CrysAtom: Distributed Representation of Atoms for Crystal Property Prediction

Shrimon Mukherjee[1]*, **Madhusudan Ghosh**[1]*, **Partha Basuchowdhuri**[1]
[1]School of Mathematical & Computational Sciences, Indian Association for the Cultivation of Science, India
{shrimonmukherjee,madhusuda.iacs}@gmail.com, partha.basuchowdhuri@iacs.res.in

Abstract

Application of artificial intelligence (AI) has been ubiquitous in the growth of research in the areas of basic sciences. Frequent use of machine learning (ML) and deep learning (DL) based methodologies by researchers has resulted in significant advancements in the last decade. These techniques led to notable performance enhancements in different tasks such as protein structure prediction, drug-target binding affinity prediction, and molecular property prediction. In material science literature, it is well-known that crystalline materials exhibit topological structures. Such topological structures may be represented as graphs and utilization of graph neural network (GNN) based approaches could help encoding them into an augmented representation space. Primarily, such frameworks adopt supervised learning techniques targeted towards downstream property prediction tasks on the basis of electronic properties (formation energy, bandgap, total energy, etc.) and crystalline structures. Generally, such type of frameworks rely highly on the handcrafted atom feature representations along with the structural representations. In this paper, we propose an unsupervised framework namely, CrysAtom, using untagged crystal data to generate dense vector representation of atoms, which can be utilized in existing GNN-based property predictor models to accurately predict important properties of crystals. Empirical results show that our dense representation embeds chemical properties of atoms and enhance the performance of the baseline property predictor models significantly.

1 Introduction

In recent years, there has been a significant surge in applying machine learning (ML) algorithms across various disciplines, including material science and chemistry, where ML advancements are leveraged to address domain-specific challenges [1–4], such as property prediction, molecule generation, and discovery of key descriptors for CO2 activation [5, 6]. Despite the reliance on density functional theory (DFT) simulations in early material science works [7], their resource-intensive nature prompted a shift towards ML-based strategies to replace high-latency computational processes with efficient approximations [8, 9]. ML techniques depend on handcrafted features, whereas deep learning algorithms learn feature representations, mitigating the need for domain-expert intervention [7]. In material science, crystal structures play a significant role for most of the downstream tasks [5]. Since the majority of the crystals are available in nature as three-dimensional (3D) structures, they are initially transformed into graphs by preserving their periodic invariance [5]. Such graphs are used in graph neural network based frameworks for solving different downstream property prediction tasks [5, 10–12].

For the downstream tasks, atoms are commonly initialized using a one-hot sparse representation, leading to suboptimal performance [9, 13]. In contrast, distributed representations encapsulate richer semantic and structural information [14, 15]. Atom2Vec [13] proposed a singular value decomposition (SVD) based distributed atom vector representation using handcrafted feature vectors, requiring domain knowledge. To alleviate this problem, SkipAtom [9] introduced a skip-gram [15] based dense representation of the atoms by learning the required feature representation from the dataset.

S. Mukherjee et al., CrysAtom: Distributed Representation of Atoms for Crystal Property Prediction. *Proceedings of the Third Learning on Graphs Conference (LoG 2024)*, PMLR 269, Virtual Event, November 26–29, 2024.

*Equal contributions

However, neither consider structural representation of the crystal materials, that can be harnessed by neural network-based models for improving dense representations. To mitigate this challenge, we investigate the feasibility of utilizing the graph structure information into the neural network framework towards generating distributed atom vector representations. On this note, we propose a novel auto-encoder-decoder based framework, namely Crystal Atom Vector Extractor (**CrysAtom**), to learn distributed representations of molecular atoms (shown in Figure 1i) by introducing a fusion mechanism by combining Self-Supervised Learning (SSL) and Unsupervised Learning (UL)-based techniques. The key distinction of our approach from existing work [16] lies in the adoption of a novel fusion mechanism, characterized by a generalized SSL loss for pretraining task. Unlike previous methods, our SSL techniques are uniquely generalized, requiring no external information such as space groups. Furthermore, we utilize our proposed distributed vector representation for different downstream property prediction tasks and analyze its performance.

Our Contributions: To the best of our knowledge, we are the first to investigate the feasibility of applying an auto-encoder-decoder based graph neural network approach to obtain a domain-independent generic distributed representation of atoms. We assess the quality of our distributed representation by comparing it with other existing state-of-the-art (SOTA) representations of atoms (Atom2Vec and SkipAtom) in multiple property prediction tasks. We use two popular benchmark materials datasets to show that our distributed representation of atoms obtained from CrysAtom helps gain substantial improvement in performance for several property predictor models (CGCNN, ALIGNN, etc) over their vanilla (from 5.21% to 21.92%), distilled and fine-tuned versions. Moreover, the property-tagged dataset suffers from error bias, as it is theoretically derived from DFT. We successfully mitigate this issue using a small set of experimental data in the training setup.

2 Related Works

In recent times, data-driven approach specifically the graph neural network based frameworks [5, 17, 18] played a crucial role to conduct the property prediction task by utilizing the topological structures of the crystal materials. Earlier studies [5, 10, 19–21] did not comprise of the periodic invariance properties. Later, Matformer [11], applied a periodic graph transformer based framework by employing both periodic invariance and periodic pattern encoding strategy. Similarly, PotNet [12] used interatomic potentials for the property prediction tasks. Recently, M3GNet [22], and CHGNet [23] introduced universal interatomic potentials into their architecture to capture complex structure of crystal materials. In a few previous works [24–26] atomistic simulations have been used to predict crystal materials properties. Additionally, there are several works which employed UL [7, 16] and SSL [27, 28] strategies to apply the pretraining task followed by task-specific finetuning. All prior works used sparse one-hot representations for node features. We propose a novel CrysAtom framework that generates distributed atom vector representations by incorporating graph structure into the encoder and introducing a generalized unsupervised contrastive loss, requiring no external information like space groups, as described by Das et.al. [16]. Also, there are several algorithms [23, 29] which use pre-train and finetune techniques. Recently, Gupta et.al. [30] proposed a transfer learning based method to predict the property of crystals by creating atom and angle based features. However, they did not generate generalized dense vector representation of atoms. In their method, atom features are generated by training on a property (formation energy) tagged data and thus their dense vector is biased towards formation energy property.

3 Methodology

This section commences by outlining the general idea of our proposed neural framework, namely CrysAtom[2], for generating dense vector representation of chemical atoms. Subsequently, a detailed discussion about the different components of our proposed framework follows. An overview of our proposed CrysAtom framework has been shown in Figure 1ii. Table 5 summarizes important terms and corresponding notations used in this work, which is present in Appendix.

3.1 CrysAtom

In this work, we propose a novel encoder-decoder based neural framework, CrysAtom, to generate dense vector representations of the chemical atoms, which can be used to enhance the performance of SOTA downstream neural property predictor models present in the literature of material science [5, 7, 11, 16]. To generate the dense vector representation, we first consider a collection of untagged crystal graphs $D_{ut} = \{G_i\}$ collected from the well-known materials database, which serves as input to our proposed CrysAtom model (f_θ). Similar to some of the state-of-the-art methods, popularly known for generating dense vector representations of atoms, such as SkipAtom and Atom2Vec, we use untagged

[2] Our source code is available at `https://github.com/shrimonmuke0202/CrysAtom`

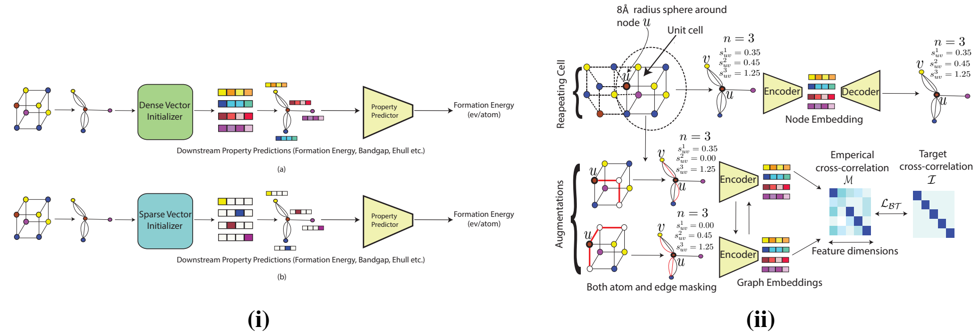
Dense Vector Initializer
Property Predictor
Formation Energy (ev/atom)
Downstream Property Predictions (Formation Energy, Bandgap, Ehull etc.)
(a)
Sparse Vector Initializer
Property Predictor
Formation Energy (ev/atom)
Downstream Property Predictions (Formation Energy, Bandgap, Ehull etc.)
(b)
(i)
8Å radius sphere around node u
Unit cell
Reapeating Cell
u
v
$n = 3$
$s^1_{uv} = 0.35$
$s^2_{uv} = 0.45$
$s^3_{uv} = 1.25$
Encoder
Decoder
Node Embedding
Augmentations
$s^1_{uv} = 0.35$
$s^2_{uv} = 0.00$
$s^3_{uv} = 1.25$
$s^1_{uv} = 0.00$
$s^2_{uv} = 0.45$
$s^3_{uv} = 1.25$
Emperical cross-correlation $\mathcal{M}$
Target cross-correlation $\mathcal{I}$
$\mathcal{L}_{\mathcal{BT}}$
Feature dimensions
Graph Embeddings
Both atom and edge masking
(ii)

Figure 1: (i) Workflow showing the downstream property prediction tasks with the (a) application of the distributed vector representation obtained from CrysAtom framework, and (b) application of sparse atom vector representation. (ii) Schematic diagram illustrating the architecture of our proposed framework, **CrysAtom**, highlighting the key components. The diagram encapsulates the novel mechanisms underlying CrysAtom's functionality.

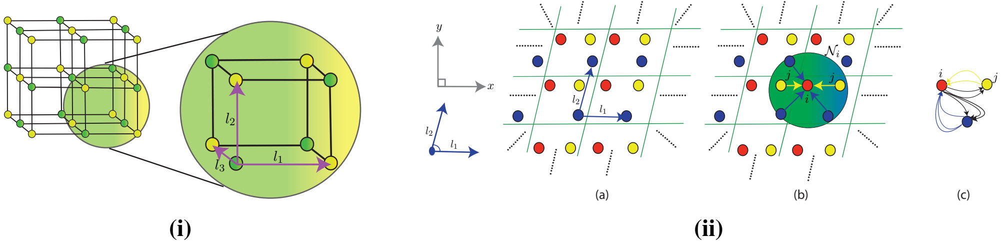

Figure 2: (i) This illustration showcases a periodic crystal structure, featuring a point cloud of atoms arranged in repeating patterns. The image includes a magnified view of a unit cell, clearly delineating the lattice vectors ($L = [l_1, l_2, l_3]$), highlighting the fundamental building blocks of the crystal's geometric arrangement. (ii) Illustration of the multigraph representation of a crystal. We use blue arrows, l_1 and l_2, for the a crystal structure in 2D. We use circles of different colors to represent different atoms and green lines to denote periodic boundaries. We use $\mathcal{N}_i$ as the neighborhood set of node i, and use yellow and blue arrows to specify the captured atomic interactions from the multigraph of the given crystal. (a) A crystal structure with periodic patterns l_1 and l_2 is shown in 2D. (b) Atomic interactions between the yellow nodes j and the center red node i, captured by the multigraph of the crystal. (c) The corresponding multigraph. All the periodic duplicates of j in the crystal are mapped to a single node j in the multigraph.

data for generating dense vectors with better generalization capabilities. Subsequently, it generates the dense vector representation of the chemical atoms by learning the intrinsic structural and chemical patterns from the input representation of the crystal graphs. Additionally, to establish the effectiveness of the vector representations generated by CrysAtom, we conduct extensive theoretical and empirical analysis. For empirical analysis, we consider chemical property prediction, which is an important and challenging task in material science literature [7, 16].

Crystal Graph Representation. As proposed by Xie et. al. [5], we use crystal graph structures $D = \{G_i = (V_i, E_i, \chi_i, F_i)\}$ to represent crystalline materials. A crystal lattice is formed by repeating a unit cell in all three dimensions as shown in Figure 2i. G_i is an undirected weighted multigraph that represents unit cell of the crystal structure. V_i denotes a set of nodes (atoms) present in the unit cell and $E_i = \{(u, v, n_{uv})\}$ denotes a set of triples, where each triple consists of three entries - a pair of nodes and the number of edges between them. It signifies that v is an atom that appears n_{uv} times in the *nearby cells* around u. The *nearby cells* are defined as the cells that are within a distance of radius r from u. Therefore, r is a hyper-parameter for this model and the set of *nearby cells* may be expressed as,

$$\mathcal{N}_r(u) = \{v \in V_i \mid dist(u, v) \leq r\},$$

where V_i is the set of atoms present in the crystal graph G_i of crystal i and $\mathcal{N}_r(u)$ is the set of *nearby cells* for u given r. In the later part of the paper, we have often used the terms nodes and atoms interchangeably. The pictorial representation of creating multigraph from crystal structure has been shown in Figure 2ii. χ_i represents node features i.e., features which comprehensively represent the chemical properties of an atom, such as atomic volume, electron affinity, etc. Finally, F_i represents a collection of edge weights between a pair of atoms in a crystal graph. In other words, $F_i = \{\{s^n\}_{(u,v)} \mid (u, v) \in E_i\}$ is a set that represents a collection of bond length values between each pair of nodes (u, v) that are connected by an edge in E_i. The bond length is denoted as s^n, where s is the length of one bond and n is the number of bonds (hence, the number of edges) between u and v. We consider bond length as a measure of distance from one atom to other atoms in close proximity. In the next section, we explain our proposed methodology for generating dense vector representation of atoms and subsequently analyze its effectiveness in property predictor models.

3.2 Atom vector formation

In this part, we discuss the architecture of the proposed CrysAtom model, as shown Figure 1ii. It consists of an auto-encoder with an SSL framework that leverages the correlations in the input to learn robust and generalizable dense vector representation of atoms.

Encoder. We develop our auto-encoder module by employing crystal graph convolutional neural network (CGCNN) [5]. We utilize CGCNN to encode the chemical and structural information of a crystal graph G. It encodes information of the l-hop neighborhood for each node by applying following equations:

$$z_{(u,v)_n}^{l-1} = x_u^{l-1} \oplus x_v^{l-1} \oplus s_{(u,v)}^n \tag{1}$$

$$x_u^l = x_u^{l-1} + \sum_{v,n} \sigma(z_{(u,v)_n}^{l-1} \Theta_c^{l-1} + b_c^{l-1}) \odot g(z_{(u,v)_n}^{l-1} \Theta_s^{l-1} + b_s^{l-1})$$

where, l is the number of CGCNN layers, x_u^{l-1} denotes the embedding of node u by aggregating the $l-1$ hop neighborhood information. The embedding of node u is initialized to a transformed node feature vector, i.e., it is a function of the atom u's chemical features such as $x_u^0 = \chi_u \Theta_\chi$ where Θ_χ is the list of trainable parameters of the transformation network and χ_u is the input node feature vector. $s_{(u,v)}^n \in F_u$ represents the overall bond length between atoms u and v. The operator $\oplus$ denotes concatenation and $\odot$ denotes element-wise multiplication. Here, Θ_c^{l-1}, Θ_s^{l-1}, b_c^{l-1}, b_s^{l-1} are the convolution matrix, self-weight matrix, convolution bias, self-bias of $(l-1)$th layer convolution, respectively. σ is a nonlinear transformation function, used to generate a real value in [0,1] indicating the edge importance and g is a feed-forward network. Finally, we collect local information at each node after aggregating the information from the neighborhood (x_u^l). We denote the set of trainable parameters for this encoder as Θ_e for future reference.

Decoder. The encoder encodes the chemical properties of an atom into a latent vector space x by learning the structural and chemical information. Subsequently, the decoder tries to decode the vectors from x, thereby enhancing the encoding capability of the encoder. As mentioned earlier, the crystal properties depend on the local chemical environment and the overall conformation of the repeating crystal cell structure. Therefore, we build our decoder framework to reconstruct two important features to capture the properties of local chemical environment: (a) the node features, which are the chemical properties of individual atoms, and (b) the local connectivity, which is the relative position of the nodes with respect to their local neighbors. We employ the node feature reconstruction strategy by computing the following equations,

$$\hat{\chi_u} = \Theta_x^T x_u^l + b_x \tag{2}$$

$$\mathcal{L}_{\mathcal{FR}} = -\chi_u \cdot \log(\hat{\chi_u}) - (1 - \chi_u) \cdot \log(1 - \hat{\chi_u}) \tag{3}$$

where Θ_x, b_x and $\mathcal{L}_{\mathcal{FR}}$ are the trainable weights, biases and the feature reconstruction loss, respectively.

Furthermore, we reconstruct the global topological information to generate the connectivity and the periodicity information of the crystal structures by employing a bilinear transformation strategy, with the help of the following equations,

$$x_{uv}^l = \sigma(x_u^l \Theta_{bl}^T x_v^l + b_{bl}) \tag{4}$$

$$\mathcal{L}_{\mathcal{CR}} = \arg\max_n \frac{e^{\beta_k(x_{uv}^l, n)}}{\sum_n e^{\beta_k(x_{uv}^l, n)}} \tag{5}$$

where Θ_{bl}, b_{bl} and $\mathcal{L}_{\mathcal{CR}}$ are the trainable weights and biases and connection reconstruction loss, respectively. The function σ is an activation function that maps the input to a value between 0 and 1. The output representation from the initial bilinear transformation layer (Θ_{bl} and b_{bl}) is passed to another linear transformation layer followed by a softmax activation. The term β_k represents a feed-forward neural network with k layers, which generates a logit vector of length n (as mentioned previously, n is the number of edges between two atoms) with the help of a softmax function. We denote the set of trainable parameters for this decoder as Θ_d for future reference.

SSL framework. Here, the SSL framework uses correlation between the actual input graph and its augmented versions to learn robust and generalized representation from the unlabeled data [28]. It provides an additional boost in terms of performance. We employ two types of augmentation techniques such as atom masking and edge masking. Atom masking randomly masks 10% of the atoms in the crystal, while edge masking randomly masks 10% of the edge features between adjacent atoms.

$$\mathcal{X}_G = \text{READOUT}(x_u^l) \tag{6}$$

Algorithm 1 Training procedure and chemical vector extraction

Input : $D = \{G_i = (V_i, E_i, \chi_i, F_i)\}$ dataset used for creating the chemical atom vector
Output : Generalized dense atom vector representations (A)

```
begin
    Initialize $\Theta_e$, $\Theta_d$ ▷ Parameters of the encoder and decoder
    $\mathcal{L}_{\mathcal{FR}}, \mathcal{L}_{\mathcal{CR}}, \mathcal{L}_{\mathcal{BT}}, A, S, N, G_j, D$ ▷ Feature reconstruction loss (Equation 3), Connection reconstruction loss (Equation 5), Barlow Twins loss (Equation 7), Set of atom vectors, Set of common atoms in one batch, Number of epochs, Batch graph (collection of 128 crystal structures), and Graph data
    for $i \leftarrow 1$ to $N$ do
        for $G_j \in D$ do
            ▷ $j$-th batch with randomly selected 128 crystals
            $H_{emd} = f_e(G_j, \Theta_e)$ ▷ $H_{emd}$: hidden representation of $G_j$
            $A_{Femd}, Adj_e = f_d(H_{emd}, \Theta_d)$ ▷ $AF_{emd}$: output of the decoder, $Adj_e$: constructed adjacency matrix
            $G_1, G_2 \leftarrow G_j$ ▷ $G_1, G_2$: augmented representations of $G_j$
            $\mathcal{X}^{G_1}, \mathcal{X}^{G_2} = f_e(G_1, G_2)$ ▷ $\mathcal{X}^{G_1}, \mathcal{X}^{G_2}$: augmented embeddings of $G_1, G_2$
            $\mathcal{L}_j = \alpha\mathcal{L}_{\mathcal{FR}} + \beta\mathcal{L}_{\mathcal{CR}} + \gamma\mathcal{L}_{\mathcal{BT}}$
            $\Theta = \Theta_e \oplus \Theta_d$
            $\Theta = \Theta - \alpha\nabla\mathcal{L}$
            $A_j \leftarrow f_{Unbatch}(H_{emd})$ ▷ $f_{Unbatch}$ generates atom representations
            $\forall\, s \in S, S_j^s = 0, count_s = 0$ ▷ $S_j^s$: cumulative representation of atom $s$ in $A_j$
            $\forall\, A_j^s \in A_j, count_s ++, S_j^s = S_j^s + A_j^s$
            $\forall\, s \in S, A_j^s = \frac{S_j^s}{count_s}$ ▷ Final representation of atom $s$ for batch $j$
            if $\mathcal{L}_j \leq \mathcal{L}_{j-1}$ then
                $A_j = A_j^s$
            else
                $A_j = A_{j-1}$
    return $A$
```

where READOUT interpretes to a global pooling function. Augmented graph representations are generated from the same crystalline structures. These augmented representations are used towards identity matrix formation. Given the problem formulation, we identified Barlow Twins [31] (BT) loss as a suitable loss function to reconstruct the graph representation of the crystals. Hereby, we utilize Barlow Twins loss function, which is based on the redundancy reduction principle by H. Barlow [32, 33]. We apply this loss function to the cross-correlation matrix that is formed from the embeddings produced by the encoder module of the auto-encoder.

$$\mathcal{L}_{\mathcal{BT}} \triangleq \underbrace{\sum_i (1 - \mathcal{M}_{ii}^2)}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} \mathcal{M}_{ij}^2}_{\text{redundancy reduction term}} \tag{7}$$

$$\mathcal{M}_{ij} \triangleq \frac{\sum_b \mathcal{X}_{b,i}^{G_1} \mathcal{X}_{b,j}^{G_2}}{\sqrt{(\mathcal{X}_{b,i}^{G_1})^2}\sqrt{(\mathcal{X}_{b,j}^{G_2})^2}} \tag{8}$$

Equation 7 describes the Barlow Twins loss function, which is used in our SSL block. It considers the cross-correlation matrix $\mathcal{M}$ of embeddings from two augmented instances, which is computed by Equation 8. The parameter λ (set to 0.0051 in the original paper [31]) is a positive constant that balances the first and second terms of the loss function. In this study, we apply Equations 1 and 6, to obtain the two augmented embeddings, namely $\mathcal{X}^{G_1}$ and $\mathcal{X}^{G_2}$. Furthermore, we utilize Equation 8 to derive the cross-correlation matrix, wherein b represents the batch index and i, j indicate the vector dimension of the projected output. Overall, the deep auto-encoder architecture is trained in an end-to-end fashion to optimize the loss function ($\mathcal{L}_{train}$) as shown in Equation 9.

$$\mathcal{L}_{train} = \underbrace{\alpha\mathcal{L}_{\mathcal{FR}} + \beta\mathcal{L}_{\mathcal{CR}}}_{\text{loss for UL}} + \underbrace{\gamma\mathcal{L}_{\mathcal{BT}}}_{\text{loss for SSL}} \tag{9}$$

$\mathcal{L}_{\mathcal{FR}}$, $\mathcal{L}_{\mathcal{CR}}$ are the reconstruction losses for node features and local connectivity, respectively. $\mathcal{L}_{\mathcal{BT}}$ is the Barlow Twins loss and α, β, γ are the weighting coefficients for each loss. We consider α = 0.25, β = 0.25 and γ = 0.5, considering a convex combination. The sequential steps for the training process and extraction of dense vector representation are stated in Algorithm 1. The number of parameters used by CrysAtom is 5.5 MB and the running time of each epoch of the training (using Algorithm 1) is approximately 30 minutes.

Atom Feature Vector Extraction. The atom vector extraction strategy in Algorithm 1 focuses on deriving atom feature vectors from the latent feature representations generated by the encoder module during each training epoch. For a batch graph G_j, the hidden representation H_{emd} is obtained using the encoder function f_e:

$$H_{emd} = f_e(G_j, \Theta_e)$$

Subsequently, the function $f_{Unbatch}$ is applied to H_{emd} to generate individual atom representation A_j.

$$A_j = f_{Unbatch}(H_{emd})$$

Here, $f_{Unbatch}$ maps the latent feature vectors H_{emd} to their corresponding atom feature vectors. If H_{emd} is a hidden representation of size $n \times d$, where n is the number of nodes (atoms) and d is the dimensionality of the feature vector, $f_{Unbatch}$ effectively extracts the important features to generate the distributed vector representation for each atom.

For each atom type s in the set of common atoms S, the proposed algorithm computes a cumulative representation S_j^s and calculates $count_s$ as,

$$S_j^s = \sum_{i=1}^{n} \nVdash_{\{A_j^i=s\}} \cdot A_j^i, \quad count_s = \sum_{i=1}^{n} \nVdash_{\{A_j^i=s\}}$$

Here, $\nVdash_{\{A_j^i=s\}}$ is an indicator function that is 1 if the atom A_j^i is of type s and 0 otherwise. Lastly, the final representation for each atom type s, in the batch j, is computed by averaging the cumulative representations.

$$A_j^s = \frac{\sum_{i=1}^{n} \nVdash_{\{A_j^i=s\}} \cdot A_j^i}{\sum_{i=1}^{n} \nVdash_{\{A_j^i=s\}}}$$

In this way, our algorithm ensures that the atom vectors are consistently updated and refined through the training epochs, ultimately storing the generalized dense atom vector representations in A.

3.3 Downstream Property Prediction Task

The objective of this study is to integrate the proposed atom feature vectors into a SOTA property predictor model to enhance the performance of the downstream task. The following steps are used for applying the dense vector representations to the downstream property prediction tasks. We extract the atom vectors from our novel CrysAtom framework. Subsequently, we train SOTA property predictor model (P_ψ) by providing property-tagged training data $D_t = \{G_i, y_i\}$ as well as the generated feature vector as initial node feature representation. Here, we consider CGCNN [5], CrysXPP [7], ALIGNN [10], M3GNet [22], Matformer [11], PotNet [12], and coGN [34] as the baseline property predictors due to its SOTA performance in property prediction tasks. Additionally, we take a pre-trained model named CHGNet [23] as baseline property prediction model. Training setup, hyper-parameter selection (Table 6), and detailed descriptions of baseline property predictors, have been presented in Appendix B and F for brevity.

4 Results

In this section, we first describe the details of our dataset used in our experiments, and then we follow up with the research questions in context to the task of atom vector generation, and the analytical discussion towards addressing those research questions.

4.1 Datasets

We use 139K unlabeled crystal graphs from the Materials Project (MP) to obtain the required dense vector representation of atoms. For our downstream property prediction, as suggested by the Yan et. al [11], we consider the datasets MP 2018.6.1 [35], JARVIS-DFT 2021.8.12 [36], and MatBench [37] to investigate the chemical rationality of our proposed vector representations. MP 2018.6.1 contains 69,239 materials with four properties, formation energy, bandgap (OPT), bulk modulus (Kv) and shear modulus (Gv), whereas the JARVIS-DFT dataset contains 55,723 materials with seven properties such as formation energy, bandgap (OPT), total energy, ehull, bandgap (MBJ), bulk modulus (Kv) and shear modulus (Gv). MatBench [37] contains total 132,752 crystals for formation energy (e_form), bandgap (gap) respectively, and 10,987 crystals for bulk (log_kvrh) and shear modulus (log_gvrh). Here, all these properties in Materials Project, JARVIS-DFT, and MatBench datasets are based on DFT calculations of crystal. To investigate how our dense vector representation mitigates DFT errors, we take a small dataset OQMD-EXP [38] containing 1,500 materials, consisting of experimental data for formation energy. Details of each of these datasets have been provided in Appendix C.

4.2 Research Questions

We pose a few important research questions (RQs), which are central to our research work.

RQ-1: Effectiveness of CrysAtom for downstream property prediction task. How effectively does CrysAtom aid the existing neural property predictors to achieve SOTA performance? Furthermore, to what extent does CrysAtom improve the performance of the distilled, pre-trained, and fine-tuned versions of the neural property predictors?

RQ-2: Robust dense vector representation. How does our dense vector representation CrysAtom fare when compared to the popular vector representations for atoms, such as, Atom2Vec or SkipAtom?

RQ-3: Removing DFT bias. How effectively does our proposed crystal atom vector representation mitigate DFT error bias, leading to SOTA results in existing neural property predictors?

RQ-4: Preserving periodic properties. How well does CrysAtom capture the periodic properties and the chemical significance of the atom?

4.3 Discussions

Property	Unit	CGCNN	CGCNN (CrysAtom)	CrysXPP	CrysXPP (CrysAtom)	ALIGNN	ALIGNN (CrysAtom)	M3GNet	M3GNet (CrysAtom)	Matformer	Matformer (CrysAtom)	PotNet	PotNet (CrysAtom)
Formation Energy	eV/atom	0.039	**0.028**	0.041	**0.030**	0.026	**0.023**	0.024	**0.022**	0.021	**0.019**	0.019	**0.018**
Bandgap (OPT)	eV	0.388	**0.270**	0.347	**0.262**	0.271	**0.252**	0.280	**0.263**	0.211	**0.206**	0.204	**0.193**
Bulk Modulus (Kv)	log(GPa)	0.054	**0.050**	0.080	**0.048**	0.051	**0.043**	0.050	**0.045**	0.043	**0.040**	0.040	**0.038**
Shear Modulus (Gv)	log(GPa)	0.087	**0.082**	0.105	**0.082**	0.078	**0.072**	0.087	**0.076**	0.073	**0.071**	0.065	**0.064**
Formation Energy	eV/atom	0.063	**0.040**	0.062	**0.041**	0.033	**0.031**	0.039	**0.031**	0.033	**0.030**	0.029	**0.028**
Bandgap (OPT)	eV	0.200	**0.143**	0.190	**0.142**	0.142	**0.133**	0.145	**0.133**	0.137	**0.135**	0.127	**0.120**
Total Energy	eV/atom	0.078	**0.043**	0.072	**0.044**	0.037	**0.035**	0.041	**0.035**	0.035	**0.031**	0.032	**0.029**
Ehull	eV	0.170	**0.124**	0.139	**0.121**	0.076	**0.066**	0.095	**0.068**	0.064	**0.057**	0.055	**0.049**
Bandgap (MBJ)	eV	0.410	**0.333**	0.378	**0.348**	0.310	**0.280**	0.360	**0.294**	0.300	**0.290**	0.270	**0.240**
Bulk Modulus (Kv)	GPa	14.47	**12.37**	13.61	**13.10**	10.40	**10.19**	12.40	**11.24**	11.21	**10.85**	10.11	**9.98**
Shear Modulus (Gv)	GPa	11.75	**10.45**	11.20	**10.44**	9.86	**9.39**	10.95	**10.29**	10.76	**9.85**	9.23	**9.13**

Table 1: Summary of the results (MAE) of different properties in Materials Project (top) and JARVIS-DFT (bottom). Model M is the vanilla variant of a SOTA model and M(CrysAtom) is a variant of the SOTA model with CrysAtom dense vectors as input. The best performance has been highlighted in **bold**.

	e_form	gap
Methods	MAE	MAE
coGN	0.0170	0.1559
coGN (CrysAtom)	0.0164 (-3.52)	**0.1530** (-1.86)

(a)

	log_kvrh	log_gvrh
Methods	MAE	MAE
coNGN	0.0491	0.0670
coNGN (CrysAtom)	**0.0485** (-1.22)	**0.0664** (-0.89)

(b)

Property	CHGNet (CrysAtom)	CHGNet
Ehull	**0.376** (-12.8)	0.431
Bandgap (MBJ)	**0.612** (-14.6)	0.717
Bulk Modulus (Kv)	**24.85** (-10.34)	27.72
Shear Modulus (Gv)	**16.31** (-7.5)	17.64

(c)

Table 2: Comparison of prediction performance (MAE) between CrysAtom and vanilla versions of coGN/-coNGN on MatBench properties: (a) e_form and gap, and (b) log_kvrh and log_gvrh. (c) Comparison of MAE for JARVIS-DFT properties between CrysAtom and vanilla CHGNet. Best results are shown in **bold**, with percentage decrease in MAE for CrysAtom relative to the vanilla model indicated in brackets.

Downstream Task Analysis. In relation to RQ-1, we compare five different SOTA frameworks for crystal property prediction such as CGCNN [5], CrysXPP [7], ALIGNN [10], Matformer [11], M3GNet [22], CHGNet [23], coGN/coNGN [34], and PotNet [12]. To train these methods for property prediction, we use the 200-dimensional dense vectors obtained for each atom using Algorithm 1. These vectors serve as input atom features, which are initialized as non-trainable node features. For each property, we trained on 80%, validated on 10% and evaluated on 10% of the data. In Table 1, we report mean absolute error (MAE) score (lower the MAE, higher the improvement) for the property prediction task. We observe that the SOTA models, when trained using CrysAtom generated vector representation as input, outperform their counterparts on the Materials Project and JARVIS-DFT datasets. Specifically, the average improvements of vanilla SOTA models such as CGCNN, CrysXPP, ALIGNN, Matformer, and PotNet are 21.92%, 23.40%, 8.63%, 6.49%, and 5.21%, respectively. These improvements are significant, considering the overall architecture of these property predictor models remain unchanged, while in the input space, we introduce CrysAtom generated feature vectors to train these models on various downstream tasks. Additionally, it is to be noted that the average relative improvement across all properties for ALIGNN (8.63%), Matformer (6.49%) and PotNet (5.21%) is lesser compared to CGCNN (21.92%) and CrysXPP (23.40%). The likely reason is that ALIGNN, Matformer, and PotNet are more complex models with higher parameter counts (97.8 MB for ALIGNN and 68.42 MB for Matformer) compared to CGCNN and CrysXPP. ALIGNN learns three-body interactions, Matformer captures periodic invariance, and PotNet incorporates interatomic potentials. Coversely, CGCNN and CrysXPP use simpler encoder architectures, primarily applying GCN layers to multi-graph crystal structures. Complex models often learn intrinsic features of crystal structures, so introducing atom vector representation alone doesn't significantly enhance performance. In contrast, simpler models benefit more from dense features as inputs for downstream tasks. Another interesting observation from Table 1 is that all SOTA models achieve an average improvement of 17.03% in formation energy prediction, when trained using CrysAtom generated atom vector representation. This improvement is likely because the formation energy of a crystal, defined as the difference between the energy of a unit cell composed of N chemical species and the sum of their chemical potentials (with units of $eV/atom$), depends on its node features [16]. In contrast to that, However, the improvement is suboptimal for mechanical properties like bulk (9.99%) and shear modulus (7.18%), as these depend more on structural information such as lattice structure and symmetry [39] than on chemical properties. In Table 1, we report the performance of M3GNet [22] trained on CrysAtom generated vectors, showing an average improvement of 12.17% over vanilla M3GNet. Similarly, in Tables 2a and 2b, we present MAE results on the MatBench dataset [37], where coGN/coNGN [34], trained on CrysAtom vectors, outperforms their vanilla versions. Additionally, in Table 2c, CHGNet [23] trained with CrysAtom vectors also surpasses

Property	CGCNN (CrysAtom)	Distilled CGCNN	Fine-tuned CrysGNN	CrysXPP	Pre-trained GNN
Formation Energy	**0.040** (-14.9)	0.047	0.056	0.062	0.764
Bandgap (OPT)	**0.143** (-10.6)	0.160	0.183	0.190	0.688
Total Energy	**0.043** (-18.9)	0.053	0.069	0.072	1.451
Ehull	0.124 (+2.4)	**0.121**	0.130	0.139	1.112
Bandgap (MBJ)	**0.333** (-2.1)	0.340	0.371	0.378	1.493
Bulk Modulus (Kv)	12.37 (+0.5)	**12.31**	13.42	13.61	20.34
Shear Modulus (Gv)	**10.45** (-3.9)	10.87	11.07	11.20	16.51

(a)

Property	CGCNN (CrysAtom)	CGCNN (SkipAtom)	CGCNN (Atom2Vec)	CGCNN (Random)
Formation Energy	**0.040** (-34.4)	0.061	0.070	0.075
Bandgap (OPT)	**0.143** (-27.0)	0.196	0.251	0.263
Total Energy	**0.043** (-38.6)	0.070	0.076	0.089
Ehull	**0.124** (-18.9)	0.153	0.160	0.164
Bandgap (MBJ)	**0.333** (-20.7)	0.420	0.529	0.569
Bulk Modulus (Kv)	**12.37** (-13.9)	14.36	15.41	15.99
Shear Modulus (Gv)	**10.45** (-9.8)	11.58	12.09	13.52

(b)

Table 3: (a) Comparison of prediction performance (MAE) for the seven properties in JARVIS-DFT between CrysAtom version of the CGCNN, Distilled CGCNN and other SOTA pre-trained/fine-tuned models. (b) Comparison of prediction performance (MAE) for the seven properties in JARVIS-DFT between variants of CGCNN with different dense vector representations, namely, CrysAtom, SkipAtom, Atom2Vec and Random. Both for (a) and (b), the best results have been shown in **bold** and the second best results have been underlined. Both for (a) and (b), percentage of decrease in MAE for CrysAtom, with respect to the best performing model from the rest, has been mentioned within the bracket.

vanilla CHGNet in the property prediction task on the JARVIS-DFT dataset.

Comparison with Existing Distilled and Pre-trained Models. To address the second part of RQ-1, we investigate the efficacy of utilizing a fixed atom feature vector representation in input space rather than applying resource intensive approaches such as knowledge distillation and task specific fine-tuning. As the encoder module of CGCNN, CrysGNN [16] and CrysXPP [7] are variants of GCN, we have shown performance comparison between these frameworks (as shown in Table 3a), where we have used CrysAtom generated dense vector representation only for CGCNN. Additionally, we consider pre-trained GNN [40], which is widely used for pre-training property predictor models. We pre-train GNN [16, 40] on 800K untagged crystal data and fine-tune it on seven different properties, as shown in Table 3a. For fine-tuned CrysGNN framework, we consider the pre-trained encoder of CrysGNN followed by a feed-forward neural network to predict a specific property. Similarly, for distilled CGCNN framework, we apply knowledge distillation using pre-trained CrysGNN model [16]. We observe that CGCNN (CrysAtom) outperforms fine-tuned CrysGNN, CrysXPP and Pre-trained GNN with a significant margin over all properties. We also notice that CGCNN (CrysAtom) outperforms distilled CGCNN by a large margin for formation energy, bandgap (OPT), total energy, bandgap (MBJ) and shear modulus. However, it produces comparable results for ehull and bulk modulus (Kv). The reason behind it could be use of a relatively small dataset of 139K untagged crystals to generate a fixed vector representation of atoms using CrysAtom model, whereas the pre-training of CrysGNN is done on a large dataset of size 800K. Descriptions of Distilled CGCNN, Fine-tuned CrysGNN, and Pre-trained GNN methods are provided in Appendix F (Table 9).

Comparison with Existing Dense Representations. In RQ-2, we compare the performance of our dense vector representation, generated by CrysAtom, against the existing dense representations of atoms. Here, we have considered CGCNN as the encoder module to conduct necessary experiments. We train CGCNN using our 200-dimensional atom vector representations, including Random[3], Atom2Vec and SkipAtom on seven properties from the JARVIS-DFT dataset and reported MAE values in Table 3b. Our results show that CGCNN, when combined with CrysAtom, significantly outperforms SkipAtom, Atom2Vec, and Random aided versions across all the properties. The Random version performs the worst, as its dense representations fail to capture essential chemical features. The Atom2Vec version also underperforms, due to its inability to capture the topological complexity of crystal materials. The SkipAtom version, which captures atomic context better than SVD-based methods, performs slightly better than Atom2Vec but still falls short due to its limitations in comprehending complex structures.

Experiment Settings	CGCNN	CGCNN (CrysAtom)	CrysXPP	CrysXPP (CrysAtom)	ALIGNN	ALIGNN (CrysAtom)	Matformer	Matformer (CrysAtom)	PotNet	PotNet (CrysAtom)
Train on DFT **Test on Experimental**	0.265	0.241 (-9.06)	0.243	0.222 (-8.6)	0.220	0.212 (-3.6)	0.218	0.213 (-2.3)	0.217	0.211 (-2.8)
Train on DFT and 20 % Experimental **Test on 80 % Experimental**	0.144	0.111 (-22.9)	0.138	0.115 (-16.7)	0.099	0.093 (-6.1)	0.098	0.094 (-4.1)	0.097	0.093 (-4.1)
Train on DFT and 80 % Experimental **Test on 20 % Experimental**	0.094	0.072 (-23.4)	0.087	0.071 (-18.4)	0.073	0.068 (-6.8)	0.072	0.069 (-4.2)	0.070	0.067 (-4.3)

Table 4: MAE of predicting experimental values by different SOTA models and their CrysAtom versions with full DFT data and different percentages of experimental data for formation energy in OQMD-EXP dataset. Percentage of decrease in MAE for CrysAtom is mentioned in bracket.

Removal of DFT Error Bias using CrysAtom. In this section we discuss RQ-3 to understand how

[3] We randomly select vector representations from $\mathbb{R}^n$. We use normal distribution for randomly drawing the values to populate the initial vector representation.

we can remove DFT error bias using experimental data with the help of dense vector representations of atoms obtained from CrysAtom. One of the fundamental issues in material science is that the experimental data instances, as described in Section 4.1, for crystal properties are scarce [7]. Hence, existing SOTA models highly rely on DFT calculated data to train its parameters. However, mathematical approximations in DFT calculation lead to erroneous prediction (error bias) in contrast to the actual experimental data [16]. Hence, DFT error bias is a common problem present in the existing SOTA frameworks. Das et.al. [7] have shown that pre-training plays a significant role in mitigating error bias when fine-tuned with experimental data. Consequently, we investigate whether DFT error bias in SOTA models can be reduced with the help of our novel atom vector representation, using a small set of experimental data instances. Here, we consider OQMD-EXP [38] dataset to conduct the relevant experiments for formation energy prediction task. We train all SOTA models and their CrysAtom variants with the complete DFT data in addition to a part of the experimental data. We report the MAE of different SOTA models and its CrysAtom variants in Table 4, where the evaluation is performed on gold standard experimental data. From Table 4, we can conclude that DFT bias error in the SOTA models have reduced with the application of CrysAtom generated vectors.

Preserving Periodic Properties of Elements using CrysAtom. In this section, we focus on RQ-4 to determine if the proposed dense representations embed equivalent atomic information. To understand the chemical significance of these vector representations, we visualize them using a lower-dimensional projection. The illustration in Figure 3, which is present in Appendix, significantly aids in qualitatively examining the chemical properties of the vector representations in relation to the periodic table. We produce 200-dimensional dense representations only for 89 atoms as shown in Figure 3. Remaining 29 atoms are rare elements and the more details is given in Appendix C.1. Our illustration shows that group-I alkali metals (Li, Na, K, Rb, Cs) form a single cluster, indicating our vector representation captures their similarity in terms of chemical properties, such as high reactivity and single valence electron [41–44]. Similarly, group-II alkaline earth metals (Ca, Sr, Mg, Ba) cluster together, indicating their lesser reactivity and electrical conductivity. Additionally, our representation, though not illustrating Be's alkaline properties, captures its diagonal relationship with Al, indicating their tendency to form covalent bonds and tetrahedral structures [41–44]. All reactive non-metals (C, N, O, F, Cl, Br, I, P, Se, and S) form a single cluster. This cluster spans groups VIA (halogens) and VIIA (chalcogens) of the periodic table due to their shared properties like reactivity and anion formation. Within this cluster, O and P are close in the embedding space, both binding with Hydrogen to form water and phosphine, respectively. Additionally, P and C exhibit a diagonal relationship [42, 44]. The figure shows that metalloid elements like Silicon (Si), Germanium (Ge), Boron (B), Antimony (Sb), and Tellurium (Te) cluster together, indicating that our vector representation captures their high semiconductivity and semi-metallic nature. Additionally, Boron and Silicon's similarity, in terms of electronegativity, allows them to form covalent bonds, showcasing their diagonal relationship. Lead (Pb), Tin (Sn), and Thallium (Tl) also form a cluster, representing post-transition metals, with Pb and Sn in close proximity due to the inert pair effect. Our representation, groups all noble gases (Ne, Ar, Kr, Xe) and captures properties of transition metals (V, Cr, Fe, Co, Ni, Cu). Lanthanides and Actinides (e.g., La, Ce, Pr, Nd, Sm, Tb, Eu, Gd, Dy, Np, Ho, Pu, Pm, Pa, U, Th, Ac, Yb, Lu, Er) cluster due to their radioactive nature. Notably, Pb and Sn also exhibit diagonal relationship properties [41–44]. Also, we visualize the other two principal components namely third and fourth principal components, which is present in Appendix H.

Ablation study. We analyze the effect of increasing dimensions (50, 100, 200) of the CrysAtom vector in Appendix I.1, the combination of UL and SSL approaches in Appendix I.2, the efficiency of the CrysAtom variant in combination with SOTA models in Appendix I.3, and the impact of increasing dataset size which helps CrysAtom to generate dense vectors in Appendix I.4.

5 Conclusion

In this study, we introduce a novel framework, CrysAtom, designed to create dense vector representations for crystal atoms. These vectors play a pivotal role for different graph neural network-based crystal property prediction tasks. Our approach uniquely combines UL and SSL techniques to generate these dense representations. Additionally, we propose a novel way to extract the generalized feature vector representation from the latent space of the encoder module of CrysAtom framework. Our empirical results demonstrate that CrysAtom significantly enhances the performance of existing neural property predictors. Experiments show that our proposed framework generates a robust and unbiased dense vector representation for atoms, effectively capturing periodic properties and chemical significance of atoms. Future directions for extending our work could be incorporating many-body interactions as a part of the system, aiming to achieve performance improvements across complex state-of-the-art models.

Acknowledgement

Shrimon Mukherjee and Madhusudan Ghosh are thankful to Indian Association for the Cultivation of Science for funding their research. We also thank Prof. Ayan Datta, Prof. Abhishek Dey, and Prof. T.K. Paine for their helpful insights. We also extend our gratitude to Kishalay Das (Corresponding author of CrysXPP [7], CrysGNN [16], and CrysMMNet [45]) for providing clarifications on various aspects of the CrysGNN code and for sharing the OQMD-EXP dataset.

References

[1] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. 1

[2] Nongnuch Artrith, Keith T Butler, François-Xavier Coudert, Seungwu Han, Olexandr Isayev, Anubhav Jain, and Aron Walsh. Best practices in machine learning for chemistry. *Nature chemistry*, 13(6):505–508, 2021.

[3] Kamal Choudhary, Brian DeCost, Chi Chen, Anubhav Jain, Francesca Tavazza, Ryan Cohn, Cheol Woo Park, Alok Choudhary, Ankit Agrawal, Simon JL Billinge, et al. Recent advances and applications of deep learning methods in materials science. *npj Computational Materials*, 8 (1):59, 2022.

[4] Shrimon Mukherjee, Madhusudan Ghosh, and Partha Basuchowdhuri. Deepglstm: deep graph convolutional network and lstm based approach for predicting drug-target binding affinity. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pages 729–737. SIAM, 2022. 1

[5] Tian Xie and Jeffrey C Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical review letters*, 120(14):145301, 2018. 1, 2, 3, 4, 6, 7, 13, 15, 16

[6] Varadarajan Rengaraj, Sebastian Jost, Franz Bethke, Christian Plessl, Hossein Mirhosseini, Andrea Walther, and Thomas D Kühne. A two-step machine learning method for predicting the formation energy of ternary compounds. *Computation*, 11(5):95, 2023. 1

[7] Kishalay Das, Bidisha Samanta, Pawan Goyal, Seung-Cheol Lee, Satadeep Bhattacharjee, and Niloy Ganguly. Crysxpp: An explainable property predictor for crystalline materials. *npj Computational Materials*, 8(1):43, 2022. 1, 2, 3, 6, 7, 8, 9, 10, 16

[8] Daniel W Davies, Keith T Butler, Adam J Jackson, Andrew Morris, Jarvist M Frost, Jonathan M Skelton, and Aron Walsh. Computational screening of all stoichiometric inorganic materials. *Chem*, 1(4):617–627, 2016. 1

[9] Luis M Antunes, Ricardo Grau-Crespo, and Keith T Butler. Distributed representations of atoms and materials for machine learning. *npj Computational Materials*, 8(1):44, 2022. 1, 15, 18

[10] Kamal Choudhary and Brian DeCost. Atomistic line graph neural network for improved materials property predictions. *npj Computational Materials*, 7(1):185, 2021. 1, 2, 6, 7, 16

[11] Keqiang Yan, Yi Liu, Yuchao Lin, and Shuiwang Ji. Periodic graph transformers for crystal material property prediction. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 2, 6, 7, 16

[12] Yuchao Lin, Keqiang Yan, Youzhi Luo, Yi Liu, Xiaoning Qian, and Shuiwang Ji. Efficient approximations of complete interatomic potentials for crystal property prediction. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 21260–21287. PMLR, 23–29 Jul 2023. 1, 2, 6, 7, 16

[13] Quan Zhou, Peizhe Tang, Shenxiu Liu, Jinbo Pan, Qimin Yan, and Shou-Cheng Zhang. Learning atoms for materials discovery. *Proceedings of the National Academy of Sciences*, 115(28): E6411–E6417, 2018. 1, 15

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 1

[15] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 1, 15

[16] Kishalay Das, Bidisha Samanta, Pawan Goyal, Seung-Cheol Lee, Satadeep Bhattacharjee, and Niloy Ganguly. Crysgnn: Distilling pre-trained knowledge to enhance property prediction for crystalline materials. *arXiv preprint arXiv:2301.05852*, 2023. 2, 3, 7, 8, 9, 10, 15

[17] Steph-Yves Louis, Yong Zhao, Alireza Nasiri, Xiran Wang, Yuqi Song, Fei Liu, and Jianjun Hu. Graph convolutional neural networks with global attention for improved materials property prediction. *Physical Chemistry Chemical Physics*, 22(32):18141–18148, 2020. 2

[18] Anthony Yu-Tung Wang, Steven K Kauwe, Ryan J Murdock, and Taylor D Sparks. Compositionally restricted attention-based network for materials property predictions. *Npj Computational Materials*, 7(1):77, 2021. 2

[19] Chi Chen, Weike Ye, Yunxing Zuo, Chen Zheng, and Shyue Ping Ong. Graph networks as a universal machine learning framework for molecules and crystals. *Chemistry of Materials*, 31 (9):3564–3572, 2019. 2

[20] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[21] Zhantao Chen, Nina Andrejevic, Tess Smidt, Zhiwei Ding, Qian Xu, Yen-Ting Chi, Quynh T Nguyen, Ahmet Alatas, Jing Kong, and Mingda Li. Direct prediction of phonon density of states with euclidean neural networks. *Advanced Science*, 8(12):2004214, 2021. 2

[22] Chi Chen and Shyue Ping Ong. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, 2022. 2, 6, 7, 16

[23] Bowen Deng, Peichen Zhong, and KyuJung Jun. Chgnet as a pretrained universal neural network potential for charge-informed atomistic. *Nature*, 2023. 2, 6, 7, 16

[24] Han Yang, Chenxi Hu, Yichi Zhou, Xixian Liu, Yu Shi, Jielan Li, Guanzhi Li, Zekun Chen, Shuizhou Chen, Claudio Zeni, et al. Mattersim: A deep learning atomistic model across elements, temperatures and pressures. *arXiv preprint arXiv:2405.04967*, 2024. 2

[25] Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M Elena, Dávid P Kovács, Janosh Riebesell, Xavier R Advincula, Mark Asta, William J Baldwin, Noam Bernstein, et al. A foundation model for atomistic materials chemistry. *arXiv preprint arXiv:2401.00096*, 2023.

[26] Duo Zhang, Xinzijian Liu, Xiangyu Zhang, Chengqian Zhang, Chun Cai, Hangrui Bi, Yiming Du, Xuejian Qin, Jiameng Huang, Bowen Li, et al. Dpa-2: Towards a universal large atomic model for molecular and material simulation. *arXiv preprint arXiv:2312.15492*, 2023. 2

[27] Hannes Stärk, Dominique Beaini, Gabriele Corso, Prudencio Tossou, Christian Dallago, Stephan Günnemann, and Pietro Liò. 3d infomax improves gnns for molecular property prediction. In *International Conference on Machine Learning*, pages 20479–20502. PMLR, 2022. 2

[28] Rishikesh Magar, Yuyang Wang, and Amir Barati Farimani. Crystal twins: self-supervised learning for crystalline material property prediction. *npj Computational Materials*, 8(1):231, 2022. 2, 4

[29] Nima Shoghi, Adeesh Kolluru, John R Kitchin, Zachary W Ulissi, C Lawrence Zitnick, and Brandon M Wood. From molecules to materials: Pre-training large generalizable models for atomic property prediction. *arXiv preprint arXiv:2310.16802*, 2023. 2

[30] Vishu Gupta, Kamal Choudhary, Brian DeCost, Francesca Tavazza, Carelyn Campbell, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. Structure-aware graph neural network based deep transfer learning framework for enhanced predictive analytics on diverse materials datasets. *npj Computational Materials*, 10(1):1, 2024. 2

[31] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 5, 18

[32] Horace B Barlow et al. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01):217–233, 1961. 5

[33] Horace Barlow. Redundancy reduction revisited. *Network: computation in neural systems*, 12 (3):241, 2001. 5

[34] Robin Ruff, Patrick Reiser, Jan Stühmer, and Pascal Friederich. Connectivity optimized nested graph networks for crystal structures. *arXiv preprint arXiv:2302.14102*, 2023. 6, 7, 16

[35] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1):011002, 2013. 6, 14, 15

[36] Kamal Choudhary, Kevin F Garrity, Andrew CE Reid, Brian DeCost, Adam J Biacchi, Angela R Hight Walker, Zachary Trautt, Jason Hattrick-Simpers, A Gilad Kusne, Andrea Centrone, et al. The joint automated repository for various integrated simulations (jarvis) for data-driven materials design. *npj computational materials*, 6(1):173, 2020. 6, 15

[37] Alexander Dunn, Qi Wang, Alex Ganose, Daniel Dopp, and Anubhav Jain. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Computational Materials*, 6(1):138, 2020. 6, 7, 15

[38] Scott Kirklin, James E Saal, Bryce Meredig, Alex Thompson, Jeff W Doak, Muratahan Aykol, Stephan Rühl, and Chris Wolverton. The open quantum materials database (oqmd): assessing the accuracy of dft formation energies. *npj Computational Materials*, 1(1):1–15, 2015. 6, 9

[39] Peter T Bobrowsky and Brian Marker. *Encyclopedia of engineering geology*. Springer Berlin, 2018. 7

[40] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020. 8, 15

[41] James E Huheey, Ellen A Keiter, Richard L Keiter, and Okhil K Medhi. *Inorganic chemistry: principles of structure and reactivity*. Pearson Education India, 2006. 9

[42] Gary L Miessler. *Inorganic chemistry*. Pearson Education India, 2008. 9

[43] John Emsley. *Nature's building blocks: an AZ guide to the elements*. Oxford University Press, USA, 2011.

[44] Eric Scerri. *The periodic table: its story and its significance*. Oxford University Press, 2019. 9

[45] Kishalay Das, Pawan Goyal, Seung-Cheol Lee, Satadeep Bhattacharjee, and Niloy Ganguly. Crysmmnet: multimodal representation for crystal property prediction. In *Uncertainty in Artificial Intelligence*, pages 507–517. PMLR, 2023. 10

[46] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 13

[47] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 13

[48] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019. 13

[49] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 17

[50] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 18

[51] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 18

A Notations

Notation	Terms
u	Source node
v	Target node
V_i	Set of atoms present in the unit cell
n_{uv}	No. of edges between u and v
E_i	A multiset of node pairs
χ_i	Node features
F_i	Collection of edge weights
r	Radius
$\mathcal{M}$	Cross-correlation matrix
READOUT	Global pooling function
Random	Vector representations drawn from $\mathbb{R}^n$ using normal distribution
$\mathcal{L}_{\mathcal{BT}}$	Barlow Twins Loss
$\mathcal{L}_{\mathcal{FR}}$	Feature reconstruction loss
$\mathcal{L}_{\mathcal{CR}}$	Connection reconstruction loss
MAE	Mean Absolute Error

Table 5: Important notations used in the paper.

B Training Setup / Hyper-parameter Details / Computational Resources used.

Training Setup/ Hyper-parameter details/ Computational Resources	**Value**
r	8
Convolution Layers	Five convolution layers [5]
Epochs	110
Optimizer	Adam [46]
Learning Rate	0.03
Embedding Dimension	200 (can be 50 and 100-dimensional dense vectors)
Batch Size	128
Weightage for α	0.25
Weightage for β	0.25
Weightage for γ	0.50 (for convex sum)
Optimizer	Adam [46]
Epochs	1000
Learning Rate	Default learning rates used in the vanilla versions
Batch Size	64
Seed Value	123
Train, Valid, Test Splits	80%, 10%, 10%
Implementation Framework	PyTorch
Computational Resources	one NVIDIA A6000 48GB GPU and one NVIDIA A100 80GB GPU

Table 6: Summary of hyper-parameter details in CrysAtom (top), CrysAtom variant of all SOTA models (middle) and computational resources (bottom) used in this work.

In this work, we employ five convolution layers [5] of the encoder model to train our framework CrysAtom. Then we train it for 110 epochs using Adam [46] for optimization with a learning rate of 0.03. We keep the embedding dimension for each node as 200 (it can be 50 and 100-dimensional dense vectors please set the embedding dimension accordingly), the batch size of 128, and equal weightage (0.25) for α, β and weightage (0.50) for γ (for convex sum) of Equation 9 (In the main manuscript). In the downstream property prediction tasks we also use the Adam [46] optimizer with weight decay [47] of 1e-5 and one cycle learning rate scheduler [48] to train our vanilla property predictors. We utilize batch size of 64 and seed value of 123. We use mean squared error as the objective function to train and mean absolute error as the evaluation metric to validate and test. We utilize PyTorch framework for our implementation. We use one NVIDIA A6000 48GB GPU, and one NVIDIA A100 80GB GPU to generate 200 dimensional-dense representation of atoms and perform the downstream property prediction task using vanilla property predictors using our generated dense vector. Training setup and hyper-parameter details are stated in detailed in Table 6.

C Datasets

C.1 Dataset used for Vector Creation

Task	Datasets	Graph Num.	Structural Info.	Properties Count	Data Type
Dense Vector Creation	MP	139K	✓	×	DFT Calculated
Property Prediction	MP 2018.6.1	69K	✓	4	DFT Calculated
	JARVIS-DFT	55K	✓	7	DFT Calculated
	MatBench	132K	✓	4	DFT Calculated
	OQMD-EXP	1.5K	✓	1	Experimental

Table 7: Datasets used for both dense vector creation and downstream tasks.

In this work, we use 139,308 untagged inorganic compounds obtained from Materials Project Database [35] to propose the 200 dimensional dense representation of chemical atoms. The number of atom elements present in our dataset is 89. These atoms are Hydrogen (H), Helium (He), Lithium (Li), Beryllium (Be), Boron (B), Carbon (C), Nitrogen (N), Oxygen (O), Fluorine (F), Neon (Ne), Sodium (Na), Magnesium (Mg), Aluminum (Al), Silicon (Si), Phosphorus (P), Sulfur (S), Chlorine (Cl), Argon (Ar), Potassium (K), Calcium (Ca), Scandium (Sc), Titanium (Ti), Vanadium (V), Chromium (Cr), Manganese (Mg), Iron (Fe), Cobalt (Co), Nickel (Ni), Copper (Cu), Zinc (Zn), Gallium (Ga), Germanium (Ge), Arsenic (As), Selenium (Se), Bromine (Br), Krypton (Kr), Rubidium (Rb), Strontium (Sr), Yttrium (Y), Zirconium (Zr), Niobium (Nb), Molybdenum (Mo), Technetium (Tc), Ruthenium (Ru), Rhodium (Rh), Palladium (Pd), Silver (Ag), Cadmium (Cd), Indium (In), Tin (Sn), Antimony (Sb), Tellurium (Te), Iodine (I), Xenon (Xe), Cesium (Cs), Barium (Ba), Lanthanum (La), Cerium (Ce), Praseodymium (Pr), Neodymium (Nd), Promethium (Pm), Samarium (Sm), Europium (Eu), Gadolinium (Gd), Terbium (Tb), Dysprosium (Dy), Holmium (Ho), Erbium (Er), Thulium (Tm), Ytterbium (Yb), Lutetium (Lu), Hafnium (Hf), Tantalum (Ta), Tungsten (W), Rhenium (Re), Osmium (Os), Iridium (Ir), Platinum (Pt), Gold (Au), Mercury (Hg), Thallium (Tl), Lead (Pb), Bismuth (Bi), Actinium (Ac), Thorium (Th), Protactinium (Pa), Uranium (U), Neptunium (Np), Plutonium (Pu). We produce 200-dimensional feature vectors only for these 89 atoms. In this work, we have not employed the atoms for training our novel atom vector extractor framework such as Polonium (Po), Astatine (At), Radon (Rn), Francium (Fr), Radium (Ra), Actinium (Ac), Rutherfordium (Rf), Dubnium (Db), Seaborgium (Sg), Bohrium (Bh), Hassium (Hs), Meitnerium (Mt), Darmstadtium (Ds), Roentgenium (Rg), Copernicium (Cn), Nihonium (Nh), Flerovium (Fl), Moscovium (Mc), Livermorium (Lv), Tennessine (Ts), Oganesson (Og), Americium (Am), Curium (Cm), Berkelium (Bk), Californium (Cf), Einsteinium (Es), Fermium (Fm), Mendelevium (Md), Nobelium (No), Lawrencium (Lr). The main reason behind not including these atoms are due to the unavailability of these atoms in compounds form. Table 7 shows the dataset statistics used for both pretraining and the downstream tasks.

Property	Unit	Data-size
Formation Energy	eV/(atom)	69239
Bandgap (OPT)	eV	69239
Bulk Modulus (Kv)	log(GPa)	5450
Shear Modulus (Gv)	log(GPa)	5450
Formation Energy	eV/(atom)	55723
Bandgap (OPT)	eV	55723
Total_Energy	eV/(atom)	55723
Ehull	eV	55371
Bandgap (MBJ)	eV	18172
Bulk Modulus (Kv)	GPa	19680
Shear Modulus (Gv)	GPa	19680
e_form	eV/(atom)	132752
gap	eV	106113
log_kvrh	log(GPa)	10987
log_gvrh	log(GPa)	10987

Table 8: Summary of different crystal properties in Materials Project (top), JARVIS-DFT (middle) datasets, and MatBench (bottom) datasets.

C.2 Datasets Used for Downstream Property Prediction Task

Materials Project (MP): This database is publicly accessible and provides a repository of crystal structures and corresponding materials properties that have been calculated [35]. The dataset is made from the result obtained with density functional theory-based calculations. The dataset is composed of electronic structure, thermodynamics, mechanical, and dielectric properties. It also has a visual web-based interface[4]. Table 8 shows statistics of the Materials Project dataset.
JARVIS: JARVIS[5] (Joint Automated Repository for Various Integrated Simulations) [36] is a data repository that incorporates not only DFT-based calculations but also data from the classical force fields and machine learning techniques. The database is free and public. Table 8 shows statistics of the JARVIS dataset.
MatBench: MatBench[6] [37] is a an automated learderboard for benchmarking SOTA machine learning algorithms predicting a diverse range of solid materials properties. This repository is maintained by Materials Project [35]. The database is free and publicly available. Table 8 shows statistics of the MatBench dataset.

D Definitions

Diagonal Relationship Diagonal relationship refers to the observation that elements located at diagonally opposite corners of the periodic table possess similarities in physical and chemical properties.
Inert Pair Effect This phenomenon describes how certain elements tend to display oxidation states that are lower than what one would expect based on their location in the periodic table.

E Baseline Representations

Atom2Vec [13] Atom2Vec proposed a singular value decomposition (SVD) based framework to generate distributed feature representation of molecular atoms to address the problem of one-hot representation.
SkipAtom [9] SkipAtom proposed framework that utilizes skip-gram [15] technique to generate dense representation of atoms.

F Baseline Property Predictors

Methods	Approaches	Pre-trained Datasets
Fine-tuned CrysGNN	In this method, we take the pre-trained encoder of CrysGNN [16], which is pre-trained using CGCNN as an encoder module [5] by incorporating UL and SSL techniques.	Materials project & OQMD (800K)
Pre-trained GNN	Here, we consider the Pre-trained GNN, which is a popular pre-training algorithm for molecules proposed by Hu et al. [40]. This approach typically leverages techniques such as Attribute Masking and Supervised Attribute Prediction etc. In our work, we extend this by passing the multi-graph structure of the crystal into the Pre-trained GNN, experimenting with different combinations of node-level pre-training strategies alongside graph-level supervised pre-training.	Materials project & OQMD (800K)

Table 9: Detailed description, Fine-tuned CrysGNN, and Pre-trained GNN.

[4] https://materialsproject.org/
[5] https://jarvis.nist.gov/
[6] https://matbench.materialsproject.org/

To evaluate the effectiveness of our dense 200-dimensional vector we use CGCNN[7], CrysXPP[8], ALIGNN[9], M3GNet [22][10], Matformer[11], PotNet[12], coGN[13], and CHGNet[14] as our base property predictor.

CGCNN [5] This research develops a multi-graph representation for crystals derived from inorganic materials and establishes a supervised model based on graph convolution to predict a range of crystal properties. Here we directly use the publicly available code of CGCNN and we change the input feature dimension to 200 to feed our 200-dimensional dense vector representation of atoms.

CrysXPP [7] In this study, the researchers develop an autoencoder named CrysAE, training it on a vast collection of unlabelled crystal graphs. The insights gained are then leveraged to prime the encoder of CrysXPP, which undergoes further refinement using data tagged with specific properties. Additionally, they create a feature selector to aid in understanding the predictions made by the model. Here we directly use the publicly available code of CrysXPP and we change the input feature dimension to 200 to feed our 200-dimensional dense vector representation of atoms.

ALIGNN [10] This study introduces line graph neural networks as a novel approach to incorporate angular information into the convolution layer by alternating message passing between the bond graph and its bond-angle line graph. Here we directly use the publicly available code of ALIGNN and we change the input feature dimension to 200 to feed our 200-dimensional dense vector representation of atoms.

M3GNet [22] This work proposed an architecture which considers the universal interatomic potentials for crystal representation learning. Here we directly use the publicly available code of M3GNet and we change the input feature dimension to 200 to feed our 200-dimensional dense vector representation of atoms.

Matformer [11] This work proposes a periodic graph transformer called Matformer, which incorporates periodic invariance and periodic pattern encoding for crystal representation learning, achieving better performance than baseline methods on various tasks and highlighting the importance of both periodic invariance and periodic pattern encoding in crystal representation learning. Here we directly use the publicly available code of Matformer and we change the input feature dimension to 200 to feed our 200-dimensional dense vector representation of atoms.

PotNet [12] This work proposed an architecture PotNet which considers the interatomic potentials for crystal representation learning, achieving better performance than baseline methods on various task and highlighting the importance of interatomic potentials in crystal representation learning. Here we directly use the publicly available code for PotNet and we change the input feature dimension to 200 to feed our 200-dimensional dense vector representation of atoms.

coGN [34] This work proposed an architecture named coGN and coNGN, which incorporates connectivity optimized crystal graph network from message passing and line graph templates, which produces SOTA results on various tasks of crystals property prediction task[15]. Here we directly use the publicly available code for coGN/coNGN and we change the input feature dimension to 200 to feed our 200-dimensional dense vector representation of atoms.

CHGNet [23] This work proposed a pre-trained universal neural network potential for atomistic modelling of crystal materials. Here we directly use the publicly available code of CHGNet and the released pre-trained model and we change the input feature dimension to 200 to feed our 200-dimensional dense vector representation of atoms.

[7] https://github.com/txie-93/cgcnn.git

[8] https://github.com/kdmsit/crysxpp

[9] https://github.com/usnistgov/alignn.git

[10] https://github.com/materialsvirtuallab/m3gnet

[11] https://github.com/YKQ98/Matformer

[12] https://github.com/divelab/AIRS/tree/main/OpenMat/PotNet

[13] https://github.com/aimat-lab/gcnn_keras/tree/v3.0.1/kgcnn/literature/coGN

[14] https://github.com/CederGroupHub/chgnet/tree/main

[15] https://matbench.materialsproject.org/

G Preserving Periodic Properties of Elements using CrysAtom

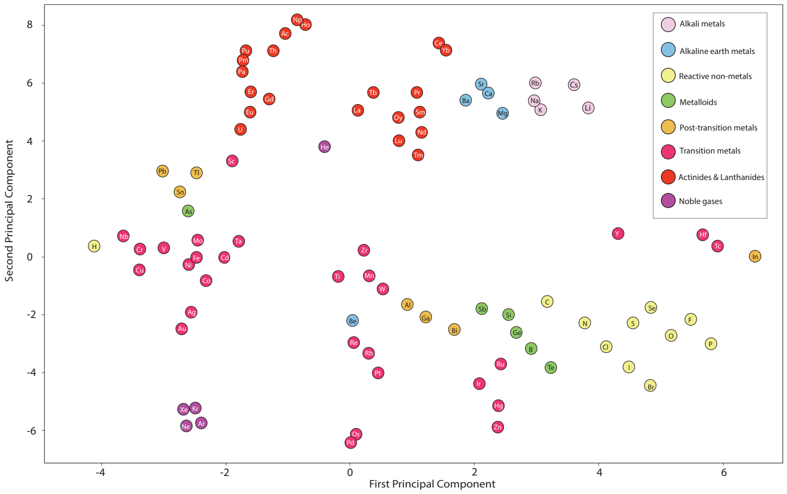

Figure 3: Dimensionally-reduced atomic vectors were obtained from 200-dimensional vectors. Subsequently, these vectors were further reduced to two dimensions using t-SNE [49] for visualization. This plot shows an approximate position of the atomic vector representations of atoms in two dimensional (first and second principal components) Euclidean space.

H Visualization of Our 200-dimensional Atom Vectors Along Third and Fourth Principal Components

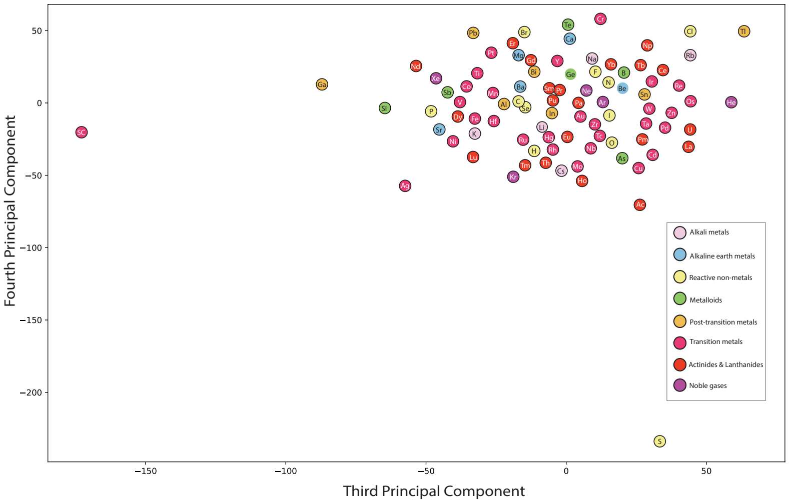

Figure 4: Dimensionally-reduced atomic vectors were obtained from 200-dimensional vectors. Subsequently, these vectors were further reduced to two dimensions using t-SNE [49] for visualization. This plot shows an approximate position of the atomic vector representations of atoms in two dimensional (third and fourth principal components) Euclidean space .

Figure 4 visualize the dense representation of our atom vectors using t-SNE [49]. Here we reduced our 200-dimensional dense vector and plot them using the other two principal components namely third and fourth. From Figure 4, we observe that the projection over the third and fourth principal components is extremely noisy. There appears to be no discernible pattern beyond the first two principal components.

I Ablation Study

Property	CGCNN (200-dim CrysAtom)	CGCNN (100-dim CrysAtom)	CGCNN (50-dim CrysAtom)
Formation Energy	**0.040**	0.044	0.046
Bandgap (OPT)	**0.143**	0.160	0.165
Total Energy	**0.043**	0.049	0.053
Ehull	**0.124**	0.127	0.130
Bandgap (MBJ)	**0.333**	0.349	0.356
Bulk Modulus (Kv)	**12.37**	13.29	13.60
Shear Modulus (Gv)	**10.45**	10.75	10.99

(a)

Property	UL + SSL (BT)	UL + SSL (VICREG)	UL + SSL (NTXent)	UL
Formation Energy	**0.040**	0.042	0.042	0.043
Bandgap (OPT)	**0.143**	0.156	0.155	0.158
Total Energy	**0.043**	0.045	0.047	0.048
Ehull	**0.124**	0.130	0.126	0.127
Bandgap (MBJ)	**0.333**	0.369	0.340	0.345
Bulk Modulus (Kv)	**12.37**	13.20	13.20	13.24
Shear Modulus (Gv)	**10.45**	10.65	10.68	10.71

(b)

Table 10: (a) Performance comparison (MAE) of various versions of CGCNN equipped with different dimensional vector representations (such as 50-dim, 100-dim, and 200-dim) obtained from CrysAtom. (b) Performance comparison (MAE) of our CrysAtom framework by introducing different SSL loss functions such as BT, VICREG, NTXent and UL (without using SSL loss). Here we use CGCNN as our encoder model to conduct the experiments. The best results have been shown in **bold** and the second best results have been underlined.

In this section, we demonstrate the effect of variation in the dimension of the dense vector representation on its capability to encode the chemical properties of an atom. We also analyze the influence of combining UL and SSL on CrysAtom performance and efficiency of different variants of CrysAtom aided SOTA models by designing the following set of ablation studies:

1. Does increase in dimensions of the vector representation entail better performance in downstream property prediction task?
2. Does hybrid learning strategy perform well when UL is aided with SSL?
3. What is efficiency of different CrysAtom versions of SOTA models?
4. Does increasing the dataset size, which is used to generate dense vector representation, impacts the performance on the baseline property predictor?

I.1 Impact of Increasing Dimensions

Results in Table 10a leads to the conclusion that increase in the dimension of the dense vectors improves the performance of the property predictor model (here, CGCNN). Our conclusion is aligned with earlier findings by Antunes et. al [9]. However, increasing dimension size of the vector representation beyond a certain point may lead to exponential growth in computational resources and may prove to be an impediment in terms of model training.

I.2 Impact of Combining UL and SSL

Table 10b demonstrates how effectively we can combine our UL and SSL based approaches in JARVIS-DFT dataset for seven properties. We observe that combination of SSL and UL based frameworks equipped with Barlow Twins [31] loss helps the existing property predictors significantly to achieve SOTA performance on downstream task, when compared with NTXent [50] and VICREG [51] losses. The common intuition is that, while training a model using Barlow Twins loss, it generates the hard negative samples by masking the nodes and edges internally for a graph to capture the overall topological structure. Whereas, SSL loss functions such as NTXent, VICREG take necessary negative samples by considering external properties such as space group information. SSL strategies, in isolation, are not used for generative tasks, thereby preventing us from using such strategies for generation of dense vector representations. However, UL is applied together with SSL, leading to a generative loss for generating a dense vector representation as a subtask of graph generation.

Method	Time/Epoch	Total Training Time	Total Testing Time	Model Para.
CGCNN (CrysAtom)	0.189 s	3.12 h	0.04 s	1.1 MB
CrysXPP (CrysAtom)	0.195 s	3.25 h	0.07 s	1.1 MB
ALIGNN (CrysAtom)	140.4 s	39 h	80.4 s	97.8 MB
Matformer (CrysAtom)	80.4 s	22 h	1.04 s	68.42 MB
PotNet (CrysAtom)	42 s	11 h	31s	42.9 MB

Table 11: Training time per epoch, total training time, total testing time, and model complexity compared with CGCNN (CrysAtom), CrysXPP (CrysAtom), ALIGNN (CrysAtom), Matformer (CrysAtom) and PotNet (CrysAtom) for formation energy on JARVIS-DFT dataset.

I.3 Efficiency of CrysAtom Variant of the SOTA Models

Table 11 shows comparison of CGCNN (CrysAtom), CrysXPP (CrysAtom), ALIGNN (CrysAtom), Matformer (CrysAtom) and PotNet (CrysAtom) in terms of training time per epoch, total training time, total testing time and model complexity for formation energy on JARVIS-DFT dataset. We clearly observe that CGCNN (CrysAtom) is the fastest model among all the CrysAtom variant of SOTA models. It uses less number of parameters compared to the other models. This led us to consider CGCNN as our encoder model to conduct the necessary experiments in ablation study which is present in the main manuscript.

Property	CGCNN (CrysAtom pre-trained on MP (139K))	CGCNN (CrysAtom pre-trained on MP+OQMD (800K))
Formation Energy	**0.040**	0.042
Bandgap (OPT)	**0.143**	0.147
Total Energy	**0.043**	0.045
Ehull	**0.124**	0.128
Bandgap (MBJ)	**0.333**	0.338
Bulk Modulus (Kv)	**12.37**	12.54
Shear Modulus (Gv)	**10.45**	10.59

Table 12: Performance comparison between CGCNN models using a 200-dimensional CrysAtom vector representation pre-trained on 139K MP instances and a 200-dimensional CrysAtom vector representation pre-trained on 800K MP+OQMD instances across various property prediction tasks. The CGCNN model with CrysAtom pre-trained on 139K MP instances consistently outperforms the one pre-trained on 800K MP+OQMD instances in all tasks.

I.4 Impact of Dataset Size on Dense Vector Generation during CrysAtom Pre-training

Table 12 presents the performance comparison between CGCNN using a 200-dimensional CrysAtom vector representation pre-trained on 139K MP instances, and CGCNN using a 200-dimensional CrysAtom vector representation pre-trained on 800K MP and OQMD instances. It is evident that CGCNN with CrysAtom pre-trained on 139K instances consistently outperforms CGCNN with CrysAtom pre-trained on 800K instances across various property prediction tasks. A possible reason for this outcome could be that the additional data from the OQMD dataset may introduce noise or irrelevant features that are not as beneficial for the specific property prediction tasks. The MP dataset is more closely aligned with our target properties, leading to better generalization when pre-trained on a smaller but more relevant dataset (139K MP instances). In contrast, the inclusion of OQMD data (800K instances) might result in a more diverse but less focused representation, which could hinder the model's performance on tasks that require the specific characteristics captured by the MP dataset. Therefore, we select the CrysAtom vector representation pre-trained exclusively on the MP dataset as our best representation. The running time of each epoch of the training of CrysAtom (using Algorithm 1) on MP+OQMD (800K) is approximately 1 hours 30 minutes and the number of parameters used by CrysAtom is 5.5 MB.