

AI-Assisted Rapid Crystal Structure Generation Towards a Target Local Environment

Osman Goni Ridwan,¹ Sylvain Pitié,² Monish Soundar Raj,³ Dong Dai,⁴ Gilles Frapper,^{2, a)} Hongfei Xue,^{3, b)} and Qiang Zhu^{1, 5, c)}

¹⁾*Department of Mechanical Engineering and Engineering Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA*

²⁾*Applied Quantum Chemistry Group, Poitiers University-CNRS, Poitiers 86073, France*

³⁾*Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA*

⁴⁾*Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA*

⁵⁾*North Carolina Battery Complexity, Autonomous Vehicle and Electrification (BATT CAVE) Research Center, Charlotte, NC 28223, USA*

(Dated: 8 September 2025)

In the field of material design, traditional crystal structure prediction approaches require structural sampling through computationally expensive energy minimization methods using either force fields or quantum mechanical simulations. While emerging artificial intelligence (AI) generative models have shown great promise in generating realistic crystal structures more rapidly, most existing models fail to account for the unique symmetries and periodicity of crystalline materials, and they are limited to handling structures with only a few tens of atoms per unit cell. Here, we present a symmetry-informed AI generative approach called Local Environment Geometry-Oriented Crystal Generator (LEGO-xtal) that overcomes these limitations. Our method generates initial structures using AI models trained on an augmented small dataset, and then optimizes them using machine learning structure descriptors rather than traditional energy-based optimization. We demonstrate the effectiveness of LEGO-xtal by expanding from 25 known low-energy sp² carbon allotropes to over 1,700, all within 0.5 eV/atom of the ground-state energy of graphite. This framework offers a generalizable strategy for the targeted design of materials with modular building blocks, such as metal–organic frameworks and next-generation battery materials.

Keywords: Materials Informatics, Crystal Structure Prediction, Machine Learning

CONTENTS

I INTRODUCTION	2	B The sp ² Carbon Allotrope Database	10
II Computational Methodology	3	1 Distribution of low-energy crystals by complexity	10
A Symmetry-based Crystal Representation	3	2 The 0-2D low-energy sp ² carbon allotropes	11
B Practical Challenges of AI Generative Models for Crystal Structure Prediction	3	3 The 3D Negative Curved Graphite allotropes	12
C Symmetry and Geometry Considerations	4		
III The LEGO-xtal framework	5	V Conclusions	13
A Training Data Augmentation & Tabularization	5	Acknowledgments	14
B Generative Model Training & Sampling	6	Data availability	14
C Structure Relaxation to Targeted Local Environment	7	Conflict of interest	14
D Energy Ranking & Database	8	Author contributions	14
E Approximated Time Costs	8		
IV Results & Discussions	9	S1 Generative Models Training and Sampling	15
A Model Performances Evaluation	9	A Generative Adversarial Network (GAN)	15
		B Variational Autoencoder (VAE)	16
		S2 The SO(3) Descriptor	17
		S3 Pre-relaxation	19
		A Descriptor-based Geometry Optimization	19
		B Geometry Optimization under Subgroup Representation	19
		C Pre-relaxation Bias Analysis	19

^{a)}Electronic mail: gilles.frapper@univ-poitiers.fr

^{b)}Electronic mail: hongfei.xue@charlotte.edu

^{c)}Electronic mail: qzhu8@charlotte.edu

D Comparison of Different Relaxation Strategies	20
E Pre-relaxation Examples on the Multi-component SiO ₂ System	21
S4 Impact of Training data	21
A Pre-filtered Structures	21
B Data Augmentation	22
C Subgroup Augmentation Details	22
S5 Analysis of Generative Model Performance	23
A Comparison with Random Sampling	23
B Saturation Analysis	23
S6 An Extended List of sp² Allotropes	23
A The 2D sp ² Allotropes	23
B The Low-energy 3D and 0D sp ² Allotropes	23
C Phonon Calculations for Selected Structures	24
REFERENCES	24

I. INTRODUCTION

For new materials design, the ability to predict the crystal structure of a material is crucial for understanding its properties and potential applications. However, crystal structure prediction (CSP) has long been considered a grand challenge in materials science. Analogous to protein structure prediction¹, solving a typical CSP problem requires searching for the minimum energy arrangement of 10–100 atoms within an unknown periodic crystal lattice². This formulation results in a combinatorial search space of astronomical size^{3–7}. Additionally, evaluating each trial solution involves time-intensive quantum mechanical simulations, making the problem even more formidable.

Around the 2000s, the CSP field experienced a rapid development of non-empirical structure search algorithms by borrowing ideas from the global optimization community². Several algorithms—including random search³, genetic/evolutionary algorithms^{4,8–10}, simulated annealing¹¹, and basin hopping¹² were successfully applied to various material systems under both ambient and extreme conditions. These successes were largely attributed to strategies aimed at reducing dimensionality at the algorithmic level^{2,3,13}. While the number of possible structural arrangements is infinite, many can be immediately discarded due to unfavorable energy costs. By incorporating reasonable initial guesses (e.g., imposing chemical and symmetry constraints^{3,13}) and efficient structure-switching mechanisms (e.g., following physical vibration modes^{14–16}, and self-learning intelligence^{4,8}), researchers were able to focus on exploring low-energy basins with a much reduced computational cost. However, these traditional approaches remain heavily reliant on high-end electronic structure prediction methods, which come with a computational cost scaling relation $O(N^3)$ with respect to the number of valence electrons of the given system. As a result, a practical CSP task is limited to 20–30 atoms per periodic system, requiring several days to weeks on a single CPU node consisting of 48–96 cores.

Recently, the advent of artificial intelligence (AI) generative models has opened up new avenues for rapid crystal structure generation. Inspired by the successes of synthetic image generation and protein structure prediction¹, various AI-based generative models have been introduced to the materials community since 2018, including Generative Adversarial Networks (GANs)^{17,18}, Variational Autoencoders (VAEs)¹⁹, and more recent active learning²⁰, diffusion^{21–27} and transformer models^{28–31}. These models offer a fundamentally new approach to exploring structure space by learning data-driven representations, allowing for the efficient generation of low-energy crystal structures at a much faster rate than traditional global optimization methods. However, the direct application of AI-based models to the CSP task faces a significant challenge: the generated structures often lack diversity or fail to satisfy essential physical constraints, as these models tend to either replicate structural prototypes from the training data, or generate invalid and unstable structure, rather than producing truly novel and physical-plausible structures³². This limitation arises because these data-driven models primarily generate crystal structures only based on the statistical patterns from the training data source without incorporating the unique characteristics of crystalline materials, particularly their symmetries. To address the challenge, several recent works have attempted to incorporate symmetry into generative AI model design with the aim of enhancing the quality and validity of generated structures^{18,24,25,28,30,31}. However, most of the previous works simply add each individual symmetry representation into the existing AI models. Consequently, the learned models may be able to generate new candidate structures with a symmetry identical to that in the training samples. Nevertheless, they are less likely to generate those closely symmetry-related structures, thus largely limiting the model’s learning efficiency.

In this work, we present a novel symmetry-informed generative model called LEGO-xtal (Local Environment Geometry-Oriented Crystal Generator) that overcomes the limitations of existing AI models in crystal structure prediction. Unlike the previous AI generative models that focus on generation of relatively small crystal systems from a large dataset without the explicit constraints on the local chemical environment, our approach is designed to generate complex crystal structures with well-defined local environments (i.e. specified chemical building blocks) while preserving the inherent symmetries and periodicity characteristic of crystalline materials. As a proof of concept, we train our generative models (VAEs and GANs) on an augmented dataset of known three-coordinated carbon allotropes—commonly referred to as sp² carbon structures—which enables the exploration of a broad range of structural configurations. Starting from only 25 known low-energy sp² carbon allotropes (as well as additional 115 high-energy examples), we demonstrate the effectiveness of our approach by generating over 1,700 new structures with varied dimensionality, topology and unit cell sizes.

The remainder of this paper is organized as follows: First, we discuss the practical challenges in using AI models for crystal structure prediction and introduce our key solutions. Then, we present the detailed methodology of the

LEGO-xtal framework. Finally, we analyze the generated structures and highlight several interesting findings.

II. COMPUTATIONAL METHODOLOGY

Unlike computer vision objects like images and point clouds that only require limited local information, crystal structures exhibit distinct long-range translational and local rotational symmetries. These symmetry elements are rigorously defined through crystallographic space groups using group theory³³. When combined with specific chemical systems, crystal chemistry provides valuable rules and heuristics to guide our understanding of crystal packing and structure formation². To design an effective structure prediction model, we need to carefully consider how to incorporate these crystallographic and chemical principles into the modeling framework.

A. Symmetry-based Crystal Representation

To store and exchange crystal structure information, the Crystallographic Information File (CIF) is the most common and standardized text-based file format, containing atomic positions, unit cell parameters, and symmetry information³⁴. For example, the well-known diamond structure can be compiled into a tabular format as shown in Figure 1. Diamond crystallizes in space group $Fd\bar{3}m$ (SG 227), with a cubic unit cell ($a = b = c = 3.567 \text{ \AA}$, and $\alpha = \beta = \gamma = 90^\circ$). In each unit cell, there are 8 symmetry-equivalent carbon atoms occupying an orbit, forming $\bar{4}3m$ site symmetry along the crystallographic [100], [111], and [110] directions³⁵. In crystallography, the Wyckoff site notation is used to describe the arrangement of atoms in a crystal structure. Each Wyckoff site is defined by its symmetry properties and multiplicity, which indicates how many equivalent positions exist for that site within the unit cell. The notation typically consists of a number indicating the multiplicity and a letter indicating the symmetry type (e.g., 8a, 4b, etc.). In this case, the diamond structure has 8 carbon atoms occupying Wyckoff site 8a, which is characterized by its high symmetry (see Figure 1b).

Using this notation, it is straightforward to consider each structure in a tabular format with the following attributes.

- *Space Group number* which ranges from 1 to 230.
- *Cell Parameters*: $a, b, c, \alpha, \beta, \gamma$, which is a 6-vector that defines the cell lengths and angles in a standard crystallographic setting.
- *Wyckoff site 1*: (index, element type, x, y, z), which represents the index of Wyckoff site and the associated fractional coordinates bounded in the (0, 1) interval.
- *Wyckoff site 2*: (index, element type, x, y, z).
- ...

Thus, one essentially needs two sets of variables to describe a structure: (i) the *discrete variables* denoting the exact space group symmetry and Wyckoff site indices; and (ii) the *continuous variables* describing the cell parameters and Wyckoff positions. For high symmetry cases, the cell parameters and (x, y, z) coordinates may have additional constraints. For instance, cubic unit cells must have $a = b = c$ and $\alpha = \beta = \gamma = 90^\circ$. The Wyckoff position may be completely or partially restricted to specific fractional coordinates due to site symmetry requirements. If input variables do not satisfy symmetry requirements, deterministic algorithms can check if variables can be symmetrized within an acceptable threshold. Considering the symmetry constraints, the free continuous variables are thus reduced to a 1-length vector [3.567] for the diamond carbon.

Therefore, we can use a fixed-length 1D vector to represent a crystal structure with a maximum number of Wyckoff positions. For a maximum of 8 Wyckoff positions, the vector length is $1 + 6 + 8 \times 5 = 47$. For a single-component system, this reduces to 39.

Although this representation is straightforward, it is not mathematically unique. In principle, one crystal structure can be represented by many different 1D vectors by choosing different choices of (x, y, z) coordinates in a given Wyckoff site. To explicitly account for the symmetry, we can augment the representation by considering all possible multiplicities in each Wyckoff site as shown in Figure 1c.

B. Practical Challenges of AI Generative Models for Crystal Structure Prediction

Using this representation, the objective of CSP is to find or generate the best combination of these variables that leads to a low-energy configuration. Inspired by the successes of AI generative models for image, point cloud, or text, we aim to develop a model to learn the statistical distribution from the training data for both discrete and continuous variables, and then generate new crystal representations. However, there are several challenges that prohibit the direct application of existing AI generative models to crystal generation.

First, there is a risk of lacking sufficient training data to describe the design space for the discrete variables (e.g., space group and Wyckoff site choices). In the context of crystal structure prediction, most materials generation tasks require the researchers to generate likely candidates from only a limited choices of good examples. However, with a total of 230 space groups and 1731 Wyckoff site choices, even for a simple unit cell of 10 atoms, the design space becomes astronomically large. With limited training data in lack of diversity and coverage, it is likely to lead to overfitting and poor generalization, resulting in generated structures that are too similar to the training data or fail to explore the design space effectively.

Second, assuming the discrete design space (i.e. space group and Wyckoff site choices) has been sufficiently covered, the next challenge is to generate a valid crystal structure under the symmetry constraints. Currently, there exist two common practices to generate the continuous variables:

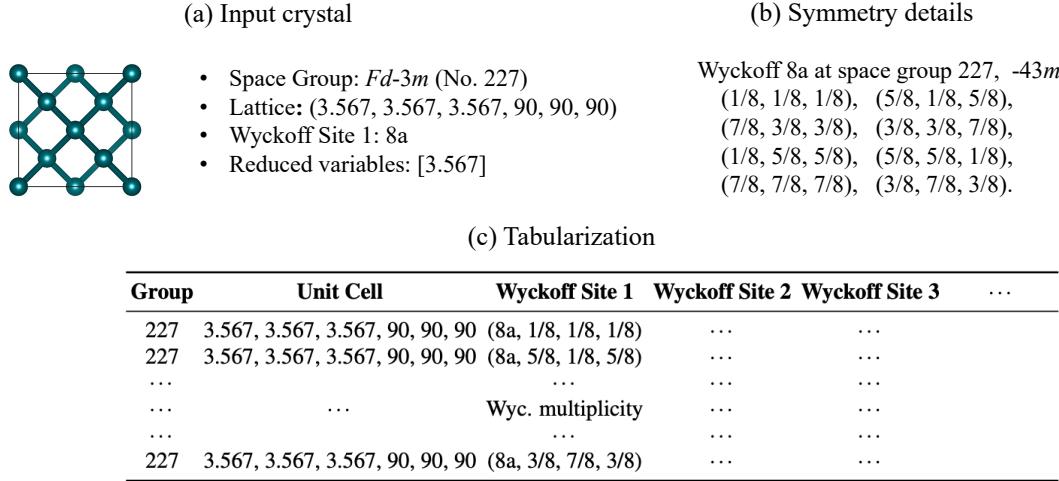


FIG. 1. Illustration of symmetry encoded crystal representation. (a) shows the example diamond crystal structure, (b) lists the associated Wyckoff symmetry operations to describe the carbon atoms in the diamond; (c) lists the complete tabular representation of a diamond crystal according to its highest space group symmetry 227.

(i) directly applying the diffusion model to generate the continuous variables for unit cell and Wyckoff sites^{24,25}; or (ii) combining random sampling with energy-based optimization to search for the low-energy structures^{18,31}. The first approach relies on the availability of sufficient training data and may not be applied to small data tasks. Although the second approach is more reliable, for a specific space group and Wyckoff site choice, we need to sample a large number of random configurations and then optimize to find the low-energy states. This is often time-consuming and the quality of the optimized solution is overly reliant on the initial guess.

Third, materials researchers may not only seek target structures optimized for a single metric such as energy or a specific property. In practice, materials are often designed to incorporate established structural motifs or to maintain a preferred local environment (e.g., specific coordination geometry). To our knowledge, existing generative models do not explicitly enforce local geometry during structure generation. As a result, generated structures may deviate from the desired local motifs, limiting their practical relevance for targeted materials design.

C. Symmetry and Geometry Considerations

To effectively address these fundamental challenges in crystal structure prediction with AI generative models, we propose two key numerical recipes that significantly improve the model's performance as illustrated in Figure 2.

Data augmentation by subgroup symmetry relations. By default, a crystal structure is assigned to the highest possible space group symmetry. However, this assignment does not convey the symmetry information comprehensively. For example, the diamond structure can be represented by its highest symmetry space group $Fd\bar{3}m$ (227) with Wyckoff site 8a. The cubic boron nitride can be viewed as a split of 8a

Wyckoff site into two Wyckoff sites (4a and 4d) in a lower space group symmetry $F\bar{4}3m$ (216). While these two structures are closely related by symmetry breaking, this relationship cannot be recognized by simply comparing their tabular representations using the highest symmetry representation. In fact, such a Wyckoff split is rather common in crystal structures, especially during phase transitions or structural transformations³⁵. Following our recent study on symmetry relations in ferroelectric phase transitions, we leverage our previously developed subgroup symmetry module within the PyXtal library³⁶ to generate a rich set of subgroup representations. Using diamond crystal as an example (see Figure 2a), we can generate multiple alternative subgroup symmetry representations to provide more training data. This approach enables sophisticated statistical models to thoroughly learn the distribution of space group/Wyckoff combinations in a more comprehensive design space. After training on the symmetry-augmented samples, the model can then generate not just exact replicates of the highest symmetry space groups from the training data, but also closely related subgroup symmetry alternatives.

Pre-relaxation with respect to the reference local environment. Once the model is capable of suggesting a promising candidate crystal, we then need to optimize the remaining continuous variables to get the realistic structure, under the conditioned space group symmetry constraint. In many practical applications, we may already know how the atoms may behave in a short-range cutoff. For instance, carbon may take sp, sp², or sp³ local bonding; Porous materials such as zeolites and metal–organic frameworks (MOFs) are often categorized by their well-defined molecular building units, making them ideal targets for generative approaches that preserve local chemical environments. Therefore, we can utilize this knowledge to predefine the reference environment according to the recently developed power spectrum descriptors from spherical harmonics expansion^{37–41}, and then relax the struc-

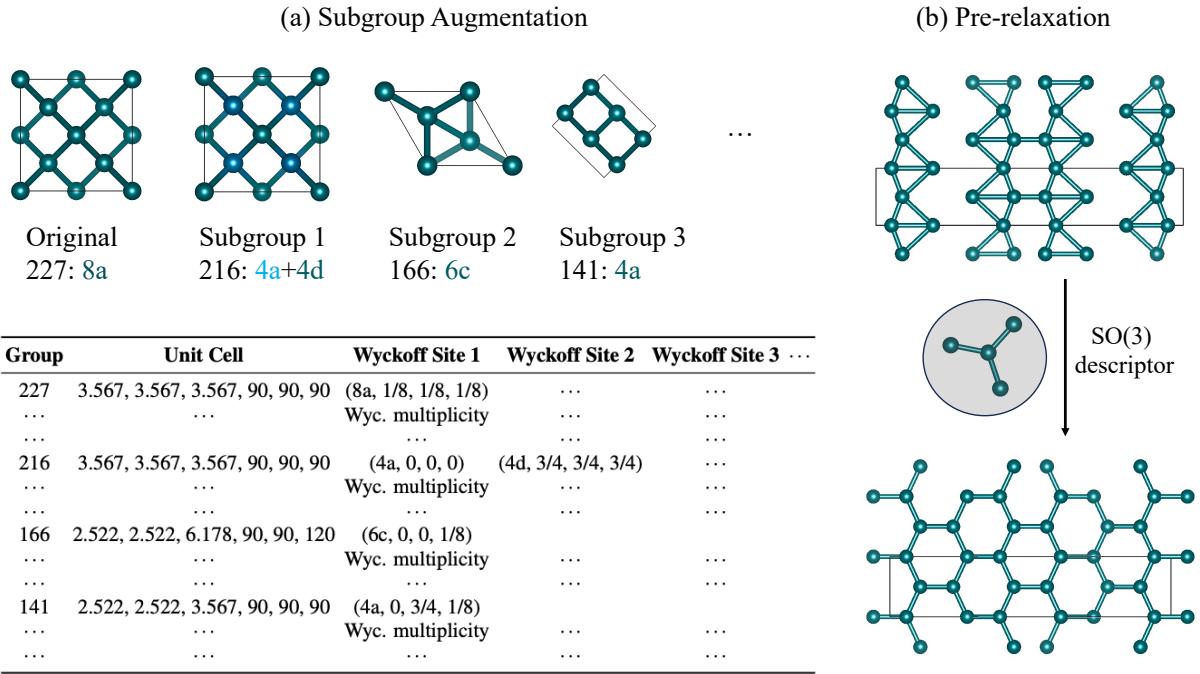


FIG. 2. Proposed symmetry augmentation and pre-relaxation approaches to enhance the learning of crystal structure representation. (a) shows the process to augment the representation of diamond crystal by using its associated subgroup symmetry structures. (b) illustrates a graphic example to pre-relax the generated crystal to the desired sp^2 local environment by using the power spectrum $SO(3)$ descriptor.

ture with respect to this reference environment. Figure 2b explains the idea of using $SO(3)$ descriptor based reference environment for a randomly generated symmetric crystal. Ideally, we aim to achieve two outcomes from the use of pre-relaxation, including (1) maintain the same chance if the initial guess is close to an ideal arrangement; and (2) enhance the chance of generating good structures even from a bad guess by pre-relaxation. The numerical details and justification can be found in the supplementary materials Section III. Hence, we can rapidly screen if the trial structure can be relaxed to the desired environment prior to further consideration with more expensive energy-based optimization. Note that similar ideas of optimizing the structures at the descriptor space have also been explored in several recent works^{42,43}.

Using the aforementioned approaches, we expect that the search space can be greatly reduced for both discrete and continuous variables that are needed to describe the crystal. Therefore, one can manipulate the structural complexity with only a few variables, and the generative model can be trained on a small dataset with a limited number of known examples.

III. THE LEGO-XTAL FRAMEWORK

Motivated by preceding discussions, we propose the Local Environment Geometry-Oriented Crystal Generator (LEGO-xtal) framework that is capable of rapidly generating many feasible crystal structures from a few known examples. As illustrated in Figure 3, it consists of four main components: (1) data collection and augmentation, (2) gen-

erative modeling training and sampling, (3) structural pre-relaxation, and (4) energy ranking with physical models. Below, we will describe the details of each component in the context of generating complex structures comprising exclusively sp^2 -hybridized carbon atoms. Although the ground state of sp^2 -carbon is well known to be the graphite structure, many metastable carbon allotropes exist with intriguing properties. The exploration of these sp^2 carbon allotropes has been highly sought after due to their fascinating electrical, optical, magnetic, thermal, and chemical properties^{44–47}. Hence, we will use the sp^2 -carbon allotropes as a testbed to demonstrate the effectiveness of the LEGO-xtal framework.

A. Training Data Augmentation & Tabularization

First, we need to gather the already known data used for training as many as possible. For carbon research, the online Samara Carbon Allotrope Database (SACADA) database, which contains all available data about 3-periodic carbon allotropes extracted from scientific literature from Web of Science and Scopus databases⁴⁴. From SACADA, we extracted 140 carbon allotropes which are made of all sp^2 carbon atoms. This includes a total of 127 unique (space group, Wyckoff sites) combinations, with energy spanning from 0 (graphite) to 1.344 eV/atom using the GGA-PBE pseudopotential⁴⁸ within the framework of density functional theory (DFT).

For each structure, we utilized PyXtal's subgroup symmetry module³⁶ to generate valid subgroup representations within constraints of 500 atoms and 8 Wyckoff sites per unit

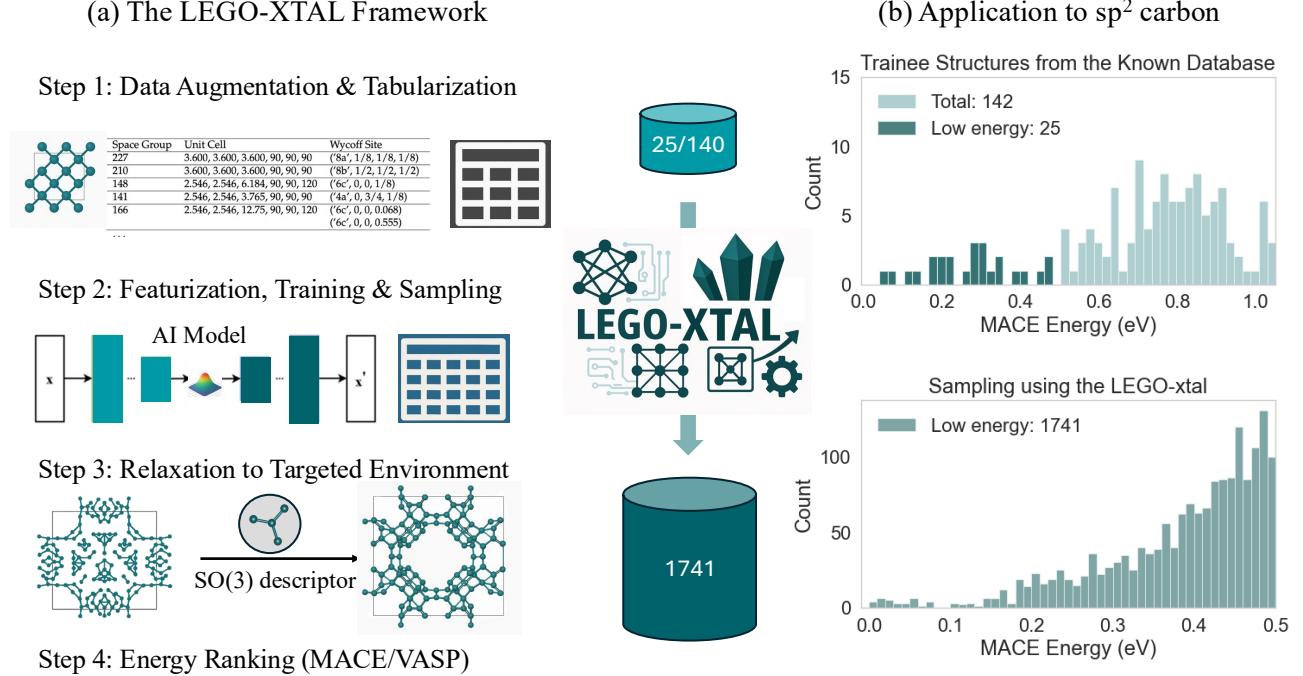


FIG. 3. **The LEGO-xtal framework and its application to search for low-energy carbon allotropes.** (a) shows the detailed pipeline of LEGO-xtal for adaptive generation of complex crystal structures. (b) summarizes the energy distribution of training dataset and newly generated structures by using this framework.

cell. Since the subgroup representation often results in larger unit cells and more Wyckoff sites, we excluded 10 structures with either low symmetry or large atom counts from the original 150 structures (see Supplementary Materials Section IV-A and Table S2). For each of the remaining structures, we generated 2–15 subgroup symmetry representations. Additionally, we randomly sampled the multiplicity of each Wyckoff site due to symmetry operations. To evaluate the effectiveness of data augmentation, we created three datasets labeled as v1-60k, v2-120k, and v3-240k, with details summarized in Table I.

TABLE I. Summary of the training dataset for sp² carbon allotropes.

Dataset	Number of samples	Number of unique symmetries ^a
Raw	140	127
v1-60k	63,114	419
v2-120k	122,322	449
v3-240k	242,766	485

^a The unique symmetry is defined according to the combination of space group and sorted list of Wyckoff indices. For instance, the diamond carbon is denoted as (227, 8a).

In each representation, the space group number and Wyckoff site column values are always treated as discrete features. The cell parameters and Wyckoff site position values are treated as continuous features by default (*cont-dataset*). To improve the model’s learning efficacy, we also explore the possibility of transforming all continuous variables to discrete

categorical values. In the discrete mode (*dis-dataset*), we divide the unit cell into a $100 \times 100 \times 100$ grid and convert each Wyckoff position into corresponding grid indices. For the cell parameters, we set up 100 bins between 0 and 35 for cell lengths, and 100 bins between 30 and 150 degrees for cell angles. The number of columns is 39 when the maximum number of Wyckoff sites is 8. Optionally, one may add some new columns, such as energy or other desired physical properties, to inject more information to guide the structure generation process.

B. Generative Model Training & Sampling

Prior to employing AI generative models, we transformed each row into an extended feature representation to enhance model learning. For discrete features, one-hot encoding is used, where each unique categorical value is mapped to an individual binary column. In this representation, a value of 1 indicates the presence of the corresponding category, while all other columns are set to 0. For continuous columns, we apply a Bayesian Gaussian Mixture Model (BGMM) approach as proposed by Patki et al⁴⁹, in which each continuous value is first scaled to the interval $[-1, 1]$, and then softly assigned to distinct Gaussian components via a probabilistic one-hot representation. This dual encoding strategy effectively captures both normalized numerical information and probabilistic cluster associations, improving the stability and efficacy of feature representation during generative model training. In the

dis-dataset, we replace the BGMM-based encoding with fixed binning. All continuous features are discretized into categorical bins (e.g., cell lengths into 100 bins between 0–35 Å), and the resulting values are one-hot encoded. This yields a fully categorical representation for the entire dataset.

After transformation, the table expands to 525 columns in the cont-dataset and 2332 columns in the dis-dataset. These two encoding schemes allow us to benchmark model performance under both mixed-type and fully discrete data conditions. Next, we explored two generative models on both dis- and cont-type datasets to learn the underlying distribution of crystal symmetry data. The models are trained to synthesize new rows of tabular features that correspond to novel crystal structures that satisfy crystallographic constraints.

- **GAN.** The Generative Adversarial Network (GAN) framework involves two neural networks—the Generator and the Discriminator—trained in a min-max game fashion, in which the Generator aims to create synthetic samples from random noise and fool the Discriminator, while the Discriminator attempts to distinguish real data from generated data. Our implementation is based on the Conditional Tabular GAN model⁵⁰, which is specifically designed to handle tabular datasets with mixed data types (both continuous and categorical). During training, we adopt the Wasserstein GAN with Gradient Penalty framework⁵¹ to improve training stability by enforcing the Lipschitz continuity condition via a gradient penalty. Once training is complete, the Discriminator is discarded, and only the trained Generator is used to produce new synthetic samples by using a random noise vectors sampled from a standard normal distribution.
- **VAE.** The Variational Autoencoder (VAE) is a probabilistic generative model that learns to map input data into a structured latent space from which new samples can be generated^{52,53}. The model consists of two neural networks: an Encoder that maps each input to the parameters (mean and variance) of a Gaussian distribution into latent space, and a Decoder that reconstructs the input from a sample drawn from this distribution. The latent space is regularized by enforcing the learned posterior distribution $q_\phi(z | x)$ to be close to a standard normal prior following the standard Normal distribution $\mathcal{N}(0, I)$, ensuring smoothness and continuity in the latent space which facilitates sampling and interpolation.

The training objective includes a reconstruction loss and a Kullback-Leibler (KL) divergence term. The reconstruction loss ensures fidelity between the input and output (negative log-likelihood for continuous features, cross-entropy for categorical ones), while the KL term encourages the approximate posterior to match the prior, acting as a form of regularization. During the inference phase, new samples are generated by drawing random noise vectors from the standard multivariate normal distribution. These latent vectors are passed through the trained Decoder network, which maps them

to the transformed tabular representation of crystal symmetry features.

For more details regarding GAN and VAE models, please refer to the Supplementary Materials Section I and Figs. S1 and S2). In a typical experiment, we train each GAN or VAE model with 250–1000 epochs, which is sufficient to converge the loss function values. Then we take the epoch that achieves the lowest loss function value to generate 100k - 500k samples for further consideration.

C. Structure Relaxation to Targeted Local Environment

Next, the structures yielded from the trained models need to be relaxed to the nearby energy minimum with the desired geometries. Instead of using the conventional structure optimization techniques based on classical force fields or electronic structure theory, we employed a descriptor-guided optimization framework that minimizes the dissimilarity in local atomic environments relative to a reference structure.

To represent the local atomic environment, the SO(3) descriptor P , with the option to include the explicit radial distribution function, is calculated to extract the radial and angular features of the neighbor density function⁴¹. For the purpose of this work, we selected a sp² carbon (i.e., one centered atom is connected to 3 neighboring carbon with 1.42 Å bond length and 120° bond angles in the same plane) to compute its power spectrum P_{ref} up to a radial cutoff of 2.1 Å. For each generated structure, we first evaluated the corresponding power spectrum P for each atom, and then sought to minimize the summed mean squared error (MSE) loss function as follows:

$$F_{\text{objective}} = \sum_i^{\text{all atoms}} \|P_i - P_{\text{ref.}}\|^2, \quad (1)$$

which quantifies the structural deviation from the standard sp² environment. The goal of the optimization procedure is to adjust reduced variables (e.g., lattice parameters and free x, y, z Wyckoff positions) to minimize this MSE loss function value, thereby steering structures toward graphite-like local order without requiring explicit energy evaluations. To perform the minimization, we utilized available optimization algorithms from the `scipy.optimize.minimize` library⁵⁴, including Nelder-Mead⁵⁵, L-BFGS-B⁵⁶, and Basin-Hopping⁵⁷.

During optimization, symmetry constraints were naturally enforced (due to the use of reduced variables) to ensure chemical realism and structural validity. Combining with the subgroup representation, this descriptor-guided strategy not only shortens the computational overhead associated with iterative energy evaluations but also allows the system to escape undesirable local minima associated with conventional force-based optimization. While we may use the subgroup symmetry as the initial guess, it is important to note that the low-symmetry structure after relaxation does not necessarily evolve to high symmetry. This is because many pathways may exist to stabilize the Wyckoff site to a target reference environment. For

more details, please refer to Fig. S3 and a motivating example in the Supplementary Section III.

D. Energy Ranking & Database

After the pre-relaxation, the valid structures need to be evaluated by the energy ranking. Given that many structures needs to be processed and it is computationally expensive to perform quantum mechanical calculations for each structure, we considered two alternatives, (i) a reactive force field (ReaxFF)⁵⁸ via the GULP package⁵⁹, a bond-order-based force field that can efficiently model chemical reactions and interatomic interactions; and (ii) a pretrained MACE model⁶⁰ via ASE⁶¹, which utilizes higher order equivariant message passing for fast and accurate predictions of interatomic interactions in a large chemical space. Both methods are significantly faster than conventional DFT calculations, making them suitable for large-scale screening of generated structures. Although most of the pre-relaxed structures should be very close to the ground state, each energy model may describe their ground state slightly differently in terms of bond lengths and angles, as well as lattice constants. Therefore, it is still necessary to run relaxation, instead of single point energy evaluation, to obtain the ground state and accurate energy ranking.

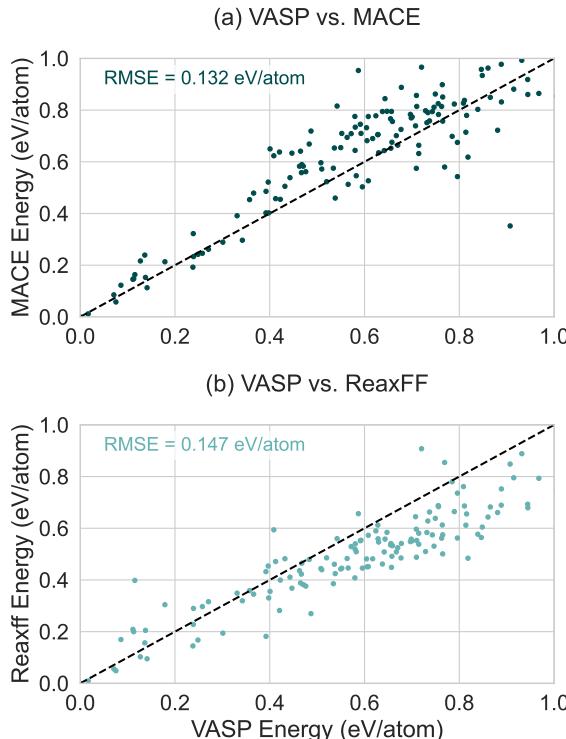


FIG. 4. Comparison of simulated energies for the selected 140 sp^2 allotropes from the SACADA database. (a) VASP vs. MACE and (b) VASP vs. ReaxFF. The dashed line indicates the ideal 1:1 correspondence.

Figure 4 compares energies calculated using MACE and

ReaxFF models against reference DFT calculations performed with the VASP code⁶² using the r^2 SCAN meta-GGA functional^{63,64} and a KSPACING value of 0.15 \AA^{-1} . In terms of the the root mean square error (RMSE) values, the MACE model (0.132 eV/atom) shows a better agreement with DFT energies compared to ReaxFF (0.147 eV/atom). Particularly, MACE describes better for the low-energy structures, making it slightly more suitable for our purposes. However, both models may stabilize structures into nonphysical configurations after pre-relaxation. Therefore, cross-checking generated structures with multiple models helps ensure physical validity. In our pipeline, we first use ReaxFF for relaxation and energy ranking, then apply MACE optimization to get the energy value as an alternative metric to infer the structural stability. According to our empirical observation, a feasible structure should be characterized by favorable energy ranking in both MACE and ReaxFF models.

The final database includes structures with MACE energies below $0.5\text{--}1.0 \text{ eV/atom}$. Beyond energy calculations, we characterize structural topology using CrystalNet.jl⁶⁵ to identify novel configurations compared to existing databases using the common RCSR notations⁶⁶, providing additional insight into the structural diversity and stability of generated structures.

E. Approximated Time Costs

Table II summarizes the approximate timing for each stage in the LEGO-xtal workflow. All timings are based on processing 1,000 samples; for larger datasets, times scale linearly.

First, Data augmentation is performed on a single CPU and typically takes 0.2–0.5 minutes per 1,000 samples. Model training on an H100 GPU requires 10–20 minutes for 250 epochs, while sampling (generation of new structures) takes 1–2 minutes per 1,000 samples.

Next, Pre-relaxation using the SO(3) descriptor on an AMD EPYC 96-core processor (2.4 GHz) takes approximately 5.5 minutes per 1,000 samples. When SO(3) pre-relaxation is followed by GULP-ReaxFF or MACE, the combined times are 7.5 and 9.5 minutes per 1,000 samples, respectively. For comparison, running GULP-ReaxFF or MACE without pre-relaxation requires 20 and 120 minutes per 1,000 samples, respectively. This suggests that pre-relaxation can effectively reduce the time cost as compared to the energy-based optimization. As shown in the Supplementary Materials Section III-C, we also provide direct evidence to demonstrate that pre-relaxation also improve the success rate of finding the unique valid structures.

In addition, we have implemented Batchwise SO(3) pre-relaxation on an H100 GPU which is even faster, taking about 1.5 minute per 1,000 samples, although this approach is still under development and requires further optimization.

TABLE II. Summary of time costs for each stage in `LEGO-xtal`.

Stage	Device	Speed (per 1k samples)
Data Augmentation	Single CPU	0.2-0.5 mins
Training	H100 GPU	10~20 mins / 250 epochs
Sampling	H100 GPU	1~2 mins
Pre-relaxation	AMD 96-core	~ 5.5 mins
Pre-relax + ReaxFF	AMD 96-core	~ 5.5+1.5 mins
Pre-relax + MACE	AMD 96-core	~ 5.5+2.5 mins
Pre-relaxation Batchwise	H100 GPU	~1.5 mins
ReaxFF w/o Pre-relax	AMD 96-core	~20 mins
MACE w/o Pre-relax	AMD 96-core	~120 mins

IV. RESULTS & DISCUSSIONS

To evaluate our framework, we trained multiple GAN and VAE models for generating sp^2 carbon allotropes, by experimenting with various hyperparameters (e.g., training dataset size, continuous versus discrete representation, neural network architectures, number of training epochs). From each model, we generated 200k samples for evaluation. After pre-relaxation, we only retained structures exhibiting valid sp^2 environments for further processing using MACE and ReaxFF. Below, we first analyze how different hyperparameter choices impact model performance, followed by a detailed examination of the sp^2 allotrope database generated using `LEGO-xtal`.

A. Model Performances Evaluation

Table III shows statistics of unique sp^2 structures from different generative models trained on the v1-60k dataset with 200~500 epochs. Compared to the original 140 unique structures in the SACADA dataset, all models except GAN-Cont. successfully regenerated over 100 training structures within 100k samples, indicating a strong capability to learn from the training data. If we allow generating more samples, we expect all the training structures would be reproduced by these models. More encouragingly, each model generated a substantial number of unique structures (GAN-Cont.: 2937, GAN-Dis.: 4514, VAE-Cont.: 4862, VAE-Dis.: 4289). The limited overlap between models, as found in Table III, suggests that each model learn different statistical patterns from the training data, leading to diverse generated structures.

TABLE III. The overlap matrix showing the number of shared unique sp^2 structures between 140 training structures and the 100k samples trained from v1-60k dataset using different models.

	Train	GAN-Cont.	GAN-Dis.	VAE-cont.	VAE-Dis.
Train	140	62	111	100	132
GAN-Cont.		2937	649	848	707
GAN-Dis.			4514	979	952
VAE-Cont.				4862	1152
VAE-Dis.					4289

Next, we performed a more extensive benchmark to assess the quality and success of our generative method with different hyperparameters. To make a meaningful comparison between different models, we introduced the following metrics:

- $N_{\text{valid_xtal}}$: Number of generated samples that match the required constraints of the intended crystal symmetry.
- $N_{\text{valid_env}}$: Number of valid structures successfully optimized to adopt the sp^2 -type bonding configuration.
- N_{unique} : Number of unique crystal configurations among the valid set, after removing duplicates.
- N_{train} : Number of structures that were regenerated from the initial 140 training set.
- $N_{\text{lowE_all}}$ and $N_{\text{lowE_cubic}}$: Number of unique (cubic) crystals with a MACE energy less than 0.55 eV/atom compared to the ground state, indicating thermodynamic stability.

These quantitative indicators collectively provide insights into how realistic, novel, and stable the generated structures are. The results, summarized in Table IV, show the relative performance of different models. Among them, the GAN models trained on continuous representation appear to perform notably worse than other models in nearly all metrics. This is likely due to the GAN's difficulty in capturing the discrete nature of crystal symmetries and Wyckoff sites, which are crucial for generating valid crystal structures.

Nevertheless, other VAE and GAN models demonstrate similar performances across various metrics. On average, we observe that around 98% success rate in generating valid crystal structures that satisfy the space group and Wyckoff site symmetry constraints, and 15-22% of these structures can be relaxed to adopt the desired sp^2 bonding configuration. The number of unique structures ranges from 3,235 to 5,156 per 100k samples, indicating a good diversity in the generated crystal configurations. As shown in Fig. S4 of the Supplementary Materials, one can continue to try more samples to get new unique structures. In addition, our analysis in Section V-A of Supplementary Materials suggests that these metrics are clearly better than the pure random sampling approaches, suggesting the efficacy of the statistical model learning.

For realistic application, the more important metric is the number of low-energy unique structures. The GAN-Dis-v2/v3 model achieved higher numbers of low-energy (cubic) structures as compared to the corresponding v1 model, suggesting that more explicit subgroup symmetries can promote the model's learning efficacy. A similar trend is also observed for the VAE-continuous models. However, this is not the case for the VAE-Discrete models, in which the model trained on the v1-60k dataset produced the highest number of low-energy structures. This may be due to the fact that the VAE-Dis. model uses the noise sampled from continuous Gaussian distribution, which may hinder the decoder from learning discrete, one-hot-like representations in the latent space, thus reducing generation fidelity on categorical outputs. A natural remedy is to adopt the Gumbel-Softmax reparameteriza-

TABLE IV. Summary of metrics evaluated on 100k generated structures across different models with the best numerical results highlighted bold.

Model	Dataset	<i>N_{valid_xtal}</i>	<i>N_{valid_env}</i>	<i>N_{unique}</i>	<i>N_{train}</i>	<i>N_{lowE_all}</i>	<i>N_{lowE_cubic}</i>
GAN-Cont.	v1-60k	95,313	10,687	2,937	62	147	30
	v2-120k	96,174	10,486	3,132	65	40	31
	v3-240k	96,404	9,621	3,235	71	43	29
GAN-Dis.	v1-60k	98,393	16,985	4,514	111	212	62
	v2-120k	98,861	18,406	4,814	120	426	69
	v3-240k	99,026	16,092	5,080	119	291	70
VAE-Cont.	v1-60k	97,657	16,524	4,862	100	57	35
	v2-120k	97,812	15,330	5,123	94	206	28
	v3-240k	98,004	15,052	5,156	101	241	28
VAE-Dis.	v1-60k	97,466	22,204	4,640	133	267	59
	v2-120k	97,419	15,685	4,649	127	53	36
	v3-240k	98,302	15,123	4,883	118	51	35

tion trick, which introduces a temperature-controlled, differentiable approximation to categorical sampling: starting from a high temperature to ensure smooth gradients, and annealing towards a low but non-zero τ to encourage near-discrete latent codes while preserving differentiability⁶⁷. Additionally, balancing the KL-divergence and reconstruction terms via β -VAE⁶⁸ or KL-annealing⁶⁹ strategies can prevent posterior collapse and improve latent expressiveness. Finally, in scenarios with severely imbalanced categorical outputs, applying class-weighted Cross-Entropy or Focal Loss⁷⁰ can stabilize training by emphasizing hard or rare classes.

Unlike VAE models that explicitly reconstruct inputs, GANs rely solely on adversarial feedback. This enables the discriminator to learn the joint distribution of the representations without being constrained to memorizing input patterns, thereby enhancing the generator's ability to generalize. This effect is particularly evident in Table IV, where the GAN achieves stronger performance on the discrete dataset compared to the Cont dataset. The smaller and more structured joint distribution space of discrete features makes it easier for the GAN to capture meaningful generalizable patterns, rather than overfitting to training samples.

While it is difficult to make a definitive recommendation on a single best model for this task, we observe a general trend that most models can rapidly generate a large number of low-energy sp² carbon allotropes with diverse structures, as compared to the original 140 training structures. Importantly, Table IV suggests that the more data is included, the chance of generating unique structures (*N_{unique}*) becomes higher regardless choice of generative models. This is likely to be associated with the increment of unique symmetries (see extended discussion in Section V-B of the Supplementary Materials). Overall, the results highlight new opportunities for effectively harnessing the power of AI generative models to explore the structural space of complicated materials. The improvements of model performance will be left for the future, as it is beyond

the scope of this work.

B. The sp² Carbon Allotrope Database

Since no single model significantly outperformed the others, we ran multiple models to comprehensively explore the structural space of sp² allotropes. After removing duplicate structures, we identified a total of 21,786 structures within a 1.0 eV MACE energy window, including 1,741 structures with energies less than 0.5 eV above the ground state. In the original 140 data used in training, there exist 25 sp² carbon allotropes that have the MACE energy less than 0.5 eV/atom higher than the ground state graphite structure (see Figure 3b). Using the LEGO-xtal, we have significantly expanded it to 1700+ distinct structures, evidently demonstrating the power of our LEGO-xtal framework to rapidly generate novel structures. To double check the accuracy of MACE energy ranking, we also manually selected 150 structures for further relaxation with DFT-r²SCAN using the VASP package. The complete database of 1,700+ low-energy sp² carbon allotropes, as well as ReaxFF, MACE and DFT-r2SCAN energies, is available at <https://lego-crystal.onrender.com>. Below, we will focus on the chemical analysis of these low-energy structures.

1. Distribution of low-energy crystals by complexity

Figure 5 displays the distribution of structures by number of atoms per unit cell, space group symmetry, and number of reduced crystal variables in each allotrope. First, Figures 5a and 5b list the distributions by unit cell size and space group symmetry. Clearly, there exists a large percentage of structures having both high symmetry and large unit cell size. The largest low-energy sp² structure we attempted has 960 atoms

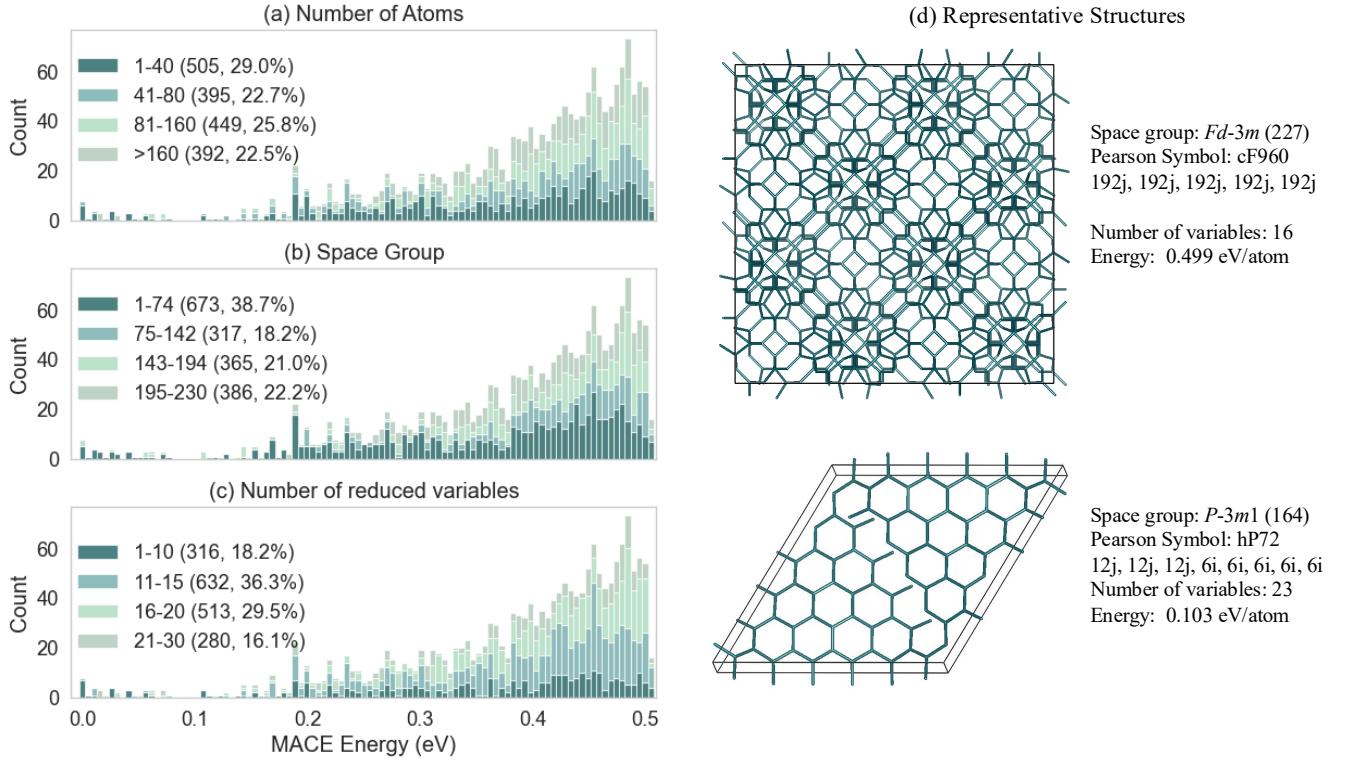


FIG. 5. Complexity analysis of the 1741 low-energy sp^2 carbon allotropes. (a)-(c) shows the distributions of the energy of the generated structures by number of atoms in the unit cell, space group symmetry, and number of reduced variables, respectively. (d) shows two representative low-energy structures demonstrating the intrinsic crystal searching challenge. In (d), the structures are labeled by their Pearson symbol and relative DFT energies as compared to the ground state graphite structure. The Pearson symbol is a shorthand notation that describes the crystal structure, including the space group and the number of atoms in the unit cell.

in the unit cell with $Fd\bar{3}m$ space group symmetry, consisting of 5 general Wyckoff sites at $192i$. To our knowledge, such a large structure has never been reported for either a traditional or AI-based crystal structure prediction method, suggesting that our new approach can efficiently handle the complex structures by taking advantage of crystallographic symmetry. Indeed, as long as the space group symmetry and Wyckoff choices are known, this 960-atom cubic structure contains only 16 reduced crystal variables (1 for the lattice parameter and 15 for (x,y,z) coordinates of the five Wyckoff sites). Therefore, the searching space can be drastically reduced. In turn, our approach can effectively generate large crystals in high symmetry, which is a significant advance as compared to recently proposed approaches that are mostly suitable for structures with no more than 20 atoms in the unit cell^{20,22,23}.

In addition, it is important to emphasize that our approach does not necessarily only favor high-symmetry structures. As shown in Figure 5b, there exist many structures with lower symmetry space groups other than the cubic symmetry. Using the number of reduced crystal variables as a metric to probe the model's predictive capabilities (Figure 5c), we find that the majority of low-energy structures have fewer than 20 reduced variables. This indicates that our framework is most effective at exploring the structural space of sp^2 carbon allotropes with a limited number of variables, less than 20. However, we can

still find that a notable portion of structures (280 out of 1741) have more than 20 reduced variables. Figure 5d highlights a representative low-energy structure with 23 reduced variables - a 3D-periodic structure containing 72 atoms per unit cell in $P\bar{3}m1$ symmetry. This finding demonstrates that our framework can effectively explore more complex structural spaces despite increasing uncertainty at higher dimensions.

2. The 0-2D low-energy sp^2 carbon allotropes

Given the nature of sp^2 bonding, layered graphite is unquestionably the ground state. Figure 6a shows the breakdown of energy distribution by detected dimensionality. The majority of low-energy structures (< 0.15 eV/atom) are characterized by stacked hexagonal graphite layers. In addition to the commonly known stacking polytypism due to layer translations¹⁶, we also discovered many variations arising from layer twisting. Among these, the most interesting structure is shown in Figure 6d, which is an AB-type stacked layered structure with about a 30-degree rotation angle between two adjacent layers. This structure is not only low in energy but also exhibits a unique 2D periodicity. This finding suggests that our framework can effectively explore the structural space of sp^2 carbon allotropes that is related to recent work on the twisted bilayer

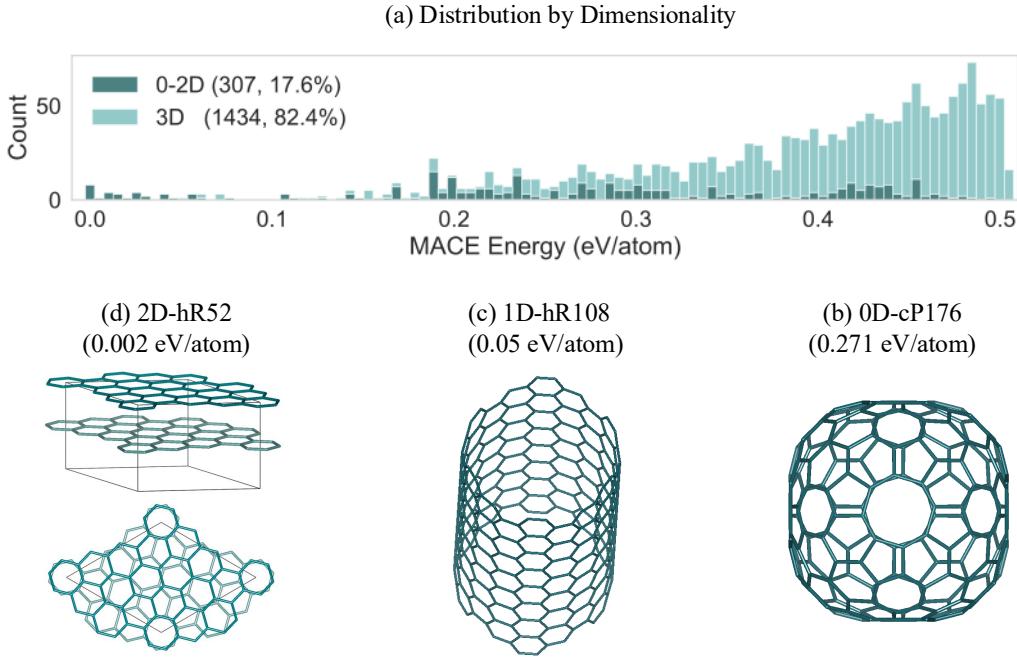


FIG. 6. The screened low-dimensional sp^2 carbon allotropes. (a) Distribution of the energy of the generated structures grouped by dimensionality. (b-d) show three representative 0D, 1D, and 2D structures labeled by the Pearson symbol and the relative DFT- r^2 SCAN energies as compared to the ground state graphite structure.

graphene structures with varying rotation angles^{71,72}.

Beyond the stacking graphite-type layers, we also identified several other planar configurations in different types of ring topology (e.g., mixed five-, seven, eight-membered rings, see Section VI and Fig. S6 in the Supplementary Materials) with an energy spanning from 0 to 0.5 eV range. Additionally, several unique 1D carbon nanotube-like structures with varied diameters (e.g., a representative model 1D-hR108 in Figure 6c) have been observed in our results. Finally, we also found two large 0D allotropes (see 0D-cP176 in Figure 6d) featured by the combination of five, six and eight-membered rings that are similar to the well known C_{60} and C_{70} ⁷³. Notably, these 0D allotropes are calculated to possess lower energies (0.2-0.3 eV/atom) than C_{60} and C_{70} . Hence, they have better stability if they can be synthesized in the experiment. These examples demonstrate our framework's capability to explore diverse structural motifs across multiple dimensionalities.

3. The 3D Negative Curved Graphite allotropes

Apart from the 300+ low-dimensional crystal packing motifs, the majority of low-energy structures possess extended networks with 3D periodicity. Not surprisingly, the lowest energy 3D sp^2 allotropes closely resemble graphite-like layered structures (see Fig. S7 in the Supplementary Materials and previous literature⁷⁴). However, the conceptually more compelling 3D sp^2 allotropes are those containing negative Gaussian curvature that form inward curved surfaces, resembling

saddle shapes or hyperbolic geometry⁴⁶. Unlike C_{60} with a positive curved graphite network, these 3D negative curved graphite (NCG) structures are more complex and feature fascinating properties due to their unique topology. Since 1990, only 15 NCG structures have been proposed through either mathematical derivation or sophisticated physical modeling approaches^{45,47,75-80}.

Within the 1741 structures, we have observed a total of 386 sp^2 cubic structures. To confirm their stability, we selected the 100 NCG candidates for further energy ranking using VASP at the DFT level. And the top 8 ranked structures are listed in Figure 7. Among them, we found four new structures (NCG1-cP192, NCG2-cP108, NCG3-cP192, and NCG4-cI144) that have neither been used in training nor reported in the previous literature. Remarkably, both NCG1-cP192 and NCG2-cP108 are ~ 0.015 eV/atom lower in energy than the previously known lowest-energy mct-cI192 NCG structure⁷⁹. Furthermore, we computed the phonon dispersions for the NCG1-NCG3 structures and confirmed their dynamical stability (see Fig. S8 in the Supplementary Materials).

In addition, there exist 41 structures with more than 300 atoms in the unit cell, highlighting that our approach can effectively handle complex structures that were seldom reported in the literature. Interestingly, by comparing with the structures used in training, we found that all newly identified structures exhibit some similar features to the training data while still maintaining unique characteristics. This indicates that our framework can effectively explore and expand upon known structural motifs rather than simply replicating existing structures. These remarkable findings demonstrate the power of

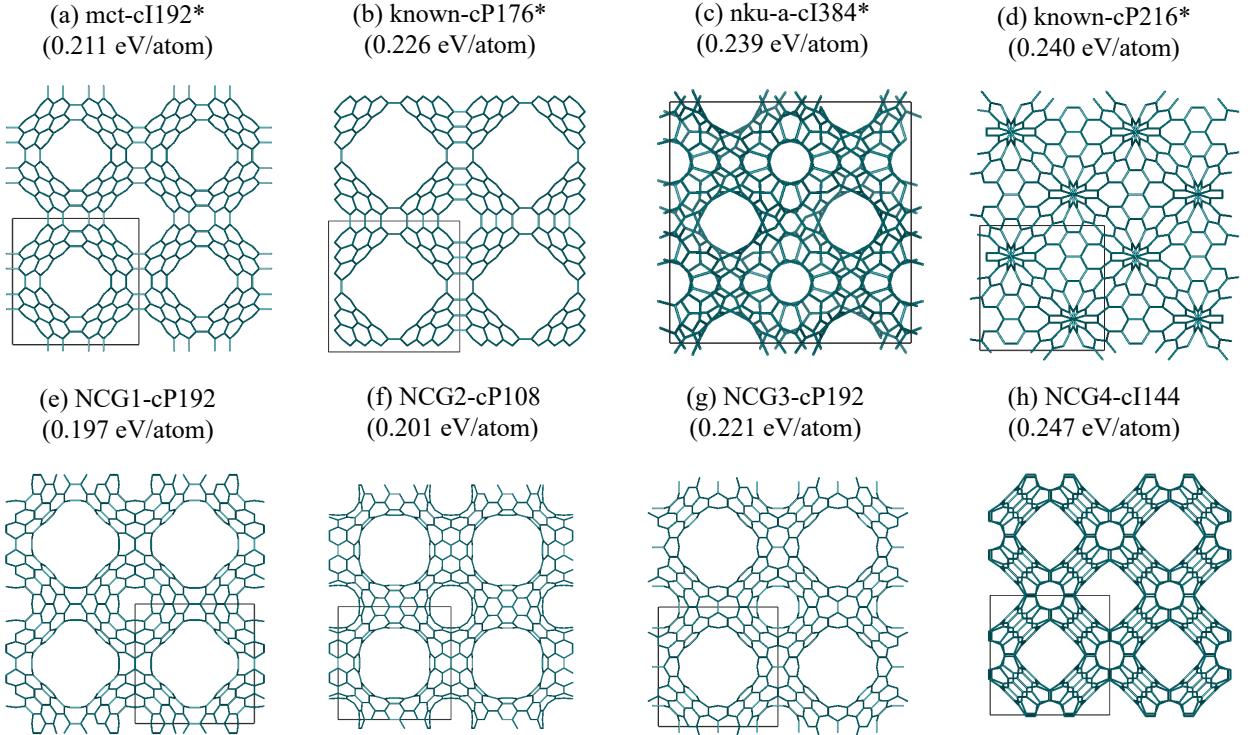


FIG. 7. **A list of low-energy negative curved graphite structures found by LEGO-xtal.** The structures are labeled by the topology according to RCSR notation⁶⁶ (when available), the Pearson symbol, and relative DFT- r^2 SCAN energies as compared to the ground state graphite structure. The previously reported structures (a-d) are also marked by the asterisks.

our framework to discover new materials with complex geometries.

V. CONCLUSIONS

In this work, we present LEGO-xtal, a novel framework that combines subgroup symmetry augmentation, descriptor-guided pre-relaxation, and modern generative components (GAN and VAE) to efficiently explore crystal structure space from limited training data. Starting from just 25 (140) known sp² carbon structures, our approach successfully generated over 1,700 (21,786) diverse low-energy allotropes, including high-symmetry structures with large unit cells (up to 960 atoms) and unique 0D, 1D, and 2D configurations. This demonstrates the framework's capability to discover complex crystal structures from a small dataset while maintaining physically desirable local environments. This clearly distinguishes our approach from previous methods^{18,20–31} that focus on generation of relatively small crystal systems from a large dataset without the explicit constraints on the local chemical environment.

The success of our framework stems from the synergistic combination of subgroup symmetry augmentation and descriptor-guided pre-relaxation. While symmetry augmentation enables efficient exploration of the structural space through reduced discrete variables, pre-relaxation ensures the

generated structures maintain desired local environments at minimal computational cost. Together, these strategies dramatically reduce the dimensionality of the crystal structure search space. Importantly, the subgroup-based data augmentation strategy introduced in our work is broadly applicable. In the supplementary materials Section IV-C, we have provided a universal interface to systematically search for subgroup symmetries, which could be integrated into existing symmetry-aware generative models to enhance their generalization capabilities^{18,24,25,28,30,31}.

Regarding the choice of generative models, this study explored both VAE and GAN architectures that offer a balance between simplicity and effectiveness. Despite their relative simplicity, both remain core building blocks in today's state-of-the-art pipelines (e.g., Stable Diffusion⁸¹, SV4D 2.0⁸², MaskGIT⁸³). We hope that LEGO-xtal will serve as a solid starting point, from which we can build toward integrating more advanced generative modeling techniques more advanced paradigms such as latent diffusion, masked modeling, or auto-regressive architectures in future work.

Due to the use of symmetrized tabular representation, this approach is likely to intrinsically favor high symmetry structures. For instance, our current models do not consider the generation of more than 8 atoms in the unit cell with P1 symmetry. While the assumption of no low-energy complex structures in low symmetry is true for the case of sp² carbon allotropes, it may not be suitable to describe other systems

where low-symmetry crystals appear more often. In that case, we recommend the use of the generative models without symmetry constraints^{22,23,26}.

Finally, we aim to extend subgroup symmetry augmentation beyond single components and reference environments, enabling the generation of more complex structures with multiple components and varied local environments in a single model. This extension requires (1) adapting pre-relaxation strategies to handle multiple reference environments, either by modifying the environmental loss function within traditional optimization routines⁴³ or by employing more flexible diffusion models that incorporate local environment constraints during the training stage and (2) the support of generative models on Wyckoff sites labeled with different elements or local environments. Conceptually, the use of descriptor-guided pre-relaxation and subgroup augmentation is expected to be even more advantageous to handle more complex compositional systems. These improvements will enable the generation of increasingly complex structures with higher dimensionality and more sophisticated local environments, which will be the focus of our future work.

ACKNOWLEDGMENTS

This research was sponsored by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, and the Established Program to Stimulate Competitive Research (EPSCoR) under the DOE Early Career Award No. DE-SC0024866, the UNC Charlotte's seed grant for data science, as well as European Union (ERDF), Région Nouvelle Aquitaine, Poitiers University, and French government programs "Investissements d'Avenir" (EUR INTREE, reference ANR-18-EURE-0010) and PRC ANR MagDesign and TcPredictor. The computing resources are provided by ACCESS (TG-DMR180040) and High-Performance Computing Centre Adastral/CINES of GENCI (projects A0140807539 and A0160815101). We also acknowledge the reviewers for their insightful comments and suggestions during the review stage. During the preparation of this work, the authors used GitHub Copilot in order to improve the code readability and documentation. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

DATA AVAILABILITY

The LEGO_xtal source code, instructions, as well as scripts used to calculate the results of this study, are available in <https://github.com/MaterSim/LEGO-xtal>. The complete list of 1741 low-energy sp² carbon allotropes, as well as ReaxFF, MACE and DFT-r²SCAN energies, is listed in <http://lego-crystal.onrender.com>.

CONFLICT OF INTEREST

All authors declare that they have no conflict of interest.

AUTHOR CONTRIBUTIONS

The concept of *Crystal Structure Generator From Chemical Building Blocks* was initially discussed by G.F. and Q.Z. (with the latter being a Visiting Professor at Poitiers University in June 2023). Q.Z. and G.F. co-conceived the idea. With the help of G.F., Q.Z. initiated the framework. Q.Z., D.D., H.X., and G.F. supervised this project. O.G.R., M.S.R., H.X., and Q.Z. implemented the code. S.P. performed the electronic structure calculations. All authors analyzed the results and contributed to manuscript writing.

Supplementary Materials

S1. GENERATIVE MODELS TRAINING AND SAMPLING

A. Generative Adversarial Network (GAN)

Generative Adversarial Networks (GANs) have emerged as powerful generative models capable of synthesizing realistic data by training two neural networks—the Generator and the Discriminator—in an adversarial setting⁸⁴. The Generator aims to create synthetic data that closely resembles real data, while the Discriminator is responsible for distinguishing between real and synthetic inputs. During training, both components are optimized in an adversarial process: the Generator progressively improves its ability to produce data that mimics real samples with increasing fidelity, while the Discriminator becomes more adept at detecting even subtle discrepancies between real and generated data. This dynamic continues until an equilibrium is reached, at which point the Generator produces outputs so realistic that the Discriminator can no longer reliably differentiate them from real data. The GAN model in our LEGO-xtal framework is directly inspired by the Conditional Tabular GAN (CTGAN) approach proposed by Xu et al.⁵⁰, specifically designed to handle structured tabular data containing both discrete and continuous attributes.

a. Generator. The Generator is responsible for transforming random noise into realistic tabular data samples that resemble the true distribution. This noise, denoted as $z \sim \mathcal{N}(0, I)$, is drawn from a standard multivariate normal distribution. The Generator architecture is composed of multiple residual blocks. Each block includes a fully connected (FC) linear transformation, followed by batch normalization (BN)⁸⁵, and a Rectified Linear Unit (ReLU) activation function. Residual (or skip) connections⁸⁶ are used to concatenate the input of each block with its output, helping preserve information flow and mitigate vanishing gradients during training. The internal computations of the Generator can be summarized as follows:

$$\begin{cases} h_0 = \mathbf{z} \in \mathbb{R}^{B \times 128} \\ r_1 = \text{ReLU} \circ \text{BN} \circ \text{FC}_{128 \rightarrow 512}(h_0) \\ h_1 = r_1 \oplus h_0 \in \mathbb{R}^{B \times 640} \\ r_2 = \text{ReLU} \circ \text{BN} \circ \text{FC}_{640 \rightarrow 512}(h_1) \\ h_2 = r_2 \oplus h_1 \in \mathbb{R}^{B \times 1152} \\ \hat{\mathbf{x}} = \text{FC}_{1152 \rightarrow d_x}(h_2) \end{cases} \quad (\text{S1})$$

Here,

- \mathbf{z} : Latent noise vector sampled from a standard normal distribution.
- B : Batch size.
- $\text{FC}_{a \rightarrow b}$: Fully connected (linear) layer with input dimension a and output dimension b .

- BN: Batch normalization layer.
- ReLU: Rectified Linear Unit activation function.
- \oplus : Concatenation operator along the feature dimension.
- \circ : Element-wise multiplication
- $\hat{\mathbf{x}}$: Output vector with dimensionality matching the trainee input data (d_x).

The generated samples lie in a transformed feature space comprising both continuous and categorical components. To handle these different types appropriately, we apply distinct activation functions based on metadata recorded during preprocessing. Each feature is tagged with a label, either `tanh` or `gumbel`, indicating the appropriate postprocessing strategy.

For continuous features, we use *cluster-based normalization*, implemented via Gaussian Mixture Modeling (GMM)⁴⁹. Each feature is modeled as a mixture of Gaussians, and transformed into a combination of two components: (1) a scalar normalized value indicating the position within a selected Gaussian, and (2) a one-hot encoded vector representing the most likely cluster. During generation, the scalar component is passed through the hyperbolic tangent activation function, `tanh`, to restrict its values to the range $(-1, 1)$, maintaining numerical stability and aligning with the transformed scale.

For categorical (discrete) features, which are represented using one-hot encoding, the Gumbel-Softmax function⁶⁷ is applied. This enables differentiable sampling from a categorical distribution, allowing training via backpropagation despite the discrete nature of these variables. The Gumbel-Softmax activation ensures the synthetic outputs closely follow the one-hot format of the original data.

Finally, a composite loss is computed by comparing the activated outputs to the real transformed inputs. Separate loss components are used for continuous and categorical features, enabling the model to learn an accurate and data-type-aware generative process.

b. Discriminator. The Discriminator is a neural network trained to distinguish between real data and data generated by the Generator. It plays a crucial role in guiding the Generator to produce more realistic samples. In our architecture, the Discriminator consists of fully connected (dense) layers with LeakyReLU activations, which help stabilize learning and reduce the chance of neurons becoming inactive during training⁸⁷. Additionally, Dropout layers are included to prevent overfitting, improving the model’s ability to generalize to unseen samples.

We also adopt the *packing* strategy (PAC)⁵⁰ which improves training stability. Instead of feeding single samples into the Discriminator, we group every 10 samples together and concatenate them into one larger vector. This allows the Discriminator to evaluate patterns across sample groups rather than treating them independently.

Mathematically, the Discriminator operates as follows:

$$\begin{cases} x_{\text{pac}} = x \in \mathbb{R}^{B \times 10 \times d_x} \rightarrow \text{reshape} \in \mathbb{R}^{B \times (10 \cdot d_x)} \\ h_1 = \text{Dropout} \circ \text{LeakyReLU}_{\gamma=0.2} (\text{FC}_{10 \cdot d_x \rightarrow 512}(x_{\text{pac}})) \\ h_2 = \text{Dropout} \circ \text{LeakyReLU}_{\gamma=0.2} (\text{FC}_{512 \rightarrow 512}(h_1)) \\ y = \text{FC}_{512 \rightarrow 1}(h_2) \end{cases}$$

Here, x_{pac} denotes the packed input with batch size B , and each group of 10 samples of original feature dimension d_x is reshaped into a vector of size $10 \cdot d_x$. This reshaped input then flows through two fully connected layers with Dropout (0.2) and LeakyReLU activation functions.

Using a small slope for negative values helps avoid the “dying ReLU”⁸⁷ problem by allowing gradients to pass through even when the input is negative, ensuring the network continues to learn effectively.

c. Loss Functions. Our model utilizes the Wasserstein GAN with Gradient Penalty (WGAN-GP) loss formulation for the Discriminator⁵¹. Specifically, the Discriminator loss L_D is defined as:

$$L_D = [D(\tilde{x})] - [D(x)] + \lambda [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \quad (\text{S2})$$

where x represents real samples, \tilde{x} represents generated samples, \hat{x} are interpolated samples between real and generated samples, D is the discriminator function, $\nabla_{\hat{x}} D(\hat{x})$ denotes the gradient with respect to \hat{x} , and λ is the gradient penalty coefficient. The Generator loss L_G aims to minimize the discriminator’s ability to distinguish generated samples as fake and is defined as:

$$L_G = -[D(\tilde{x})] \quad (\text{S3})$$

d. Training Model. To enable reproducibility of our results, we provide a simple Python script demonstrating how to retrain the GAN-based synthesizer on the processed training data. The model is implemented using our `lego.GAN` module and supports discrete column conditioning. Below is an example of how the model can be trained and used to generate new samples:

```
# https://github.com/MaterSim/LEGO-xtal
from lego.GAN import GAN
import pandas as pd

# Load processed training data
df = pd.read_csv('data/train/train-v4.csv')

# Specify discrete columns used for conditioning
discrete_columns = ['spg', 'wp0', 'wp1', 'wp2', 'wp3',
                     'wp4', 'wp5', 'wp6', 'wp7']

# Initialize and train the synthesizer
synthesizer = GAN()
synthesizer.fit(df, discrete_columns=discrete_columns)

# Sample synthetic data and save to file
df_synthetic = synthesizer.sample(samples=1000)
```

```
df_synthetic.to_csv('synthetic_structures_GAN.csv',
index=False, header=True)
```

Listing S1. A Python script to train the GAN model in the LEGO-xtal Framework.

During training, the generator and discriminator losses were recorded across 500 epochs. The plot in Figure S1 illustrates the convergence behavior of the GAN. The generator loss steadily improves, while the discriminator loss stabilizes, indicating a balanced adversarial training process.



FIG. S1. **GAN training losses over 500 epochs.** The generator loss (blue) improves over time, while the discriminator loss (orange) stabilizes.

B. Variational Autoencoder (VAE)

The Variational Autoencoder (VAE) is a deep generative model that learns a probabilistic mapping between high-dimensional observed data and a lower-dimensional latent space from which synthetic data can be generated^{52,53}. The VAE consists of two neural networks: an encoder that transforms input data into the parameters of a latent Gaussian distribution, and a decoder that reconstructs data from samples drawn from this distribution. To enable gradient-based optimization, the reparameterization trick is employed, which makes the sampling process differentiable.

In this work, we utilize the Tabular Variational Autoencoder (TVAE) architecture as implemented in the open-source SDV project⁴⁹, which extends the VAE framework to structured tabular datasets. We adopt this architecture within our proposed LEGO-xtal framework to model the underlying distribution of crystallographic symmetry features and generate novel candidate crystal structures.

a. Encoder $q_\phi(z | x)$. Let $\mathbf{x} \in \mathbb{R}^{B \times d_x}$ denote a batch of input data with B samples, each having d_x features. In the encoder network, the input data \mathbf{x} is passed through two FC + ReLU layers to produce a hidden representation h_2 . From this, two separate FC layers compute the mean $\mu \in \mathbb{R}^{B \times 128}$ and the log-variance $\log \sigma^2 \in \mathbb{R}^{B \times 128}$ of the latent distribution.

$$\begin{cases} x &= \mathbf{x} \in \mathbb{R}^{B \times d_x} \\ h_1 &= \text{ReLU} \circ \text{FC}_{d_x \rightarrow 512}(x) \\ h_2 &= \text{ReLU} \circ \text{FC}_{512 \rightarrow 512}(h_1) \\ \mu &= \text{FC}_{512 \rightarrow 128}(h_2) \\ \log \sigma^2 &= \text{FC}_{512 \rightarrow 128}(h_2) \end{cases} \quad (\text{S4})$$

To sample the latent vector \mathbf{z} , the reparameterization trick is used:

$$\mathbf{z} = \mu + \exp\left(\frac{1}{2} \log \sigma^2\right) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$$

where \odot denotes element-wise multiplication.

b. Decoder $p_\theta(x | z)$. The decoder takes the latent variable $\mathbf{z} \in \mathbb{R}^{B \times 128}$ and reconstructs the input using two additional FC + ReLU layers, followed by a final FC layer producing the output $\hat{\mathbf{x}}_{\text{out}} \in \mathbb{R}^{B \times d_x}$. Additionally, a learnable parameter vector $\boldsymbol{\sigma}_{\text{out}} \in [0.01, 1.0]^{d_x}$ is used to model the per-feature uncertainty for continuous-valued outputs. This parameter is jointly optimized during training to improve reconstruction quality.

$$\begin{cases} z &= \mathbf{z} \in \mathbb{R}^{B \times 128} \\ h_1 &= \text{ReLU} \circ \text{FC}_{128 \rightarrow 512}(z) \\ h_2 &= \text{ReLU} \circ \text{FC}_{512 \rightarrow 512}(h_1) \\ \hat{\mathbf{x}}_{\text{out}} &= \text{FC}_{512 \rightarrow d_x}(h_2) \end{cases} \quad (\text{S5})$$

c. Loss Function. The training objective of the VAE model combines two components: the reconstruction loss and the Kullback-Leibler divergence (KLD). Given an input batch $\mathbf{x} \in \mathbb{R}^{B \times d_x}$, the VAE aims to reconstruct \mathbf{x} from the latent space while ensuring that the learned posterior distribution remains close to a unit Gaussian prior.

For each input feature, the type of reconstruction loss is chosen based on the activation function defined in the data transformer. The total loss is computed as:

$$\mathcal{L}_{\text{total}} = \frac{\lambda}{B} \sum_{i=1}^B \mathcal{L}_{\text{recon}}^{(i)} + \frac{1}{B} \sum_{i=1}^B D_{\text{KL}} \left(q_\phi(z^{(i)} | x^{(i)}) \| \mathcal{N}(0, \mathbf{I}) \right) \quad (\text{S6})$$

Reconstruction Loss $\mathcal{L}_{\text{recon}}$. The reconstruction loss is computed feature-wise:

- For continuous features, we model the output as a Gaussian distribution with mean predicted by the decoder and learnable standard deviation $\sigma \in [0.01, 1.0]$, producing:

$$\mathcal{L}_{\text{cont}} = \sum_{j \in \mathcal{C}} \left(\frac{(x_{\text{in}} - \tanh(\hat{x}_{\text{out}}))^2}{2\sigma_{\text{out}}^2} + \log \sigma_{\text{out}} \right) \quad (\text{S7})$$

Here, \mathcal{C} denotes the set of continuous feature indices, and \hat{x}_{out} is the raw output of the decoder before activation. The \tanh function ensures bounded output for numeric stability.

- For categorical features, the cross-entropy loss is applied between the target one-hot vector x_{out} and the predicted logits \hat{x}_{in} :

$$\mathcal{L}_{\text{cat}} = \sum_{j \in \mathcal{D}} \text{CrossEntropy}(\hat{x}_{\text{out}}, \arg \max(x_{\text{in}})) \quad (\text{S8})$$

Here, \mathcal{D} denotes the set of categorical feature indices.

KL Divergence. The Kullback-Leibler divergence encourages the encoder's learned latent distribution $q_\phi(z | x) = \mathcal{N}(\mu, \text{diag}(\sigma^2))$ to remain close to the prior $\mathcal{N}(0, \mathbf{I})$:

$$D_{\text{KL}} = -\frac{1}{2} \sum_{k=1}^{d_z} (1 + \log \sigma_k^2 - \mu_k^2 - \sigma_k^2) \quad (\text{S9})$$

The total loss is scaled by a user-defined factor λ (default: 2.0) to balance reconstruction and regularization effects.

d. Training. The model is trained for 250 epochs using the Adam optimizer⁸⁸ with L2 weight regularization. Input data is preprocessed, which contains normalized continuous features and one-hot encoded categorical features. During training, the loss is computed on mini-batches of size 500. To ensure numerical stability, the per-feature variance parameters σ_{out} are clamped between 0.01 and 1.0. The average training loss per epoch is recorded, showing a smooth convergence trend (see Figure S2). Below is an example of how the model can be trained and used to generate new samples using the LEGO-xtal framework:

```
# https://github.com/MaterSim/LEGO-xtal
from lego.VAE import VAE
import pandas as pd

# Load processed training data
df = pd.read_csv('data/train/train-v4.csv')

# Specify discrete columns used for conditioning
discrete_columns = ['spg', 'wp0', 'wp1', 'wp2', 'wp3', 'wp4', 'wp5', 'wp6', 'wp7']

# Initialize and train the synthesizer
synthesizer = VAE()
synthesizer.fit(df, discrete_columns=discrete_columns)

# Sample synthetic data and save to file
df_synthetic = synthesizer.sample(samples=1000)
df_synthetic.to_csv('synthetic_structures_VAE.csv', index=False, header=True)
```

Listing S2. A Python script to train the VAE model in the LEGO-xtal Framework.

S2. THE SO(3) DESCRIPTOR

In 2012, Bartók et al. introduced an improved many-body descriptor that explicitly incorporates both radial and angular components³⁹. This approach overcomes the limitation of

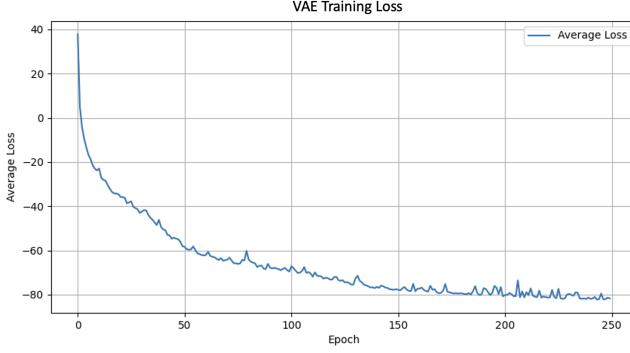


FIG. S2. The VAE Model training loss over 250 epochs.

describing neighbor density with a Dirac delta function by replacing the delta function with a Gaussian function of limited width α . This smoothing allows for a more realistic representation of how atoms are distributed around a reference atom. According to Bartók, the modified neighbor density function is given by:

$$\rho'(\mathbf{r}) = \sum_i^{r_i \leq r_c} e^{(-\alpha|\mathbf{r}-\mathbf{r}_i|^2)} = \sum_i^{r_i \leq r_c} e^{-\alpha(r^2+r_i^2)} e^{2\alpha\mathbf{r}\cdot\mathbf{r}_i} \quad (\text{S10})$$

Expanding the exponential of a dot product in spherical coordinates:

$$e^{2\alpha\mathbf{r}\cdot\mathbf{r}_i} = e^{2\alpha rr_i \cos(\gamma)} = 4\pi \sum_{l=0}^{\infty} \sum_{m=-l}^l I_l(2\alpha rr_i) Y_{lm}^*(\hat{\mathbf{r}}_i) Y_{lm}(\hat{\mathbf{r}}). \quad (\text{S11})$$

In which, we used the general formula addition theorem for spherical harmonics,

$$e^{z\cos(\gamma)} = 4\pi \sum_{l=0}^{\infty} \sum_{m=-l}^l I_l(z) Y_{lm}^*(\hat{\mathbf{r}}_i) Y_{lm}(\hat{\mathbf{r}}). \quad (\text{S12})$$

This expression can be further expanded as:

$$\rho'(\mathbf{r}) = \sum_i^{r_i \leq r_c} \sum_{lm} 4\pi e^{-\alpha(r^2+r_i^2)} I_l(2\alpha rr_i) Y_{lm}^*(\hat{\mathbf{r}}_i) Y_{lm}(\hat{\mathbf{r}}), \quad (\text{S13})$$

where the first part $I_l(2\alpha rr_i)$ is the modified spherical Bessel function of the first kind (governed by $2\alpha rr_i$), providing the radial dependence, and the second part captures the angular dependence of the vectors \mathbf{r} and \mathbf{r}_i .

Bartók also introduced a set of polynomials, $g_n(r)$, which help describe the radial component in a more refined way:

$$\phi_\alpha(r) = (r_c - r)^{\alpha+2} / N_\alpha$$

where N_α is a normalization factor given by:

$$N_\alpha = \sqrt{\int_0^{r_c} r^2 (r_c - r)^{2(\alpha+2)} dr}$$

These polynomials are orthonormalized to ensure that the radial functions $g_n(r)$ form a basis. The orthonormalization process is performed through linear combinations of $\phi_\alpha(r)$, and the coefficients are obtained from

$$g_n(r) = \sum_{\alpha=1}^{n_{\max}} W_{n\alpha} \phi_\alpha(r),$$

where W is constructed from the inverse square root of the overlap matrix S ,

$$S_{\alpha\beta} = \int_0^{r_c} r^2 \phi_\alpha(r) \phi_\beta(r) dr \\ = \frac{\sqrt{(2\alpha+5)(2\alpha+6)(2\alpha+7)(2\beta+5)(2\beta+6)(2\beta+7)}}{(5+\alpha+\beta)(6+\alpha+\beta)(7+\alpha+\beta)}$$

This overlap matrix describes how different radial functions overlap with each other and ensures that the final radial basis functions $g_n(r)$ are orthonormal.

The neighbor density function $\rho'(\mathbf{r})$ can then be expanded in terms of both the radial basis $g_n(r)$ and the spherical harmonics:

$$c_{nlm} = \langle g_n(r) Y_{lm}(\hat{\mathbf{r}}) | \rho'(\mathbf{r}) \rangle \quad (\text{S14}) \\ = \int d^3r g_n(r) Y_{lm}(\hat{\mathbf{r}}) \sum_{r_i \leq r_c} \sum_{l' m'} 4\pi e^{-\alpha(r^2+r_i^2)} I_{l'}(2\alpha rr_i) Y_{l'm'}^*(\hat{\mathbf{r}}_i) Y_{l'm'}(\hat{\mathbf{r}})$$

When integrating over the angular variables $\hat{\mathbf{r}}$, only the terms with $l' = l$ and $m' = m$ will survive, due to orthogonality.

$$c_{nlm} = 4\pi \sum_i^{r_i \leq r_c} Y_{lm}^*(\hat{\mathbf{r}}_i) \int_0^{r_c} r^2 g_n(r) e^{-\alpha(r^2+r_i^2)} I_l(2\alpha rr_i) dr \quad (\text{S15}) \\ = 4\pi \sum_i^{r_i \leq r_c} e^{-\alpha r_i^2} Y_{lm}^*(\hat{\mathbf{r}}_i) \int_0^{r_c} r^2 g_n(r) e^{-\alpha r^2} I_l(2\alpha rr_i) dr$$

Finally, the rotation-invariant power spectrum is obtained by combining these expansion coefficients:

$$p_{n_1 n_2 l} = \sum_{m=-l}^{+l} c_{n_1 lm} c_{n_2 lm}^*, \quad (\text{S16})$$

where the expansion coefficients c_{nlm} are projections of the neighbor density onto a set of orthonormal radial basis functions $g_n(r)$ and spherical harmonics $Y_{lm}(\hat{\mathbf{r}})$. This descriptor effectively encodes the local atomic geometry, including coordination shells and angular distributions, while ensuring invariance to global rotations.

It has been suggested that pure spherical harmonics descriptor cannot reconstruct the frequency components uniquely up to rotation³⁸. To remedy this issue, the SO(3) descriptor can be further complemented with the radial distribution function

(RDF) to capture the radial distribution of atoms around a central atom. The RDF is defined as:

$$g(r) = \frac{1}{N} \sum_{i=1}^N \sum_{j \neq i} \delta(r - |\mathbf{r}_i - \mathbf{r}_j|), \quad (\text{S17})$$

where N is the total number of atoms, \mathbf{r}_i and \mathbf{r}_j are the positions of atoms i and j , and δ is the Dirac delta function. The RDF provides additional information about the distribution of atoms at various distances from a central atom, complementing the SO(3) descriptors angular information.

S3. PRE-RELAXATION

A. Descriptor-based Geometry Optimization

After generating a crystal structure with the desired local environment, the next step is to optimize the geometry of the crystal structure. The optimization process is crucial for refining the generated structure to ensure it is energetically favorable and stable. In this work, we employ a descriptor-based optimization approach that utilizes the SO(3) descriptor to guide the optimization process via `scipy.optimize.minimize`.

- **Nelder-Mead** (100 iterations): a gradient-free simplex algorithm that is robust for local non-smooth landscapes and suitable for early-stage refinement,
- **L-BFGS-B** (100 iterations): a quasi-Newton method with bound constraints, used for smooth, efficient convergence in later stages.
- **Basin-hopping**: optionally used for global optimization, combining local minimization with random jumps to escape local minima.

More details about the optimization can be found in the documentation of `pyxtal.lego.builder.optimize_xtal`³⁶.

B. Geometry Optimization under Subgroup Representation

Figure S3 illustrates how to optimize the geometry of a crystal structure under subgroup representation. If one performs the symmetry-constrained optimization starts from a diamond structure with $Fd\text{-}3m$ symmetry, neither the local environment nor the energy landscape is suitable to obtain the sp^2 allotropes. If one starts from a subgroup $R\text{-}3m$ symmetry, the energy-based optimization will retain the diamond structure packing with the sp^3 local environment. However, the descriptor-based optimization can lead to a local environment that is more compatible with sp^2 allotropes. This is because the subgroup representation allows for a more reduced crystal variables that allows for the atomic transition to the sp^2 allotropes, while the original $Fd\text{-}3m$ symmetry is too rigid and

does not allow for such flexibility. Using the subgroup representation, one can effectively tune the local environment even for a low-energy crystal structure with unwanted local environment.

The above demonstration can be achieved using the following codes (Listing S3) based on PyXtal.

```
# pip install pyxtal
from pyxtal.lego.builder import builder
from pyxtal import pyxtal

# Get the graphite reference environment and set up
# optimizer
xtal = pyxtal()
xtal.from_prototype('graphite')
cif_file = xtal.to_pymatgen()
bu = builder(['C'], [1], db_file='test.db')
bu.set_descriptor_calculator(mykargs={'rcut': 2.0})
bu.set_reference_environments(cif_file)
print(bu)

# Get the diamond crystal
xtal = pyxtal()
xtal.from_prototype('diamond')
print(xtal)
print(xtal.get_1d_rep_x())
_, _, _ = bu.optimize_xtal(xtal)

# Subgroup conversion
sub_xtal = xtal.subgroup_once(H=166, eps=1e-4)
sub_xtal.to_file('sp3.cif')
print(sub_xtal)
print(sub_xtal.get_1d_rep_x())

# Optimization
sub_xtal, loss, _ = bu.optimize_xtal(sub_xtal)
print(sub_xtal)
print(sub_xtal.get_1d_rep_x())
sub_xtal.to_file('sp2.cif')
```

Listing S3. A Python script to perform geometry optimization using the subgroup representation.

C. Pre-relaxation Bias Analysis

As we described in the main text, we propose the use of pre-relaxation to achieve two goals

1. Maintain the same chance if the initial guess is close to an ideal arrangement,
2. Enhance the chance of generating good structures even from a bad guess by pre-relaxation.

To verify if the pre-relaxation process introduces unwanted bias, we first performed a local perturbation (up to 0.5 Å) to the 140 known sp^2 carbon allotropes. After the pre-relaxation, all of structures returned to the original geometry, suggesting that this approach yields similar relaxation outcome to the energy-based approach when the initial structure is close to the ideal sp^2 environment. When the initial guess is far from the ideal case, the pre-relaxation not only saves a lot of computational time, but also effectively further reduces the searching space by excluding the possibility of handling low-energy configurations with an undesired local environment.

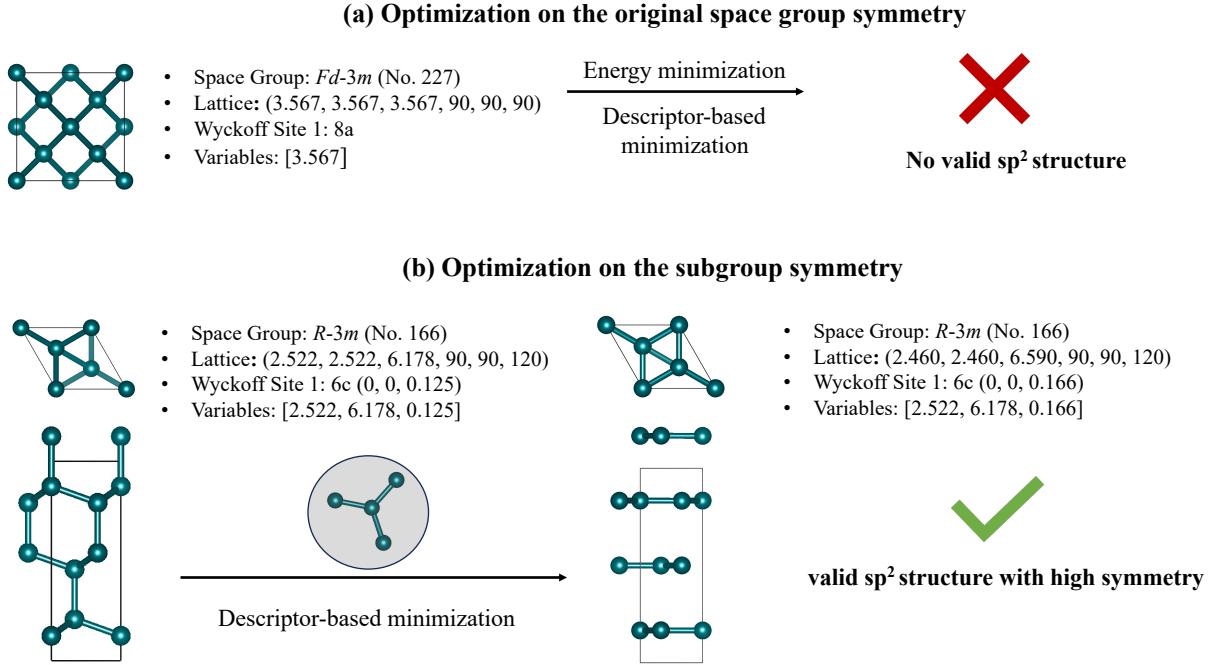


FIG. S3. Geometry optimization under the symmetry constraint from diamond structure to the sp^2 environment. (a) optimization from the original $Fd\bar{3}m$ symmetry; (b) optimization from the subgroup $R\bar{3}m$ symmetry.

Furthermore, Fig. 2 and Fig. 3 of the main text show two pre-relaxation examples that turn the seemingly irrelevant structures to perfect sp^2 -bonding carbon networks. This may look striking from the energy perspective since the initial chemical bonding is hard to break if one seeks to minimize the energy. However, it is feasible if the target is to optimize the geometry as defined by the descriptor. In Fig. S3, we also show that one can even relax a perfect sp^3 -based diamond structure to a sp^2 network.

In summary, we indeed introduced biases to force the system to transform to the desired local environment. However, this should not create artifacts if the initial guess is close to the target environment.

D. Comparison of Different Relaxation Strategies

In Table III of the manuscript, we report the timing for the MACE or ReaxFF based on the remaining valid sp^2 structures after pre-relaxation. The use of pre-relaxation is crucial for the success of the subsequent MACE or ReaxFF relaxation. This is because the VAE/GAN generated structures may contain unphysical features, such as very short interatomic distances or highly distorted bond angles, which can lead to convergence issues or unrealistic results during direct relaxation with MACE or ReaxFF. The pre-relaxation step helps to correct these issues by adjusting atomic positions and lattice parameters to more physically reasonable values, thereby providing a better starting point for the more accurate but computationally intensive MACE or ReaxFF methods. If we directly

use MACE or ReaxFF to perform the relaxation on the VAE/GAN generated samples, both the timing and success rate will become significantly worse.

To illustrate this concept, we used the VAE-cont-60k model to generate 10K structures, and then relaxed them with (1) descriptor-based pre-relaxation followed by GULP relaxation; (2) descriptor-based pre-relaxation followed by MACE relaxation; (3) GULP-ReaxFF only relaxation and (4) MACE relaxation only. The results are summarized in Table S1.

TABLE S1. Comparison of relaxation strategies for 1k samples on a 96-core CPU node.

Method	Time per 1k samples (mins)	Valid unique sp^2 structures (%)
Method		
Pre-relax + GULP	5.5 + 1.5	7.83
Pre-relax + MACE	5.5 + 2.5	7.83
GULP-only	>20	1.03
MACE-only	>120	1.83

Clearly, the GULP-only and MACE-only relaxation strategies are approximately 3 and 15 times slower, respectively, than those incorporating pre-relaxation. For both GULP-only and MACE-only relaxations, a timeout of 60 minutes was enforced to prevent calculations from stalling—a common occurrence for random structures. Furthermore, if a significant fraction of very complex structures (with more than 100 atoms per unit cell) is present, the actual computational cost may increase to 5–10 for GULP and 15–50 for MACE.

More importantly, direct relaxation using MACE or GULP-ReaxFF often yields low-energy structures with mixed sp , sp^2 , and sp^3 environments, resulting in a low success rate (less than

2%) for generating valid sp^2 structures, as compared to the 7.83% with the inclusion of pre-relaxation. This success rate is expected to decrease further when the initial structure contains multiple Wyckoff sites. Thus, the pre-relaxation step not only accelerates the process but also substantially improves the success rate.

Last, the batch-wise relaxation has been implemented, reducing the pre-relaxation time from 5 minutes per 1,000 samples to less than 1–2 minutes per 1,000 samples under the GPU environment. These improvements are available at [LEGO-xtal](#) repository and will continue to be refined. Conceptually, further speed enhancements are anticipated as GPU acceleration is adopted.

E. Pre-relaxation Examples on the Multi-component SiO_2 System

While the descriptor-based pre-relaxation method has been primarily demonstrated on single-element systems, it is equally applicable to multi-component systems. The key requirement is the availability of a reference structure that accurately represents the local environments of all constituent elements. This reference structure serves as a template for defining the desired coordination environments and bonding characteristics for each element in the multi-component system.

To demonstrate the pre-relaxation process for multi-component systems, we provide an example using SiO_2 . The reference structure is set to α -cristobalite, which contains two Wyckoff positions: one for Si (4a) and another for O (8b). The coordination numbers are specified as 4 for Si and 2 for O, with the exclusion of direct O-O bonding. The builder is configured to generate structures with space group 154 ($P3_121$) and Wyckoff positions 3a and 6c. The optimization process is performed using the descriptor-based method. The complete code is provided in Listing S4.

```
# pip install pyxtal
from pyxtal.lego.builder import builder
from pyxtal import pyxtal

# Define the reference template environments
xtal = pyxtal()
xtal.from_prototype('a-cristobalite')
print(xtal)

# Initialize the builder
bu = builder(['Si', 'O'], [1, 2],
            db_file='sio2.db',
            log_file='sio2.log',
            verbose=True)
bu.set_descriptor_calculator(mykargs={'rcut':
    2.4})
bu.set_reference_environments(xtal.to_ase())
bu.set_criteria(CN={'Si': [4], 'O': [2]},
                exclude_ii=True)
print(bu)

# Test pre-relaxatin on the known quartz structure
quartz = pyxtal()
quartz.from_prototype('a-quartz')
bu.optimize_xtal(quartz)
```

```
# Random Generation of using basin-hopping
bu.generate_xtal(spg=154,
                  wps=[['3a'], ['6c']],
                  niter=20,
                  early_quit=0.02)
```

Listing S4. A Python script to handle the SiO_2 system

The output should look like the following:

```
-----MOF Builder-----
System: Si1 O2
Database: sio2.db
Log_file: sio2.log
Descriptor: SO3 descriptor with Cutoff: 2.400
lmax: 4, nmax: 2, alpha: 1.500
Reference enviroments (2, 15)
Criterion_CN: {'Si': [4], 'O': [2]}
Criterion_cutoff: None
Criterion_exclude_ii: True

Test pre-relaxation on the existing quartz
* 9 6 152 P3121 2.46 True 14.36 => 0.00 3a 6c

Test random generation using basin-hopping
3 154 [['3a'], ['6c']] 0.2 5 0.02
* 9 6 154 P3221 2.46 0.000 3a 6c
* 9 6 154 P3221 2.46 True 0.00 => 0.00 3a 6c
```

Clearly, it suggests our code can handle multiple environment in a straight manner for relaxation, as well as generative task by using the Basin Hopping approach. The structure generation process can also be extended to the use of AI generative models such as GAN and VAE. To enable this function in LEGO-xtal workflow, one can train a VAE or GAN model using a dataset of multi-component structures. The trained model can then be used to generate new candidate structures, which can subsequently undergo descriptor-based pre-relaxation and final relaxation & energy ranking using GULP or MACE. This approach allows for the exploration of a wide range of multi-component crystal structures while ensuring that the generated structures are physically reasonable and energetically favorable.

S4. IMPACT OF TRAINING DATA

A. Pre-filtered Structures

To construct the training dataset, we initially extracted a total of 150 sp^2 crystal structures from the SACADA database. Upon careful examination, we identified 10 structures that were unsuitable for inclusion due to one or more of the following criteria: (1) a large number of atoms per unit cell (>500), (2) an excessive number of Wyckoff sites (>8), or (3) a high number of irreducible variables (>24). These 10 structures were therefore excluded from the final training set. Details of the excluded structures are provided in Table S2.

For reference, the lowest-energy sp^2 structure in the database—graphite—has a VASP energy of -9.355 eV/atom . Of the neglected structures, six exhibit relatively high energies

TABLE S2. Summary of neglected sp² structures used for training.

ID	# Atoms	DOF	Space Group	List of Wyckoff Sites	VASP energy eV/atom
1	96	18	66 (<i>Cccm</i>)	16m 16m 16m 8j 8j 8k 8k 8k	-9.123
2	40	29	10 (<i>P2/m</i>)	4o 4o 4o 4o 4o 2j 2j 2j 2j 2k 2k 2k 2l 2l	-9.106
3	72	38	150 (<i>P321</i>)	6g	-9.051
4	24	19	10 (<i>P2/m</i>)	4o 4o 4o 21 21 2l 2j 2j 2i	-9.047
5	512	11	227 (<i>Fd-3m</i>)	32e 96g 96g 192i 96g	-8.852
6	8	30	1 (<i>P1</i>)	1a 1a 1a 1a 1a 1a 1a 1a	-8.646
7	8	30	1 (<i>P1</i>)	1a 1a 1a 1a 1a 1a 1a 1a	-8.353
8	672	22	203 (<i>Fd-3</i>)	96g 96g 96g 96g 96g 96g 96g	-8.353
9	60	29	12 (<i>C2/m</i>)	8j 4i 8j 4i 4i 8j 4i 8j	-8.415
10	64	26	24 (<i>I2_12_12_1</i>)	8d 8d 8d 8d 8d 4c 4c 8d 8d	-8.086

TABLE S3. Performance of VAE-cont models trained on datasets with different sizes and unique symmetries.

Train Size	Unique Symmetries	N _{Valid}	N _{unique}	N _{train}
3K	139	19637	1852	51
8K	155	18466	2105	50
3K	320	15958	2899	61
12K	341	18149	4059	73
60K	419	16524	4862	100
120K	449	15330	5123	94
240K	485	15052	5156	101

(> 0.5 eV/atom above graphite), making them less relevant for the generation of low-energy sp² allotropes. The remaining four structures, while lower in energy, are highly complex and were excluded for simplicity. In future work, we may plan to incorporate these complex structures into the training dataset to enhance the model's ability to generate a broader range of sp² allotropes.

B. Data Augmentation

The data augmentation comes from two sources:

1. Subgroup representation is used to improve the number of unique symmetries
2. Swap of coordinates to represent symmetry operation in each Wyckoff position

Conceptually, adding more unique symmetries are very important to help the model's diversity. On the other hand, adding the swap of coordinates may be applied to the same (space group, Wyckoff positions) multiple times in current data augmentation strategy. So there may be a redundancy here.

To verify our hypothesis, we prepared seven datasets (1) 3K with 139 unique symmetries; (2) 8K with 155 unique symmetries; (3) 3K with 320 unique symmetries; (4) 12K with 341 unique symmetries; (5) 60K with 419 unique symmetries; (6) 120K with 449 unique symmetries; (7) 240K with 485 unique symmetries to train the VAE-cont models. We then generated 100,000 samples from each model and test their performances thereafter. The results are summarized in Table S3.

The results reveal a clear trend: as the training dataset size and the number of unique symmetries increase, the success rate for generating valid sp² structures decreases, likely due to increased data complexity. Conversely, models trained on smaller datasets tend to produce more duplicate structures, resulting in fewer valid unique sp² crystals and a higher rate of reproducing training data. Thus, a sufficiently large and diverse training dataset is essential for enhancing model performance in terms of both novelty and reproducibility.

Based on the test performances, we recommend using the 240k dataset. Although this dataset may include redundant information due to repeated permutations of the same Wyckoff positions, subgroup augmentation clearly enhances model performance. This is evident by directly comparing the results trained on 3K-139 and 3K-320 datasets, in which the latter (2899) generates more unique sp² crystals than the former (1852). This is not surprising since the models have learned more symmetry examples from the training data. This approach enables the model to learn both subgroup representations and the statistical data distribution of Wyckoff sites in the given space group, thereby increasing the likelihood of generating crystals with subgroup symmetries.

Finally, subgroup augmentation is expected to be even more advantageous for multicomponent systems, as it facilitates learning structural analogues (e.g., from diamond to cubic boron nitride as described in our introduction section), thereby broadening the scope of generative crystal design.

C. Subgroup Augmentation Details

To perform subgroup augmentation, we utilized the PyXtal package³⁶ to generate subgroups for each structure in our initial dataset using a generic way as described in Listing S5.

```
# pip install pyxtal
from pyxtal import pyxtal

# load a graphite crystal
xtal=pyxtal()
xtal.from_prototype('graphite')

print("Derive subgroup graphite structures")
sub_t = xtal.subgroup(eps=0.01, group_type='t')
print("t_subgroup xtals", len(sub_t))
```

```
sub_k = xtal.subgroup(eps=0.01, group_type='k',
                     max_cell=9)
print("k_subgroup xtals", len(sub_k))
```

Listing S5. A Python script to generate the subgroup structures

In this work, we report the results by using a very small random distortion (0.001 \AA). Conceptually, the benefit of subgroup augmentation is mainly to encourage the model to try new choices for the discrete combinations of space group number and Wyckoff site choices. Hence, the numerical noises on continuous variables should not impact too significantly. To verify this hypothesis, we tried to apply a larger distortion up to (0.2 \AA) and train the generative models. There seems to be no significant impact on the results, thus confirming this hypothesis.

S5. ANALYSIS OF GENERATIVE MODEL PERFORMANCE

A. Comparison with Random Sampling

Our data augmentation strategy leverages subgroup representation to enhance the diversity of generated structures, particularly by increasing the number of unique symmetries. However, it is important to assess whether a generative model trained on lattice parameters and reduced Wyckoff coordinates truly outperforms random sampling.

To address this concern, we performed a sanity check using 1,000 samples. When lattice parameters and Wyckoff positions were randomly generated for a fixed space group and Wyckoff site, only 8 unique valid structures were obtained. In contrast, our trained VAE model produced 108 unique structures under the same conditions. Furthermore, fully randomizing the tabular representation (without respecting space group or Wyckoff constraints) yielded only 2 valid structures.

These results confirm that learning a distribution over lattice and reduced coordinates substantially improves both structural diversity and validity compared to naive random sampling. Nevertheless, we acknowledge that the model may be overfitted and could generate data within a limited range. Therefore, we retain the option to allow random sampling for further exploration.

B. Saturation Analysis

As summarized in Table IV of the main text, approximately 4,000 - 6,000 unique sp^2 structures per 100,000 samples appears to be the upper limit observed across all strategies tested. However, true diversity does not fully saturate within this range. By generating increasingly larger sample sets using the VAE model trained on the v1-60k-cont dataset (specifically, batches of 1k, 10k, 50k, 100k, 200k, 300k, 400k, and 500k samples), we observe a continued increase in the total number of unique structures. Figure S4 illustrates this trend: although the rate of discovering new unique sp^2 structures decreases as

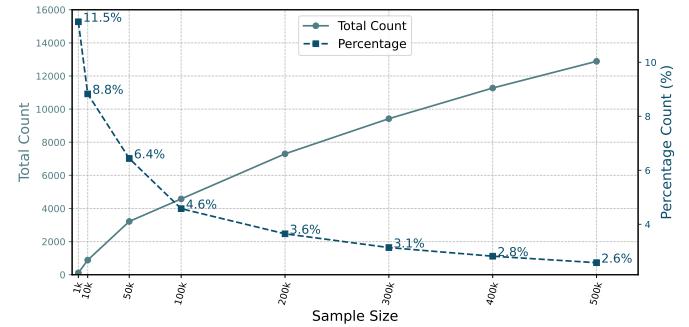


FIG. S4. **Saturation Analysis.** The number of unique sp^2 structures generated by the VAE model as a function of the total number of samples. The curve shows that while the rate of discovering new unique structures decreases with more samples, saturation is not reached, indicating continued diversity in generated structures.

more samples are generated, saturation is not reached, indicating persistent diversity in the generated dataset.

S6. AN EXTENDED LIST OF SP^2 ALLOTROPIES

The full list of structural information can be found at <https://lego-crystal.onrender.com>. Below we highlight several representative structures and their dynamical stability.

A. The 2D sp^2 Allotropes

Out of 1741 structures from our database, there are a total of 248 2D sp^2 allotropes, including (1) 53 structures featured by different kinds of graphene (hcb topology) stacking variants; (2) 6 hybrid stacking between hcb and other 2D layer motifs; (3) 189 other 2D motifs. Among the 189 other 2D motifs, they include many well known topologies (e.g., hnb, hnc, hae, hae, etc.), as well as many defective-graphene structures with various types of non-hexagonal rings (e.g., 5-7, 5-8, 4-6, 4-8, etc.). The energy versus MACE energy plot is shown in Figure S5.

Fig. S6 displays various 2D layer motifs found in the LEGO- sp^2 database. While these motifs possess higher energies than the ground state graphite layer, they represent distinct topological arrangements that could exhibit interesting electronic properties worthy of further investigation.

B. The Low-energy 3D and 0D sp^2 Allotropes

Fig. S7 displays two notable structures discussed in the main text. Fig. S7a shows the lowest-energy 3D sp^2 allotrope, featuring alternating graphite layers connected by single bonds. These structures exhibit distinctive electronic properties due to their unique connectivity pattern⁷⁴. Fig. S7b shows a 0D cage structure containing 152 carbon atoms, which represents a novel addition to the fullerene family.

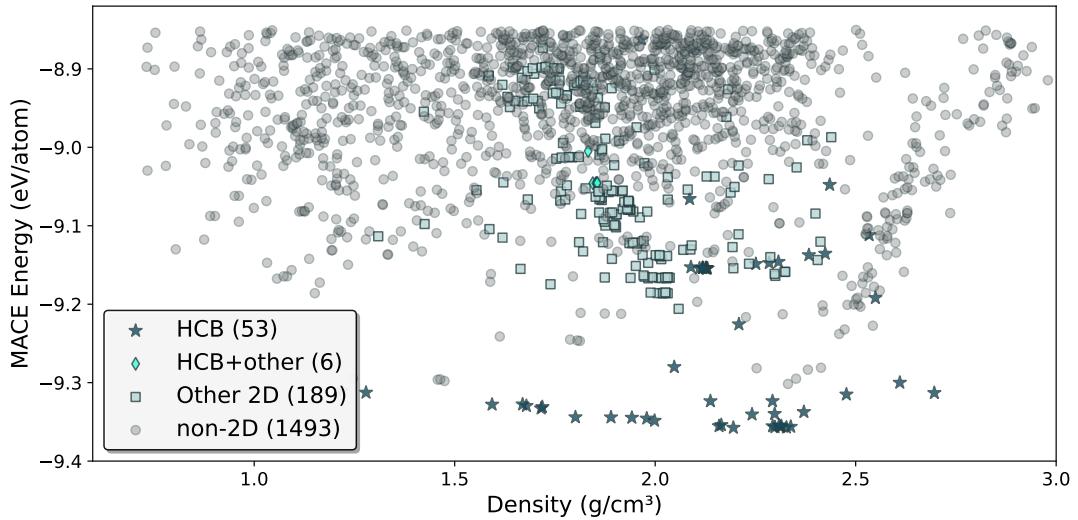


FIG. S5. Energy versus density plot for 2D sp^2 allotropes. The distribution of energies and densities for 248 unique 2D sp^2 carbon structures generated by the VAE model. The plot highlights the diversity of topologies, including graphene stacking variants and other low-energy motifs, as compared to the ground state graphite layer.

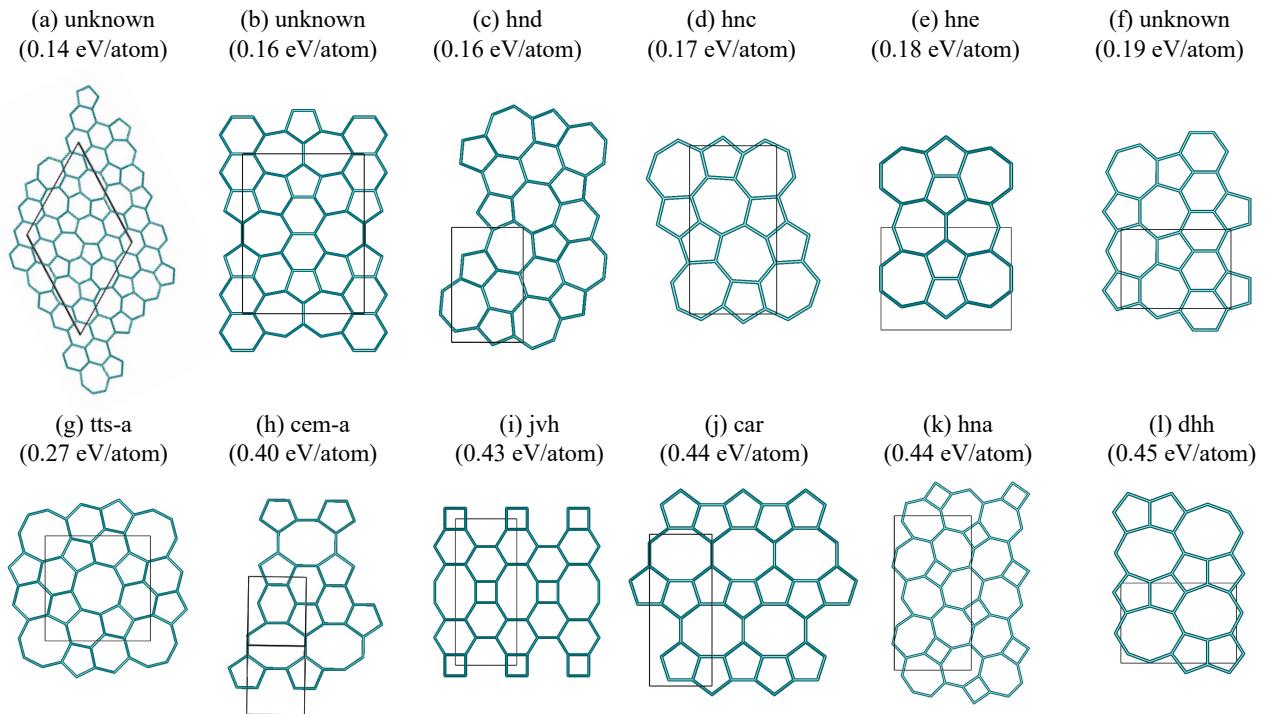


FIG. S6. An extended list of of low-energy 2D carbon sp^2 layers. The known topologies are denoted according to notation used in RCSR⁶⁶. The energy is provided based on the MACE model as compared to the ground state graphite (hcb) layer.

C. Phonon Calculations for Selected Structures

Fig. S8 displays the calculated phonon for 4 representative low-energy sp^2 allotropes that are reported in this work for the first time, including (a) 2D-hP52, (b) NCG1-cP192, (c) NCG2-cP108 and (d) NCG3-cP192.

REFERENCES

- ¹J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al., “Highly accurate protein structure prediction with alphafold,” *Nature* **596**, 583–589 (2021).

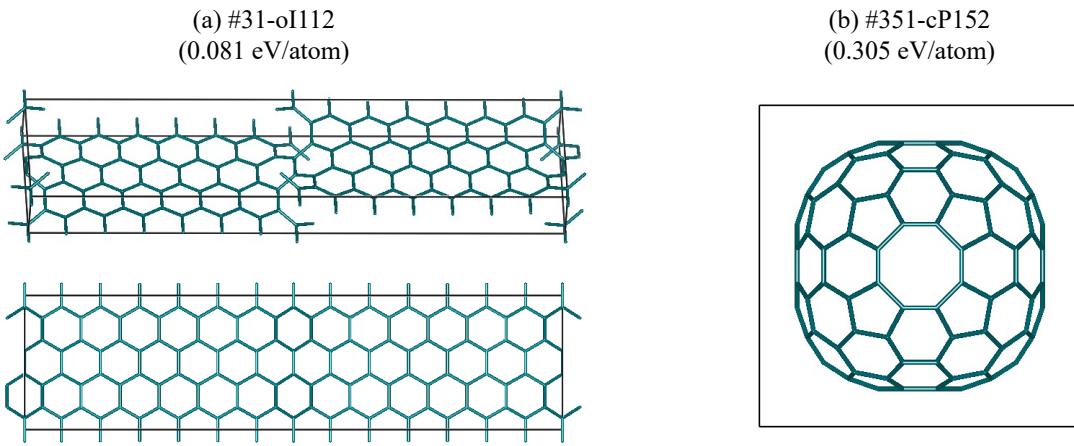


FIG. S7. Two low-energy structures. (a) the lowest-energy 3D structure with alterative graphite layer stacking; (b) a 0D structure with the C₁₅₂ cage. The indices in the labels correspond to the entry ids in <https://lego-crystal.onrender.com> and the energy values are shown as the relative DFT-r²SCAN energies as compared to the ground state graphite structure.

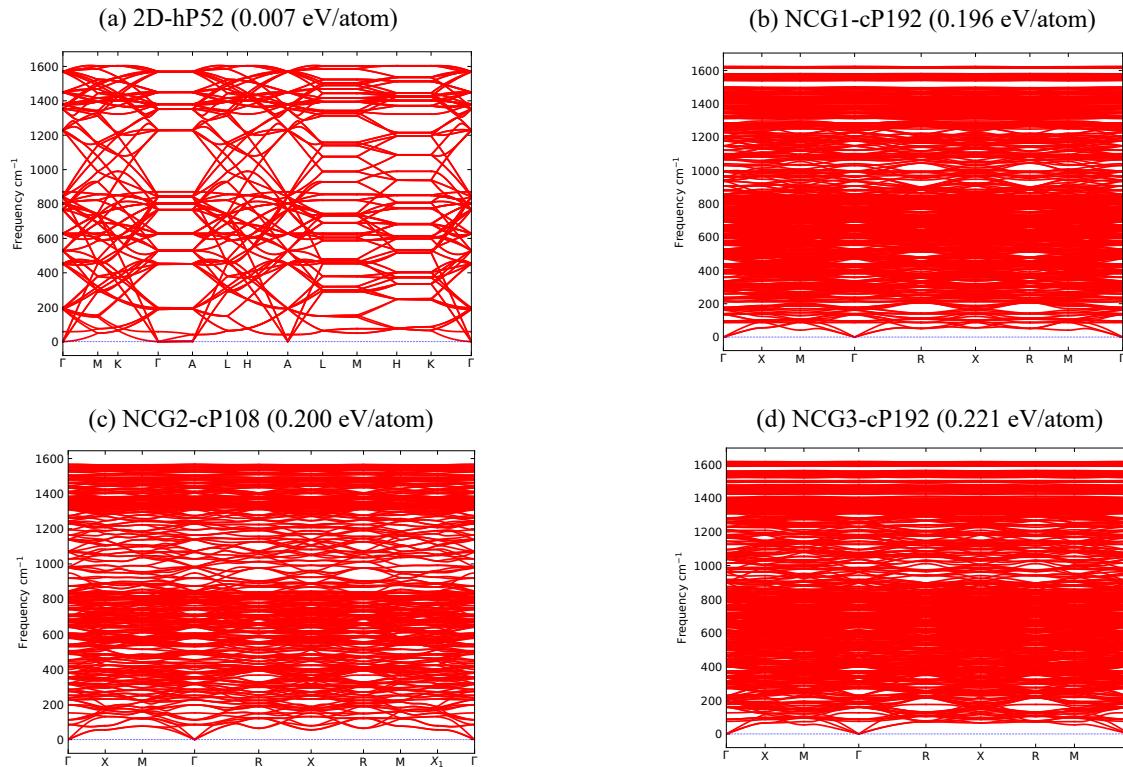


FIG. S8. Calculated phonon dispersions for a few selected low-energy structures at the PBE level of theory. (a) is the twisted graphene shown in Figure 6b. (b)-(d) are NCG1-3 structures shown in Figure 7.

²A. R. Oganov, C. J. Pickard, Q. Zhu, and R. J. Needs, “Structure prediction drives materials discovery,” *Nat. Rev. Mater.* **4**, 331–348 (2019).

³C. J. Pickard and R. Needs, “Ab initio random structure searching,” *J. Phys.: Condens. Matter* **23**, 053201 (2011).

⁴A. R. Oganov and C. W. Glass, “Crystal structure prediction using ab initio evolutionary techniques: Principles and applications,” *J. Chem. Phys.* **124**, 244704 (2006).

⁵A. R. Oganov, A. O. Lyakhov, and M. Valle, “How evolutionary crystal structure prediction works and why,” *Acc. Chem. Res.* **44**, 227–237 (2011).

⁶F. H. Stillinger, “Exponential multiplicity of inherent structures,” *Phys. Rev. E* **59**, 48 (1999).

⁷S. Martiniani, K. J. Schrenk, J. D. Stevenson, D. J. Wales, and D. Frenkel, “Structural analysis of high-dimensional basins of attraction,” *Phys. Rev. E* **94**, 031301 (2016).

- ⁸Y. Wang, J. Lv, L. Zhu, and Y. Ma, "Crystal structure prediction via particle-swarm optimization," *Phys. Rev. B* **82**, 094116 (2010).
- ⁹D. C. Lonie and E. Zurek, "Xtalopt: An open-source evolutionary algorithm for crystal structure prediction," *Comput. Phys. Commun.* **182**, 372–387 (2011).
- ¹⁰B. C. Revard, W. W. Tipton, and R. G. Hennig, "Genetic algorithm for structure and phase prediction," (2018).
- ¹¹J. Pannetier, J. Bassas-Alsina, J. Rodriguez-Carvajal, and V. Caignaert, "Prediction of crystal structures from crystal chemistry rules by simulated annealing," *Nature* **346**, 343 (1990).
- ¹²A. Banerjee, D. Jasrasaria, S. P. Niblett, and D. J. Wales, "Crystal structure prediction for benzene using basin-hopping global optimization," *J. Phys. Chem. A* **125**, 3776–3784 (2021).
- ¹³Y. Han, C. Ding, J. Wang, H. Gao, J. Shi, S. Yu, Q. Jia, S. Pan, and J. Sun, "Efficient crystal structure prediction based on the symmetry principle," *Nat. Comput. Sci.* **5**, 1–13 (2025).
- ¹⁴A. O. Lyakhov, A. R. Oganov, and M. Valle, "How to predict very large and complex crystal structures," *Comput. Phys. Commun.* **181**, 1623–1632 (2010).
- ¹⁵Q. Zhu, A. R. Oganov, and A. O. Lyakhov, "Evolutionary metadynamics: a novel method to predict crystal structures," *CrystEngComm* **14**, 3596–3601 (2012).
- ¹⁶Q. Zhu, A. R. Oganov, A. O. Lyakhov, and X. Yu, "Generalized evolutionary metadynamics for sampling the energy landscapes and its applications," *Phys. Rev. B* **92**, 024106 (2015).
- ¹⁷S. Kim, J. Noh, G. H. Gu, A. Aspuru-Guzik, and Y. Jung, "Generative adversarial networks for crystal structure prediction," *ACS Central Sci.* **6**, 1412–1420 (2020).
- ¹⁸R. Zhu, W. Nong, S. Yamazaki, and K. Hippalgaonkar, "Wycryst: Wyckoff inorganic crystal generator framework," *Matter* **7**, 3469–3488 (2024).
- ¹⁹J. 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* **1**, 1370–1384 (2019).
- ²⁰A. Merchant, S. Batzner, S. S. Schoenholz, M. Aykol, G. Cheon, and E. D. Cubuk, "Scaling deep learning for materials discovery," *Nature* **624**, 80–85 (2023).
- ²¹T. 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), 10.48550/arXiv.2110.06197.
- ²²R. Jiao, W. Huang, P. Lin, J. Han, P. Chen, Y. Lu, and Y. Liu, "Crystal structure prediction by joint equivariant diffusion," in *Advances in Neural Information Processing Systems*, Vol. 36, edited by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Curran Associates, Inc., 2023) pp. 17464–17497.
- ²³C. Zeni, R. Pinsler, D. Zügner, A. Fowler, M. Horton, X. Fu, Z. Wang, A. Shysheya, J. Crabbé, S. Ueda, et al., "A generative model for inorganic materials design," *Nature* **639**, 624–632 (2025).
- ²⁴D. Levy, S. S. Panigrahi, S.-O. Kaba, Q. Zhu, M. Galkin, S. Miret, and S. Ravanbakhsh, "SymmCD: Symmetry-preserving crystal generation with diffusion models," in *AI for Accelerated Materials Design - NeurIPS 2024* (2024).
- ²⁵R. Jiao, W. Huang, Y. Liu, D. Zhao, and Y. Liu, "Space group constrained crystal generation," *arXiv preprint arXiv:2402.03992* (2024), 10.48550/arXiv.2402.03992.
- ²⁶B. K. Miller, R. T. Q. Chen, A. Sriram, and B. M. Wood, "FlowMM: Generating materials with riemannian flow matching," in *Forty-first International Conference on Machine Learning* (2024).
- ²⁷P. Zhong, X. Dai, B. Deng, G. Ceder, and K. A. Persson, "Practical approaches for crystal structure predictions with inpainting generation and universal interatomic potentials," *arXiv preprint arXiv:2504.16893* (2025), 10.48550/arXiv.2504.16893.
- ²⁸L. M. Antunes, K. T. Butler, and R. Grau-Crespo, "Crystal structure generation with autoregressive large language modeling," *Nat. Commun.* **15**, 1–16 (2024).
- ²⁹N. Gruver, A. Sriram, A. Madotto, A. G. Wilson, C. L. Zitnick, and Z. Ulissi, "Fine-tuned language models generate stable inorganic materials as text," *arXiv preprint arXiv:2402.04379* (2024), 10.48550/arXiv.2402.04379.
- ³⁰Z. Cao, X. Luo, J. Lv, and L. Wang, "Space group informed transformer for crystalline materials generation," *arXiv preprint arXiv:2403.15734* (2024), arXiv:2403.15734 [cond-mat.mtrl-sci].
- ³¹N. Kazeev, W. Nong, I. Romanov, R. Zhu, A. Ustyuzhanin, S. Yamazaki, and K. Hippalgaonkar, "Wyckoff transformer: Generation of symmetric crystals," *arXiv preprint arXiv:2503.02407* (2025), 10.48550/arXiv.2503.02407.
- ³²A. K. Cheetham and R. Seshadri, "Artificial intelligence driving materials discovery? perspective on the article: Scaling deep learning for materials discovery," *Chem. Mater.* **36**, 3490–3495 (2024).
- ³³B. Souvignier, H. Wondratschek, M. I. Aroyo, G. Chapuis, and A. M. Glazer, "Chapter 1.4. space groups and their descriptions," *International Tables for Crystallography* , 42–74 (2016).
- ³⁴S. R. Hall, F. H. Allen, and I. D. Brown, "The crystallographic information file (CIF): a new standard archive file for crystallography," *Acta Cryst. Sec. A* **47**, 655–685 (1991).
- ³⁵H. Wondratschek and U. Müller, *Symmetry relations between space groups* (International Union of Crystallography, 2006).
- ³⁶S. Fredericks, K. Parrish, D. Sayre, and Q. Zhu, "Pyxtal: A python library for crystal structure generation and symmetry analysis," *Comput. Phys. Commun.* **261**, 107810 (2021).
- ³⁷P. J. Steinhardt, D. R. Nelson, and M. Ronchetti, "Bond-orientational order in liquids and glasses," *Phys. Rev. B* **28**, 784–805 (1983).
- ³⁸M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3 d shape descriptors," in *Symposium on geometry processing*, Vol. 6 (2003) pp. 156–164.
- ³⁹A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," *Phys. Rev. B* **87**, 184115 (2013).
- ⁴⁰D. Zagaceta, H. Yanxon, and Q. Zhu, "Spectral neural network potentials for binary alloys," *J. Appl. Phys.* **128**, 045113 (2020).
- ⁴¹H. Yanxon, D. Zagaceta, B. Tang, D. S. Matteson, and Q. Zhu, "Pyxtal_ff: a python library for automated force field generation," *Machine Learning: Sci. Tech.* **2**, 027001 (2020).
- ⁴²S. Tao, X. Shao, and L. Zhu, "Accelerating structural optimization through fingerprinting space integration on the potential energy surface," *J. Phys. Chem. Lett.* **15**, 3185–3190 (2024).
- ⁴³C. J. Pickard, "Beyond theory-driven discovery: introducing hot random search and datum-derived structures," *Faraday Discussions* **256**, 61–84 (2025).
- ⁴⁴R. Hoffmann, A. A. Kabanov, A. A. Golov, and D. M. Proserpio, "Homo citans and carbon allotropes: for an ethics of citation," *Angew. Chem. Int. Ed.* **55**, 10962–10976 (2016).
- ⁴⁵T. Lenosky, X. Gonze, M. Teter, and V. Elser, "Energetics of negatively curved graphitic carbon," *Nature* **355**, 333–335 (1992).
- ⁴⁶S. H. Pun and Q. Miao, "Toward negatively curved carbons," *Acc. Chem. Res.* **51**, 1630–1642 (2018).
- ⁴⁷E. Braun, Y. Lee, S. M. Moosavi, S. Barthel, R. Mercado, I. A. Baburin, D. M. Proserpio, and B. Smit, "Generating carbon schwarzites via zeolite-templating," *Proc. Nat. Acad. Sci.* **115**, E8116–E8124 (2018).
- ⁴⁸J. P. Perdew, K. Burke, and M. Ernzerhof, "Generalized gradient approximation made simple," *Phys. Rev. Lett.* **77**, 3865–3868 (1996).
- ⁴⁹N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault," in *IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (2016) pp. 399–410.
- ⁵⁰L. Xu, M. Skouliaridou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," in *Advances in Neural Information Processing Systems* (2019).
- ⁵¹I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, Vol. 30, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017).
- ⁵²D. P. Kingma, M. Welling, et al., "Auto-encoding variational bayes," (2013).
- ⁵³D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. ICML*, Proceedings of Machine Learning Research, Vol. 32, edited by E. P. Xing and T. Jebara (PMLR, Beijing, China, 2014) pp. 1278–1286.
- ⁵⁴P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, I. Polat, Y. Feng, E. W. Moore,

- J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors, “SciPy 1.0—Fundamental Algorithms for Scientific Computing in Python,” *arXiv e-prints*, arXiv:1907.10121 (2019).
- ⁵⁵J. A. Nelder and R. Mead, “A simplex method for function minimization,” *The Computer Journal* **7**, 308–313 (1965).
- ⁵⁶C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, “Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization,” *ACM Transactions on mathematical software (TOMS)* **23**, 550–560 (1997).
- ⁵⁷D. J. Wales and J. P. Doye, “Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms,” *J. Phys. Chem. A* **101**, 5111–5116 (1997).
- ⁵⁸A. C. Van Duin, S. Dasgupta, F. Lorant, and W. A. Goddard, “Reaxff: a reactive force field for hydrocarbons,” *J. Phys. Chem. A* **105**, 9396–9409 (2001).
- ⁵⁹J. D. Gale, “Gulp: A computer program for the symmetry-adapted simulation of solids,” *J. Chem. Soc., Faraday Trans.* **93**, 629–637 (1997).
- ⁶⁰I. Batatia, D. P. Kovacs, G. N. C. Simm, C. Ortner, and G. Csanyi, “MACE: Higher order equivariant message passing neural networks for fast and accurate force fields,” in *Advances in Neural Information Processing Systems*, edited by A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho (2022).
- ⁶¹A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dulak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, et al., “The atomic simulation environment—a python library for working with atoms,” *J. Phys.: Condens. Matter* **29**, 273002 (2017).
- ⁶²G. Kresse and J. Furthmüller, “Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set,” *Phys. Rev. B* **54**, 11169–11186 (1996).
- ⁶³J. W. Furness, A. D. Kaplan, J. Ning, J. P. Perdew, and J. Sun, “Accurate and numerically efficient r²scan meta-generalized gradient approximation,” *J. Phys. Chem. Lett.* **11**, 8208–8215 (2020).
- ⁶⁴J. W. Furness, A. D. Kaplan, J. Ning, J. P. Perdew, and J. Sun, “Correction to ‘accurate and numerically efficient r²scan meta-generalized gradient approximation’,” *J. Phys. Chem. Lett.* **11**, 9248–9248 (2020).
- ⁶⁵L. Zoubritzky and F.-X. Coudert, “Crystalnets. jl: identification of crystal topologies,” *SciPost Chem.* **1**, 005 (2022).
- ⁶⁶M. O’keeffe, M. A. Peskov, S. J. Ramsden, and O. M. Yaghi, “The reticular chemistry structure resource (rcsr) database of, and symbols for, crystal nets,” *Acc. Chem. Res.* **41**, 1782–1789 (2008).
- ⁶⁷E. Jang, S. Gu, and B. Poole, “Categorical reparameterization with gumbel-softmax,” in *International Conference on Learning Representations* (2017).
- ⁶⁸I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, “beta-vae: Learning basic visual concepts with a constrained variational framework,” in *International conference on learning representations* (2017).
- ⁶⁹H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, “Cyclical annealing schedule: A simple approach to mitigating kl vanishing,” *arXiv preprint arXiv:1903.10145* (2019).
- ⁷⁰T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision* (2017) pp. 2980–2988.
- ⁷¹R. Bistritzer and A. H. MacDonald, “Moiré bands in twisted double-layer graphene,” *Proc. Nat. Acad. Sci.* **108**, 12233–12237 (2011).
- ⁷²Y. Cao, V. Fatemi, S. Fang, K. Watanabe, T. Taniguchi, E. Kaxiras, and P. Jarillo-Herrero, “Unconventional superconductivity in magic-angle graphene superlattices,” *Nature* **556**, 43–50 (2018).
- ⁷³A. D. Haymet, “C120 and c60: Archimedean solids constructed from sp2 hybridized carbon atoms,” *Chem. Phys. Lett.* **122**, 421–424 (1985).
- ⁷⁴Z. Zhao, Z. Zhang, and W. Guo, “A family of all sp 2-bonded carbon allotropes of topological semimetals with strain-robust nodal-lines,” *J. Mater. Chem. C* **8**, 1548–1555 (2020).
- ⁷⁵A. Mackay and H. Terrones, “Diamond from graphite,” *Nature* **352**, 762–762 (1991).
- ⁷⁶M. O’Keeffe, G. B. Adams, and O. F. Sankey, “Predicted new low energy forms of carbon,” *Phys. Rev. Lett.* **68**, 2325–2328 (1992).
- ⁷⁷A. L. Mackay and H. Terrones, “Hypothetical graphite structures with negative gaussian curvature,” *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences* **343**, 113–127 (1993).
- ⁷⁸S. J. Townsend, T. J. Lenosky, D. A. Muller, C. S. Nichols, and V. Elser, “Negatively curved graphitic sheet model of amorphous carbon,” *Phys. Rev. Lett.* **69**, 921–924 (1992).
- ⁷⁹R. Phillips, D. A. Drabold, T. Lenosky, G. B. Adams, and O. F. Sankey, “Electronic structure of schwarzite,” *Phys. Rev. B* **46**, 1941–1943 (1992).
- ⁸⁰M. Tagami, Y. Liang, H. Naito, Y. Kawazoe, and M. Kotani, “Negatively curved cubic carbon crystals with octahedral symmetry,” *Carbon* **76**, 266–274 (2014).
- ⁸¹R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proc. CVPR* (2022) pp. 10684–10695.
- ⁸²C.-H. Yao, Y. Xie, V. Voleti, H. Jiang, and V. Jampani, “Sv4d 2.0: Enhancing spatio-temporal consistency in multi-view video diffusion for high-quality 4d generation,” *arXiv* (2025), 10.48550/arXiv.2503.16396.
- ⁸³H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, “Maskgit: Masked generative image transformer,” in *Proc. CVPR* (2022) pp. 11315–11325.
- ⁸⁴I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *Commun. ACM* **63**, 139–144 (2020).
- ⁸⁵S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 37, edited by F. Bach and D. Blei (PMLR, Lille, France, 2015) pp. 448–456.
- ⁸⁶K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016) pp. 770–778.
- ⁸⁷A. L. Maas, A. Y. Hannun, A. Y. Ng, et al., “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. ICML*, Vol. 30 (Atlanta, GA, 2013) p. 3.
- ⁸⁸D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” (2017), arXiv:1412.6980.