Xtal2DoS: Attention-based Crystal to Sequence Learning for Density of States Prediction

Anonymous Author(s)
Affiliation
Address
email

Abstract

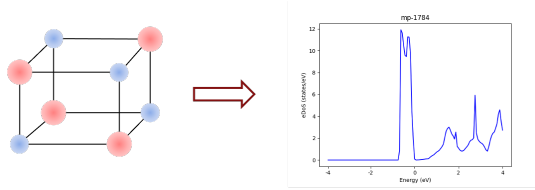
Modern machine learning techniques have been extensively applied to materials science, especially for property prediction tasks. A majority of these methods address scalar property predictions, while more challenging spectral properties remain less emphasized. We formulate a crystal-to-sequence learning task and propose a novel attention-based learning method, Xtal2DoS, which decodes the sequential representation of the material density of states (DoS) properties by incorporating the learned atomic embeddings through attention networks. Experiments show Xtal2DoS is faster than the existing models, and consistently outperforms other state-of-the-art methods on four metrics for two fundamental spectral properties, phonon and electronic DoS.

1 Introduction

Machine learning for materials discovery has risen to prominence recently, with impressive applications to materials' property predictions [1, 2]. While the performance greatly exceeds the existing models, most of the achievements are confined to individual scalar quantities [3–6]. For conventional machine learning models, some other structured properties are often hard to deal with, for example spectral properties [7]. Two fundamental spectral properties are phonon density of states (phDoS) [8] and electronic density of states (eDoS) [9]. In a given frequency interval, phDoS describes the number of phonon modes if the density of wave vectors in the Brillouin zone is homogeneously distributed. The phonon density of states is typically normalized to one [10]. Similarly, eDoS depicts the number of different states at a particular energy level that electrons are allowed to occupy (namely, the number of electron states per unit volume per unit energy) [11]. Mathematically, the density of states can be regarded as a sequence depicting the fundamental characteristics of particles. Each crystalline material is associated with unique density of states (phDoS, eDoS). The goal of density of states prediction is to estimate this sequence given an input structure.

There are two challenges in this sequence prediction problem. First, the input crystals are 3D structures, which are very different from typical machine learning formats (i.e., vectors or matrices). Featurizing the crystal structures is the cornerstone of the whole learning process. If useful information cannot be extracted effectively, the follow-up predictions would be severely hampered. Essentially, each crystal structure is a collection of atoms. Each atomic site contains a specific element, and these sites can be seen as nodes on a graph. Atoms also have interactions with each other [12]. Close atoms often have strong bonds. These bonds can be represented by the connecting edges on the graph [3]. Conventional CNN [13] and RNN [14] are not applicable to graphs, while graph neural networks [15] are specifically designed for such structures. The pioneering work applying GNN networks to

Submitted to NeurIPS 2022 AI for Science Workshop.



$D(E) = N(E)/V$

Figure 1: An illustration of predicting eDoS from the crystal structure for material *CsF*.

materials property prediction was CGCNN [3], where the crystal unit cell was formalized as a graph with atoms as nodes and bonds as edges. MEGNet [16] further introduced a global state to add extra features like temperature, pressure, and entropy. More recently, GATGNN [17] suggested using an attention mechanism to capture the global crystal structure beyond the local atomic environments. In GATGNN, the local augmented graph-attention layers encode the properties of local environments for each atom to obtain an abstract feature, and a global attention layer aggregates these atomic vectors by weighted averaging to produce the global feature vector for prediction. This exploitation of structure markedly facilitated feature extraction from the crystalline systems, and built an extensible foundation for various predictions.

The second challenge in sequence prediction is the target format. Works like GATGNN focused on the prediction of scalar properties such as formation energies, band gaps and elastic moduli. The performance of these models degrades when applied to spectral properties like eDoS and phDoS [18]. The simplest way to adapt the previous methods to predict sequences is to expand the last layer to output multiple values instead of a single value. The general idea is similar to predicting multiple scalar properties simultaneously. However, such parallel prediction overlooks the sequential nature of the target DoS. The smoothness property cannot be enforced onto the predicted targets either. Mat2Spec [18] observes this flaw and re-formulates the problem as a multi-target regression task. Mat2Spec keeps the framework of expanding the last layer to output more values, but additionally uses contrastive learning [19] and a probabilistic variational autoencoder [20] to model the correlations among different locations of the sequence.

In this work, we introduce Crystal-to-DoS (Xtal2DoS) learning, to improve both the graph encoding and the sequence decoding for phDoS and eDoS predictions. First, we inherit the general framework of GATGNN and employ graph attention networks (GAT) [21] to learn both the local and global attention. Second, we adopt the attention-based transformer to decode the target sequence [22]. The transformer reads the one-hot position embedding together with the global crystal representation, and decodes left-to-right auto-regressively. Differing from the multi-target regression perspective, Xtal2DoS takes the sequence-to-sequence (seq2seq) learning perspective. In seq2seq, the decoding process incorporates both self-attention and source-attention. Self-attention retrieves the historical memory from the previous steps, and source-attention derives the similarity between the decoding sequence and the input sequence. In this DoS prediction task, the input is a graph rather than a sequence, so the source-attention attends each position with all the atom embeddings. The resulting model, Xtal2DoS, thus improves both the encoding and decoding over prior models, for this graph-to-sequence learning problem. Experimental results show our method can outperform the existing state-of-the-art models on all four evaluation metrics by over 14% on average for phDoS, and over 4% on average for eDoS.

2 Crystal-to-DoS Learning

2.1 Problem Formulation

The input crystal material x is denoted as a graph $G_x = (V, E)$. $V = \{u_i\}_{i=1}^n$ is the collection of n atoms where u_i is an atom embedding learnt from CGCNN. $E = \{e_{ij}\}_{1 \leq i,j \leq n}$ denotes the bonds connecting atoms. e_{ij} depicts the distance between two atoms i, j. For a crystalline system, the

crystal structure repeats in the 3D space and each atom has a 3D coordinate. Though theoretically, each atom has connections with any other atom, for simplicity and generality, we choose n_{max_nbr} neighbors to be the upper bound for the number of neighbors of each atom. In other words, the sum of each row or column of G_x's adjacency matrix is smaller than or equal to n_{max_nbr}. Our goal is to predict y for each x. y is the desired density of states (phDoS or eDoS), with length l_y. $l_y = 51$ for phDoS and $l_y = 128$ for eDoS. l is predefined for each type of DoS. The learnt model maps x to y ($f : x \rightarrow y$). Figure 1 presents an example (caesium fluoride , *CsF*) of DoS prediction.

2.2 Graph Attention Encoder

As the prototype of GAT, graph convolutional network (GCN) [23] combines the local graph structure and node-level features to yield impressive node predictions. However, the feature aggregation of GCN depends on structure and is mostly linear, constraining effectiveness of message passing on the graph. In GCN, a graph convolution operation can be written as

$$h_i^{l+1} = \sigma(\sum_{j \in N(i)} \frac{1}{c_{ij}} W^l h_j^l) \tag{1}$$

$\sigma(\cdot)$ is a non-linear activation function. W_l is the feature transformation matrix in the l-th layer. h_i^l is the latent node embedding for node i in the l-th layer, with $h_i^0 = u_i$. $c_{ij} = \sqrt{|N(i)||N(j)|}$ is a normalization constant and $N(i)$ is the set of node i's neighbors.

In Xtal2DoS, we use a variant of GAT, *UniMP* [24], to encode the crystalline graph. Compared to GAT, UniMP adds support for edge features. For each layer in UniMP, h_i^l is first linearly transformed to queries, keys and values

$$q_i^l = W_q^l h_i^l + b_q^l, \qquad k_i^l = W_k^l h_i^l + b_k^l, \qquad v_i^l = W_v^l h_i^l + b_v^l. \tag{2}$$

Similarly, we also transform the edge features between node i and node j

$$g_{ij} = W_e e_{ij} + b_e. \tag{3}$$

The edge feature transformation is not layer-dependent and W_e, b_e are shared for all edges. The attention coefficients are then computed as follows

$$\alpha_{ij}^l = \frac{\langle q_i^l,\ k_j^l + g_{ij} \rangle}{\sum_{p \in N(i)} \langle q_i^l,\ k_p^l + g_{ip} \rangle} \tag{4}$$

where $\langle q, k \rangle = \exp(\frac{q^T k}{\sqrt{d}})$ and d is the dimensionality of q_i^l, k_i^l or v_i^l. Afterwards, the embeddings from neighbors are aggregated, together with the edge embeddings, weighted by attention scores

$$\hat{h}_i^{l+1} = \sum_{j \in N(i)} \alpha_{ij}^l \cdot (v_j^l + e_{ij}). \tag{5}$$

From Eq. 1 and Eq. 5, GAT considers similarity scores and edge information when performing the aggregation to gain more expressiveness.

Finally, motivated by the success of residual neural networks [25], UniMP appends a gated residual connection before passing $\hat{h}_i^{l+1}$ to the next layer

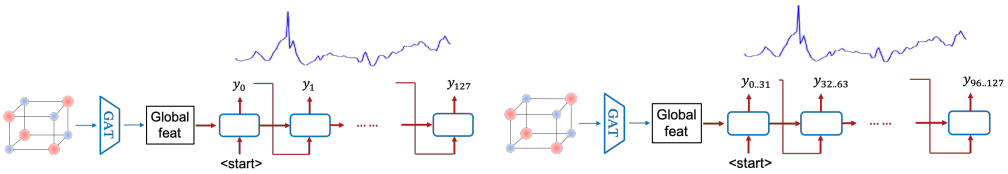
$$\begin{aligned} r_i^l &= W_r^l\, h_i^l + b_r^l, \qquad \beta_i^l = \text{sigmoid}(a^l\ [\hat{h}_i^{l+1}\ ;\ r_i^l\ ;\ \hat{h}_i^{l+1} - r_i^l]) \\ h_i^{l+1} &= \sigma(f_{LN}((1-\beta_i^l)\ \hat{h}_i^{l+1} + \beta_i^l\ r_i^l)) \end{aligned} \tag{6}$$

where ; denotes concatenation, a^l is the learnable weight parameter, and f_{LN} represents the layer norm. Note that for the last layer of UniMP, $\sigma(\cdot)$ and f_{LN} are skipped.

2.3 Sequential Decoding

GAT learns embeddings for each node, h_i^L, if there are L layers in total. GATGNN and Mat2Spec simply take the average of all the embeddings as the global feature vector

$$h_x = \frac{1}{n} \sum_{i=1}^{n} h_i^L \tag{7}$$



(a) The global feature vector is fed into RNN as the initial state. The decoding unrolls step by step.

(b) Chunk RNN decoder is an RNN decoder that predicts a segment instead of a single value at each step.

Figure 2: Vanilla RNN, chunk RNN (without attentions).

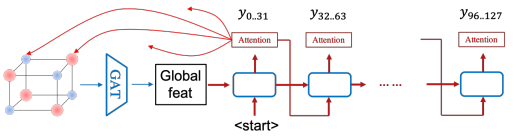


Figure 3: Adding an attention mechanism to the chunk RNN decoder.

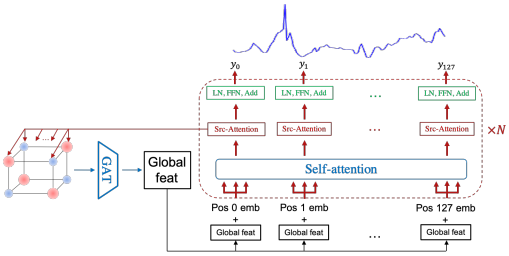


Figure 4: Fully attention-based transformer decoding for DoS prediction. All the recurrent components are replaced with self-attention.

Then the global feature vector is used to decode the target DoS, using MLP or VAE.

We compared various sequence models including RNN and transformer for decoding. By default, we used the GRU model for RNN.

RNN Figure 2a sketches the RNN decoding of the GAT-encoded crystal. RNN is naturally suitable for sequence decoding. It reads the global feature vector as the cell state and a special $\langle start \rangle$ symbol to indicate the start of the decoding. The output at each step is also the input for the next step. However, RNN is criticized for lacking long-term memory [26]. When we decode the last step (y_{127}), the memory of the first step (y_0) is probably degraded. This might hurt the prediction quality for long sequences. Another drawback of RNN is the speed. Each step has to wait for the previous output, and thus the decoding process could take a very long time.

Chunk RNN Figure 2b gives an overview of a chunk RNN, which is a simple RNN variant. Chunk RNN predicts a segment rather than a single value at each step. The benefit is that the decoding time could be greatly shortened. For instance, on eDoS, if the segment length is 32, then sequence length can be reduced from 128 to 4. On the other hand, each segment still waits for the generation of the previous segment, and longer segments harm the accuracy gradually.

Chunk RNN + Attention To compensate for the performance degradation brought by chunk RNN, we further introduce the attention mechanism into the sequence model (Figure 3). The attention module directly attends to the latent sequence states with the atom embeddings. The latent states can take specific atoms into consideration and build more fine-grained correlations. In practice, we also find including the attention module boosts the evaluation results.

Transformer Given that we have already introduced the attention mechanism, it is intuitive to completely drop the recurrent framework and embrace the fully attentional transformer [22]. The input at each step is the concatenation of the positional embeddings and the global feature vector. A positional embedding is simply a one-hot vector of length l_y indicating the index of the current step. The global feature vector h_x additionally provides the global context for decoding. Formally, let $z_i = [s_i \; ; \; h_x]$ where s_i is the one-hot position embedding with length l_y. z_i is the initial state for the first self-attention layer. As in UniMP, we derive collections of queries, keys and values for the hidden states of each self-attention layer, denoted by Q, K, and V, respectively.

Transformer adopts dot-product attention to calculate the outputs

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T/\sqrt{d})\, V \tag{8}$$

where d is the latent dimensionality. Multiple sets of Q, K, and V matrices can be learned, and each set is known as a head. Multiple heads (multi-head) can give self-attention greater power of discrimination. The outputs of all the heads are concatenated, layer-normalized and decoded through a fully connected feed-forward network. A more detailed depiction can be found in [22].

During the decoding phase, each attention layer also computes the source-attention between the output states and the input atom nodes. If we denote the queries, keys, values in decoding as $Q_{dec}, K_{dec}, V_{dec}$, and the ones from the input graph encoding as $Q_{enc}, K_{dec}, V_{dec}$, we have

$$\begin{aligned} \text{Self-Attention}(Q_{dec}, K_{dec}, V_{dec}) &= \text{softmax}(\frac{Q_{dec}K_{dec}^T}{\sqrt{d}})V_{dec}, \\ \text{Source-Attention}(Q_{dec}, K_{enc}, V_{enc}) &= \text{softmax}(\frac{Q_{dec}K_{enc}^T}{\sqrt{d}})V_{enc}, \end{aligned} \tag{9}$$

for the sequential decoding. Source-attention usually happens after self-attention. Furthermore, the transformer model adds masking to prevent the current positions from attending to the future positions, to prevent leftward information flow in the decoder, and to preserve the auto-regressive property, ensuring each decoded state only knows the history but not the future. The final generated length-l_y sequence $\hat{y}$ is compared with the ground-truth y through Kullback–Leibler divergence (KL-div) [27] or its generalized version, *Bregman Divergence* (BD) [28], resulting in the loss term, $\mathcal{L} = D_{KL}(y \,||\, \hat{y})$. We chose KL-div and BD because they are more suitable than other measures for data containing peaks, which are ubiquitous in DoS.

Our model is a synergistic combination of a GAT (UniMP) encoder and a transformer decoder. The whole model is entirely based on attentions on graphs and sequences. Figure 4 illustrates the main idea of Xtal2DoS.

3 Related Works

Learning for 3D structures has been a long-standing problem. In materials discovery, two atomic systems can be exactly the same even though the coordinates are totally different in the 3D space if they can be transformed to one another through translation, rotation or inversion. For standard machine learning models, it is very hard to capture such invariance and symmetry [29]. There are in general three strategies to address the concern.

First, the same models can be trained using more data [30]. The structured inputs can be randomly transformed without changing the labels. Some prior works successfully applied data augmentation to exploring materials design spaces [31]. On the other hand, in materials science, even with augmentation, it is not easy for machine learning models to comprehend the crystalline system, let alone the data collection and augmentation require enormous domain knowledge and expert effort [2].

Second, rather than expanding the scale on the dataset side, some researchers have explored the possibility of enforcing the invariance or equivariance inside the model. This kind of model handles the coordinates directly and are often called equivariant neural networks [32, 33]. One notable model for materials science applications is E3NN [34]. E3NN performs training based on radial functions and spherical harmonics [35]. Intuitively, E3NN represents the atoms with the 3D coordinates, mass and velocity, and tells the equivariant convolutional filters how the crystals might transform. Thus, E3NN can capture the full crystal symmetry, while reducing training set sample complexity. However, radial functions and spherical harmonics may limit the expressiveness of the model. In some scenarios where the data is abundant, E3NN's benefits might become marginal.

Third, it is often helpful to pre-process the data for simplicity [36]. Instead of using the coordinates, we can re-format the crystals to graphs by retaining only the atoms and the bonds between them.

Some of the structure information is lost from the input, such as the lattice angles, but this could also reduce the risk of overfitting. Once the inputs are converted to graphs, mature graph neural network techniques [37, 38] can be applied. A representative work is GATGNN [17]. GATGNN is one of the primary methods to encode the atom embeddings and extract the global feature vector. It still leads in the performance of many scalar property prediction tasks including absolute energy, Fermi energy, band gap and Poisson-ratio. Nevertheless, in sequence property predictions, its performance degrades significantly [18]. Mat2Spec [18] uses the same graph encoder as GATGNN, but improves the decoder. GATGNN only uses an MLP, which is incapable of sequence decoding. Mat2Spec not only aligns the inputs and targets in a latent multivariate Gaussian space, but also learns an embedding for each energy level through contrastive learning. Mat2Spec views the DoS prediction as a multi-target regression problem, yet ignores the sequential structure of DoS. Besides, Mat2Spec simply deepens the MLP decoder and plugs contrastive modules into the decoder, while sequence modeling is cleaner and simpler.

Xtal2DoS follows the design of GATGNN and Mat2Spec because the graph encoding is shown to outperform the coordinate encoding [18]. On top of the existing designs, Xtal2DoS further replaces GAT encoder with a more advanced variant, UniMP, and replaces the decoder with a powerful sequence model, transformer. Evaluation results and ablation studies prove the effectiveness and efficiency of our introduced model.

4 Experiments on DoS Prediction

We validate the performance of Xtal2DoS on two datasets, phDoS and eDoS. The task of directly predicting phDoS in crystalline solids from atomic species and positions was first proposed in E3NN [1] [34]. We also include a new version of E3NN, denoted by E3NN-new [2] which improves over the previous version. phDoS is normalized to 1 since phDoS gives the number of modes per unit frequency per unit volume of real space and is essentially a distribution. phDoS is critical for calculating properties such as the average phonon frequency and the heat capacity at 300K. The analogous task of predicting eDoS for nonmagnetic materials was proposed in [18]. eDoS is the same as phDoS except that it depicts electrons. Materials scientists are interested in a small energy range between -4 to 4 eV with respect to band edges with 63 meV intervals, including both unoccupied and occupied states. Fermi energy and band edges are all set to 0 eV on this energy grid. Predicting eDoS accurately can be very valuable to bandgap energy estimation. Both phDoS and eDoS are continuous functions in nature. We sample 51 points on phDoS and 128 points on eDoS spanning the frequency or energy range. In other words, $l_y = 51$ for phDoS and $l_y = 128$ for eDoS.

4.1 Evaluation Metrics

Four metrics are adopted for evaluation: coefficient of determination (R^2), mean absolute error (MAE), mean squared error (MSE) and Wasserstein distance (WD).

R^2 measures the proportion of the variation in the ground-truth targets y that is predictable from the predictions $\hat{y}$, $R^2 = 1 - \frac{\sum_{i=1}^{l_y}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{l_y}(y_i - \bar{y})^2}$, where $\bar{y} = \frac{1}{l_y}\sum_i y_i$. In the best case when y and $\hat{y}$ match exactly, R^2 would reach the maximum possible value, 1. If $y_i = \bar{y}$, $R^2 = 0$. Note that R^2 score can be negative if the predictions are worse.

MAE and MSE are standard metrics for regression. We also adopt them for evaluation.

Wasserstein distance is another metric defined between probability distributions. If we view the two distributions as two piles of earth, WD is the minimum amount of dirt to move in order to turn one pile to be the same as the other. Therefore, WD is also known as earth mover's distance. We define

[1] https://github.com/zhantaochen/phonondos_e3nn/blob/main/phdos_train.ipynb
[2] https://docs.e3nn.org/en/stable/api/nn/models/gate_points_2101.html

model	$R^2 \uparrow$	MAE $\downarrow$	MSE $\downarrow$	WD $\downarrow$
E3NN	0.560	0.00869	0.000389	0.1217
E3NN-new	0.621	0.00833	0.000275	0.0765
GATGNN	0.457	0.00976	0.000405	0.1376
E3NN2DoS	<u>0.660</u>	<u>0.00740</u>	<u>0.000247</u>	0.0740
Mat2Spec	0.641	0.00773	0.000264	<u>0.0736</u>
Xtal2DoS	**0.734**	**0.00663**	**0.000193**	**0.0631**
improve	11.21%	10.41%	21.86%	14.27%

Table 1: The evaluation results of four metrics on phDoS. (Best in bold; 2nd best underlined.)

model	$R^2 \uparrow$	MAE $\downarrow$	MSE $\downarrow$	WD $\downarrow$
E3NN	0.410	4.8968	99.852	0.467
E3NN-new	<u>0.555</u>	3.9190	<u>74.949</u>	0.233
GATGNN	0.492	4.2965	85.555	0.285
E3NN2DoS	0.552	<u>3.8900</u>	75.453	0.236
Mat2Spec	0.545	3.8939	76.624	<u>0.230</u>
Xtal2DoS	**0.577**	**3.7845**	**71.213**	**0.218**
improve	3.96%	2.71%	4.98%	5.22%

Table 2: The evaluation results of four metrics on eDoS. (Best in bold; 2nd best underlined.)

model	$R^2 \uparrow$	MAE $\downarrow$	MSE $\downarrow$	WD $\downarrow$
RNN	0.663	0.00794	0.000245	0.0797
RNN+Attn	0.668	0.00726	0.000241	0.0664
Chunk RNN+Attn	<u>0.693</u>	<u>0.00720</u>	<u>0.000223</u>	<u>0.0656</u>
Xtal2DoS	**0.734**	**0.00663**	**0.000193**	**0.0631**
improve	5.92%	7.92%	13.45%	3.81%

Table 3: Ablation study on phDoS.

model	$R^2 \uparrow$	MAE $\downarrow$	MSE $\downarrow$	WD $\downarrow$
Chunk RNN+Attn	0.558	3.9125	75.013	0.221
Xtal2DoS	**0.577**	**3.7845**	**71.213**	**0.218**
improve	3.41%	3.27%	5.07%	1.36%

Table 4: Ablation study on eDoS.

the WD as

$$\mathrm{WD} = \int_{-\infty}^{+\infty} |P - Q| \tag{10}$$

for distributions p and q with cumulative distribution functions (CDF) P and Q, respectively. The continuous form of WD can also be adapted for discrete cases.

4.2 Implementation

The graph encoder is built with PyTorch Geometric [39] modules. The encoder is composed of three GAT/UniMP layers, each followed by batch normalization [40] and a non-linear activation [41]. The sequence decoder is a stack of six self-attention layers, where each layer has subsequent layer normalization and feed-forward layers. The whole model is trained with the Adam optimizer [42], on Nvidia V100 GPU.

4.3 Prediction Results

Xtal2DoS is first evaluated on phDoS, compared with other state-of-the-art baselines, including E3NN, GATGNN, Mat2Spec and some of their variants. Among all the compared methods, Xtal2DoS gives the best numbers. On R^2, MAE, MSE and WD, the relative improvements over the best compared baseline are as large as 11.21%, 10.41%, 21.86%, 14.27%, respectively. The average improvement is 14.44%.

Table 2 shows the performance gain achieved by Xtal2DoS over other competitive baselines on the more challenging eDoS dataset. Xtal2DoS outperformed the best baseline on the four metrics by 3.96%, 2.71%, 4.98%, 5.22% respectively. The average improvement is 4.22%, which is a noticeable increment especially when the training speed is faster (see section 4.6).

4.4 Ablation Study

Encoder To validate the neural module selection of Xtal2DoS, we compare E3NN graph encoder (E3NN2DoS) and UniMP (Xtal2DoS) graph encoder with transformer decoder fixed. E3NN2DoS adopts the new version of E3NN. From Table 1 and 2, Xtal2DoS consistently outperforms E3NN2DoS. Though E3NN preserves the extra information like angles, these features might also lead to overfitting. E3NN could potentially work better when the data are more scarce and the properties of interest are scalars. Similar findings can also be found in [43].

model	E3NN	Mat2Spec	RNN	RNN+Attn	Chunk RNN+Attn	**Xtal2DoS**
Time per epoch (s)	31	33	54	69	42	**11**

Table 5: The training time for Mat2Spec, RNN-based sequence models and Xtal2DoS. For fair comparison, we illustrate the training cost for each epoch in seconds. They take a similar number of epochs to converge. So, overall, Xtal2DoS is the fastest method among all the models.

Decoder We further compare the performance of RNN, RNN+attention, Chunk RNN+attention, and the transformer (Xtal2DoS). We use the same four metrics to compare these settings. In Table 3 and 4, transformers always dominate other RNN-based models on all the metrics. The improvements shown in the last row indicate the comparison between Xtal2DoS and the best compared model (usually Chunk RNN+Attn).

4.5 Qualitative Results

We also show some qualitative results on both phDoS and eDoS in Figure 5 and 6 (in Appendix). Both figures demonstrate the accuracy of our predictions. The red and blue lines are always closely correlated.

4.6 Training Speed

One major advantage of the transformer models is parallelism. Unlike RNNs where each step has to wait for the previous output, the transformer computes the outputs at all the steps simultaneously and enforces auto-regressiveness through masking. These operations can be performed in matrix form, taking advantage of powerful GPUs, and thus substantially reduce the training cost. Table 5 shows the training time per epoch for Mat2Spec, RNN-based sequence models and Xtal2DoS. Mat2Spec does not involve any sequential model and generates outputs through MLP, so it is faster than any RNN-based sequence decoder model. Mat2Spec instead integrates probabilistic sampling and contrastive learning, so it is not surprising that it takes more time than Xtal2DoS. Unsurprisingly, RNN trains more than 50% slower than Mat2Spec. Adding the attention module further prolongs the time cost. RNN+Attention doubles the training cost compared with Mat2Spec. This also motivated us to attempt chunk RNN. As a result, chunk RNN+Attention can complete a training epoch faster than RNN and RNN+Attention, but is still slower than Mat2Spec. It only takes one third of Mat2Spec's training time, one fifth of RNN's training time, and one fourth of chunk RNN+Attention's training time. Xtal2DoS excels at speed because of its parallel design and its succinct structure to avoid extra neural module. Faster models are often more advantageous in training, tuning and deploying.

5 Conclusion

In this work, we introduce a novel graph-to-sequence learning framework, *Xtal2DoS*, for predicting the spectral properties of materials, *phDoS* and *eDoS*, from the input crystalline structures. Our model adopts an improved graph attention network (to handle edges) as the graph encoder, and a transformer as the decoder for sequential decoding. Xtal2DoS is fully attentional and captures all the atom-atom, atom-sequence and step-step correlations. Therefore, it considerably exceeds other existing state-of-the-art baselines, in quantitative evaluation metrics, qualitative matching results and training speed.

References

[1] Yue Liu, Tianlu Zhao, Wangwei Ju, and Siqi Shi. Materials discovery and design using machine learning. *Journal of Materiomics*, 3(3):159–177, 2017.

[2] Jonathan Schmidt, Mário RG Marques, Silvana Botti, and Miguel AL Marques. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5(1):1–36, 2019.

[3] 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.

[4] Olexandr Isayev, Corey Oses, Cormac Toher, Eric Gossett, Stefano Curtarolo, and Alexander Tropsha. Universal fragment descriptors for predicting properties of inorganic crystals. *Nature communications*, 8(1):1–12, 2017.

[5] Maarten De Jong, Wei Chen, Randy Notestine, Kristin Persson, Gerbrand Ceder, Anubhav Jain, Mark Asta, and Anthony Gamst. A statistical learning framework for materials science: application to elastic moduli of k-nary inorganic polycrystalline compounds. *Scientific reports*, 6(1):1–11, 2016.

[6] Ya Zhuo, Aria Mansouri Tehrani, and Jakoah Brgoch. Predicting the band gaps of inorganic solids by machine learning. *The journal of physical chemistry letters*, 9(7):1668–1673, 2018.

[7] MT Eismann. Spectral properties of materials'. *Hyperspectral Remote Sensing*, pages 133–198, 2012.

[8] R Osborn, EA Goremychkin, AI Kolesnikov, and DG Hinks. Phonon density of states in mgb 2. *Physical Review Letters*, 87(1):017005, 2001.

[9] Joseph P Heremans, Vladimir Jovovic, Eric S Toberer, Ali Saramat, Ken Kurosaki, Anek Charoenphakdee, Shinsuke Yamanaka, and G Jeffrey Snyder. Enhancement of thermoelectric efficiency in pbte by distortion of the electronic density of states. *Science*, 321(5888):554–557, 2008.

[10] Krzysztof Parlinski. Computing for materials, 2010.

[11] M Ali Omar. *Elementary solid state physics: principles and applications*. Pearson Education India, 1975.

[12] Richard M Martin. *Electronic structure: basic theory and practical methods*. Cambridge university press, 2020.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[15] Will Hamilton et al. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[16] 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.

[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.

[18] Shufeng Kong, Francesco Ricci, Dan Guevarra, Jeffrey B Neaton, Carla P Gomes, and John M Gregoire. Density of states prediction for materials discovery via contrastive learning from probabilistic embeddings. *Nature communications*, 13(1):1–12, 2022.

[19] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, pages 4182–4192. PMLR, 2020.

[20] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[21] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[23] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[24] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[26] Jingyu Zhao, Feiqing Huang, Jia Lv, Yanjie Duan, Zhen Qin, Guodong Li, and Guangjian Tian. Do rnn and lstm have long memory? In *International Conference on Machine Learning*, pages 11365–11375. PMLR, 2020.

[27] Solomon Kullback. *Information theory and statistics*. Courier Corporation, 1997.

[28] Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

[29] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

[30] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[31] Yongtae Kim, Youngsoo Kim, Charles Yang, Kundo Park, Grace X Gu, and Seunghwa Ryu. Deep learning framework for material design space exploration using active transfer learning and data augmentation. *npj Computational Materials*, 7(1):1–7, 2021.

[32] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[33] Weitao Du, He Zhang, Yuanqi Du, Qi Meng, Wei Chen, Nanning Zheng, Bin Shao, and Tie-Yan Liu. Se (3) equivariant graph neural networks with complete local frames. In *International Conference on Machine Learning*, pages 5583–5608. PMLR, 2022.

[34] 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.

[35] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68, 2018.

[36] Dhairya Kumar. Introduction to data preprocessing in machine learning, 2018.

[37] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[38] Zhongyu Huang, Yingheng Wang, Chaozhuo Li, and Huiguang He. Going deeper into permutation-sensitive graph neural networks. *arXiv preprint arXiv:2205.14368*, 2022.

[39] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[40] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

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

[43] Keqiang Yan, Yi Liu, Yuchao Lin, and Shuiwang Ji. Periodic graph transformers for crystal material property prediction. *arXiv preprint arXiv:2209.11807*, 2022.

A Qualitative Results

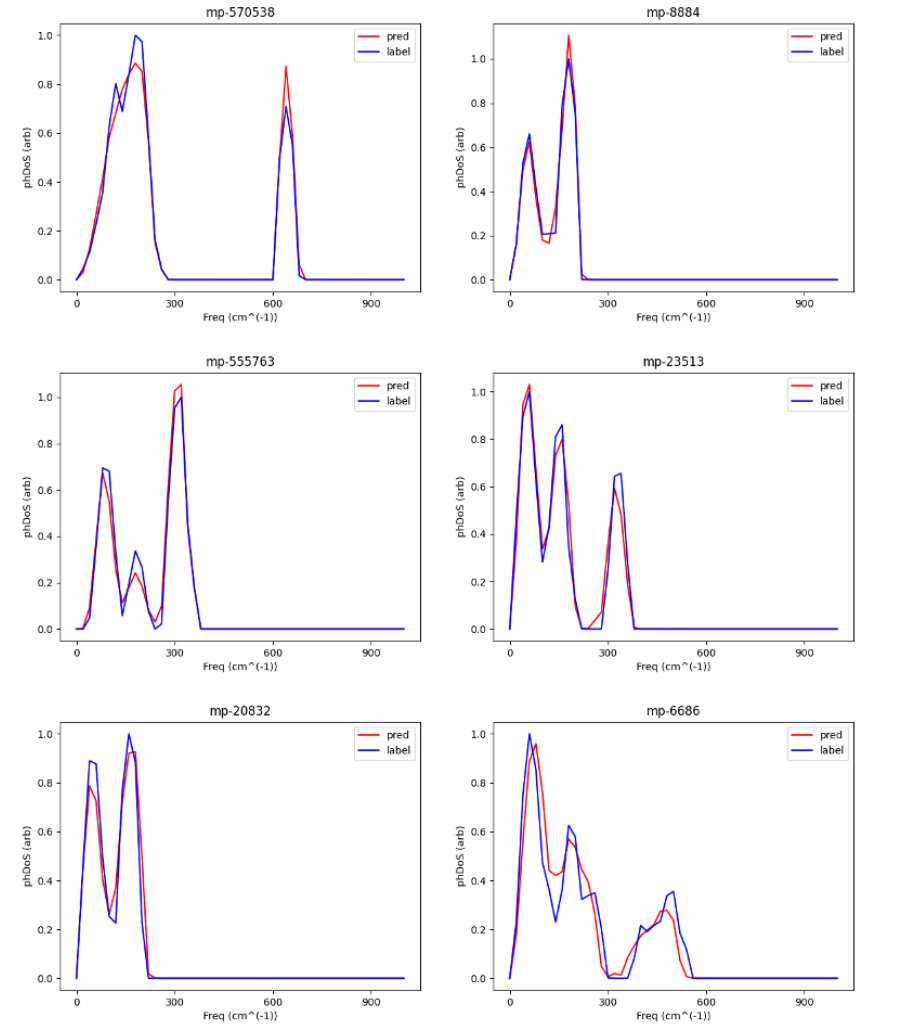


Figure 5: Comparison of the phDoS predictions and ground-truth labels on some crystalline systems. The red line denotes the prediction and the blue line denotes the ground-truth.

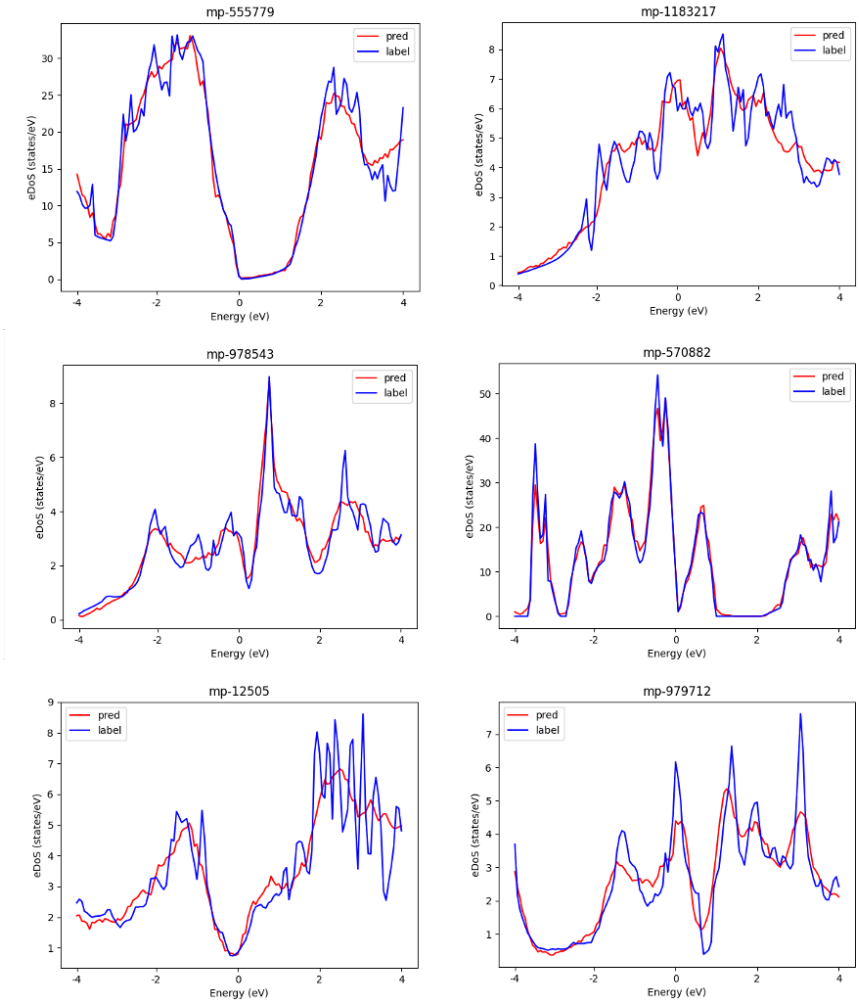


Figure 6: The comparison between the eDoS predictions and ground-truth labels on some crystalline systems. The red line denotes the prediction and the blue line denotes the ground-truth.