Generative adversarial networks and diffusion models in material discovery

Michael Alverson[1*], Sterling G Baird[1], Ryan Murdock[1], (Enoch) Sin-Hang Ho[2], Jeremy Johnson[2] and Taylor D Sparks[1]

[1]Department of Materials Science & Engineering, University of Utah, Salt Lake City, 84112, Utah, USA.
[2]Department of Chemistry & Biochemistry, Brigham Young University, Provo, 84602, Utah, USA.

*Corresponding author(s). E-mail(s): michaeldavidalverson@gmail.com;
Contributing authors: sterling.baird@utah.edu; rynmurdock@gmail.com; enochho201315@gmail.com; jjohnson@chem.byu.edu; sparks@eng.utah.edu;

Abstract

The idea of materials discovery has excited and perplexed research scientists for centuries. Several different methods have been employed to find new types of materials, ranging from the arbitrary replacement of atoms in a crystal structure to advanced machine learning methods for predicting entirely new crystal structures. In this work, we pursue three primary objectives. I) Introduce `CrysTens`, a crystal encoding that can be used in a wide variety of deep learning generative models. II) Investigate and analyze the relative performance of Generative Adversarial Networks (GANs) and Diffusion Models to find an innovative and effective way of generating theoretical crystal structures that are synthesizable and stable. III) Show that the models that have a better "understanding" of the structure of `CrysTens` produce more symmetrical and realistic crystals and exhibit a better apprehension of the dataset as a whole. We accomplish these objectives using over fifty thousand Crystallographic Information Files (CIFs) from Pearson's Crystal Database.

1 Introduction

Materials discovery is an enormous open problem in the field of materials informatics and a wide variety of techniques have been used to address it. Some methods include large-scale combinatorial synthesis simulations [1, 2] to create crystals that are synthesizable and stable. However, in order for combinatorial synthesis methods to be successful, thousands or millions of compounds must be generated in order to identify useful candidates. Furthermore, known materials may only account for a minuscule fraction of the possible number of synthesizable and stable crystals [3]. Discovering new structures in an efficient and effective way requires a method of processing enormous amounts of data, quickly identifying patterns that are present within the dataset, and extrapolating those patterns outside of the dataset so that new materials can be discovered. Generative models have been implemented for a variety of topics, including composition [4, 5, 6, 7, 8, 9], molecules [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23], and crystal structures [24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41]. In this work, we focus on crystal structure generative models.

Crystal generative models typically use variational auto-encoders (VAEs) [27, 31, 36, 42, 38], generative adversarial networks (GANs) [30, 32, 38, 43, 41], genetic algorithms [28], or substitution [29]. One example of a crystal generative model based on VAE is the Fourier-Transformed Crystal Properties (FTCP) representation [42]. The FTCP representation is a crystal representation that not only incorporated both chemical and structural crystal properties but was invertible as well (property → chemistry + structure). When combined with a jointly trained VAE model, a probabilistic property-structured latent space was obtained that allowed for the generation of novel crystals with user-defined properties. Another important work in this area of study is the Crystal Diffusion Variational Autoencoder (CDVAE) [40]. By leveraging a diffusion process that pushes atomic coordinates to lower energy states and iterates atom types to satisfy bonding preferences, CDVAE significantly outperforms past attempts to perform material generation. Furthermore, CDVAE was capable of optimizing crystals for a given property and reconstructing a material from its latent space representation. Many examples exist for GANs as well. Constrained crystals deep convolutional generative adversarial network (CCDCGAN) [32] uses a post-processing filtering criteria to remove compounds with large formation energies based on a cutoff threshold. In another example, crystal structure is represented in a lean fashion by a set of atomic coordinates and unit cell parameters, with significantly reduced memory requirements compared to a voxel-based generative model, iMatGen [36]. Physics Guided Crystal Generative Model (PGCGM) [41] expands on CubicGAN [43] while using base site atoms with physics-guided loss functions. For example, structures with atoms that are too crowded or too far apart are penalized via two terms in the loss function. We also note that normalizing flows, another promising direction, have been used in some crystal structure contexts [39, 44] including an application to molecular crystals [44].

In this work, we compare the use of Generative Adversarial Networks and Diffusion Models in unstructured crystal generation. We illustrate the shortcomings of GANs in this space and show that Diffusion Models may offer a very promising alternative to previous methods. Furthermore, we introduce a standardized crystal embedding representation (`CrysTens`) that can be used in a wide array of different models.

2 Background

Creating an effective method of materials discovery is an extraordinarily important issue to address in the realm of materials science. Efficient materials discovery could revolutionize material science by not only yielding new materials, but also by providing new insights into the different ways that crystal structures *can* form. Perhaps the most impactful outcome would be in the area of inverse design wherein materials are tailor designed to meet specific property criteria. Generating new stable crystal structures has proven a stubbornly difficult task, let alone to do so with targeted properties. An outstanding challenge in this area has been the periodic nature of crystal structures. Only recently were representation approaches developed that encode both the symmetry and composition information contained within the Crystallographic Information File (CIF) [45, 46, 47, 48].

New representations that capture chemistry (composition) and structure (periodicity) are important because they then allow us to utilize machine learning algorithms to identify and exploit patterns in data. Even the most experienced materials scientist domain expert would be unable to fully comprehend and leverage all the patterns in high-dimensional materials data for hundreds of thousands of crystal structures, and then use the discovered patterns to generate novel materials. Instead, scientists have relied on very low-order approximations and simplifications for generating new materials. Or, alternatively, they have relied on screening down lists of already discovered materials to identify those candidates that most closely match the desired properties using empirical relationships or computational materials science techniques such as molecular dynamics, Density Functional Theory (DFT), etc. However, given the immense size of the chemical space that is believed to exist (10^{60} materials) and our current microscopic subset of known materials (10^5 - 10^6 materials) [3], it is unlikely that screening efforts alone are sufficient to find the new materials necessary to answer society's most pressing technological needs. Therefore it is abundantly clear that an intuitive and efficient method is needed to scan over the vast regions of untapped chemical space and select the groupings of materials it deems as stable and synthesizable.

Fortunately, machine learning (ML) and more specifically deep learning (DL) methods, have emerged as powerful complements to the human capacity for materials design [49, 50]. Within the field of material discovery, generative ML models are currently being investigated by a wide variety of research teams [51, 52, 45, 53]. Previously, the two most common types of generative ML models were the VAE and GANs. VAEs attempt to encode a sample of data into a lower dimensional latent space. The encoded samples are then decoded from latent space into potential new samples. However, VAEs differ from traditional auto-encoders because they are simultaneously attempting to structure the latent space according to a predefined probability distribution. This makes VAEs an exciting candidate for efficient inverse design because materials that exhibit ideal properties may be located at the intersection of each property within latent space [54, 55].

The other common generative ML model for material discovery is GANs. GANs differ from VAEs in several key ways. First, GANs are composed of two separate neural network architectures: the generator and the discriminator. The generator's task is to create realistic samples of whatever data distribution it is trying to model. In the case of this work, the generator is attempting to create realistic crystal structures. The discriminator's job is to differentiate between samples that are real (taken from the original dataset) or fake (created by the generator). The discriminator and generator train against each other, continually improving until training is finished and the generator can be separated and used to create realistic samples. By using an adversarial approach, the generator can construct its own probability distribution of the data instead of requiring a pre-defined probability distribution like those used within VAEs. Although the training of GANs is game-theoretic in nature, they are not guaranteed to converge to a Nash Equilibrium [56] which can lead to performance issues. A more advanced variation of the traditional GAN is included in this work known as the Wasserstein GAN or WGAN. The differences between WGANs and GANs are discussed later [57, 58].

Recently, however, with the success of OpenAI's DallE-2 [59] and Google's Imagen [60], Diffusion Models have quickly risen to achieve state-of-the-art performance for many types of generative modeling. Diffusion Models are inspired by non-equilibrium thermodynamics and operate by gradually destroying input training data by adding Gaussian noise (forward process) only to learn the transformations necessary to reconstruct each sample (backward process). By continually repeating this process, and incrementally adding more noise in each iteration, fully trained Diffusion Models are able to completely reconstruct a data sample from nothing more than noise. Diffusion Models take longer to train than GANs but do not suffer from many of the same deficiencies that GANs do such as mode collapse and extreme instability.

3 Methods

3.1 `CrysTens` Representation

We chose to use Pearson's Crystal Data (PCD) as the primary dataset of CIFs for this work. PCD contains over 140,000 unique CIFs, however with the constraints of `CrysTens`, the crystal embedding representation explained below (See Fig. 3 and Fig. 4), we were left with 53,856 CIFs. Each CIF is used to create a `CrysTens` and these tensors are concatenated together to form our training set.

CIFs contain a simple body of text that entirely capture the fundamental chemistry and structure of a crystal structure. Software such as `VESTA` takes CIFs as input and outputs useful and aesthetically pleasing crystal visualizations that can aid in crystal chemistry research and education. In order for `VESTA` to create such visualizations, there are several key attributes that are needed within CIFs [61]. First, the lattice parameters and their angles with respect to one another are needed to establish the periodicity inherent in crystalline lattices. These lattice parameters also create the three-dimensional "bounding box" for the repeating unit cell and are generally represented with the variables a, b, and c with lattice angles α, β, and γ. Following the lattice information, the space group number is used to indicate which space group a particular crystal structure belongs to. The space group essentially encapsulates the symmetry properties of a given crystal structure or put more formally the space groups summarize the total number of three-dimensional patterns that are found in crystal structures [62]. The final information is the basis describing the arrangement of atoms associated with each lattice point. The basis allows us to distinguish different crystal structures having unique chemistries and atomic positions while retaining identical symmetry. The basis and the symmetry operations are combined to generate the exact atomic positions for all atoms within the unit cell.

With all of the crystal information organized within a CIF text file, `VESTA` is able to create visualizations as seen in Fig. 1 that assist materials scientists in determining the structural components of a given crystal. Since the process of experimentally solving atomic positions is more challenging than solving the lattice parameters and space group, not all CIF entries include the basis. As a result, from the 140,000 CIF entries, only approximately 50,000 CIFs contain all the required information needed for our `CrysTens` representation and satisfy the representations' constraints. `Pymatgen`, a materials informatics Python library, was used to extract relevant information from each CIF during `CrysTens` construction using a Python programming interface and the `Pymatgen` Structure object (See Fig. 2) [63].

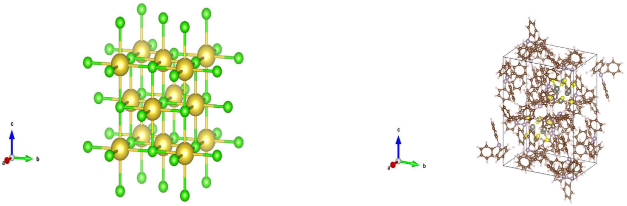
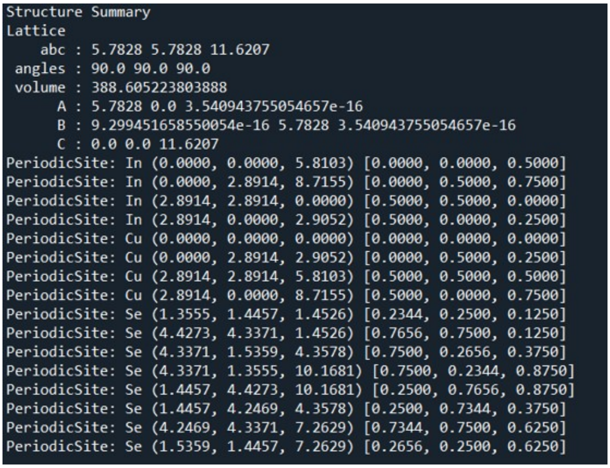


Fig. 1: (left) A `VESTA` visualization of $NaCl$
(right) A `VESTA` visualization of $W_6S_8(PC_{18}H_{15})_6(C_6H_6)$ [61].



```
Structure Summary
Lattice
    abc : 5.7828 5.7828 11.6207
 angles : 90.0 90.0 90.0
 volume : 388.605223803888
      A : 5.7828 0.0 3.540943755054657e-16
      B : 9.299451658550054e-16 5.7828 3.540943755054657e-16
      C : 0.0 0.0 11.6207
PeriodicSite: In (0.0000, 0.0000, 5.8103) [0.0000, 0.0000, 0.5000]
PeriodicSite: In (0.0000, 2.8914, 8.7155) [0.0000, 0.5000, 0.7500]
PeriodicSite: In (2.8914, 2.8914, 0.0000) [0.5000, 0.5000, 0.0000]
PeriodicSite: In (2.8914, 0.0000, 2.9052) [0.5000, 0.0000, 0.2500]
PeriodicSite: Cu (0.0000, 0.0000, 0.0000) [0.0000, 0.0000, 0.0000]
PeriodicSite: Cu (0.0000, 2.8914, 2.9052) [0.0000, 0.5000, 0.2500]
PeriodicSite: Cu (2.8914, 2.8914, 5.8103) [0.5000, 0.5000, 0.5000]
PeriodicSite: Cu (2.8914, 0.0000, 8.7155) [0.5000, 0.0000, 0.7500]
PeriodicSite: Se (1.3555, 1.4457, 1.4526) [0.2344, 0.2500, 0.1250]
PeriodicSite: Se (4.4273, 4.3371, 1.4526) [0.7656, 0.7500, 0.1250]
PeriodicSite: Se (4.3371, 1.5359, 4.3578) [0.7500, 0.2656, 0.3750]
PeriodicSite: Se (4.3371, 1.3555, 10.1681) [0.7500, 0.2344, 0.8750]
PeriodicSite: Se (1.4457, 4.4273, 10.1681) [0.2500, 0.7656, 0.8750]
PeriodicSite: Se (1.4457, 4.2469, 4.3578) [0.2500, 0.7344, 0.3750]
PeriodicSite: Se (4.2469, 4.3371, 7.2629) [0.7344, 0.7500, 0.6250]
PeriodicSite: Se (1.5359, 1.4457, 7.2629) [0.2656, 0.2500, 0.6250]
```

Fig. 2: A CIF represented as a `Pymatgen` Structure object.

Finding a concise, efficient, and structurally informative representation for each CIF was an important step in the process of crystal structure generative modeling. After many iterations, including just simply listing the crystal structure parameters in a list, we eventually decided upon a representation that captured both the intricate parameters of each crystal as well as their interatomic components. The representation is a tensor of shape 64x64x4 and can be visualized in Fig. 3 and Fig. 4. The top twelve rows and leftmost twelve columns are symmetrical and list out all of the CIF-extracted information of a given crystal. The top (or leftmost) list is the atomic number of each of the atoms present in the crystal. They are listed from left to right (or top to bottom) until either fifty-two spots have been filled or the crystal has run out of atoms to place. If there is leftover space in the representation, zeros will be filled in for the remainder. Structures with more than 52 atoms in the basis are excluded. The same process is repeated for the three fractional coordinates (x, y, and z), the three lattice parameters (a, b, and c), the three lattice angles (α, β, and γ), and the space group number. Finally, a padding layer is inserted to separate the CIF-extracted information from the interatomic portion of the representation. For the first layer of the representation, the bottom-right 52x52 matrix is used to encode a pairwise distance matrix that relates each of the atoms together by their Euclidean distance. Within the latter three layers of the representation, a distance graph for each of the dimensions is represented to show the uni-dimensional relative distance between each atom.

The motivation behind the structure of this representation is to highlight the major components of crystal structures. There is a structural component that encodes symmetry, basis, and lattice information and an interatomic component that encodes the relative distances between atoms into the representation itself. With only the interatomic distances of a given crystal, it would be impossible to reconstruct the atomic numbers, space group, and various lattice parameters. Likewise, with only the structural components, it may be challenging for a convolution-based generative modeling algorithm to encode the relationships between various components which may lead to generated structures with unreasonable interatomic distances.

There are questions that naturally arise when one is first introduced to this representation. One of which is, “Why is each layer 64x64?” The reason that we selected 64 as the length and width of each of our layers is because of the commonly used deep learning heuristic of selecting powers of two. It could easily have been another value, however, the focus of this work is to show the structure of the `CrysTens` representation rather than the numerical intricacies or optimization of the representation. Furthermore, since 64 was selected and there are eleven different parameters plus a padding layer, only crystal structures with 52 atoms or less were selected for training and analysis. Another question that may arise is “Why use four layers?” In this work, our goal was to fully capture all of the different structural aspects of a given crystal which is why we redundantly included both the pairwise distance matrix and the three-dimensional graphs, however, the representation can be changed to fit given constraints. We anticipate that future work could add layers to encode aspects related to chemistry such as the Oliynyk, magpie, mat2vec vector constituents in order to ensure realistic atom assignment.

3.2 Crystal Reconstruction

Once we have trained a model with the `CrysTens` representation, it would be desirable to transform the output of each model at inference time back into a CIF for visualization and analysis purposes. The process of transforming back from the `CrysTens` representation can also provide insights into the level of "understanding" of the representation itself that each model displays. `CrysTens` representations generated from CIF files are by definition symmetric, but `CrysTens`' that are output from generative models are not necessarily symmetric due to the stochastic nature of deep learning algorithms. Instead, we found that even our most symmetrical generated `CrysTens`' had a small amount of noise that was mitigated by the redundancy of our representation.

Given a generated `CrysTens`, there will be a number of atoms that it will have predicted to be present in a crystal. We will refer to this value as A, and $0 < A \leq 52$ due to the nature of the current `CrysTens` representation. If the generative model was able to understand the symmetrical relationship within the `CrysTens` representation, then it will have A non-empty columns from left to right, and A non-empty rows from top to bottom for each of the four layers. Similarly, the single-valued structure parameters such as the lattice parameters (a, b, c), the lattice angles (α, β, γ), and the space group number will be repeated $2A$ times for each layer, a total of $8A$ occurrences. These values can be averaged to find the value that will ultimately be used when generating the output CIF. The variance among these repeated values is a good indicator of whether a given model "understands" the structure of the representation. If almost all $8A$ space group numbers are between 219 and 221, then there is a good chance that the model intended on a space group number of 220 and knew where to place these values. Generated `CrysTens` representations with small variances between values that are meant to be the same tended to produce more symmetrical and better-looking crystals, shown in **Results**. However, if there is a space group number spread between 110 and 220 there is a very good chance that the model is not able to pick up on the relationship between the different space group rows and columns, and the quality of the finalized crystal is likely to reflect that. Values that do not represent a single value throughout the entire `CrysTens` such as atomic number or fractional coordinates are instead only repeated a total of 8 times (two times for each layer) because each one corresponds to a single atom. The same variance check can be performed on each of these values as well.

However, the single best indicator of model performance comes from the use of the directional graphs (Layers 2 - 4). Each spot in the directional graphs represents the relative difference between two atoms in all three dimensions. Therefore, with the knowledge of one atom and its relationship to another atom with the direction graph, the second atom's location can be deduced. Therefore, if we have 10 atoms predicted in a `CrysTens` representation, each with their respective fractional coordinates and a directional graph relating each of them, we can have 10 "guesses" as to where a given atom is supposed to be in the crystal (one guess from its own x, y, z coordinates and nine guesses from the relative distances of other atoms). This process is highlighted in two dimensions in Fig. 5. If there is a high degree of consistency between all of the coordinate predictions, the crystal that is produced tends to be more symmetrical and realistic. The average of the coordinate predictions for each atom is ultimately chosen as the point where a given atom is placed during the CIF reconstruction. Furthermore, with these reconstructed coordinates, a pairwise distance matrix can be created and compared to the generated pairwise distance matrix in Layer 1 as an additional metric for model performance.

We are now left with an averaged value for the lattice parameters, lattice angles, and the space group number. We also have an average value for each of the coordinates of a given atom. In order to produce clean crystals that do not reflect the noise of the generative model, we added a few more post-processing steps.

As far as the atomic numbers are concerned, we found that even when there was low variance within the predicted atomic numbers we could still observe lists of atomic numbers such as $(12.2, 12.6, 12.4, 12.5, 7.7, 7.4, 8.4, 8.5)$. When rounded to the nearest atomic number, this list would reflect $(12, 13, 12, 13, 8, 7, 8, 9) \rightarrow$ (Mg, Al, Mg, Al, O, N, O, F) instead of the list that would correspond to rule of parsimony: $(12, 12, 12, 12, 8, 8, 8, 8) \rightarrow$ (Mg, Mg, Mg, Mg, O, O, O, O). In order to rectify this inconsistency, we elected to use K-Means Clustering with K = 3. For similar reasons, we chose to use K-Means Clustering with K = 6 - 12 for the coordinate values of x, y, and z across the different atoms. Now we could create a list of elements in the CIF as well as another list corresponding to their atomic positions. Using the averaged values from above, a `Pymatgen` Lattice object can be constructed which is then used in conjunction with the element and coordinate lists to create a `Pymatgen` Structure object. The Structure object can be used to create a CIF.

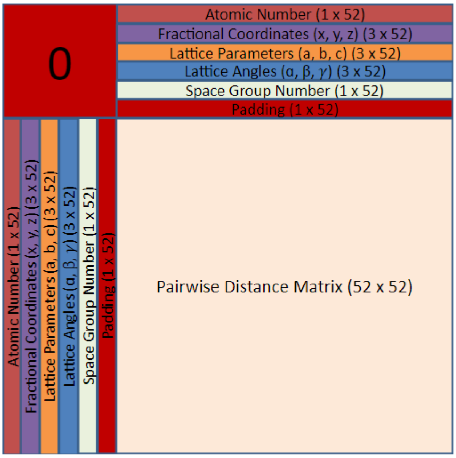


Fig. 3: Layer 1 of the crystal representation containing symmetrical rows and columns for structural information and a pairwise distance matrix with relative distances between all atoms in the basis.

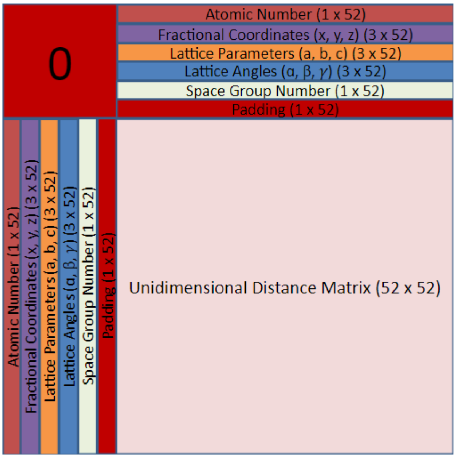


Fig. 4: Layers 2 - 4 of the crystal representation containing the same symmetrical rows and columns for structural information, but also having a directional graph for each dimension: x, y, and z.

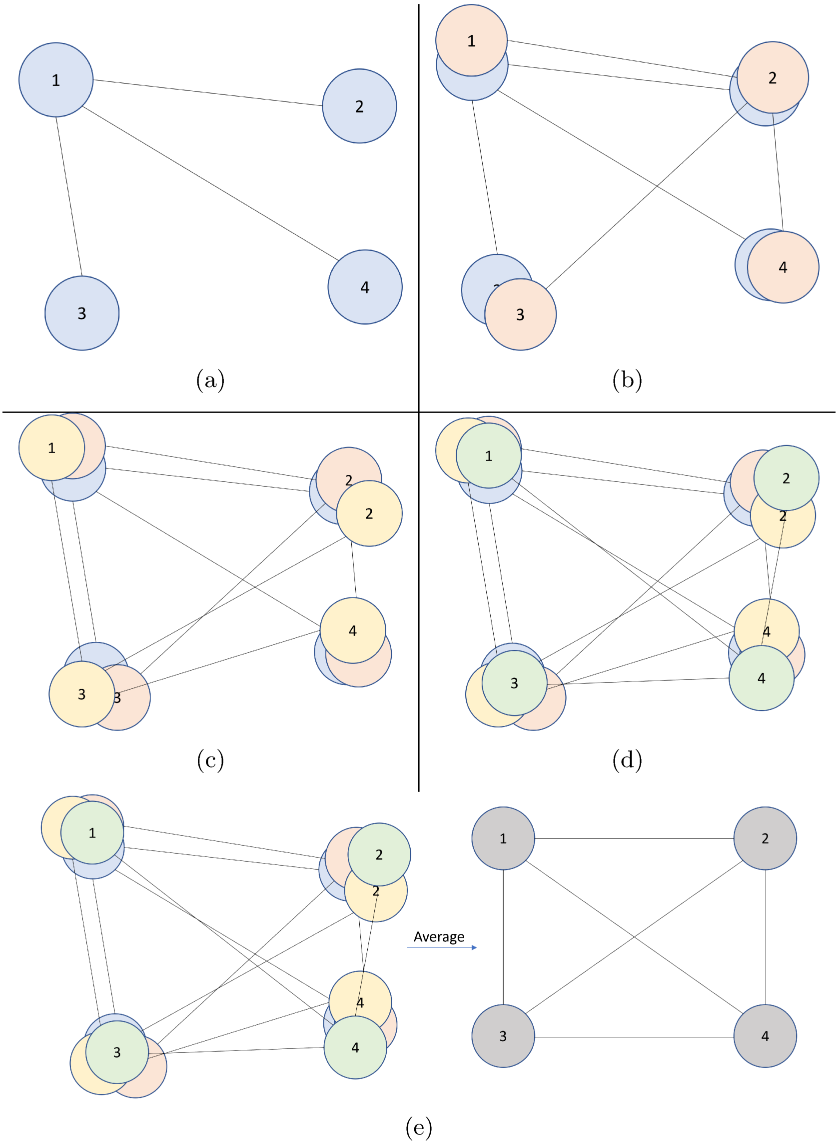


Fig. 5: (a) The location of Atom 1 is chosen based on its predicted fractional coordinates (x_1, y_1, and z_1). The location of Atom 2 is chosen by taking Atom 1's location and adding the relative distance between Atom 1 and Atom 2, found in Layers 2 - 4 of the `CrysTens` representation ($x_1 + \Delta_{2,x}$, $y_1 + \Delta_{2,y}$ and $z_1 + \Delta_{2,z}$) The locations of Atoms 3 and 4 are found the same way. (b) The location of Atom 2 is chosen based on its predicted fractional coordinates (x_2, y_2, and z_2) and predictions for Atoms 1, 3, and 4 are found using Layers 2 - 4 of the `CrysTens` representation. (c) The same method is applied. (d) The same method is applied. (e) Finally, each of the positions are averaged and the final fractional coordinates are used in the construction of each CIF.

3.3 Model Overviews

3.3.1 Vanilla Generative Adversarial Networks

Generative Adversarial Networks depend on an adversarial approach to generate new data. The two neural networks that GANs are composed of, the generator and the discriminator, work against each other. The generator attempts to generate realistic data and the discriminator attempts to differentiate between generated and real data. In order for the discriminator to predict whether a given sample is real or fake, a sigmoid activation function is used. When a prediction is above or equal to 0.5 on the sigmoid function, the sample is labeled as real, and if the prediction is below 0.5, the sample is labeled as fake or generated. The more sure the discriminator is of its prediction, the more it will tend to predict closer to 1 or 0 as opposed to around 0.5.

The major shortcoming associated with the sigmoid activation function directly corresponds to the difficulty of the tasks assigned to the two neural networks. It is far easier to classify if a given sample is real or fake than it is to generate an entirely new sample. During the initial training epochs, before the generator has had any time to calculate the weights necessary for accurate sample generation, it will produce samples that are very obviously fake. When the discriminator has to decide between something that is clearly real and something that is obviously fake, it will start performing very well and will begin predicting with higher confidence (a value closer to either 1 or 0).

When plotted on a sigmoid activation function, low confidence predictions correspond to a large gradient (closer to 0.5) while high confidence predictions correspond to a progressively diminishing gradient (closer to 1 or 0). A prediction plotted within the large gradient zone of a sigmoid activation function corresponds with more-useful information being given back to the generator about how to improve its weights. Inversely, a prediction plotted within the diminishing gradient zone will provide the generator with less-useful information. The more confident the discriminator becomes with its predictions, the more the gradient and the usability of the information coming from the discriminator will be decreased (vanishing gradient problem). This leads to a repeating cycle where the discriminator continually improves and the performance of the generator ultimately stagnates.

3.3.2 Wasserstein Generative Adversarial Networks

To rectify the issue mentioned in the previous section, researchers have implemented a technique that takes advantage of the Earth Mover's Distance (EMD). EMD is a method of understanding the dissimilarity between two multidimensional sets of data samples. In our case, the two data sets are the real `CrysTens`' constructed from PCD and the samples produced by the generator. The theoretical "distance" between the two sets of data samples can be used to construct a loss function called the Wasserstein Loss, and the GANs that take advantage of this are known as Wasserstein GANs (WGANs). Using Wasserstein Loss, WGANs no longer need to predict if a sample is real or fake based on a probability between 0 and 1. Rather, the mathematics of the EMD equation allows WGANs to instead predict the "realness" of a sample. The discriminator is replaced with a critic function by changing the last layer from a sigmoid activation function to a linear activation function. This adjustment to the training scheme of the GAN rectifies the problem discussed earlier because the linear activation function has the same slope no matter what x-value is used.

Since the slope of the line is constant with respect to the relative performance between the critic and the generator (there is no low-gradient zone), the critic will never cease to grant the generator with useful data in terms of how to adjust its weights in order to produce more realistic samples.

3.3.3 Diffusion Models

Unlike GANs, Diffusion Models do not depend on adversarial processes to generate outputs. Diffusion Models generate data based on non-equilibrium thermodynamics. Diffusion Models are composed of two separate stages: the forward and reverse diffusion processes. The forward diffusion process is responsible for the addition of Gaussian noise to a given sample while the backward diffusion process is the reconstruction of a sample from a noisier sample. Diffusion Models operate on a series of time steps, from 1 to t, where an increased time step indicates another addition of Gaussian noise. This process is treated as a Markov Chain, where the sample at time step t only depends on the sample from time step t - 1. The forward process is fixed, however, the model attempts to learn the necessary operations to perform on a given sample at time step t to reconstruct the sample at t - 1. Once training is complete, the model should be able to generate a sample similar to those within the original distribution from complete Gaussian noise. While GANs may suffer from mode collapse and instability during training, Diffusion Models allow for stable training of large models on diverse data [64]. While sampling from Diffusion Models requires many more forward passes when compared to GANs single-pass during inference, this allows for refinement of outputs and is not a major drawback for this particular use [65, 66].

4 Results

Several models from each category (Vanilla GANs, WGANs, and Diffusion Models) were trained using the `CrysTens` representations in an attempt to understand which deep learning method should be explored further for material discovery. The models were evaluated on several different metrics and we found a strong correlation between a general "understanding" of our representation and the output CIFs that were received. The GAN and WGAN used were custom convolution-based models. The Diffusion Model is the model found at https://github.com/lucidrains/imagen-pytorch. The details and hyperparameters of each model can be found in **Appendix A**.

Table 1 is used to show the ability of each model to capture the symmetrical characteristics of the `CrysTens`' structure. The average variance among each repeated parameter, angle, space group number, and fractional coordinates in the `CrysTens` representation for one thousand generated CIFs was calculated. Table 2 shows the average agreement between absolute and relative coordinates in a given `CrysTens` as to where to place each atom, as well as the average difference between the reconstructed pairwise distance matrix and the generated pairwise distance matrix for one thousand generated CIFs.

Furthermore, we wanted to investigate how well each model modeled the PCD dataset as a whole. To do this, the distribution proportions of the parameters, angles, space group number, and atomic numbers were found for the 53,856 CIFs in our dataset. The same distribution proportions were calculated for the set of one thousand generated CIFs generated by each model and a similarity score was calculated for each one (Fig. 6, Fig. 7, and Fig. 8).

Finally, to investigate whether the correlation between `CrysTens` symmetry and CIF quality was reflected on an external metric, we used the Crystal Graph Convolutional Neural Network (CGCNN) to calculate the distribution of predicted formation energy, final energy, band gap, bulk moduli, shear moduli, Fermi Energy, and Poisson Ratio for the real CIFs as well as the generated CIFs for each model (Fig. 9) [67]. We also used CGCNN to calculate decomposition energy using the model found at https://github.com/sparks-baird/mat_discover/tree/cgcnn-decomposition-energy/models/CGCNN. Finally, several generated CIFs from each model are shown to showcase the visual differences in the generated crystal quality (Fig. 10, Fig. 11, and Fig. 12). The produced `CrysTens` representations of the crystals shown, used 3-Means Clustering for the atomic numbers and 6-Means Clustering for the potential distinct x, y, and z coordinates.

Table 1: Comparison between GANs, WGANs, and Diffusion Models on `CrysTens` Lattice Parameters, Lattice Angles, Space Group, and Fractional Coordinate Variance

Model	$\sigma^2_{Parameter}$	σ^2_{Angle}	$\sigma^2_{SpaceGroup}$	$\sigma^2_{FractionalCoordinates}$
GAN	582.23	1338.11	6322.1	5.57e-2
WGAN	31.18	695.03	3024.27	5.9e-2
Diffusion	**3.5e-1**	**8.46e-1**	**4.56**	**4.36e-4**

Note: One-thousand `CrysTens` representations were generated and the variance was averaged over all of them.

Table 2: Comparison between GANs, WGANs, and Diffusion Models on `CrysTens` Coordinate Agreement and Pairwise Distance Matrix Difference

Model	$\Delta_{Coordinate}/Atom$[1]	$\Sigma(\Delta_{Pairwise})/Atom$[2]
GAN	9.62e-2	3.84
WGAN	1.37e-2	2.49
Diffusion	**5.51e-4**	**4.73e-1**

[1] The difference between the “absolute” x, y, and z values predicted and the “relative” position predicted by the direction graph.

[2] The sum of the difference between the reconstructed pairwise distance matrix (by the final coordinates) and the generated pairwise distance matrix in Layer 1 of `CrysTens`.

Note: One-thousand `CrysTens` representations were generated and the values was averaged over all of them.

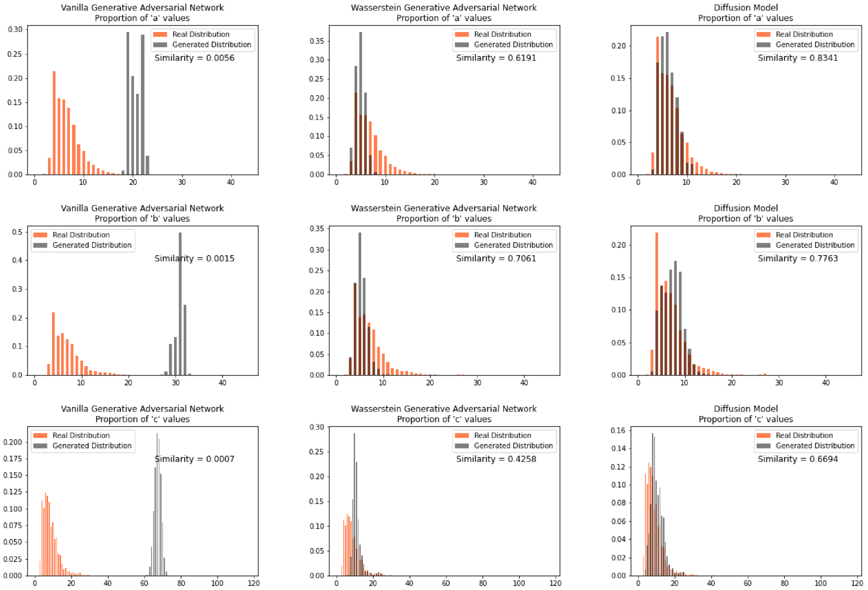


Fig. 6: The distribution proportion for the parameters a, b, and c for the real CIFs vs the generated CIFs.

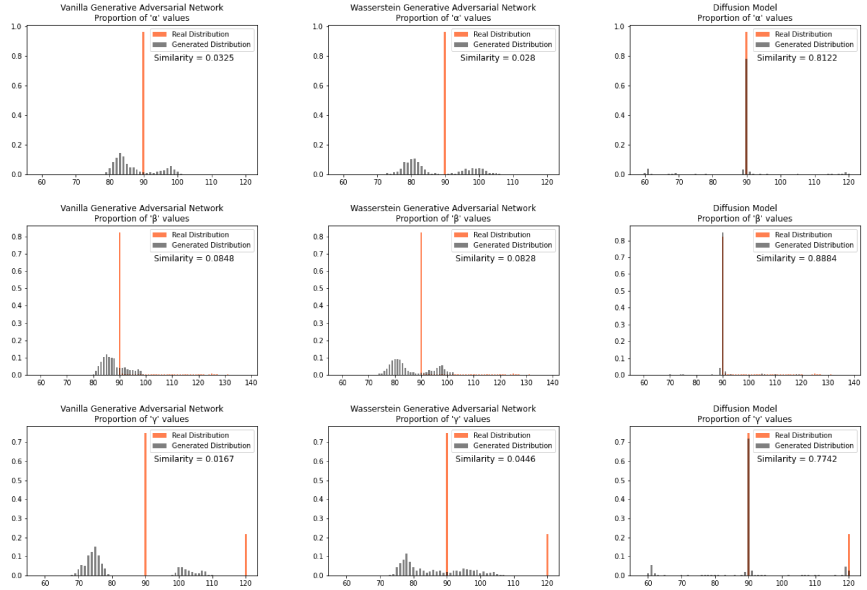


Fig. 7: The distribution proportion for the angles α, β, and γ for the real CIFs vs the generated CIFs.

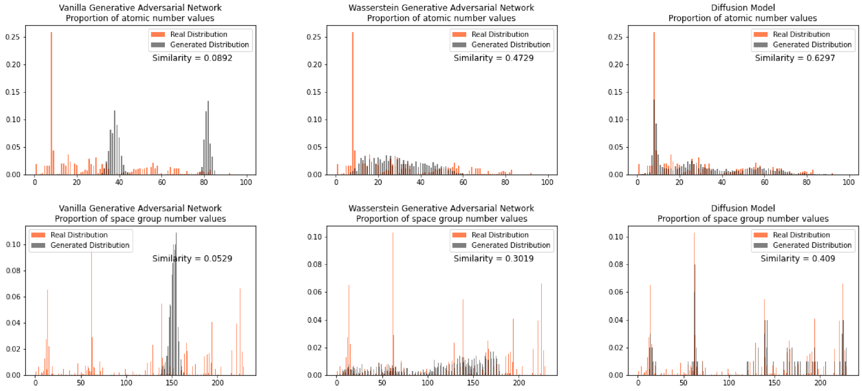


Fig. 8: The distribution proportion for the space group numbers and atomic numbers.

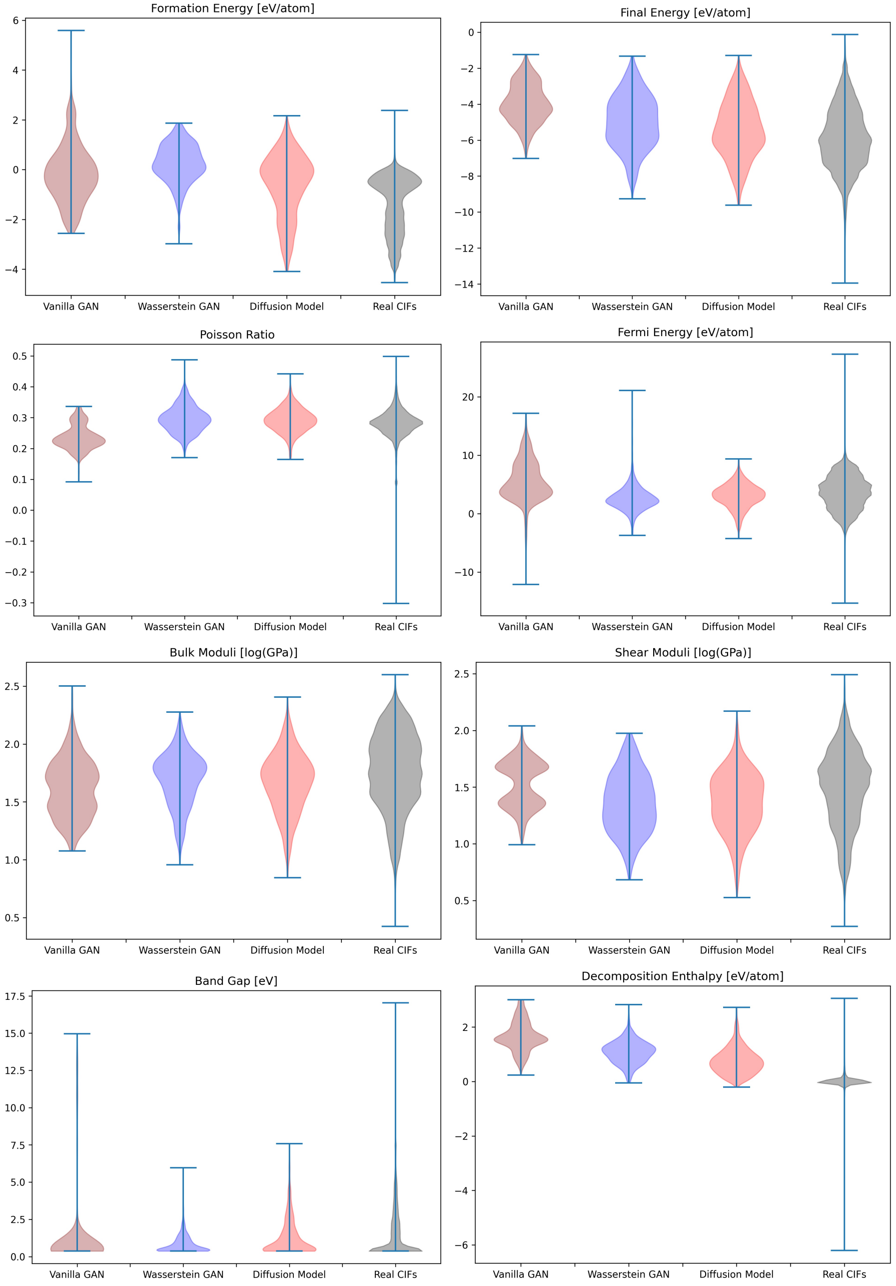


Fig. 9: The distribution of CGCNN predicted values for the real CIFs and the CIFs generated by each of the models.

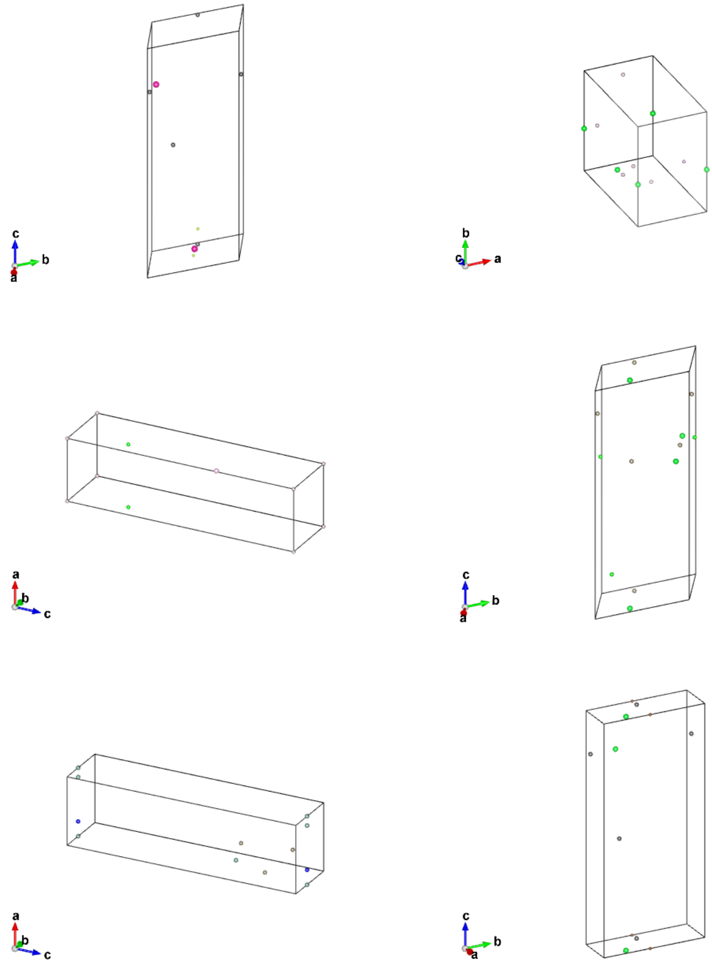


Fig. 10: CIFs generated from the Vanilla Generative Adversarial Network Model. The CIFs had a predicted decomposition energy (in eV/atom) as follows, from left to right then top to bottom: 1.03, 1.49, 1.35, 1.7, 2.43, 1.49

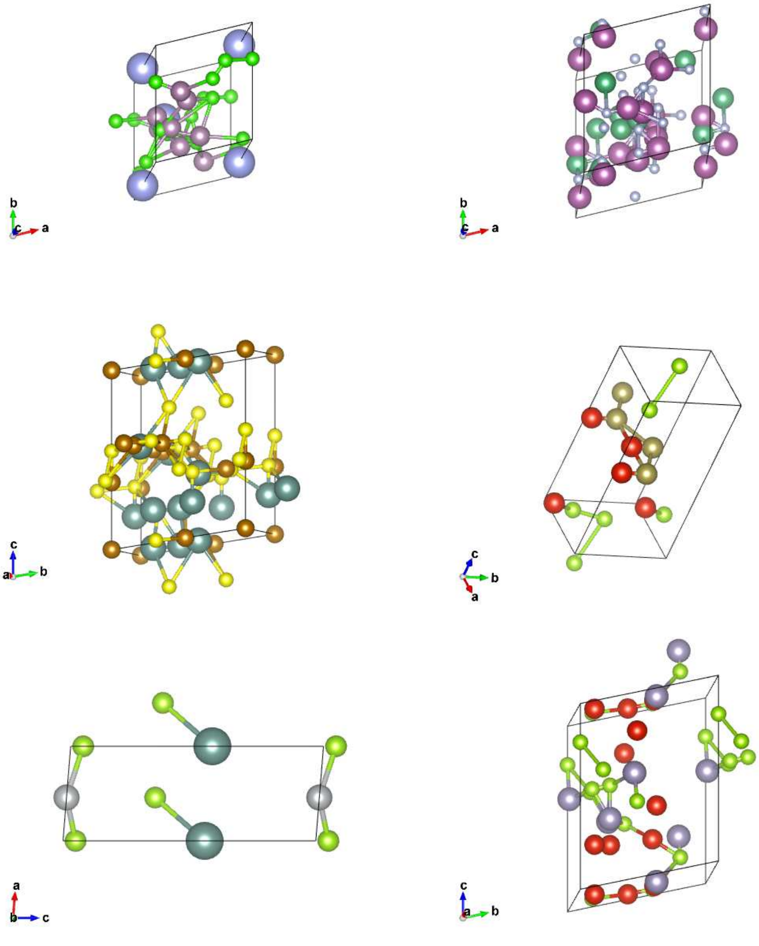


Fig. 11: CIFs generated from the Wasserstein Generative Adversarial Network Model. The CIFs had a predicted decomposition energy (in eV/atom) as follows, from left to right then top to bottom: 1.7, 1.2, 1.26, 0.98, 0.28, 1.42

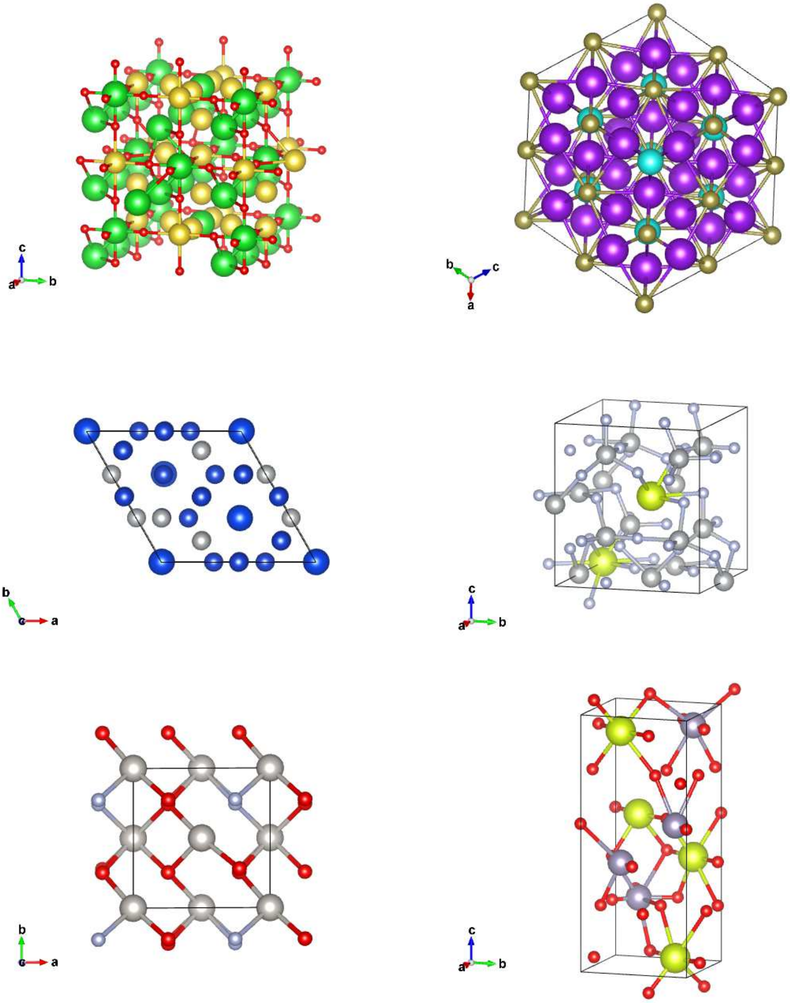


Fig. 12: CIFs generated from the Diffusion Model. The CIFs had a predicted decomposition energy (in eV/atom) as follows, from left to right then top to bottom: 0.09, 0.03, 0.02, 0.42, 1.67, 0.59

5 Discussion

The Diffusion Model outperformed Vanilla GAN and WGAN by several orders of magnitude in minimizing variance across lattice parameters, lattice angles, space group number, and fractional coordinate variances as well as the agreement between absolute and relative fractional coordinates and pairwise distance matrix difference. The Vanilla GAN was orders of magnitude worse than both the WGAN and Diffusion Models. We hypothesized that low variances across parameters, angles, space group number, and fractional coordinates and low differences between relative and absolute fractional coordinates would correspond to better performance in other metrics as well. With this hypothesis, we were expecting to see the Diffusion Models create CIFs that shared a higher degree of similarity with the real PCD distribution than the WGAN and GAN as well as produce higher quality CIFs.

When the distribution of lattice parameters, lattice angles, space group numbers, and atomic numbers of each model was checked against the values in the real distribution, we found that once again, our Diffusion Model performed best. In all areas, Diffusion Models performed exceedingly well, even capturing the peaks in the distribution of space group and atomic numbers (See Fig 8), WGANs were not able to do this and Vanilla GAN distributions were even worse with indications of severe mode collapse.

Using pre-trained CGCNN models, several material properties were predicted for each CIF. The distribution of each predicted property for each model was compared against the distribution of that property across real CIFs. Both Diffusion Models and WGANs were capable of approximating the real distribution of each predicted property to an adequate degree, while GANs struggled, once again showing the mode collapse that occurred.

Finally, a set of CIFs was shown for each model. The CIFs generated by the Vanilla GAN represented exactly what was exhibited in the parameter and CGCNN prediction distributions. All of the CIFs found had very similar parameters and space group numbers. There was an extremely low degree of symmetry and general realness. Furthermore, the bounding boxes outlined by the lattice parameters were enormous in every CIF that we observed. Each of the CIFs had a predicted decomposition energy (eV/atom) of above 1. The lack of “understanding” of `CrysTens` that the Vanilla GAN showed reflected in the CIFs it was capable of producing.

The CIFs produced by our WGAN were a positive shift in the correct direction. There was far more variability in the lattice parameters, lattice angles, space groups, and elements present when CIFs were generated. There are echoes of symmetrical components visible within the CIFs, however, symmetry would not be a word used to describe these CIFs. Although four of the CIFs shown still had decomposition energy (eV/atom) above 1, two of them did not. There is clearly room for improvement for the CIFs generated from our WGAN.

The CIFs generated by our Diffusion Models not only have a diverse set of lattice parameters, lattice angles, space group numbers, and elements present, but they are symmetrical and realistic looking. Although the CIFs produced are not perfect, there is a clear distinction between the CIFs produced by our Diffusion Model and our WGAN. Of the decomposition energies of the six CIFs listed, five of them had decomposition energies (eV/atom) below 1, and three of them even had decomposition energies below 0.1, once again reinforcing the realistic characteristics of the CIFs produced by the Diffusion Models.

6 Further Validation

Given the success of our Diffusion Model on our metrics, we attempted to validate some of the CIFs generated by our Diffusion Model. M3GNet is a materials graph neural network that incorporates three-body interactions. M3GNet can be used as an additional predictor of formation energy for all of the CIFs generated by the Diffusion Model [68].

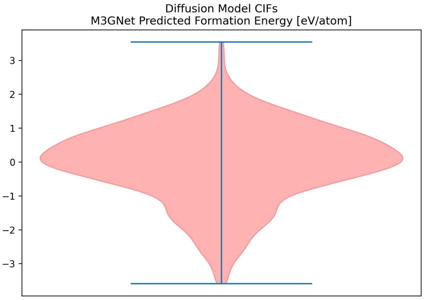
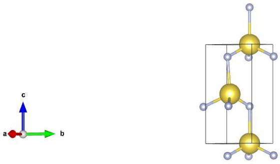
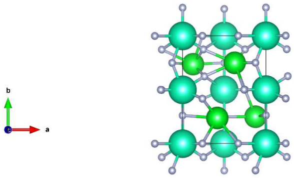


Fig. 13: The distribution of formation energy, as predicted by M3GNet for the CIFs generated by the Diffusion Model.

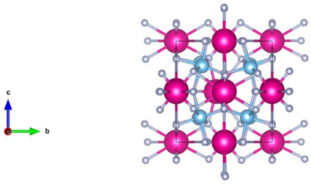
Among the top 35 CIFs with the lowest M3GNet-predicted formation energy, we selected 6 of them to be rigorously analyzed using Vienna Ab Initio Simulation Package (VASP). To prepare the 6 CIFs, slight manual tuning was applied after post-processing to maximize the likelihood that the structure would be realistic and stable. The manual tuning consisted of rounding fractional coordinates to more probable locations such as $(0.011, 0.502, 0.009) \rightarrow (0.0, 0.5, 0.0)$, and correcting potentially erroneous atomic number assignments as a result of the post-processing (36:Kr → 37:Rb). We calculated the free energy of the 6 selected CIFs, and then performed 4 consecutive structure relaxations to allow the atomic positions to be slightly adjusted each time. The resulting free energy and external pressure values after each relaxation are used to evaluate their stability. The calculated free energy for the pre-relaxed structures (E_0) , after the 4th relaxation (E_4) and the external pressure on the unit cell in kb after the 1st (P_1) and 4th (P_4) relaxation are shown below:



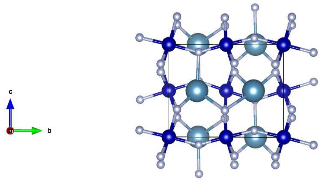
(a) NaF
E_0 = -16.6 eV, P_1 = 0 kb
E_4 = -17.3 eV, P_4 = -0.51 kb



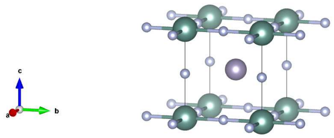
(b) $CsSrF_3$
E_0 = -59.9 eV, P_1 = 18.5 kb
E_4 = -101.4 eV, P_4 = -4.95 kb



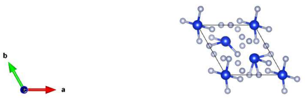
(c) $TiRbF_3$
E_0 = -162.0 eV, P_1 = 2.45 kb
E_4 = -227.1 eV, P_4 = -0.04 kb



(d) $CaCoF_3$
E_0 = -103.5 eV, P_1 = -9.66 kb
E_4 = -108.2 eV, P_4 = -1.40 kb



(e) $SnYF_3$
E_0 = -29.0 eV, P_1 = -0.30 kb
E_4 = -29.9 eV P_4 = 0.30 kb



(f) SiF_3
E_0 = -124.6 eV, P_1 = -11 kb
E_4 = –131.1 eV, P_4 = -4.90 kb

Fig. 14: VASP calculated free energy and external pressure for 6 CIFs produced by the Diffusion Model

As expected, all the free energy are lower and the external pressure values are closer to zero for the 4th relaxation, which indicates the VASP optmized the CIFs into even more stable structures. However, it is observed that the 6 CIFs exhibited low free energy and external pressure on the unit cell even from the pre-relaxation and 1st optimization. This is especially apparent for the NaF and $SnYF_3$ structures, which E_0 and E_4 are very close and the external pressure values after the 1st optimization are very close to 0kb. These are strong indications of the stability of the produced CIFs.

7 Conclusion

In order to perform efficient material discovery via deep learning, it is important to find a representation that is capable of capturing all aspects of a given crystal structure. `CrysTens` encodes all of the pertinent values of a structure such as lattice parameters, lattice angles, and space group numbers as well as an interatomic component that is composed of a pairwise distance matrix as well as a dimensional graph for each dimension. The redundant aspects of `CrysTens` not only allow generative models many opportunities to mitigate noise when generating a crystal but it can also provide a way for measuring the performance of a given model. The variance of generated lattice parameters, lattice angles, space groups, and fractional coordinates is correlated with a model's ability to produce realistic and symmetrical crystals that have parameters similar to those found in the real distribution and even match the real distribution of CGCNN predicted values. We found that Vanilla GANs struggled in this space, often falling victim to training instability and mode collapse that ultimately lead to poor generated CIF quality. Many of these problems were rectified with the implementation of EMD and Wasserstein Loss, creating a WGAN. However, although WGANs did not struggle with the same training instability and mode collapse that Vanilla GANs did, they failed to consistently produce symmetrical crystals. We found that Diffusion Models performed the best in this space. They performed the best in all of our metrics and consistently produced the most realistic-looking and symmetrical CIFs. The enhanced performance of Diffusion Models over GANs holds true in the image synthesis domain as well [64]. The use of Diffusion Models is extremely promising in the field of materials informatics and the improvement of such models could not only provide an efficient method of materials discovery but could revolutionize inverse design as well.

8 Future Works

The performance of Diffusion Models in the materials discovery space creates a lot of opportunity for future works. The method of Diffusion Model generation we used in our work is known as unconditional generation. Conditional generation is the method that was responsible for all of the text-to-image breakthroughs that underpin powerful tools such as DallE-2 [59] and Imagen [60]. It is possible to apply conditional generation to our work, as chemical formula-to-crystal generation. This could work by taking any chemical formula from PCD such as $Ca_3AlB_2[OH]_{15}[H_2O]_{11}$ and transforming it into a natural language analog such as "three calcium atoms, one aluminum atom, two boron atoms, fifteen hydroxide (one oxygen atom, one hydrogen atom) molecules, eleven water (one oxygen atom, two hydrogen atoms) molecules." This, along with increasing the complexity of our Diffusion Models and adding chemical descriptors as additional layers to `CrysTens` will be our next step in this space.

9 Acknowledgements

This work was supported by the National Science Foundation under CAREER Award 1651668. The Google TPU Research Cloud Program provided TPUs for training and testing.

10 CRediT Statement

Michael D. Alverson: Supervision, Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Visualization, Writing - Original Draft & Subsequent Edits. **Sterling G. Baird**: Methodology, Software, Writing - Review & Editing. **Ryan Murdock**: Software, Writing - Review & Editing. **Sin Hang Ho**: VASP Calculations, Writing - Editing. **Jeremy Johnson**: VASP Calculations. **Taylor D. Sparks**: Supervision, Project administration, Funding acquisition, Conceptualization, Resources, Writing – Review & Editing.

11 Data Availability

The dataset used was Pearson's Crystal Data. All code and data can be found at https://github.com/michaeldalverson/CrysTens.

References

[1] A. Ludwig, “Discovery of new materials using combinatorial synthesis and high-throughput characterization of thin-film materials libraries combined with computational methods.,” *npj Comput Mater*, vol. 5, no. 70, 2019.

[2] A. L. Greenaway, A. L. Loutris, K. N. Heinselman, C. L. Melamed, R. R. Schnepf, M. B. Tellekamp, R. Woods-Robinson, R. Sherbondy, D. Bardgett, S. Bauers, A. Zakutayev, S. T. Christensen, S. Lany, and A. C. Tamboli, “Combinatorial synthesis of magnesium tin nitride semiconductors,” *Journal of the American Chemical Society*, vol. 142, no. 18, pp. 8421–8430, 2020. PMID: 32279492.

[3] R. Bohacek, C. McMartin, and W. Guida, “The art and practice of structure-based drug design: a molecular modeling perspective,” *Medicinal research reviews*, vol. 16, p. 3—50, January 1996.

[4] Y. Dan, Y. Zhao, X. Li, S. Li, M. Hu, and J. Hu, “Generative adversarial networks (GAN) based efficient sampling of chemical composition space for inverse design of inorganic materials,” *npj Computational Materials*, vol. 6, pp. 1–7, June 2020.

[5] V. Korolev, A. Mitrofanov, A. Eliseev, and V. Tkachenko, “Machine-learning-assisted search for functional materials over extended chemical space,” *Mater. Horiz.*, vol. 7, no. 10, pp. 2710–2718, 2020.

[6] Y. Sawada, K. Morikawa, and M. Fujii, “Study of Deep Generative Models for Inorganic Chemical Compositions,” *arXiv:1910.11499 [cond-mat, physics:physics]*, Oct. 2019.

[7] Q. Sun and S.-H. Wei, “Inverse design of stable spinel compounds with high optical absorption *via* materials genome engineering,” *J. Mater. Chem. A*, p. 10.1039.D2TA02306G, 2022.

[8] L. Wei, Q. Li, Y. Song, S. Stefanov, E. M. D. Siriwardane, F. Chen, and J. Hu, “Crystal Transformer: Self-learning neural language model for Generative and Tinkering Design of Materials,” Apr. 2022.

[9] R. Xin, E. M. D. Siriwardane, Y. Song, Y. Zhao, S.-Y. Louis, A. Nasiri, and J. Hu, “Active-Learning-Based Generative Design for the Discovery of Wide-Band-Gap Materials,” *J. Phys. Chem. C*, vol. 125, pp. 16118–16128, July 2021.

[10] Z. Alperstein, A. Cherkasov, and J. T. Rolfe, “All SMILES Variational Autoencoder,” June 2019.

[11] N. Anand, R. Eguchi, and P.-S. Huang, “Fully differentiable full-atom protein backbone generation,” Mar. 2019.

[12] N. Anand and P. Huang, “Generative modeling for protein structures,” p. 12.

[13] R. R. Eguchi, C. A. Choe, and P.-S. Huang, “Ig-VAE: Generative Modeling of Protein Structure by Direct 3D Coordinate Generation,” preprint, Bioinformatics, Aug. 2020.

[14] N. C. Frey, V. Gadepally, and B. Ramsundar, “FastFlows: Flow-Based Models for Molecular Graph Generation,” *arXiv:2201.12419 [physics]*, Jan. 2022.

[15] W. Gao, S. P. Mahajan, J. Sulam, and J. J. Gray, "Deep Learning in Protein Structural Modeling and Design," *Patterns*, vol. 1, p. 100142, Dec. 2020.

[16] M. Hoffmann and F. Noé, "Generating valid Euclidean distance matrices," Nov. 2019.

[17] H. Huang and X. Gong, "A Review of Protein Inter-residue Distance Prediction," *Current Bioinformatics*, vol. 15, no. 8, pp. 821–830.

[18] D. Lemm, G. F. von Rudorff, and O. A. von Lilienfeld, "Machine learning based energy-free structure predictions of molecules, transition states, and solids," *Nat Commun*, vol. 12, p. 4468, Dec. 2021.

[19] Z. Li, S. P. Nguyen, D. Xu, and Y. Shang, "Protein Loop Modeling Using Deep Generative Adversarial Network," in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, (Boston, MA), pp. 1085–1091, IEEE, Nov. 2017.

[20] S. Ovchinnikov and P.-S. Huang, "Structure-based protein design with deep learning," *Current Opinion in Chemical Biology*, vol. 65, pp. 136–144, Dec. 2021.

[21] B. Sanchez-Lengeling and A. Aspuru-Guzik, "Inverse molecular design using machine learning: Generative models for matter engineering," p. 7, 2018.

[22] J. Westermayr, J. Gilkes, R. Barrett, and R. J. Maurer, "High-throughput property-driven generative design of functional organic molecules," *Nat Comput Sci*, pp. 1–10, Feb. 2023.

[23] F. Zhai and Q. Li, "A Euclidean distance matrix model for protein molecular conformation," *J Glob Optim*, vol. 76, pp. 709–728, Apr. 2020.

[24] R. Ahmad and W. Cai, "Free energy calculation of crystalline solids using normalizing flow," Nov. 2021.

[25] S. G. Baird, K. M. Jablonka, M. D. Alverson, H. M. Sayeed, M. F. Khan, C. Seegmiller, B. Smit, and T. D. Sparks, "Xtal2png: A Python package for representing crystalstructure as PNG files," *JOSS*, vol. 7, p. 4528, Aug. 2022.

[26] G. Bergami, "Gyankos/DGSOL," June 2022.

[27] C. J. Court, B. Yildirim, A. Jain, and J. M. Cole, "3-D Inorganic Crystal Structure Generation and Property Prediction via Representation Learning," *J. Chem. Inf. Model.*, vol. 60, pp. 4518–4535, Oct. 2020.

[28] S. Fredericks, K. Parrish, D. Sayre, and Q. Zhu, "PyXtal: A Python Library for Crystal Structure Generation and Symmetry Analysis," *Computer Physics Communications*, vol. 261, p. 107810, Apr. 2021.

[29] R. E. A. Goodall, A. S. Parackal, F. A. Faber, R. Armiento, and A. A. Lee, "Rapid discovery of stable materials by coordinate-free coarse graining," *Sci. Adv.*, vol. 8, p. eabn4117, July 2022.

[30] S. Kim, J. Noh, G. H. Gu, A. Aspuru-Guzik, and Y. Jung, "Generative Adversarial Networks for Crystal Structure Prediction," *ACS Cent. Sci.*, vol. 6, pp. 1412–1420, Aug. 2020.

[31] I.-H. Lee and K. Chang, “Crystal structure prediction in a continuous representative space,” *Computational Materials Science*, vol. 194, p. 110436, June 2021.

[32] T. Long, N. M. Fortunato, I. Opahle, Y. Zhang, I. Samathrakis, C. Shen, O. Gutfleisch, and H. Zhang, “Constrained crystals deep convolutional generative adversarial network for the inverse design of crystal structures,” *npj Comput Mater*, vol. 7, p. 66, Dec. 2021.

[33] T. Long, Y. Zhang, N. M. Fortunato, C. Shen, M. Dai, and H. Zhang, “Inverse design of crystal structures for multicomponent systems,” *Acta Materialia*, vol. 231, p. 117898, June 2022.

[34] P. Lyngby and K. S. Thygesen, “Data-driven discovery of novel 2D materials by deep generative models,” June 2022.

[35] P. Lyngby and K. S. Thygesen, “Data-driven discovery of 2D materials by deep generative models,” *npj Comput Mater*, vol. 8, p. 232, Nov. 2022.

[36] J. Noh, J. Kim, H. S. Stein, B. Sanchez-Lengeling, J. M. Gregoire, A. Aspuru-Guzik, and Y. Jung, “Inverse Design of Solid-State Materials via a Continuous Representation,” *Matter*, vol. 1, pp. 1370–1384, Nov. 2019.

[37] 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, “An invertible crystallographic representation for general inverse design of inorganic crystals with targeted properties,” *Matter*, vol. 5, pp. 314–335, Jan. 2022.

[38] H. Türk, E. Landini, C. Kunkel, J. T. Margraf, and K. Reuter, “Assessing Deep Generative Models in Chemical Composition Space,” *Chem. Mater.*, Oct. 2022.

[39] P. Wirnsberger, G. Papamakarios, B. Ibarz, S. Racanière, A. J. Ballard, A. Pritzel, and C. Blundell, “Normalizing flows for atomic solids,” *Mach. Learn.: Sci. Technol.*, vol. 3, p. 025009, June 2022.

[40] T. Xie, X. Fu, O.-E. Ganea, R. Barzilay, and T. Jaakkola, “Crystal Diffusion Variational Autoencoder for Periodic Material Generation,” *arXiv:2110.06197 [cond-mat, physics:physics]*, Mar. 2022.

[41] Y. Zhao, E. M. D. Siriwardane, Z. Wu, M. Hu, N. Fu, and J. Hu, “Physics Guided Generative Adversarial Networks for Generations of Crystal Materials with Symmetry Constraints,” Mar. 2022.

[42] 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, “An invertible crystallographic representation for general inverse design of inorganic crystals with targeted properties,” *Matter*, vol. 5, pp. 314–335, Jan. 2022.

[43] Y. Zhao, M. Al-Fahdi, M. Hu, E. M. Siriwardane, Y. Song, A. Nasiri, and J. Hu, “High-throughput discovery of novel cubic crystal materials using deep generative neural networks,” *Advanced Science*, vol. 8, p. 2100566, Oct. 2021.

[44] J. Köhler, M. Invernizzi, P. de Haan, and F. Noé, “Rigid body flows for

sampling molecular crystal structures," Jan. 2023.

[45] Z. Ren, J. Noh, S. Tian, F. Oviedo, G. Xing, Q. Liang, A. Aberle, Y. Liu, Q. Li, S. Jayavelu, *et al.*, "Inverse design of crystals using generalized invertible crystallographic representation," *arXiv preprint arXiv:2005.07609*, 2020.

[46] S. G. Baird, K. M. Jablonka, M. D. Alverson, H. M. Sayeed, M. F. Khan, C. Seegmiller, B. Smit, and T. D. Sparks, "xtal2png: A python package for representing crystal structure as png files," *Journal of Open Source Software*, vol. 7, no. 76, p. 4528, 2022.

[47] Y. Zhao, M. Al-Fahdi, M. Hu, E. Siriwardane, Y. Song, A. Nasiri, and J. Hu, "High-throughput discovery of novel cubic crystal materials using deep generative neural networks," *Advanced Science*, vol. 8, 10 2021.

[48] E. Siriwardane, Y. Zhao, I. Perera, and J. Hu, "Generative design of stable semiconductor materials using deep learning and dft," 01 2022.

[49] A. S. Fuhr and B. G. Sumpter, "Deep generative models for materials discovery and machine learning-accelerated innovation," *Frontiers in Materials*, vol. 9, 2022.

[50] P. J. S. P. e. a. Pyzer-Knapp, E.O., "Accelerating materials discovery using artificial intelligence, high performance computing and robotics," *npj Comput Mater*, vol. 8, no. 84, 2022.

[51] R. S. K. J. e. a. Lim, J., "Molecular generative model based on conditional variational autoencoder for de novo molecular design.," *Journal of Cheminformatics*, vol. 10, no. 31, 2018.

[52] T. Xie, X. Fu, O.-E. Ganea, R. Barzilay, and T. Jaakkola, "Crystal diffusion variational autoencoder for periodic material generation," *arXiv preprint arXiv:2110.06197*, 2021.

[53] S. Kim, J. Noh, G. H. Gu, A. Aspuru-Guzik, and Y. Jung, "Generative adversarial networks for crystal structure prediction," *ACS Central Science*, vol. 6, no. 8, pp. 1412–1420, 2020. PMID: 32875082.

[54] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[55] D. P. Kingma, M. Welling, *et al.*, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.

[56] F. Farnia and A. E. Ozdaglar, "Do gans always have nash equilibria?," in *ICML*, 2020.

[57] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[58] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, pp. 214–223, PMLR, 2017.

[59] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[60] 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, *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *arXiv preprint arXiv:2205.11487*, 2022.

[61] K. Momma and F. Izumi, "Vesta 3 for three-dimensional visualization of crystal, volumetric and morphology data," *Journal of applied crystallography*, vol. 44, no. 6, pp. 1272–1276, 2011.

[62] T. R. J. D., *Crystals and crystal structures.* Wiley, 2020.

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

[64] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[65] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[66] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*, pp. 2256–2265, PMLR, 2015.

[67] T. Xie and J. C. Grossman, "Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties," *Physical review letters*, vol. 120, no. 14, p. 145301, 2018.

[68] C. Chen and S. P. Ong, "A universal graph deep learning interatomic potential for the periodic table," *Nature Computational Science*, vol. 2, no. 11, pp. 718–728, 2022.

Appendix A Model Details

All of the code used for training the models can be found at https://github.com/michaeldalverson/CrysTens

A.1 Vanilla GAN

The Vanilla GAN was created using TensorFlow. It was trained for 100 epochs with a learning rate of $1e-5$ for both the discriminator and the generator. The latent dimension of the generator was 128. The loss function for the discriminator was binary cross-entropy. Both the discriminator and the generator contained a mixture of three-dimensional convolutional (transposed in the case of the generator) and dense layers. ReLU was used as the intermediate activation function in the generator and Leaky ReLU was used in the discriminator. Both neural networks used batch normalization.

A.2 Wasserstein GAN

The Wasserstein GAN was created using TensorFlow. It was trained for 100 epochs with a learning rate of $1e-4$ for both the critic and the generator. The loss function used was the Wasserstein Loss. Gradient penalty was also used (to stabilize training) with a coefficient of $\lambda = 10$. Both the critic and the generator contained a mixture of three-dimensional convolutional (transposed in the case of the generator) and dense layers. The critic was trained 5 times as much as the generator. Leaky ReLU is used for both the critic and the discriminator and a mixture of layer normalization and batch normalization was used.

A.3 Diffusion Model

The Diffusion model was created using Imagen-Pytorch. The model was comprised of two Unets, which form the basis for the denoising diffusion probabilistic models. Each Unet has 256 base channels. The first Unet created a "low-resolution" version of the `CrysTens` that was size 32x32x4. The second Unet, took the output of the first Unet and created the actual `CrysTens` of size 64x64x4. Each Unet was trained separately for 250,000 optimization steps with batch size 4. The original code is found at https://github.com/lucidrains/imagen-pytorch.