Unleashing the power of novel conditional generative approaches for new materials discovery

Anonymous Author(s)
Affiliation
Address
email

Abstract

For a very long time, computational approaches to the design of new materials have relied on an iterative process of finding a candidate material and modeling its properties. AI has played a crucial role in this regard, helping to accelerate the discovery and optimization of crystal properties and structures through advanced computational methodologies and data-driven approaches. To address the problem of new materials design and fasten the process of new materials search, we have applied latest generative approaches to the problem of crystal structure design, trying to solve the inverse problem: by given properties generate a structure that satisfies them without utilizing supercomputer powers. In our work we propose two approaches: 1) conditional structure modification: optimization of the stability of an arbitrary atomic configuration, using the energy difference between the most energetically favorable structure and all its less stable polymorphs and 2) conditional structure generation. We used a representation for materials that includes the following information: lattice, atom coordinates, atom types, chemical features, space group and formation energy of the structure. The loss function was optimized to take into account the periodic boundary conditions of crystal structures. We have applied Diffusion models approach, Flow matching, usual Autoencoder (AE) and compared the results of the models and approaches. As a metric for the study, physical pymatgen matcher was employed: we compare target structure with generated one using default tolerances. So far, our modifier and generator produce structures with needed properties with accuracy 41% and 82% respectively. To prove the offered methodology efficiency, inference have been carried out, resulting in several potentially new structures with formation energy below the AFLOW-derived convex hulls.

1 Introduction

The search for novel materials with specified properties has been a cornerstone of scientific exploration for decades. From the discovery of semiconductors revolutionizing electronics to the development of superalloys enhancing aerospace technologies, the synthesis of new materials has continually propelled technological advancements.

However, traditional methods for material discovery often employ exhaustive trial and error experimental approaches. In turn, computational efforts, relying on density functional theory (DFT)[1] approaches, usually require huge amounts of computing power. In this regard, automatic descriptor generators[2], GNNs[3][4] and transferable GNN models [5] fueled combination of these methods and machine learning (ML) approaches. In particular, the

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

utilization of generative machine learning models, such as Variational Autoencoder[6] and GANs[7], presents a paradigm shift in how crystal structures are generated and optimized. By harnessing the power of data-driven approaches, we can navigate the vast landscape of possible crystal structures with unprecedented efficiency and precision.

Recent advancements in the field of materials discovery have yielded promising results through various innovative approaches. For instance, FTCP[6] utilizes Autoencoders for uncovering new materials, while CubicGAN[7] leverages GANs for the discovery of cubic crystal materials. Additionally, Physics Guided Crystal Generative Model (PGCGM)[8] has introduced a method for generating crystal structures based on specific space groups encoding. DP-CDVAE[9] is a model, that combines VAE and diffusion approaches. MatterGen[10] employed equivariant GNNs as score matching function in diffusion processes for crystal structure generation.

One of the most discussed frameworks is GNoME[11] that has made most recent and large advancements in the field of the new materials discovery employs a sophisticated pipeline to discover new materials, particularly focusing on inorganic crystals. This allows for the discovery of innovative materials beyond known structures.

After generating candidate structures through both pipelines, GNoME evaluates their stability by predicting their formation energies. Based on the comparison of the obtained formation energy with those of the known competing phases (i.e. stability assessment), the model selects the most promising candidates for further evaluation using known theoretical frameworks.

The question of the completeness of chemical space arises due to two main concerns with GNoME-derived stable structures. Firstly, they mostly contain three or more unique elements, while ternary and quaternary structures are less explored than binary compounds. Secondly, the comparison of GNoME-discovered structures to the Materials Project, which has 154,718 materials, is flawed since larger databases like AFLOW, NOMAD, and the Open Quantum Materials Database contain millions of entries. This raises questions about the novelty of the discovered materials.

In this study, we present an end-to-end framework for the generation of crystal structures with specified properties using advanced generative AI techniques. The basis architecture of the models is Autoencoder, enabling encoding and decoding structural representations. Then, the most commonly used generative approaches in image generation were utilized to model probability distribution transformations, and to capture complex underlying structure-property relationships within our dataset: Flow Matching[12], Denoising Diffusion Probabilistic Models(DDPM)[13], and Denoising Diffusion Implicit Models(DDIM)[14]. Through the integration of these techniques, we aim to transcend conventional limitations in materials discovery, paving the way for accelerated predictions of materials with desired properties.

To employ model architectures often used for image/video generation, a matrix representation of crystal structures was developed, containing crucial information such as chemical composition, atomic coordinates, symmetries (space group), and formation energies. Within the approach proposed, it has become important to develop a novel metric for assessing the similarity between generated structures and target configurations. This metric obviates the need for computationally expensive DFT calculations, allowing for rapid validation and refinement of generated structures. Furthermore, we introduce a loss function that accounts for the periodic boundary conditions inherent in crystal lattices, ensuring the fidelity of the generated structures.

Our study explores two distinct approaches for crystal structure prediction: 1) conditional structure modification and 2) conditional structure generation. The former involves optimizing the stability of existing structures by generating more stable polymorphs, while the latter entails the generation of entirely new structures based on user-defined criteria. Through rigorous analysis, we demonstrate the efficacy of our approach in discovering novel materials with desired properties.

Importantly, to validate the utility of our framework, we conducted a series of generation experiments using the Vienna Ab initio simulation package(VASP)[15] as a tool for inference validation. Remarkably, our method facilitated the discovery of 6 structures below the corresponding convex hull. This significant outcome underscores the remarkable potential of

our framework in uncovering thermodynamically stable materials, thereby offering promising avenues for advanced materials discovery and design.

2 Data, Dataset

2.1 Data overview

In this study, the AFLOW database[16] was utilized as a source of data on the structures and properties of materials. AFLOW is an extensive and comprehensive database that consolidates a vast array of materials-related information, offering an expansive repository for crystallographic data, computed properties, and various other materials-science-related datasets. AFLOW database contains more than 3.5 million structures.

From the extensive collection housed within AFLOW, the focus was narrowed to select only polymorphs, because models are trained to distinguish composition-property and structure-property relations with numerous structures of the same chemical composition. Specifically, the selection process targeted polymorphic structures with 4 to 60 atoms within their unit cells. This criterion aimed to encompass a diverse yet manageable subset of structures, balancing complexity with computational feasibility. By filtering polymorphs based on their atom count, the dataset was balanced.

Moreover, in order to decrease the complexity of the data, we have removed all structures containing elements and space groups found in less than 1% of all structures. The entire dataset consisted of more than 85000 polymorph groups including more than 2.1 million structures. The minimum size of group of polymorphs was 7 samples and the maximum one was 71 samples. The total number of space groups was 19 and the total number of chemical species over the dataset was 55. Each structure S in the dataset is described by the following features:

- Fractional coordinates of atoms in the lattice basis $\mathrm{X}_{\mathrm{coord}}$ (has 60 rows with 3 coordinates x, y, z each) and $\mathrm{X}_{\mathrm{lattice}}$ (matrix 3 by 3 constructed of 3 base vectors). Overall matrix X of structure is constructed as

$$\underset{64\times3}{\mathrm{X}} = concatatenation(\underset{60\times3}{\mathrm{X}_{\mathrm{coord}}}, \underset{1\times3}{\mathrm{padding}}, \underset{3\times3}{\mathrm{X}_{\mathrm{lattice}}}) \tag{1}$$

- Chemical elements which are presented as a one-hot matrix $elements_{ij}$ of size 64×103 (including padding), where ones are positioned at the indices corresponding to the position of a certain chemical element in the periodic table.

$$elements_{ij} = \begin{cases} 1 & \text{if } i\text{-th atom's element number from the periodic table} = j \\ 0 & \text{otherwise} \end{cases}$$

- Elemental property matrix $elementalProperties$ containing 22 chemical features encoding chemical elements obtained from [8]. The properties of each element were calculated using Mendeleev package[17].
- Space group spg of a structure. We use the space group encoding method presented in [8], when each space group is represented by a $192 \times 4 \times 4$ matrix, which corresponds to 192 possible symmetry operations.
- Structure formation energy E
- Nsites - number of atoms in a crystal lattice.

2.2 Data representation. Modification task

The crystal pair sampling strategy involves handling a potential data leakage: possible inclusion of structures from the same polymorph group but with different energies into training and validation subsets. To mitigate this issue, the polymorph group formulas were initially divided into distinct training and validation sets, ensuring a relatively balanced distribution of chemical elements across these subsets. Subsequently, the pairs were categorized into two groups: those with low-energy (lowest energy in polymorph group) targets designated as

$lowestEnergyPairs = (S_i, S_0) \forall i \in [1, ..., structuresNum]$ and those with non-low-energy targets, all structures except the most optimal one, formed as $nonLowestEnergyPairs = (S_i, S_j) |\ i > j > 0$. The validation set was constructed as a subset of $lowestEnergyPairs$. The training set was dynamically constructed every epoch from $lowestEnergyPairs$ and $nonlowestEnergyPairs$, preserving equal numbers of pairs sampled and maintaining a limited count per polymorph group. This strategy ensured a robust separation between training and validation sets, thus preventing data leakage and improving model performance.

Each pair sample $\{S_{init}, S_{target}\} \in pairDataset$ consisted of the information about each structure (hereinafter, we will call them initial and target structures). The following data was used:

- Coordinates and lattice information of initial and target structures X_{init}, X_{target}
- Difference in formation energies between initial and target structures $E_{diff} = E_{target} - E_{init}$
- Space group of target structure spg_{target}
- Elements matrix $elemetsMatrix$, elemental property matrix $elementalProperties$ and number of sites $numSites$, which are the same for initial and target structure because of identical chemical composition.

The modification task involved transforming the input structure X_{init} into the target structure X_{target}.

2.3 Data representation. Generation task

In its tern the generation task receives normal or uniform (depends on a model) noise as input from which the structure is generated, which is akin to the image generation processes in computer vision tasks.

For the generation task, an additional dataset was constructed. Data for the generation task is slightly simpler, while it considers only $\{S_{target}\}$. Therefore, the models can be trained on all structures available, rather than pairs. The following data is used:

- Coordinates and lattice information of target structure X_{target}
- Formation energy of target structure E_{target}
- Space group of target structure spg_{target}
- Elements matrix $elemetsMatrix$, elemental property matrix $elementalProperties$ and number of sites $numSites$ of target structure.

3 Loss and metrics

3.1 Atomic coordinates

The atomic coordinates are represented as a 60×3 matrix, where each row corresponds to the coordinates of an atom. The L1 loss was utilized during the training of a model for predicting atomic coordinates.

$L_1(preds, target)_i = ||preds_i - target_i||_1 = \sum_{j=1}^{3} |preds_{ij} - target_{ij}|$, where $target$ and $pred$ are target and predicted atomic coordinate matrices.

3.2 Lattice

The lattice itself is represented as a 3x3 matrix, where each row signifies a directing basis vector. In this case, we have also used the L1 norm as a loss function.

3.3 Periodic boundary condition loss

This section presents an enhanced loss function, designed for the regression model (see Section 5.1), that addresses this challenge by integrating periodic boundary conditions into

the loss calculation, outperforming the conventional L1 loss function. In the field of ML applied to atomic structures, even slight displacement of atomic coordinates is crucial and employing appropriate loss functions that consider the periodic nature of atomic structures increases the flexibility of model predictions.

In the dataset representing atomic structures, it is crucial to acknowledge the presence of atoms residing at various positions within the lattice framework. Certain atoms are positioned at the vertices, edges, or faces of the lattice. According to periodic boundary conditions (PBC), identical atoms in the vicinity of vertices, edges, or faces but also exist in analogous positions across the lattice. Implementation of such an invariance within the loss function helps in effectively capturing periodic pattern of crystals, enhancing the model's capability to learn and predict atomic structures more comprehensively.

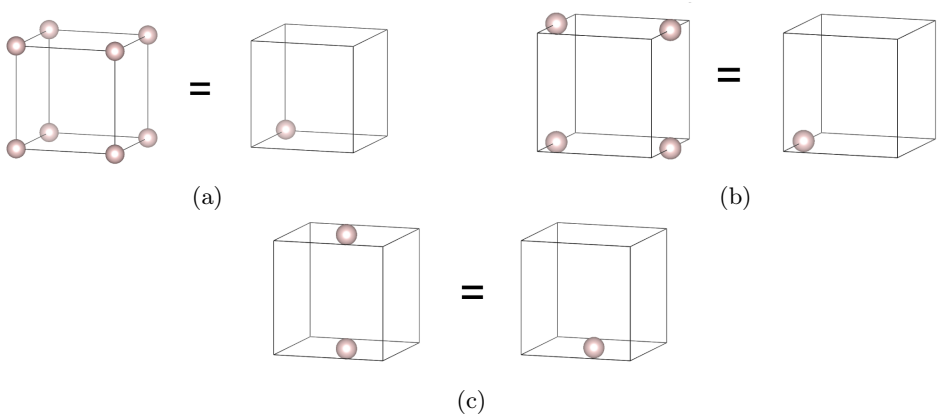


Figure 1: Illustration of atoms at a)vertices, b)edges, and c)faces of lattice under periodic boundary conditions

The loss function is being calculated as minimum of distances from predicted point to the target one taking into account 26 its periodic images (according to PBC) A.4.

The empirical validation of this enhanced loss function showcases its superiority(Figure4) in capturing discrepancies within atomic structures, thus indicating its potential as a robust tool for improving the accuracy of ML models in materials science applications.

3.4 Metric

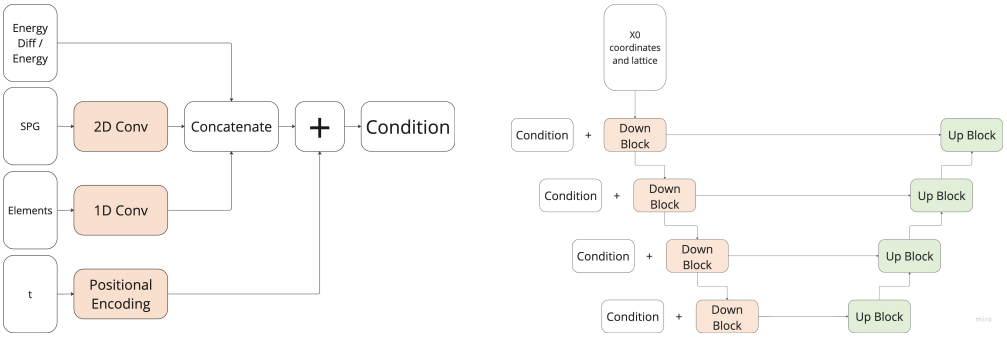
As a metric, we have chosen an analogue of accuracy: the generated structures are compared to the target structures using a specialized matcher, yielding the proportion of structures that successfully pass the matching process. For metric calculation, we employed the Pymatgen StructureMatcher with the default set of parameters ($ltol = 0.2, stol = 0.3, angle\_tol = 5$). Although this approach is less accurate than structure relaxation using ab initio calculations and comparing the structure formation energy with the energy above the hull, it enables model validation to be performed orders of magnitude faster than the traditional method.

4 Model

For experiments, a 1d UNet model (see Figure2 (b)) architecture similar to the 2d UNet model described in [18] was utilized along with 2D and 1D convolutional neural networks (CNNs) for the space group and element matrix embeddings, respectively. Based on this model, 3 different training processes have been developed: ordinary regression model, Conditional Flow Matching (CFM)[19] model, and diffusion model.

The model was conditioned (see Figure2 (a)) on the following data: time condition (t), the same as in [18], element condition (el), formation energy difference condition (E_{diff}), and desirable space group (spg). el_{emb}, spg_{emb} and E_{diff} are concatenated into one embedding

C_{emb}. t is fed into the Transformer Positional Encoding Layer and transformed into an embedding T_{emb}. The two embeddings: C_{emb} and T_{emb} are then applied into one condition.



(a) Condition block (b) UNet

Figure 2: a)Formation of conditions using formation energy, space group, and elemental representation, and b)Schematic depiction of the model architecture

5 Methodology

In this work, two approaches are proposed: crystal structure generation and crystal structure modification. For the generation approach, crystal structures are generated from normal or uniform noise and conditioned to t, el, E, spg. Within the generation, we employed three algorithms: DDPM, DDIM, and CFM models. For the modification approach, crystal structures are generated by modifying other structures, while conditioning to el, E_{diff}, spg (and optionally t, not used in ordinary regression UNet). For the modification task, we have employed three algorithms: UNet Regression model, diffusion model, based on Palette[20] approach, and CFM model. For the generation task, we have employed four algorithms: diffusion models with DDPM and DDIM samplers, and CFM models on Uniform and Normal noise.

5.1 Regression model

During the training stage, the structure coordinates and lattice x_0, elements features el, space group spg and E_{diff} are used as conditions. The model is trained to return x_1 structure coordinates and lattice (Algorithm 1). As for the inference process, one can see the details in the Algorithm 2

5.2 Conditional Flow Matching models

CFM is a fast method for training Continuous Normalizing Flows (CNF)[21] models without the need for simulations. It offers a training objective that enables conditional generative modeling and accelerates both training and inference.

The basic way of training CFM model (Algorithm 3) organized as follows: during the training stage, x_0 and x_1 are sampled from the source distribution and the target distribution respectively, then a linear interpolation x_t is calculated as $x_t = tx_1 + (1-t)x_0$ (exponential moving average between distributions x_0 and x_1; t is sampled from a uniform distribution $\mathcal{U}(0,1)$), and afterwards pass the x_t and t as inputs to our model f_θ, forcing the model to predict a velocity from the distribution x_0 to x_1. Therefore, the loss for CFM model is the following: $L_{CFM} = E_{t,x_1,x_0}[||f_\theta(x_t,t) - (x_1 - x_0)||^2] = E_{t,x_1,x_0}[||f_\theta(tx_1 + (1-t)x_0, t) - (x_1 - x_0)||^2]$

For the modification approach, x_0 and x_1 are both sampled from our dataset distribution according to the sampling strategy for modification mentioned in 2.2. Also, the model is conditioned to el, spg_1, E_{diff}, besides t (see Algorithm 4)

For the generation approach, we tested two noise distributions for the x_0: normal distribution $\mathcal{N}(0, 1)$ and uniform noise distribution $\mathcal{U}(0, 1)$, which resulted in significantly better performance. The intuition for using uniform distribution instead of normal one was inspired by the diagram of x, y, z coordinate distribution (Figure 3). The model is also conditioned to el, spg_1, E, and t (see Algorithm 5)

During the sampling stage, we generate X_1 structure by the given X_0 by solving the following ordinary differential equation (ODE): $dx_t = f_\theta(x_t, t, el, spg_1, E)dt$, beginning with x_0. In order to solve the ODE, the Euler method was employed: $x_{t+h} = x_t + hf_\theta(x_t, t, el, spg_1, E)$ (Algorithm 6)

5.3 Diffusion models

In our work, we observe diffusion models. Diffusion models generate samples from a target distribution x_1, starting from a source distribution $x_0 \sim \mathcal{N}(0, I)$.

During training, these models are trained to reverse a Markovian forward process, which adds noise x_0 to the data step by step. Meaning, diffusion models are trained to predict the noise added to the data samples x_1. In order to train a model in this setup, the following loss function is used, $L_{simple} = E_{t,x_1,x_0}[||x_0 - f_\theta(\sqrt{\bar{\alpha}_t}x_1 + \sqrt{1-\bar{\alpha}_t}x_0, t)||^2]$ where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ and $\alpha_t = 1 - \beta_t$ (β_t is the variance by which added noise is being scheduled on each step t).

Our modification approach is based on Palette, which enables sample-to-sample generation (from noise $\epsilon \sim \mathcal{N}(0, 1)$) using x_0 structure coordinates and lattice, el, spg_1, E_{diff} and t as conditions for generation of x_1 using the DDPM algorithm. Sampling stage is performed by a backward diffusion process with linear scheduler (see Algorithms 7, 8).

For the generation approach ((Algorithm 9), x_0 is sampled from a normal distribution and el, spg_1, E, t are fed into the model as conditions. During our experiments, we tested 2 approaches: DDPM(Algorithm 10) classic approach and DDIM(Algorithm 11) which results in usage of smaller number of sampling steps in order to speed up the generation process. Moreover, DDIM enables the process of generating samples from random noise to be deterministic.

6 Experiment Results

All the models presented in tables (Table 1 and Table 2) have been trained with the same hyperparameters and architectures. The metric used is described in Section 3.4. We also provide all experiment details in A.3.

Table 1: Validation metrics on generation task

DDPM	DDIM	CFM $\mathcal{N}(0, 1)$	CFM $\mathcal{U}(0, 1)$
0.8074	**0.82**	0.482	0.8097

Table 2: Validation metrics on modification task

Ordinary Model	Diffusion	CFM
0.4148	0.3653	0.2059

7 Inference

In order to demonstrate a potential of the proposed approaches, we have chosen a chemical composition, containing numerous variations and phases of structures composed of [W, B, Ta] with well-explored convex hull. Structures that lie on the convex hull are considered to be thermodynamically stable, and the ones above it are either metastable or unstable.

7.1 Inference pipeline

The proposed testing procedure involves generating test conditions for structures, passing them to the trained generative models, pre-optimizing the generated structures to accelerate the following ab initio calculations, and final relaxation and formation energy calculating using VASP. Although in this work two approaches were proposed: Generation and Modification, the following pipeline has only been applied to generation models, due to the fact, that modification approach is based on structure-polymorphs, which leads to the necessity to have at least one structure with needed composition, which is not always so. That fact makes generation models much more flexible in generation structures not only with needed properties, but also with needed composition. Another advantage of the generation models is value of metric that is two times bigger than in modification tasks. The inference algorithm is as follows:

1. Test Condition Formation:
   - The chosen chemical formulas were utilized for feature extraction of *el*. Three chemical compositions have been used: 1) $Ta_1W_1B_6$, 2) $Ta_1W_2B_5$ and 3) $Ta_2W_1B_5$.
   - We have taken *spg* presented in the dataset as an additional condition, obtaining 19 space groups.
   - Finally, a set of target formation energies E has been formed. We have carried out three experiments: 1) starting from the energy of the convex hull and decreasing with a step of 0.01 eV/atom, 2) starting from the energy of the convex hull and decreasing with a step of 0.1 eV/atom, and 3) starting from the energy 1 eV/atom less than the energy of the convex hull and decreasing with a step of 0.01 eV/atom. In total, 21 energy values were used for every inference run.
   - Final inference conditions were obtained by making all possible combinations of *spg* and E for a certain composition *el*
2. Model Inference: The conditions from the step 1 have been put to one of the trained models, resulting in the generation of structures. Two models have been employed: Diffusion approach and Flow matching
3. Pre-Optimization: Following the generation of all structures, each structure has been pre-optimized using the PyMatGen structure relaxation method. The method used m3gnet [22] model with default parameters. PyMatGen pre-optimization contributed to overall speedup of further VASP relaxation.
4. Structure relaxation: Pre-optimized structures were relaxed using VASP (the recommended pseudopotentials, plane wave energy cutoff of 500 eV, Ediif and Ediffg convergence criteria of 10^{-5} and -10^{-2} were used).

7.2 Inference results

To summarize, 6 experiments have been carried out for two different models and for three formation energy conditionings. Every experiment includes 3*380 structures, per 380 structures for every single chemical composition. The results of experiments can be seen in Table 3

As can be seen, 4 structures were obtained with formation energies significantly lower than those obtained from the AFLOW-derived convex hull. Thus, it can be concluded that this observation indicates the potential stability of the generated structures rather than differences in the computational methods used in this work and during AFLOW generation. Another four structures also have energies below the convex hull, but in the vicinity of it. Thus, their potential stability should be interpreted with caution.

8 Data availability

The raw crystal dataset is downloaded from
https://aflowlib.org

Table 3: Inference results. Each matrix element corresponds to either the minimal energy above the hull achieved in an experiment or the energy above the hull of structures with energies below the hull.

		Ta1W1B6, meV/atom	Ta1W2B5, meV/atom	Ta2W1B5, meV/atom
Diffusion	energy step = 0.01 energy gap = 0	10,41	3,275	13,079
	energy step = 0.1 energy gap = 0	11,835	3,83	**−0, 042**
	energy step = 0.01 energy gap = 1	97,676	**−1, 409**	5,539
Flow-Matching	energy step = 0.01 energy gap = 0	11,981	**−0, 483** **−0, 466** **−0, 387**	**−5, 426**
	energy step = 0.1 energy gap = 0	11,286	0,037	**−5, 497**
	energy step = 0.01 energy gap = 1	9,529	**−4, 852**	1,029

9 Code availability

The source code for training and inferencing our models can be obtained from GitHub at https://github.com/AIRI-Institute/conditional-crystal-generation

10 Conclusion

In this article, we have offered two approaches to generate crystal structures: conditional generation and conditional modification. The first approach is significantly more flexible as it does not require structure-polymorphs, enabling the generation of structures without restrictions on chemical composition, which can be crucial in certain scenarios. Another advantage of the first approach is the simplicity of data preprocessing; it only requires the chemical composition, space group, atom coordinates, and formation energies.

Our methodology has experimentally proven its effectiveness, resulting in four confident potentially new crystal structures with the following energies above the hull: {-1.409, -5.497, -5.426, and -4.852} meV/atom, and four uncertain candidates with energies of {-0.483, -0.466, -0.387, and -0.042} meV/atom. We have demonstrated that conditional generation approaches, commonly used in image generation, are also fruitful in the design of new materials.

Although the proposed methodology demonstrates its efficiency in generating potentially new crystal structures, it has certain limitations. Firstly, the data is represented in a matrix form, which does not account for all possible symmetries of the crystal structures. Secondly, the structures in the dataset range from 4 to 60 atoms per unit cell, with most structures containing fewer than 8 atoms per unit cell. However, to perform well on structures with a large number of atoms per unit cell, the models just should be pretrained on a dataset that includes larger structures.

Furthermore, despite the limited number of experiments(6) and structures generated (7182), we succeeded in identifying hypothetically new structures. We hope that our article will help to reveal the potential of generative AI in design of new materials with targeted thermodynamic properties and inspire other researchers to be part of this innovative journey in materials design. We believe that rapid and efficient generation of novel materials can lead to breakthroughs in various fields such as electronics, pharmaceuticals, and energy storage. This can accelerate technological advancements and make cutting-edge technologies more accessible and affordable.

References

[1] Diola Bagayoko. Understanding density functional theory (dft) and completing it in practice. AIP Advances, 4(12), 2014.

[2] Alexander Dunn, Qi Wang, Alex Ganose, Daniel Dopp, and Anubhav Jain. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. npj Computational Materials, 6(1):138, 2020.

[3] Roman A Eremin, Innokentiy S Humonen, Alexey A Kazakov, Vladimir D Lazarev, Anatoly P Pushkarev, and Semen A Budennyy. Graph neural networks for predicting structural stability of cd-and zn-doped γ-cspbi3. Computational Materials Science, 232:112672, 2024.

[4] Alexey N Korovin, Innokentiy S Humonen, Artem I Samtsevich, Roman A Eremin, AI Vasilev, Vladimir D Lazarev, and Semen A Budennyy. Boosting heterogeneous catalyst discovery by structurally constrained deep learning models. Materials Today Chemistry, 30:101541, 2023.

[5] Alexandre Duval, Simon V Mathis, Chaitanya K Joshi, Victor Schmidt, Santiago Miret, Fragkiskos D Malliaros, Taco Cohen, Pietro Liò, Yoshua Bengio, and Michael Bronstein. A hitchhiker's guide to geometric gnns for 3d atomic systems. arXiv preprint arXiv:2312.07511, 2023.

[6] Zekun Ren, Siyu Isaac Parker Tian, Juhwan Noh, Felipe Oviedo, Guangzong Xing, Jiali Li, Qiaohao Liang, Ruiming Zhu, Armin G Aberle, Shijing Sun, et al. An invertible crystallographic representation for general inverse design of inorganic crystals with targeted properties. Matter, 5(1):314–335, 2022.

[7] Yong Zhao, Mohammed Al-Fahdi, Ming Hu, Edirisuriya MD Siriwardane, Yuqi Song, Alireza Nasiri, and Jianjun Hu. High-throughput discovery of novel cubic crystal materials using deep generative neural networks. Advanced Science, 8(20):2100566, 2021.

[8] Yong Zhao, Edirisuriya M Dilanga Siriwardane, Zhenyao Wu, Nihang Fu, Mohammed Al-Fahdi, Ming Hu, and Jianjun Hu. Physics guided deep learning for generative design of crystal materials with symmetry constraints. npj Computational Materials, 9(1):38, 2023.

[9] Teerachote Pakornchote, Natthaphon Choomphon-Anomakhun, Sorrjit Arrerut, Chayanon Atthapak, Sakarn Khamkaeo, Thiparat Chotibut, and Thiti Bovornratanaraks. Diffusion probabilistic models enhance variational autoencoder for crystal structure generative modeling. Scientific Reports, 14(1):1275, 2024.

[10] Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Sasha Shysheya, Jonathan Crabbé, Lixin Sun, Jake Smith, et al. Mattergen: a generative model for inorganic materials design. arXiv preprint arXiv:2312.03687, 2023.

[11] Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. Nature, 624(7990):80–85, 2023.

[12] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. arXiv preprint arXiv:2210.02747, 2022.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[14] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In International Conference on Learning Representations, 2021.

[15] Guangyu Sun, Jenö Kürti, Péter Rajczy, Miklos Kertesz, Jürgen Hafner, and Georg Kresse. Performance of the vienna ab initio simulation package (vasp) in chemical applications. Journal of Molecular Structure: THEOCHEM, 624(1-3):37–45, 2003.

[16] Stefano Curtarolo, Wahyu Setyawan, Gus LW Hart, Michal Jahnatek, Roman V Chepulskii, Richard H Taylor, Shidong Wang, Junkai Xue, Kesong Yang, Ohad Levy, et al. Aflow: An automatic framework for high-throughput materials discovery. Computational Materials Science, 58:218–226, 2012.

[17] Łukasz Mentel. mendeleev – a python resource for properties of chemical elements, ions and isotopes.

[18] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In International Conference on Machine Learning, pages 8162–8171. PMLR, 2021.

[19] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In The Eleventh International Conference on Learning Representations, 2023.

[20] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models, 2022.

[21] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 31. Curran Associates, Inc., 2018.

[22] Chi Chen and Shyue Ong. A universal graph deep learning interatomic potential for the periodic table. Nature Computational Science, 2:718–728, 11 2022.

A Appendix section 1

A.1 Distribution of atomic coordinates

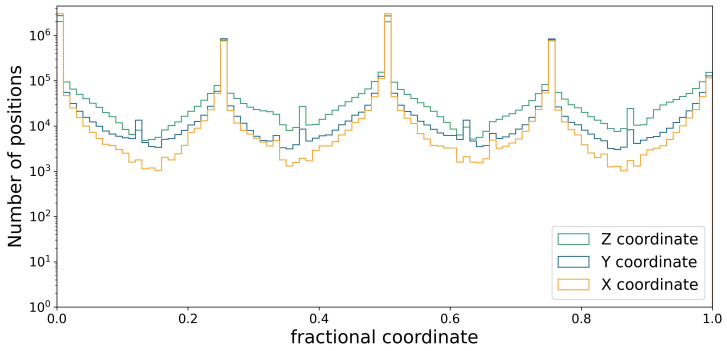


Figure 3: Distribution of the components of fractional atomic coordinates (X, Y, Z)

A.2 Pseudocode

Algorithm 1 Training Regression Modification Model

repeat
  $x_0 \sim q(x_0); x_1 \sim q(x_1); el \sim q(el); spg_1 \sim q(spg_1); E \sim q(E)$
  $\mathcal{L} \leftarrow ||x_1 - f_\theta(x_0, t, el, spg_1, E)||$
  $\theta \leftarrow Update(\theta, \nabla_\theta \mathcal{L}(\theta))$
until converge

Algorithm 2 Inferencing Regression Modification Model

$x_0 \sim q(x_0); el \sim q(el); spg_1 \sim q(spg_1); E \sim q(E)$
$x_1 = f_\theta(x_0, t, el, spg_1, E)$
return x_1

Algorithm 3 CFM Training

repeat
  $x_0 \sim q(x_0); x_1 \sim q(x_1)$
  $t \sim \mathcal{U}(0, 1)$
  $x_t = tx_1 + (1 - t)x_0$
  $\mathcal{L}_{CFM} \leftarrow ||f_\theta(x_t, t) - (x_1 - x_0)||$
  $\theta \leftarrow Update(\theta, \nabla_\theta \mathcal{L}_{CFM}(\theta))$
until converge

Algorithm 4 Training CFM for Modification

repeat
  $x_0 \sim q(x_0); x_1 \sim q(x_1); el \sim q(el); spg_1 \sim q(spg_1); E \sim q(E)$
  $t \sim \mathcal{U}(0, 1)$
  $x_t = tx_1 + (1 - t)x_0$
  $\mathcal{L}_{CFM} \leftarrow ||f_\theta(x_t, t, el, spg_1, E) - (x_1 - x_0)||$
  $\theta \leftarrow Update(\theta, \nabla_\theta \mathcal{L}_{CFM}(\theta))$
until converge

Algorithm 5 Training CFM for Generation

repeat
  $x_0 \sim \mathcal{N}(0,1)$ or $x_0 \sim \mathcal{U}(0,1)$
  $x_1 \sim q(x_1); el \sim q(el); spg_1 \sim q(spg_1); E \sim q(E)$
  $t \sim \mathcal{U}(0,1)$
  $x_t = tx_1 + (1-t)x_0$
  $\mathcal{L}_{CFM} \leftarrow ||f_\theta(x_t, t, el, spg_1, E) - (x_1 - x_0)||$
  $\theta \leftarrow Update(\theta, \nabla_\theta \mathcal{L}_{CFM}(\theta))$
until converge

Algorithm 6 Sampling with CFM for Modification or Generation

$h = \frac{1}{T}$
$x_0 \sim q(x_0)$ or $x_0 \sim \mathcal{N}(0,1)$ or $x_0 \sim \mathcal{U}(0,1)$
$el \sim q(el); spg_1 \sim q(spg_1); E \sim q(E)$
for $t = 1, \ldots, T$ do
  $x_{t+1} = x_t + hf_\theta(x_t, t, el, spg_1, E)$
end for
return x_1

Algorithm 7 Training DM for Modification

repeat
  $x_0 \sim q(x_0); x_1 \sim q(x_1); el \sim q(el); spg_1 \sim q(spg_1); E \sim q(E)$
  $t \sim \mathcal{U}(\{1, \ldots, T\})$
  $\epsilon \sim \mathcal{N}(0, I)$
  $\mathcal{L}_D \leftarrow ||\epsilon - f_\theta(\sqrt{\bar{\alpha}_t}x_1 + \sqrt{1-\bar{\alpha}_t}\epsilon, x_0, t, el, spg_1, E)||$
  $\theta \leftarrow Update(\theta, \nabla_\theta \mathcal{L}_D(\theta))$
until converge

Algorithm 8 Sampling with DM for Modification

$x_T \sim \mathcal{N}(0, I)$
for $t = T, \ldots 1$ do
  $x_0 \sim q(x_0); x_1 \sim q(x_1); el \sim q(el); spg_1 \sim q(spg_1); E \sim q(E)$
  $z \sim \mathcal{N}(0, I)$ if t > 1 else z $= 0$
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} f_\theta(x_t, x_0, t, el, spg_1, E)) + \sqrt{1-\alpha_t}z$
end for
return x_1

Algorithm 9 Training DM for Generation

repeat
  $x_1 \sim q(x_1); el \sim q(el); spg_1 \sim q(spg_1); E \sim q(E)$
  $t \sim \mathcal{U}(\{1, \ldots, T\})$
  $\epsilon \sim \mathcal{N}(0, I)$
  $\mathcal{L}_D \leftarrow ||\epsilon - f_\theta(\sqrt{\bar{\alpha}_t}x_1 + \sqrt{1-\bar{\alpha}_t}\epsilon, t, el, spg_1, E)||$
  $\theta \leftarrow Update(\theta, \nabla_\theta \mathcal{L}_D(\theta))$
until converge

Algorithm 10 DDPM Sampling

$x_T \sim \mathcal{N}(0, I)$
for $t = T, \ldots 1$ do
  $x_1 \sim q(x_1); el \sim q(el); spg_1 \sim q(spg_1); E \sim q(E)$
  $z \sim \mathcal{N}(0, I)$ if t > 1 else z $= 0$
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} f_\theta(x_t, t, el, spg_1, E)) + \sqrt{1-\alpha_t}z$
end for
return x_1

Algorithm 11 DDIM Sampling

$x_T \sim \mathcal{N}(0, I)$
for dot $= T, \dots 1$ with step C do
  $x_0 \sim q(x_0); x_1 \sim q(x_1); el \sim q(el); spg_1 \sim q(spg_1); E \sim q(E)$
  $z \sim \mathcal{N}(0, I)$ if t > 1 else z $= 0$
  $x_\theta = f_\theta(x_t, t, el, spg_1, E)$
  $x_{t-1} = \sqrt{\alpha_{t-1}}(\frac{x_t - \sqrt{1-\alpha_t} x_\theta}{\sqrt{\alpha_t}}) + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} x_\theta + \sigma_t z$

end for
return x_1

A.3 Experiment Details

All the experiments use the same hyperparameters for the model:

- num_res_blocks = 7
- attention_resolution = (1, 2, 4, 8)
- model_channels = 128

In all the experiments models are trained with the same training parameters:

- optimizer = Adam
  - betas = (0.9, 0.999)
  - eps = 1e-08
  - weight_decay = 0
- batch_size = 256
- epochs = 400
- learning_rate = 1e-4
- lr_warmup_steps = 500
- random_state = 42

An important note, that all our experiments have been conducted in mixed precision in fp16.

Generation task: Diffusion Model (DDPM):

- num_train_timesteps = 1000 (diffusion process discretization)
- beta_start = 0.0001
- beta_end = 0.02
- num_inference_steps = 100
- beta_schedule = "squaredcos_cap_v2" (cosine)

Diffusion Model (DDIM):

- num_train_timesteps = 1000 (diffusion process discretization)
- beta_start = 0.0001
- beta_end = 0.02
- num_inference_steps = 100
- beta_schedule = "squaredcos_cap_v2" (cosine)

Flow Matching $x_0 \sim \mathcal{N}(0, 1)$:

- num_inference_steps = 100

Flow Matching $x_0 \sim \mathcal{U}(0, 1)$:

- num_inference_steps = 100

Modification task:

Regression UNet:

- num_inference_steps = 1

Diffusion Model:

- num_train_timesteps = 1000 (diffusion process discretization)
- beta_start = 0.0001
- beta_end = 0.02
- num_inference_steps = 100
- beta_schedule = "squaredcos_cap_v2" (cosine)

Flow Matching:

- num_inference_steps = 100

A.4 PBC Loss details

The PBC loss function operates through several steps:

1. Vertices evaluation: If the target coordinate of the atom is lattice vertex (all 3 coordinates $x, y, z$are equal to 1 or 0), then loss between prediction point $preds_i$ and target point $target_i$ is being calculated using following formula:

$$L_{vertex}(preds_i, target_i) = \min_{v \in vertices} ||preds_i - v||,$$

   where $vertices$ is a set of 8 possible positions according to PBC ($\{0,0,0\}$, $\{0,0,1\}$, ..., $\{1,1,1\}$).

2. Edges evaluation: If the target coordinate of the atom is located on lattice edge (two coordinates are equal to 1 or 0 and one is not). For example, a lattice edge atom at point $\{0,1,0.3\}$ has identical atoms at points $\{0,0,0.3\}, \{1,0,0.3\}, \{1,1,0.3\}$. As we can see, in this example z-coordinate is fixed but x and y are exchangeable. Therefore, if the target point is represented as $\{x,y,z\}$, we can use the following formula:

$$L_{edge}(preds_i, target_i) = \min_{e \in edgePoints} ||preds_i - e||,$$

   where $edgePoints$ is a set of 4 possible positions according to PBC.

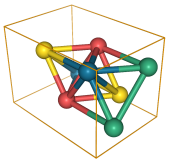
   - Case of fixed point x: $edgePoints = \{\{x,0,0\}, \{x,0,1\}, \{x,1,0\}, \{x,1,1\}\}$
   - Case of fixed point y: $edgePoints = \{\{0,y,0\}, \{0,y,1\}, \{1,y,0\}, \{1,y,1\}\}$
   - Case of fixed point z: $edgePoints = \{\{0,0,z\}, \{0,1,z\}, \{1,0,z\}, \{1,1,z\}\}$

3. Sides evaluation: If the target coordinate of the atom is located on lattice side (one coordinate is equal to 1 or 0 and two are not). For example, lattice side atom at point $\{0,0.5,0.3\}$ has identical atom at point $\{1,0.5,0.3\}$. In this example y and z coordinates are fixed but x is exchangeable. Therefore, if the target point is represented as $\{x,y,z\}$, we can use the following formula:

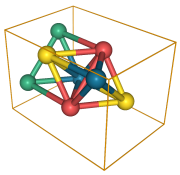
$$L_{size}(preds_i, target_i) = \min_{s \in sidePoints} ||preds_i - s||,$$

   where $sidePoints$ is a set of 2 possible positions according to PBC.

   - Case of exchangeable point x: $sidePoints = \{\{0,y,z\}, \{1,y,z\}\}$
   - Case of exchangeable point y: $sidePoints = \{\{x,0,z\}, \{x,1,z\}\}$
   - Case of exchangeable point z: $sidePoints = \{\{x,y,0\}, \{x,y,1\}\}$



(a) Target structure



(b) Prediction

Figure 4: Example of using PBC-aware loss. The depicted structures (Mo2Nb2Ta2W2) are visually different, but in fact they are exact the same. It is confirmed by insignificant value of PBC-aware loss

4. Points, which don't belong to the groups above, are processed using the default loss function.

Since the $\min(x_1, x_2, ..., x_n)$ function is undifferentiable at multiple points ($x_i = x_j \ \forall i \neq j$), it makes a loss function to have a more complicated surface. Therefore, we used a norm function with order $k \to -\infty$ which is differentiable at all points as a replacement.

$$min_{diff}(x_1, x_2, ..., x_n) = (\sum_{i=1}^{n} |x_i|^k)^{\frac{1}{k}} \quad k \longrightarrow -\infty$$

Therefore, overall PBC-aware loss for a structure is represented as:

$$\begin{aligned} L_{PBC}(preds, target) = \sum_{i=1}^{n} & \mathbb{I}(target_i \text{ is vertex point}) L_{vertex}(preds_i, target_i) \\ & + \mathbb{I}(target_i \text{ is edge point}) L_{edge}(preds_i, target_i) \\ & + \mathbb{I}(target_i \text{ is side point}) L_{side}(preds_i, target_i) \\ & + \mathbb{I}(target_i \text{ is usual point}) L_2(preds_i, target_i) \end{aligned}$$

As the count of atoms varies across different structures, the L_{PBC} metric tends to yield higher values for structures featuring a larger number of atoms. Thus, it is important to normalize the loss function with the number of atoms in the structure if it would be used in batches with structures with different number of atoms. Therefore, a PBC-aware loss for a batch of structures is formulated as:

$$\begin{aligned} & L_{batchPBC}(batchPreds, batchTargets) = \\ & \sum_{i=1}^{batchSize} \frac{1}{numSites_i} L_{PBC}(batchPreds_i, batchTargets_i) \end{aligned}$$

Compute resources

For our computational needs in model training and inference, we deployed a total of three GPU servers with the following configurations:

Server 1:

- GPU: NVIDIA A100/80G
- CPU: 8vCPU of Intel(R) Xeon(R) Gold 6248R @ 3.00 GHz
- RAM: 64Gb

Server 2:

- GPU: NVIDIA V100 (32GB)
- CPU: 8vCPU of Intel(R) Xeon(R) Gold 6278C @ 2.60 GHz
- RAM: 64Gb

Server 3:

- GPU: NVIDIA V100 (32GB)
- CPU: 8vCPU of Intel(R) Xeon(R) Gold 6278C @ 2.60 GHz
- RAM: 64Gb

Every model training time consumed up to 2 weeks employing computing power of one GPU server.

For the ab-initio calculations implemented in VASP, we deployed a total of 5 identical CPU servers with the following configurations:

- CPU: 64vCPU of Intel(R) Xeon(R) Gold 6278C CPU @ 2.60GHz
- RAM: 256Gb

Structure relaxation with VASP for all six experiments mentioned in Table 3 took more that 180 thousand CPU hours. Computing power of all CPU servers was employed.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: The papers not including the checklist will be desk rejected. The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- Delete this instruction block, but keep the section heading “NeurIPS paper checklist",
- Keep the checklist subsection headings, questions/answers and guidelines below.
- Do not modify the questions and only use the provided macros for your answers.

1. Claims

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes] ,

   Justification: The main claims made in the abstract and introduction do accurately reflect the paper's contributions and scope. Every aspects mentioned in the abstract and introduction are further revealed in the main paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: All the limitations are discussed in the conclusion section. More specifically, limitations are 1) in number of atoms per unit cell of crystal structure that model can work with and 2) symmetries that our structure representation is able to encode. For example, crystals have infinite periodic structure. There is no possible way to

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA] .

Justification: Our work does not include significant theoretical results due to the fact that it is mainly focused on experiments.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.

- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all the information needed to reproduce the main experimental results. The information includes: 1) all hyperparameters for the models, including the random seed they were performed on 2) crystal structures used as a data, 3) VASP settings.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The raw crystal dataset can be downloaded from https://aflowlib.org. The source code for training and inferencing our models can be obtained from GitHub at https://github.com/AIRI-Institute/conditional-crystal-generation. Both data and code are also referenced in the main paper's Section 9 and Section 9 .

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All the training and testing details and all the hyperparameters for the experiments are mentioned in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Firstly, calculating statistical significance for all our experiments is computationally expensive. Secondly, experimental results(crystal structures obtained from model inference) have been approved by further ab-initio calculations implemented in VASP.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In the paper, we state all the compute powers that we have employed for AI model training and ab-initio calculations. One can observe it in Section A.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research, working process, data and code correspond with NeurIPS Code of Ethics in https://neurips.cc/public/EthicsGuidelines

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The broad societal impacts are discussed int the conclusion section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The models utilized in this research are appropriately cited within the text, including references to the AFLOW database. We have no commercial interests related to the use of these models and the crystal structure database. The code used in this study was entirely developed by our team.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New Assets

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The primary assets of our research are: 1) the methodology, and 2) the code for training models and performing inference. Both are thoroughly documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and Research with Human Subjects

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.