Bench datasets. During model development, CryinGAN models were trained with different λ values to study the effect of λ on model performance. The models were trained with a batch size of 32 for 100 000 epochs. Adam optimizers with learning rate of 5×10^{-5} were used for the generator and discriminator(s). The generator was trained only once every 5 batches to help stabilize the training. CryinGAN-comb, CryinGAN-max, and CryinGAN-mix models were trained using the same procedure. CryinGAN-comb uses only a single combined discriminator, so no λ testing was required. CryinGAN-max and CryinGAN-mix were trained using $\lambda = 0$ and 0.05. For $\lambda = 0$, max pooling and mix pooling were used for the fractional coordinate discriminator of CryinGAN-max and CryinGAN-mix respectively. For $\lambda = 0.05$, max pooling and mix pooling were used for the bond distance discriminator of CryinGAN-max and CryinGAN-mix respectively, whereas average pooling was used for the fractional coordinate discriminator of both models. The various GAN models were compared using 1000 structures generated from each model.

We also tested a graph convolutional neural network architecture by adapting CGCNN[59] as the discriminator. Instead of predicting material properties, the output of CGCNN was used to estimate the Wasserstein distance between the training and generated structures. Default values for the hyperparameters were used, as provided in the code repository of CGCNN. The GAN was optimized using the Adam optimizer and a batch size of 32, where we tested learning rates of 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5}. We found that the losses diverged and the GAN was unable to generate meaningful structures. We did not further pursue the development of graph convolutional neural networks as discriminators, considering similar failures reported in literature for point cloud GANs.[58]

The best GAN architecture was found to be CryinGAN using both discriminators. For the disordered interface, $\lambda = 0.05$ was found to be optimum (see ESI Fig. S15†). For the alloys, $\lambda = 0.1$ was found to be optimum (see ESI Fig. S25†). For amorphous Si, CryinGAN was unable to generate meaningful structures (see Fig. 3), so it was not benchmarked for amorphous Si. Benchmark models were trained on the Dismai-Bench disordered interface and alloy datasets using $\lambda = 0.05$ and 0.1 respectively. For the disordered interface dataset, we did not perform any data augmentation. CryinGAN is permutationally invariant, and the model does not need to learn to generate rotated interface structures. Each structure in the dataset was originally generated with the LCO slab fixed and the LSC slab randomly displaced laterally (parallel to the interface). Training models on a dataset with lateral translation augmentations was found to slow down CryinGAN's learning, since it had to learn to generate structures with displaced LCO (and LSC) slabs. Therefore, we did not apply any translation augmentation as well. For the alloy datasets, we performed data augmentation by translating the structures by integer multiples of the unit cell lattice constant (3.6 Å). Each structure is a $4 \times 4 \times 4$ supercell, giving $4^3 - 1 = 63$ unique translations, so we augmented each structure 63 times. The disordered interface and alloy benchmark models were trained for 100 000 and 1500 epochs respectively, and models from the last epoch were used for benchmarking. All models were trained with a batch size of 32.

4.3 Dismai-Bench benchmarking

Benchmarking was performed by training all generative models on each dataset separately from scratch, such that the models only generate one type of structure at a time. For each generative model architecture, three separate models were trained for each dataset, and the benchmark metrics were averaged. 1000 structures were generated for each model to calculate the benchmark metrics.

4.3.1 **Disordered LSC–LCO interface.** The generated interface structures were first post-processed by moving apart atoms that were too close together using an iterative algorithm. In each iteration, the algorithm determines all unique pairs of atoms too close together, and increases the magnitude of their bond vectors. The algorithm repeats itself until all atomic distances are >1.5 Å. We set a maximum of 100 iterations, and any structure that still had atoms too close together after 100 iterations was rejected. The structures were then relaxed using the M3GNet interatomic potential, allowing only the atom positions to relax. The relaxations were stopped when the force on each atom was below 0.05 eV Å^{-1}. The percentage of failed structures was calculated.

For each successfully relaxed structure, the CrystalNNFingerprint[55] was calculated for the cations (Li, Co, and Sc), allowing only the anions (Cl and O) to be considered as neighbors. We also appended the fraction of Cl and O neighbors in each motif to the fingerprints, so that the fingerprints contain both chemical and coordination information. Average fingerprints were calculated by averaging the site fingerprints of each structure, then averaging across all structures. The Euclidean distances between the average fingerprint of the generated structures and the training structures were calculated.

4.3.2 Amorphous Si. The generated amorphous Si structures were post-processed to move apart atoms too close together, using the same procedure described for the disordered interfaces. The structures were then relaxed using the SOAP-GAP interatomic potential, allowing only the atom positions to relax. The relaxations were stopped when the force on each atom was below 0.05 eV Å^{-1}. The percentage of failed structures was calculated.

For each successfully relaxed structure, the CrystalNNFingerprint[55] was calculated for all Si atoms. The Euclidean distances between the average fingerprint of the generated structures and the training structures were calculated. The RDF of each structure was calculated using the vasppy[85] package. The RDFs were calculated between 0.0 Å and 10.0 Å using a bin width of 0.02 Å. The RDFs were averaged across structures, and the Euclidean distances between the average RDF of the generated structures and the training structures were calculated. The bond angles of each structure was also calculated. The neighbors of each atom were determined using the CrystalNN[55] algorithm. The bond angle distribution was calculated by binning the bond angles using a bin width of 0.5°. The bond angle distributions were averaged across structures, and the Euclidean distances between the average bond angle distribution of the generated structures and the training structures were calculated.

4.3.3 Disordered stainless steel alloy. The generated alloy structures were post-processed to remove atoms not on lattice sites. An atom was considered to be on a lattice site if it was within a 0.8 Å radius of the lattice site. Atoms were assigned to lattice sites iteratively, and if a lattice site already had an atom assigned to it, no other atom can be assigned to the site. Any structure with >50 atoms removed (∼20% of all atoms) was considered a failed structure and rejected. The percentages of failed structures and accepted structures with site vacancies were calculated.

The clusters (1 monomer and 7 dimers) of each structure were counted, where we considered up to the 7th neighbor shell. A fingerprint was constructed for each structure using the vector of conditional probabilities of observing each cluster (*e.g.*, the probability of observing a 1st nearest neighbor Cr–Fe dimer given two nearest neighbor sites). The Euclidean distances between the average cluster fingerprint of the generated structures and the training structures were calculated.

4.4 Detailed evaluation of CryinGAN-generated interfaces

We further evaluated the disordered interface structures generated by CryinGAN. We trained a CryinGAN model on the dataset of DFT-relaxed interface structures with $\lambda = 0.05$. The generator was trained once every 2 batches to speed up the training. We found the training to be stable at this frequency as shown by the Wasserstein distance plot in ESI Fig. S26a.† Although the Wasserstein distance plateaued early in the training, the percentage of (relaxed) low-interface-energy structures continued to gradually increase (see ESI Fig. S26b and c†). We stopped the training when the energy improvements have mostly plateaued. Structures were generated using the model from the last epoch, and relaxed using M3GNet followed by DFT calculations. Low-interface-energy structures were filtered from the relaxed structures.

Three datasets were used to perform the structural analysis: (1) CryinGAN-generated structures with low interface energy (≤0 J m^{-2}), (2) randomly generated structures with low interface energy (*i.e.*, the training structures), and (3) randomly generated structures with high interface energy (>0 J m^{-2}). Each dataset contains 1500 structures that were relaxed using M3GNet followed by DFT calculations. The coordination environment of atoms in the interface region was compared between the datasets. The coordination motif fingerprints of Li and Sc atoms in the interface region were calculated for each dataset and averaged. The Euclidean distance and cosine similarity between the average fingerprint of the CryinGAN/high energy dataset and the training dataset were calculated. To visualize the coordination motif distribution of the datasets, the most likely coordination motif of each interface Li/Sc atom was identified by first selecting the coordination number with the highest likelihood, then selecting the coordination motif of this coordination number with the highest local structure order parameter (LoStOP).[55] We only plotted coordination motifs that appeared in >1% of the interface Li/Sc atoms in the training structures. 95% bootstrap pivotal confidence intervals[86] were calculated using 1000 bootstrap samples with 1500 structures in each sample.

4.5 DFT calculations

All DFT calculations were performed using the Vienna *Ab initio* Simulation Package (VASP),[87–90] with the projector augmented-wave (PAW) method.[91,92] The Li (1s^2 2s^1), Sc (3s^2 3p^6 3d^2 4s^1), Co (3d^8 4s^1), Cl (3s^2 3p^5), and O (2s^2 2p^4) electrons were treated as valence electrons in the pseudopotentials. The generalized gradient approximation (GGA) with the Perdew–Burke–Ernzerhof (PBE) exchange-correlation functional[93] was used. The orbitals were expanded using a plane wave basis with cutoff energy of 520 eV for bulk structures, and 450 eV for interface structures (to lower computational cost). The DFT + *U* approach[94] was used to account for the electron localization of the Co-3d states, and we selected a *U* value of 4 eV as reported in other literature.[95]

We first performed structural relaxations on unit cells of LCO ($R\bar{3}m$) and LSC ($C2/m$[41]), allowing the cell shapes, cell volumes, and atom positions to relax, until the force on each atom was below 0.001 eV Å^{-1}. The Brillouin zone was sampled using a (9 × 9 × 1) gamma-centered *k*-point grid for LCO, and a (6 × 4 × 6) Monkhorst–Pack *k*-point grid for LSC. The relaxed unit cells were used to construct the LCO(110) and LSC(100) slabs as described in Section 4.1.3.

To generate the dataset for training the M3GNet interatomic potential, LSC(100)–LCO(110) interfaces were randomly generated as described in Section 4.1.3. A total of 350 structures were generated without mutual exchanges, and 600 structures were generated with mutual exchanges. Random mutual exchanges were performed between atoms in the interface region and the top layer of LCO, where we allowed up to 3 Sc ↔ Co, 3 Li ↔ Co, and 6 Cl ↔ O exchanges per structure. Structural relaxations were performed on all interface structures with the cell shapes and cell volumes fixed. The relaxations were performed in two stages. In the first stage, a kinetic energy cutoff of 374.3 eV was used, and only the atoms in the interface region were allowed to relax for 50 ionic steps. In the second stage, a kinetic energy cutoff of 450 eV was used, and all atoms were allowed to relax until the force on each atom was below 0.1 eV $Å^{-1}$. The Brillouin zone was sampled at the gamma point only for both stages.

For structures that have been pre-relaxed using M3GNet (after the M3GNet model was trained), DFT relaxations were performed without the first stage relaxation (second stage only). The interface dataset for Dismai-Bench was created by relaxing randomly generated structures with M3GNet only. The interface dataset for developing CryinGAN was created by relaxing randomly generated structures with M3GNet followed by DFT calculations. No mutual ion exchanges were performed when the structures were generated since the exchanges mostly led to high-interface-energy structures. The initial 950 structures that were relaxed using DFT calculations only were just used for training the M3GNet model, but not any generative model.

4.6 M3GNet

We trained the M3GNet interatomic potential[49] on the LSC–LCO interface structures to allow us to perform relaxations quickly. The dataset used for M3GNet training and evaluation included the 950 relaxed interface structures and 14 534 intermediate ionic steps from the DFT relaxations (second stage only). The intermediate steps were sampled starting from the 5th ionic step of every relaxation with an interval of 10 steps. The dataset was split into 80% training, 10% validation, and 10% test data. We used the Adam optimizer to optimize the loss function, L, as follows:

$$L = \mathrm{MSE_E} + \mathrm{MSE_F} + 0.1(\mathrm{MSE_S}) \tag{7}$$

where $\mathrm{MSE_E}$, $\mathrm{MSE_F}$, and $\mathrm{MSE_S}$ are the mean squared error of energy, force and stress respectively. We tested different learning rates and batch sizes, and trained M3GNet models for 24 hours each. We found that the losses plateaued within the given training time, and there was little difference in errors between the different hyperparameters (see ESI Table S6†). We chose the model with the smallest loss (learning rate = 0.001, batch size = 4) as the working interatomic potential of this work (see ESI Fig. S27† for the loss curves). The test set mean absolute errors for energy, force, and stress are 2.70 meV per atom, 20.9 meV $Å^{-1}$, and 0.0146 GPa respectively.

Table 5 Performance of M3GNet relaxations in reaching DFT convergence. Interface structures are first optimized using M3GNet relaxations followed by DFT relaxations. n_{steps} is the number of DFT ionic steps required to fully relax the M3GNet-optimized structures. $|\Delta E_{\text{M-opt}}|$ is the absolute energy difference between the M3GNet and DFT energies of the M3GNet-optimized structure. $|\Delta E_{\text{D-opt}}|$ is the absolute energy difference between the M3GNet energy of the M3GNet-optimized structure and the DFT energy of the final DFT-optimized structure. The mean and standard deviation for each quantity are listed

Structure type	n_{steps}		$\lvert\Delta E_{\text{M-opt}}\rvert$ (meV per atom)		$\lvert\Delta E_{\text{D-opt}}\rvert$ (meV per atom)	
	Mean	Std	Mean	Std	Mean	Std
Low energy	17.6	25.7	3.14	3.41	2.15	1.42
High energy	56.1	51.8	11.0	15.3	4.75	4.51
All	31.7	41.8	6.03	10.4	3.10	3.21

We compared interface structures that were pre-relaxed with M3GNet with their final structures after subsequent DFT relaxation. Table 5 shows the errors between the M3GNet- and DFT-calculated total energies. The errors are generally low even when compared to the final DFT-relaxed structures, showing that the M3GNet relaxations are able to give accurate predictions of the energy, and yield structures close to DFT convergence.

Data availability

The datasets and interatomic potentials used are available openly at https://doi.org/10.5281/zenodo.12710372. The Dismai-Bench benchmarking and generative model code used in this work are available at https://github.com/ertekin-research-group/Dismai-Bench. The CryinGAN code is available separately at https://github.com/ertekin-research-group/CryinGAN.

Author contributions

A. X. B. Y. and E. E. conceived the idea. A. X. B. Y. designed the benchmark, wrote the code, carried out the calculations/experiments, and performed the analysis. T. S. assisted with the alloy dataset preparation and code for cluster counting. E. E. supervised and guided the project. The manuscript was prepared by A. X. B. Y. and T. S. All authors reviewed and edited the manuscript.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This work was supported by the Defense Advanced Research Projects Agency (DARPA) HR00112220028. We also acknowledge partial financial support from the U. S. Army CERL, awarded under Grant No. W9132T-19-2–0008, and the US Department of Energy H2@Scale program, through award DE-EE0008832. This work made use of the Illinois Campus Cluster, operated by the Illinois Campus Cluster Program (ICCP) in conjunction with the National Center for Supercomputing Applications (NCSA) and supported by the University of Illinois at Urbana-Champaign. This work also utilized the Hardware-

Accelerated Learning (HAL) cluster,[96] supported by the National Science Foundation's Major Research Instrumentation program, grant #1725729, as well as the University of Illinois at Urbana-Champaign. This work also used Bridges-2 (ref. 97) at Pittsburgh Supercomputing Center through allocation MAT220011 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

Notes and references

1 D. M. Anstine and O. Isayev, *J. Am. Chem. Soc.*, 2023, **145**, 8736–8750.

2 A. S. Fuhr and B. G. Sumpter, *Front. Mater.*, 2022, **9**, 1–13.

3 A. Kadurin, A. Aliper, A. Kazennov, P. Mamoshina, Q. Vanhaelen, K. Khrabrov and A. Zhavoronkov, *Oncotarget*, 2017, **8**, 10883–10890.

4 A. Kadurin, S. Nikolenko, K. Khrabrov, A. Aliper and A. Zhavoronkov, *Mol. Pharm.*, 2017, **14**, 3098–3104.

5 S. Sinai, E. Kelsic, G. M. Church and M. A. Nowak, *Variational auto-encoding of protein sequences*, 2018.

6 J. Hoffmann, L. Maestrati, Y. Sawada, J. Tang, J. M. Sellier and Y. Bengio, *Data-Driven Approach to Encoding and Decoding 3-D Crystal Structures*, 2019.

7 J. Noh, J. Kim, H. S. Stein, B. Sanchez-Lengeling, J. M. Gregoire, A. Aspuru-Guzik and Y. Jung, *Matter*, 2019, **1**, 1370–1384.

8 A. Nouira, N. Sokolovska and J.-C. Crivello, *CrystalGAN: Learning to Discover Crystallographic Structures with Generative Adversarial Networks*, 2019.

9 Z. Ren, S. I. P. Tian, J. Noh, F. Oviedo, G. Xing, J. Li, Q. Liang, R. Zhu, A. G. Aberle, S. Sun, X. Wang, Y. Liu, Q. Li, S. Jayavelu, K. Hippalgaonkar, Y. Jung and T. Buonassisi, *Matter*, 2022, **5**, 314–335.

10 Y. Zhao, M. Al-Fahdi, M. Hu, E. M. D. Siriwardane, Y. Song, A. Nasiri and J. Hu, *Advanced Science*, 2021, **8**, 2100566.

11 C. J. Court, B. Yildirim, A. Jain and J. M. Cole, *J. Chem. Inf. Model.*, 2020, **60**, 4518–4535.

12 T. Long, N. M. Fortunato, I. Opahle, Y. Zhang, I. Samathrakis, C. Shen, O. Gutfleisch and H. Zhang, *npj Comput. Mater.*, 2021, **7**, 66.

13 M. F. Thorpe and L. Tichy, *Properties and Applications of Amorphous Materials*, Springer Dordrecht, 2001, vol. 9.

14 T. Yang, Y. L. Zhao, W. P. Li, C. Y. Yu, J. H. Luan, D. Y. Lin, L. Fan, Z. B. Jiao, W. H. Liu, X. J. Liu, J. J. Kai, J. C. Huang and C. T. Liu, *Science*, 2020, **369**, 427–432.

15 Y. Xie, J. Cai, Y. Wu, X. Hao, Z. Bian, S. Niu, X. Yin, Z. Pei, D. Sun, Z. Zhu, Z. Lu, D. Niu and G. Wang, *ACS Mater. Lett.*, 2021, **3**, 1738–1745.

16 Y. G. Chung, J. Camp, M. Haranczyk, B. J. Sikora, W. Bury, V. Krungleviciute, T. Yildirim, O. K. Farha, D. S. Sholl and R. Q. Snurr, *Chem. Mater.*, 2014, **26**, 6185–6192.

17 R. Thyagarajan and D. S. Sholl, *Chem. Mater.*, 2020, **32**, 8020–8033.

18 H. Zheng, E. Sivonxay, M. Gallant, Z. Luo, M. McDermott, P. Huck and K. A. Persson, *The ab initio amorphous materials database: Empowering machine learning to decode diffusivity*, 2024.

19 M. Kilgour, N. Gastellu, D. Y. T. Hui, Y. Bengio and L. Simine, *J. Phys. Chem. Lett.*, 2020, **11**, 8532–8537.

20 V. S. C. Kolluru, D. G. Unruh, J. T. Paul and M. K. Chan, *APS March Meeting*, 2023.

21 J. Guo, A. Mannodi-Kanakkithodi, F. G. Sen, E. Schwenker, E. S. Barnard, A. Munshi, W. Sampath, M. K. Y. Chan and R. F. Klie, *Appl. Phys. Lett.*, 2019, **115**, 1–5.

22 E. Schwenker, V. S. C. Kolluru, J. Guo, R. Zhang, X. Hu, Q. Li, J. T. Paul, M. C. Hersam, V. P. Dravid, R. Klie, J. R. Guest and M. K. Y. Chan, *Small*, 2022, **18**, 2102960.

23 M. Comin and L. J. Lewis, *Phys. Rev. B*, 2019, **100**, 094107.

24 B. Kim, S. Lee and J. Kim, *Sci. Adv.*, 2020, **6**, eaax9324.

25 S. Kim, J. Noh, G. H. Gu, A. Aspuru-Guzik and Y. Jung, *ACS Cent. Sci.*, 2020, **6**, 1412–1420.

26 T. Xie, X. Fu, O.-E. Ganea, R. Barzilay and T. Jaakkola, *International Conference on Learning Representations*, 2022.

27 R. Jiao, W. Huang, P. Lin, J. Han, P. Chen, Y. Lu and Y. Liu, *Crystal Structure Prediction by Joint Equivariant Diffusion*, 2024.

28 M. Alverson, S. G. Baird, R. Murdock, S.-H. Ho, J. Johnson and T. D. Sparks, *Digital Discovery*, 2024, **3**, 62–80.

29 M. Yang, K. Cho, A. Merchant, P. Abbeel, D. Schuurmans, I. Mordatch and E. D. Cubuk, *Scalable Diffusion for Materials Generation*, 2023.

30 C. Zeni, R. Pinsler, D. Zügner, A. Fowler, M. Horton, X. Fu, S. Shysheya, J. Crabbé, L. Sun, J. Smith, B. Nguyen, H. Schulz, S. Lewis, C.-W. Huang, Z. Lu, Y. Zhou, H. Yang, H. Hao, J. Li, R. Tomioka and T. Xie, *MatterGen: a generative model for inorganic materials design*, 2024.

31 D. Flam-Shepherd and A. Aspuru-Guzik, *Language models can generate molecules, materials, and protein binding sites directly in three dimensions as XYZ, CIF, and PDB files*, 2023.

32 L. M. Antunes, K. T. Butler and R. Grau-Crespo, *Crystal Structure Generation with Autoregressive Large Language Modeling*, 2024.

33 N. Gruver, A. Sriram, A. Madotto, A. G. Wilson, C. L. Zitnick and Z. Ulissi, *Fine-Tuned Language Models Generate Stable Inorganic Materials as Text*, 2024.

34 C. R. Qi, H. Su, K. Mo and L. J. Guibas, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

35 N. Kim, B. J. Blankenau, T. Su, N. H. Perry and E. Ertekin, *Comput. Mater. Sci.*, 2022, **202**, 110969.

36 T. Su, B. J. Blankenau, N. Kim, J. A. Krogstad and E. Ertekin, *Acta Mater.*, 2024, **276**, 120088.

37 J. M. Cowley, *Phys. Rev.*, 1965, **138**, A1384–A1389.

38 V. L. Deringer, N. Bernstein, G. Csányi, C. Ben Mahmoud, M. Ceriotti, M. Wilson, D. A. Drabold and S. R. Elliott, *Nature*, 2021, **589**, 59–64.

39 A. P. Bartók, J. Kermode, N. Bernstein and G. Csányi, *Phys. Rev. X*, 2018, **8**, 041048.

40 Y. Lyu, X. Wu, K. Wang, Z. Feng, T. Cheng, Y. Liu, M. Wang, R. Chen, L. Xu, J. Zhou, Y. Lu and B. Guo, *Adv. Energy Mater.*, 2021, **11**, 2000982.

41 J. Liang, X. Li, S. Wang, K. R. Adair, W. Li, Y. Zhao, C. Wang, Y. Hu, L. Zhang, S. Zhao, S. Lu, H. Huang, R. Li, Y. Mo and X. Sun, *J. Am. Chem. Soc.*, 2020, **142**, 7012–7022.

42 X. Li, J. Liang, X. Yang, K. R. Adair, C. Wang, F. Zhao and X. Sun, *Energy Environ. Sci.*, 2020, **13**, 1429–1461.

43 C. Wang, J. Liang, J. T. Kim and X. Sun, *Sci. Adv.*, 2022, **8**, eadc9516.

44 B. Zahiri, A. Patra, C. Kiggins, A. X. B. Yong, E. Ertekin, J. B. Cook and P. V. Braun, *Nat. Mater.*, 2021, **20**, 1392–1400.

45 T. Yang, Y. L. Zhao, W. P. Li, C. Y. Yu, J. H. Luan, D. Y. Lin, L. Fan, Z. B. Jiao, W. H. Liu, X. J. Liu, J. J. Kai, J. C. Huang and C. T. Liu, *Science*, 2020, **369**, 427–432.

46 G. H. Balbus, J. Kappacher, D. J. Sprouster, F. Wang, J. Shin, Y. M. Eggeler, T. J. Rupert, J. R. Trelewicz, D. Kiener, V. Maier-Kiener and D. S. Gianola, *Acta Mater.*, 2021, **215**, 116973.

47 D. Hudry, R. Popescu, D. Busko, M. Diaz-Lopez, M. Abeykoon, P. Bordet, D. Gerthsen, I. A. Howard and B. S. Richards, *J. Mater. Chem. C*, 2019, **7**, 1164–1172.

48 S. R. Spurgeon, T. C. Kaspar, V. Shutthanandan, J. Gigax, L. Shao and M. Sassi, *Adv. Mater. Interfaces*, 2020, **7**, 1901944.

49 C. Chen and S. P. Ong, *Nat. Comput. Sci.*, 2022, **2**, 718–728.

50 B. Deng, P. Zhong, K. Jun, J. Riebesell, K. Han, C. J. Bartel and G. Ceder, *Nat. Mach. Intell.*, 2023, **5**, 1031–1041.

51 E. A. Holm, G. S. Rohrer, S. M. Foiles, A. D. Rollett, H. M. Miller and D. L. Olmsted, *Acta Mater.*, 2011, **59**, 5250–5256.

52 A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder and K. A. Persson, *APL Mater.*, 2013, **1**, 1–11.

53 P. Wang, *imagen-pytorch*, 2024, https://github.com/lucidrains/imagen-pytorch.

54 J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi and D. J. Fleet, *Video Diffusion Models*, 2022.

55 N. E. R. Zimmermann and A. Jain, *RSC Adv.*, 2020, **10**, 6063–6081.

56 M. Arjovsky, S. Chintala and L. Bottou, *Proceedings of the 34th International Conference on Machine Learning*, 2017.

57 I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. Courville, *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

58 H. Wang, Z. Jiang, L. Yi, K. Mo, H. Su and L. J. Guibas, *Rethinking Sampling in 3D Point Cloud Generative Adversarial Networks*, 2020.

59 T. Xie and J. C. Grossman, *Phys. Rev. Lett.*, 2018, **120**, 145301.

60 P. Dhariwal and A. Nichol, *Adv. Neural Inf. Process. Syst.*, 2021, 8780–8794.

61 G. Müller-Franzes, J. M. Niehues, F. Khader, S. T. Arasteh, C. Haarburger, C. Kuhl, T. Wang, T. Han, T. Nolte, S. Nebelung, J. N. Kather and D. Truhn, *Sci. Rep.*, 2023, **13**, 12098.

62 D. Davies, K. Butler, A. Jackson, A. Morris, J. Frost, J. Skelton and A. Walsh, *Chem*, 2016, **1**, 617–627.

63 S. De, A. P. Bartók, G. Csányi and M. Ceriotti, *Phys. Chem. Chem. Phys.*, 2016, **18**, 13754–13769.

64 J. Behler, *J. Chem. Phys.*, 2011, **134**, 1–13.

65 V. Fung, S. Jia, J. Zhang, S. Bi, J. Yin and P. Ganesh, *Mach. learn.: sci. technol.*, 2022, **3**, 045018.

66 Y. Zhao, E. M. D. Siriwardane, Z. Wu, N. Fu, M. Al-Fahdi, M. Hu and J. Hu, *npj Comput. Mater.*, 2023, **9**, 38.

67 R. Jiao, W. Huang, Y. Liu, D. Zhao and Y. Liu, *Space Group Constrained Crystal Generation*, 2024.

68 J. M. Sanchez, F. Ducastelle and D. Gratias, *Phys. A*, 1984, **128**, 334–350.

69 N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, *J. Chem. Phys.*, 1953, **21**, 1087–1092.

70 K. Kawasaki, *Phys. Rev.*, 1966, **145**, 224.

71 *Conjugate Gradient Methods*, ed. J. Nocedal and S. J. Wright, Springer New York, New York, NY, 2006, pp. , pp. 101–134.

72 A. Hjorth Larsen, J. Jørgen Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. Bjerre Jensen, J. Kermode, J. R. Kitchin, E. Leonhard Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. Bergmann Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng and K. W. Jacobsen, *J. Phys.: Condens. Matter*, 2017, **29**, 273002.

73 P. J. Bouwman, B. A. Boukamp, H. J. M. Bouwmeester and P. H. L. Notten, *J. Electrochem. Soc.*, 2002, **149**, A699.

74 P. W. Tasker, *J. Phys. C: Solid State Phys.*, 1979, **12**, 4977–4984.

75 J. D. Gale and A. L. Rohl, *Mol. Simul.*, 2003, **29**, 291–341.

76 K. Mathew, A. K. Singh, J. J. Gabriel, K. Choudhary, S. B. Sinnott, A. V. Davydov, F. Tavazza and R. G. Hennig, *Comput. Mater. Sci.*, 2016, **122**, 183–190.

77 A. Zur and T. C. McGill, *J. Appl. Phys.*, 1984, **55**, 378–386.

78 Y. Wang, J. Lv, L. Zhu and Y. Ma, *Phys. Rev. B*, 2010, **82**, 094116.

79 Y. Wang, J. Lv, L. Zhu and Y. Ma, *Comput. Phys. Commun.*, 2012, **183**, 2063–2070.

80 B. Gao, P. Gao, S. Lu, J. Lv, Y. Wang and Y. Ma, *Sci. Bull.*, 2019, **64**, 301–309.

81 E. J. Cheng, N. J. Taylor, J. Wolfenstine and J. Sakamoto, *J. Asian Ceram. Soc.*, 2017, **5**, 113–117.

82 M. Jiang, S. Mukherjee, Z. W. Chen, L. X. Chen, M. L. Li, H. Y. Xiao, C. Gao and C. V. Singh, *Phys. Chem. Chem. Phys.*, 2020, **22**, 22758–22767.

83 C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, T. Salimans, J. Ho, D. J. Fleet and M. Norouzi, *Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding*, 2022.

84 D. P. Kingma and J. Ba, *International Conference on Learning Representations*, 2015.

85 B. J. Morgan, *vasppy*, 2021, https://pypi.org/project/vasppy/.

86 L. Wasserman, in *The Bootstrap*, ed. L. Wasserman, Springer New York, New York, NY, 2004, pp. 107–118.

87 G. Kresse and J. Hafner, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1993, **47**, 558–561.

88 G. Kresse and J. Hafner, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1994, **49**, 14251–14269.

89 G. Kresse and J. Furthmüller, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1996, **54**, 11169–11186.

90 G. Kresse and J. Furthmüller, *Comput. Mater. Sci.*, 1996, **6**, 15–50.

91 P. E. Blöchl, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1994, **50**, 17953–17979.

92 G. Kresse and D. Joubert, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1999, **59**, 1758–1775.

93 J. P. Perdew, K. Burke and M. Ernzerhof, *Phys. Rev. Lett.*, 1996, **77**, 3865–3868.

94 S. L. Dudarev, G. A. Botton, S. Y. Savrasov, C. J. Humphreys and A. P. Sutton, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1998, **57**, 1505–1509.

95 C. Wang, J. Liang, M. Jiang, X. Li, S. Mukherjee, K. Adair, M. Zheng, Y. Zhao, F. Zhao, S. Zhang, R. Li, H. Huang, S. Zhao, L. Zhang, S. Lu, C. V. Singh and X. Sun, *Nano Energy*, 2020, **76**, 105015.

96 V. Kindratenko, D. Mu, Y. Zhan, J. Maloney, S. H. Hashemi, B. Rabe, K. Xu, R. Campbell, J. Peng and W. Gropp, *Practice and Experience in Advanced Research Computing*, New York, NY, USA, 2020, pp. , pp. 41–48.

97 S. T. Brown, P. Buitrago, E. Hanna, S. Sanielevici, R. Scibek and N. A. Nystrom, *Practice and Experience in Advanced Research Computing*, New York, NY, USA, 2021.