Self-supervised learning for crystal property prediction via denoising

Alexander New [1] **Nam Q. Le** [1] **Michael J. Pekala** [1] **Christopher D. Stiles** [1]

Abstract

Accurate prediction of the properties of crystalline materials is crucial for targeted discovery, and this prediction is increasingly done with data-driven models. However, for many properties of interest, the number of materials for which a specific property has been determined is much smaller than the number of known materials. To overcome this disparity, we propose a novel self-supervised learning (SSL) strategy for material property prediction. Our approach, crystal denoising self-supervised learning (CDSSL), pretrains predictive models (e.g., graph networks) with a pretext task based on recovering valid material structures when given perturbed versions of these structures. We demonstrate that CDSSL models out-perform models trained without SSL, across material types, properties, and dataset sizes.

1. Introduction

Recent years have seen the development of efficient and accurate machine learning (ML) methods for predicting properties of crystalline materials using descriptors based on composition (Ward et al., 2016; Goodall & Lee, 2020; Wang et al., 2021; Pogue et al., 2023) and structure (Xie & Grossman, 2018; Chen et al., 2019; Choudhary & DeCost, 2021; New et al., 2022; Ruff et al., 2024). These methods have demonstrated success across different material classes and properties (Dunn et al., 2020). They typically rely on graph networks (GNs) (Battaglia et al., 2018), in which nodes are atom, and edges capture inter-atom distances.

However, for many properties of interest, the number of material structures for which a property value is known is much less than the total number of stable materials that have a known structure. For example, the Novel Materials Discovery (NOMAD) computational database (Scheidgen et al., 2023) contains more than three million materials, and the Open Quantum Materials Database (OQMD) (Kirklin et al., 2015) contains more than one million. However, the shear modulus dataset in MatBench (Dunn et al., 2020) contains only ten thousand materials. This disparity in relative sizes will only increase as generative models are increasingly used to predict novel material structures (Xie et al., 2022; Zhao et al., 2023; New et al., 2023; Zeni et al., 2024).

In order to make use of these large general-purpose databases and to avoid needing to invest time and effort in annotation for properties with little data, a natural solution is self-supervised learning (SSL) (Balestriero et al., 2023). Unlike traditional supervised learning (SL), in SSL, models are trained on pretext tasks without need for labels, and then they are fine-tuned on the prediction task of interest. SSL has been used widely in conjunction with GNs (Xie et al., 2023), especially in the context of molecular property prediction (Hu et al., 2020; Godwin et al., 2022). Some work has also used SSL for crystalline material property prediction (Magar et al., 2022; Huang et al., 2024).

Zaidi *et al.* recently developed a novel SSL method for molecular property prediction based on structure-denoising (Zaidi et al., 2023). In particular, they showed that perturbing the atom positions in a molecule with noise and then training a model to predict that noise corresponded to learning an approximate force field for that molecule. This enabled accurate predictions of varied molecular properties.

In this work, we develop a similar denoising SSL approach for crystalline structures. Our method, crystal denoising self-supervised learning (CDSSL), works by perturbing the position of atoms in a material structure multigraph (Section 2.1 and Figure 1) and then trains a model to predict the original structure's inter-atom distances (Section 2.2). In Section 2.3, we demonstrate how to combine CDSSL with crystal property prediction models for specific prediction tasks. In Section 3.2, we evaluate CDSSL, including assessments that vary the amount of training data, the material class of interest, and the target property. We show that CDSSL consistently yields more accurate property-prediction models than those only using SL. In Section 3.3, we demonstrate that the CDSSL representation space captures some variation in properties even without finetuning.

[1]Research and Exploratory Development Department, Johns Hopkins University Applied Physics Laboratory, 11100 Johns Hopkins Rd, Laurel, MD 20723, USA. Correspondence to: Alexander New <alex.new@jhuapl.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

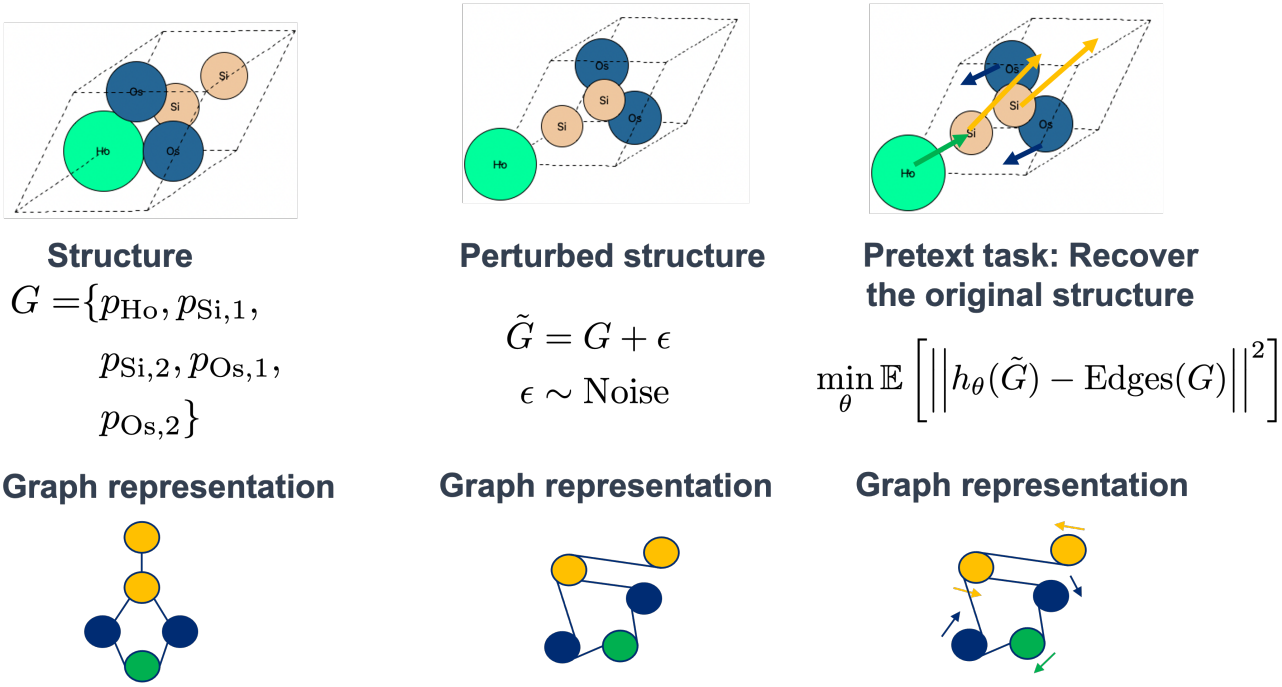

Figure 1. We summarize CDSSL. The node positions of a structure G are perturbed with Gaussian noise to create a structure $\tilde{G}$. The ML model h_θ takes the perturbed structure $\tilde{G}$ as input and seeks to output the edge embeddings of the original structure G.

2. Methods

2.1. Multigraph representations of materials

Let $\mathcal{M}$ be a space of material crystal structures. We follow the crystal graph convolutional neural network (CGCNN) (Xie & Grossman, 2018) approach and represent materials g as directed multigraphs $G = (V, E)$ consisting of sets of nodes $V = \{v\}$ and edges $E = \{(v, v', k)\}$. Each node v has a node embedding $x_v \in \mathbb{R}^{d_V}$ and a Cartesian position vector $p_v \in \mathbb{R}^3$, and each edge (v, v', k) has an edge embedding $e_{v,v',k} \in \mathbb{R}$. Because G describes a periodic tiling of a unit cell, a pair of nodes v and v' may be connected by multiple edges, indexed by $k = 1, \ldots, K_{v,v'}$.

In the CGCNN multigraph construction, the edge embedding $e_{v,v',k}$ reflects the distance between v and v': it is a function of both the node positions p_v and $p_{v'}$ and the edge index k. Given a material structure, edges (v, v', k) are constructed using nearest-neighbor calculations based on a given structure's lattice. See Table 5 in Appendix A for further details on the multigraph representation.

2.2. Crystal denoising self-supervised learning

Figure 1 outlines CDSSL. Given a graph G, we generate a perturbed version $\tilde{G}$ of it by perturbing each node's position with Gaussian noise $\tilde{p}_v \sim \mathcal{N}(p_v, \sigma^2 I)$, for some variance σ^2. Note that, compared to G, $\tilde{G}$ has the same nodes, node embeddings, and edges, but different node positions and edge embeddings.

Let $h_\theta : G \mapsto \{y_{v,v',k}\}$ be a neural network (NN), parameterized by a vector θ, that maps a graph G to a set of scalars $\{\hat{e}_{v,v',k}\}$, one for each of G's edges. Then we define the CDSSL pretext task as the following minimization problem:

$$\hat{\theta} \quad = \quad \arg\min_\theta \mathbb{E}_{p(\tilde{G}|G)p(G)} \left[\left|\left|h_\theta(\tilde{G}) - \bar{E}\right|\right|^2\right]. \quad (1)$$

Here, $p(G)$ samples graphs G from the training set, $p(\tilde{G}|G)$ generates perturbed graphs, the loss is calculated as $||h_\theta(\tilde{G}) - \bar{E}||^2 = \sum_{v,v',k} |\hat{e}_{v,v',k} - \bar{e}_{v,v',k}|^2$, and $\bar{E}$ is the set of normalized edge embeddings for G:

$$\bar{E} \quad = \quad \{\bar{e}_{v,v',k}\} = \left\{\frac{e_{v,v',k} - \mathrm{mean}\,\{e_{v,v',k}\}}{\mathrm{std}\,\{e_{v,v',k}\}}\right\}. \quad (2)$$

When every edge of a graph G has the same embedding, we set each entry of $\bar{E}$ to 0.

We present an interpretation of eq. 1. If the training set consists of structures at equilibrium, then the perturbation moves them away from locally minimizing the potential energy distribution. Thus, a model h_θ that solves eq. 1 has learned to identify small shifts $\hat{e}_{v,v',k}$ that move a non-equilibrium structure into equilibrium. As has been argued in previous work for non-periodic molecules (Zaidi et al., 2023), learning this task of predicting equilibrium structures for arbitrary materials is equivalent to learning to minimize

general interatomic potential functions. This justifies why we expect such an h_θ to have learned a general-purpose representation of materials space.

The CDSSL noise hyperparameter scale σ requires tuning. If it is too small, then the perturbed edge distances $\hat{e}_{v,v',k}$ are too similar to the original edge distances $e_{v,v',k}$, and CDSSL pretraining objective (eq. 1) is minimized when the network h_θ memorizes the training set structures. If σ is too large, then the perturbed $\tilde{G}$ is too different from the original structure G for the objective to be minimizable. Here, we show results only for a single value of σ and leave further exploration to future work.

The construction of the CDSSL pretext task is independent of the precise form of h_θ. All that is required of h_θ is that it can ingest node and/or edge embeddings and output per-edge quantities. This means that CDSSL can be used in conjunction with general structure-based property prediction architectures, such as CGCNNs (Xie & Grossman, 2018), MatErials Graph Networks (MEGNets) (Chen et al., 2019), universal material graph with three-body interactions neural networks (M3GNets) (Chen & Ong, 2022), atomistic line graph neural networks (ALIGNNs) (Choudhary & DeCost, 2021), crystal Hamiltonian graph neural networks (CHGNets) (Deng et al., 2023), or others.

2.3. Crystal denoising with MEGNets

In this work, we focus on using MEGNets as the base for h_θ. We summarize our approach in Figure 2. The MEGNet graph convolution is defined by:

$$(\{x_v\}_v, \{e_{v,v',k}\}) \quad \mapsto \quad (\{\hat{x}_v\}, \{\hat{u}_{v,v',k}\}, \hat{s}), \qquad (3)$$

where $\hat{x}_v$ are node-level output vectors, $\hat{u}_{v,v',k}$ are edge-level output vectors, and $\hat{s}$ is a graph-level output vector. During pretraining, we map edge-level output vectors to predicted edge embeddings $\hat{e}_{v,v',k}$ with a linear layer:

$$\hat{e}_{v,v',k} \quad = \quad \text{Linear}(u_{v,v',k}). \qquad (4)$$

When finetuning a pretrained model for a property prediction task, we follow typical practice for MEGNets and use Set2Set (Vinyals et al., 2016) modules to aggregate $\{\hat{x}_v\}$ and $\{\hat{u}_{v,v',k}\}$ into single vectors, and then we predict properties $\hat{y}$ with multi-layer perceptrons (MLPs):

$$\hat{y} \quad = \quad \text{MLP}(\text{Set2Set}(\{\hat{x}_v\}), \text{Set2Set}(\{\hat{u}_{v,v',k}\}), \hat{s}). \qquad (5)$$

We jointly train the MEGNet module, the Set2Set modules, and MLP for a target property, using the mean-squared error (MSE) of standardized property values as the loss.

3. Results

3.1. Evaluation details

We rely on Materials Graph Library (MatGL) (Ko et al., 2021) for general data ingestion and loading procedures and for the MEGNet (Chen et al., 2019) implementation. In particular, this leverages `pymatgen` (Ong et al., 2013) to ingest crystallographic information files (CIFs). The `pymatgen` structures are then converted into Deep Graph Library (DGL) (Wang et al., 2020) multigraphs.

We evaluate CDSSL on a variety of scalar regression material property-prediction tasks provided by MatMiner (Ward et al., 2018)[1]. The dataset, dataset size, and target property are given in Table 1. We use the matbench_mp_e_form dataset (which contains 132,752 structures) as the training set for pretraining with CDSSL. Further details on the datasets are in Table 6 in Appendix A.

All hyperparameters for pretraining and training are in Appendix A. We pretrain and train with Adam (Kingma & Ba, 2014); MEGNets use $\text{SoftPlus2}(x) = \log(\exp(x) + 1) - \log(2)$ activation. We set the CDSSL noise scale to $\sigma = 0.5$, which we chose after experimentation using matbench_log_grvh as the pretraining and evaluation dataset.

SSL is especially relevant in the setting where the amount of available labeled data is very small. Thus, we evaluate CDSSL in both low-data and high-data settings. Specifically, we vary the training dataset size to be between 10% and 70% of the total dataset size (in increments of 10%).

Dataset	Size	Property
boltztrap_mp	8,924	s_n
dielectric_constant	1,056	log10(poly_total)
jarvis_dft_2d	636	exfoliation_en
matbench_log_gvrh	10,987	log10(G_VRH)
matbench_log_kvrh	10,987	log10(K_VRH)
matbench_perovskites	18,928	e_form
matbench_phonons	1,265	log10(last phdos peak)
matbench_mp_e_form	132,752	Not used

Table 1. The dataset name, size, and target property used in our evaluation of CDSSL. The matbench_mp_e_form dataset is used only for pretraining and not for property prediction. Datasets range in size from less than a thousand structures to tens of thousands of structures, and target properties include mechanical, electronic, and thermodynamic quantities.

3.2. Evaluation results

We use the matbench_mp_e_form dataset for pretraining a MEGNet with CDSSL, with 80% for training and 20% for validation. In Figure 3, we show that the CDSSL pretraining

[1] https://hackingmaterials.lbl.gov/matminer/dataset_summary.html

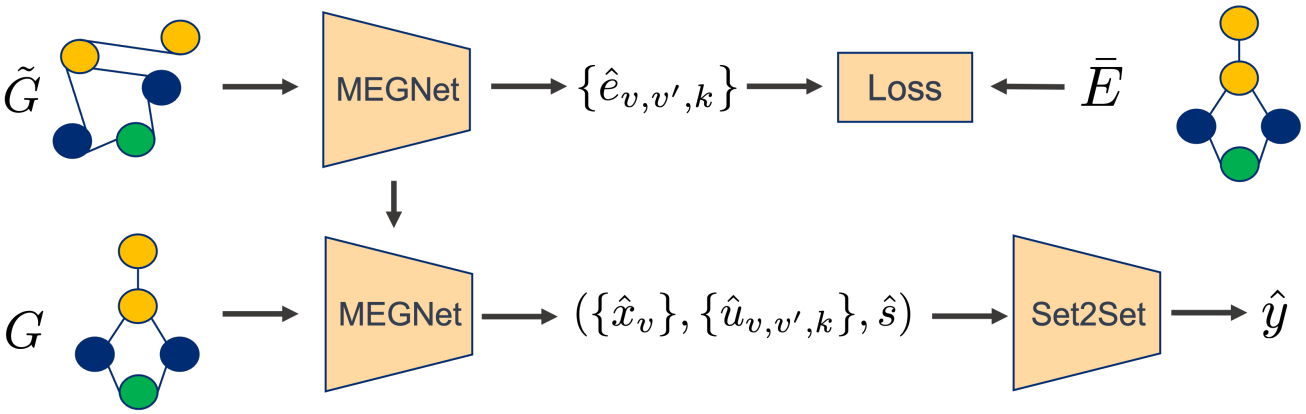

Figure 2. We summarize the application of our CDSSL framework to a property prediction task. In the top row, a MEGNet is trained to denoise crystal structures (Figure 1 and eq. 1) with predicted edge embeddings $\hat{e}_{v,v',k}$. Once the MEGNet module has been trained, we can finetune it on property prediction. This entails passing the node-level outputs $\hat{x}_v$, edge-level outputs $\hat{u}_{v,v',k}$, and graph-level output $\hat{s}$ through Set2Set (Vinyals et al., 2016) modules to output the predicted property $\hat{y}$.

task can be solved over the course of training and demonstrates no evidence of overfitting. However, it has some instability in later epochs. Thus, when finetuning CDSSL models on SL tasks, we identify the checkpoint that attained minimal pretraining loss and use that as the initial model.

Figure 4 shows the results of our study. In particular, MEGNets finetuned after pretraining with CDSSL achieve lower evaluation error than MEGNets trained only with SL in 37 out of the 49 (dataset, dataset size) configurations. This is an improvement in error across a wide variety of material classes, dataset sizes, and material property types. This suggests that a model pretrained with CDSSL could be the basis for general-purpose material property prediction.

3.3. Assessing the CDSSL representation space

Our hypothesis for why CDSSL works is that the pretext task (eq. 1) enables h_θ to learn a general representation of materials space. To test this hypothesis, we assess the quality of the CDSSL's learned representation for prediction tasks without additional fine-tuning. In particular, we choose 4,096 points from the validation split of matbench_mp_e_form and calculate their CDSSL embeddings:

$$\hat{z} \quad = \quad \text{Concat}\left(\text{mean}_v\{\hat{x}_v\}, \text{mean}_{v,v',k}\{\hat{u}_{v,v',k}\}, \hat{s}\right) \quad (6)$$

where $\hat{x}$, $\hat{u}_{v,v',k}$, and $\hat{s}$ are the outputs of the MEGNet graph convolution module, as in Section 2.3.

We can estimate how informative the CDSSL representation space is for material properties by using a linear probing strategy (Balestriero et al., 2023). We use 80% of these points to train ridge regressors to predict the log-transformed structure densities and volume, as calculated by `pymatgen` (Ong et al., 2013). We find that the density-prediction regressor attains an R^2 of 70.3%, and the volume-prediction regressor attains an R^2 of 75.6%.

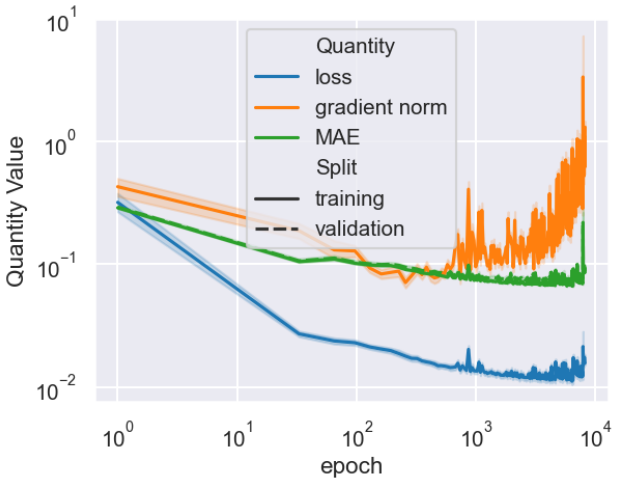

Figure 3. We show metrics from pretraining a MEGNet with the CDSSL pretraining objective. Training yields slow but consistent decreases in both the training loss (eq. 1) and the MAE of the $h_\theta(\tilde{G}) - \bar{E}$ quantity (for both the training and validation set). MAEs for the training and validation set overlap, indicating that overfitting is not happening. The CDSSL pretraining task retains instabilities during training, as evidenced by the jump in metrics and gradient norm of the loss at the end of training.

These results show that, even without the additional expressivity granted by MEGNet's Set2Set modules, the CDSSL pretraining task enables models to learn representations of materials space. In Figure 5, we present additional evidence for this claim, where we use uniform manifold approximation and projection (UMAP) (McInnes et al., 2020) to visually show that the CDSSL space captures variation in material density.

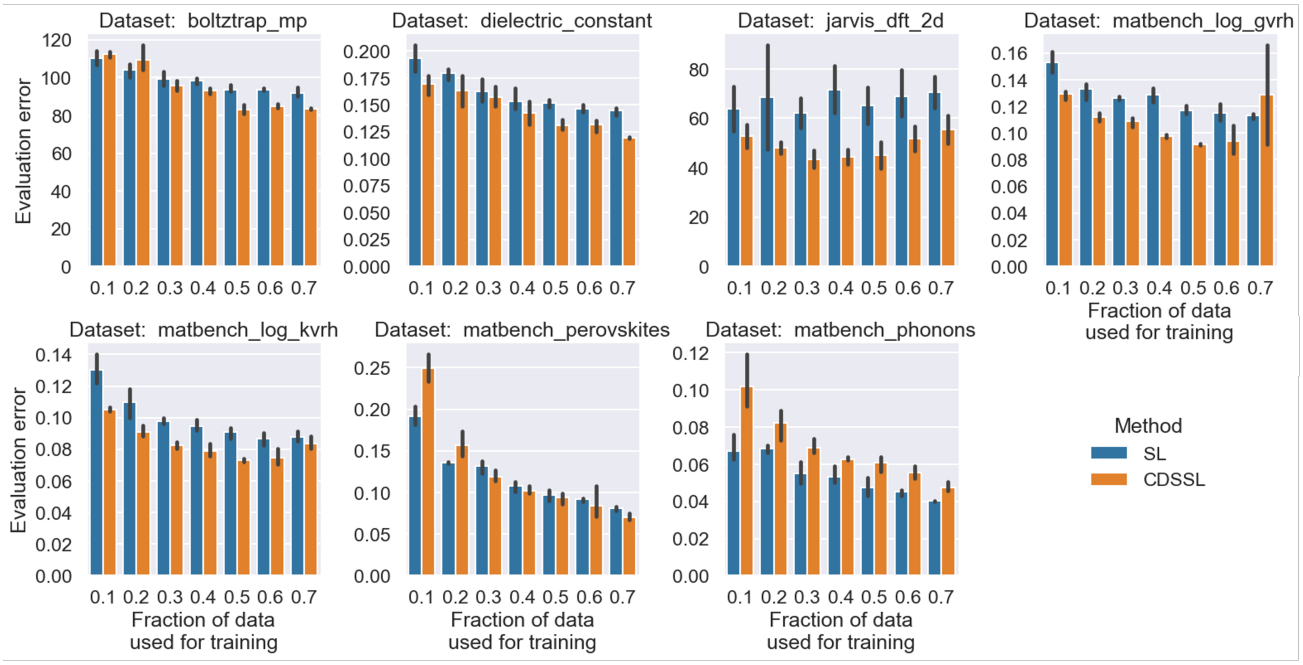

Figure 4. We demonstrate the effects of using CDSSL vs. SL across a variety of datasets and dataset sizes. Each bar reports error on the evaluation set, averaged over 3 data splits and network initializations, and error bars show standard errors in estimating that mean accuracy. The model finetuned after CDSSL has a lower error than the SL model in 37 out of 49 (dataset, dataset size) configurations.

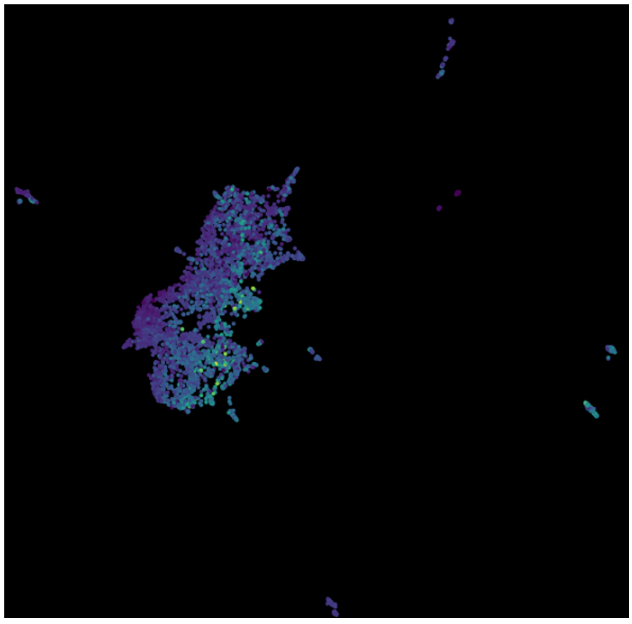

Figure 5. We use UMAP (McInnes et al., 2020) to learn a reduced representation of the matbench_mp_e_form dataset used for pretraining with CDSSL (eq. 6). We shade points by their corresponding structure's density. Within the reduced representation, structures with similar densities are near each other. This suggests that the representation space learned via CDSSL has captured general notions of material properties. Error metrics are reported in the unit of each dataset's property.

4. Discussion

In this work, we have introduced a novel method, CDSSL, for pretraining material property-prediction models. Our method works by taking a crystal structure, perturbing the positions of its constituent atoms with noise, and then tasking the predictive model to recover the structure's original edge embeddings. This enables the predictive model to learn a general, property-agnostic representation of material structure space. CDSSL is generally applicable to structure-based property prediction models, but here we focused on the MEGNet (Chen et al., 2019) architecture.

We showed that using CDSSL for pretraining MEGNets yielded an increase in accuracy across a variety of datasets and material properties, compared to a MEGNet trained only with SL. However, we believe further work can enhance the effectiveness of CDSSL. In particular, modification of the CDSSL training loss might make its minimization process more stable. Such modification might come from the development of theory that can rigorously link the denoising task to statistical mechanics.

Acknowledgements

This work was supported by internal research and development funding from the Research and Exploratory Development Mission Area of the Johns Hopkins University Applied Physics Laboratory.

References

Balestriero, R., Ibrahim, M., Sobal, V., Morcos, A., Shekhar, S., Goldstein, T., Bordes, F., Bardes, A., Mialon, G., Tian, Y., Schwarzschild, A., Wilson, A. G., Geiping, J., Garrido, Q., Fernandez, P., Bar, A., Pirsiavash, H., LeCun, Y., and Goldblum, M. A cookbook of self-supervised learning, 2023.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., et al. Relational inductive biases, deep learning, and graph networks, 2018. URL https://arxiv.org/abs/1806.01261.

Chen, C. and Ong, S. P. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, Nov 2022. ISSN 2662-8457. doi: 10.1038/s43588-022-00349-3. URL https://doi.org/10.1038/s43588-022-00349-3.

Chen, C., Ye, W., Zuo, Y., Zheng, C., and Ong, S. P. Graph networks as a universal machine learning framework for molecules and crystals. *Chemistry of Materials*, 2019. doi: 10.1021/acs.chemmater.9b01294. URL https://doi.org/10.1021/acs.chemmater.9b01294.

Choudhary, K. and DeCost, B. Atomistic line graph neural network for improved materials property predictions. *npj Computational Materials*, 7(1):185, Nov 2021. ISSN 2057-3960. doi: 10.1038/s41524-021-00650-1. URL https://doi.org/10.1038/s41524-021-00650-1.

Deng, B., Zhong, P., Jun, K., Riebesell, J., Han, K., Bartel, C. J., and Ceder, G. Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, pp. 1–11, 2023. doi: 10.1038/s42256-023-00716-3.

Dunn, A., Wang, Q., Ganose, A., Dopp, D., and Jain, A. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Computational Materials*, 6(1):138, Sep 2020. ISSN 2057-3960. doi: 10.1038/s41524-020-00406-3. URL https://doi.org/10.1038/s41524-020-00406-3.

Godwin, J., Schaarschmidt, M., Gaunt, A. L., Sanchez-Gonzalez, A., Rubanova, Y., Veličković, P., Kirkpatrick, J., and Battaglia, P. Simple GNN regularisation for 3d molecular property prediction and beyond. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1wVvweK3oIb.

Goodall, R. E. A. and Lee, A. A. Predicting materials properties without crystal structure: deep representation learning from stoichiometry. *Nature Communications*, 11(1):6280, Dec 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-19964-7. URL https://doi.org/10.1038/s41467-020-19964-7.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., and Leskovec, J. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJlWWJSFDH.

Huang, H., Magar, R., and Barati Farimani, A. Pretraining strategies for structure agnostic material property prediction. *Journal of Chemical Information and Modeling*, 64(3):627–637, 2024. doi: 10.1021/acs.jcim.3c00919. URL https://doi.org/10.1021/acs.jcim.3c00919. PMID: 38301621.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

Kirklin, S., Saal, J. E., Meredig, B., Thompson, A., Doak, J. W., Aykol, M., Rühl, S., and Wolverton, C. The Open Quantum Materials Database (OQMD): assessing the accuracy of DFT formation energies. *npj Computational Materials*, 1(1):15010, Dec 2015. ISSN 2057-3960. doi: 10.1038/npjcompumats.2015.10. URL https://doi.org/10.1038/npjcompumats.2015.10.

Ko, T. W., Nassar, M., Miret, S., Liu, E., Qi, J., and Ong, S. P. Materials graph library, 2021.

Magar, R., Wang, Y., and Barati Farimani, A. Crystal twins: self-supervised learning for crystalline material property prediction. *npj Computational Materials*, 8(1):231, Nov 2022. ISSN 2057-3960. doi: 10.1038/s41524-022-00921-5. URL https://doi.org/10.1038/s41524-022-00921-5.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

New, A., Pekala, M. J., Le, N. Q., Domenico, J., Piatko, C. D., and Stiles, C. D. Curvature-informed multi-task learning for graph networks. In *ICML 2022 2nd AI for Science Workshop*, 2022. URL https://openreview.net/forum?id=m5RYtApKFOg.

New, A., Pekala, M., Pogue, E. A., Le, N. Q., Domenico, J., Piatko, C. D., and Stiles, C. D. Evaluating the diversity and utility of materials proposed by generative models. In *1st Workshop on the Synergy of Scientific and Machine Learning Modeling @ ICML2023*, 2023. URL https://openreview.net/forum?id=2ZYbmYTKoR.

Ong, S. P., Richards, W. D., Jain, A., Hautier, G., Kocher, M., Cholia, S., Gunter, D., Chevrier, V. L., Persson, K. A., and Ceder, G. Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science*, 68:314–319, 2013. ISSN 0927-0256. doi: https://doi.org/10.1016/j.commatsci.2012.10.028. URL https://www.sciencedirect.com/science/article/pii/S0927025612006295.

Pogue, E. A., New, A., McElroy, K., Le, N. Q., Pekala, M. J., McCue, I., Gienger, E., Domenico, J., Hedrick, E., McQueen, T. M., Wilfong, B., Piatko, C. D., Ratto, C. R., Lennon, A., Chung, C., Montalbano, T., Bassen, G., and Stiles, C. D. Closed-loop superconducting materials discovery. *npj Computational Materials*, 9 (1):181, Oct 2023. ISSN 2057-3960. doi: 10.1038/s41524-023-01131-3. URL https://doi.org/10.1038/s41524-023-01131-3.

Ruff, R., Reiser, P., Stühmer, J., and Friederich, P. Connectivity optimized nested line graph networks for crystal structures. *Digital Discovery*, 3:594–601, 2024. doi: 10.1039/D4DD00018H. URL http://dx.doi.org/10.1039/D4DD00018H.

Scheidgen, M., Himanen, L., Ladines, A. N., Sikter, D., Nakhaee, M., Ádám Fekete, Chang, T., Golparvar, A., Márquez, J. A., Brockhauser, S., Brückner, S., Ghiringhelli, L. M., Dietrich, F., Lehmberg, D., Denell, T., Albino, A., Näsström, H., Shabih, S., Dobener, F., Kühbach, M., Mozumder, R., Rudzinski, J. F., Daelman, N., Pizarro, J. M., Kuban, M., Salazar, C., Ondračka, P., Bungartz, H.-J., and Draxl, C. Nomad: A distributed web-based platform for managing materials science research data. *Journal of Open Source Software*, 8(90):5388, 2023. doi: 10.21105/joss.05388. URL https://doi.org/10.21105/joss.05388.

Vinyals, O., Bengio, S., and Kudlur, M. Order matters: Sequence to sequence for sets, 2016.

Wang, A. Y.-T., Kauwe, S. K., Murdock, R. J., and Sparks, T. D. Compositionally restricted attention-based network for materials property predictions. *npj Computational Materials*, 7(1):77, May 2021. ISSN 2057-3960. doi: 10.1038/s41524-021-00545-1. URL https://doi.org/10.1038/s41524-021-00545-1.

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J., and Zhang, Z. Deep graph library: A graph-centric, highly-performant package for graph neural networks, 2020.

Ward, L., Agrawal, A., Choudhary, A., and Wolverton, C. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Computational Materials*, 2(1):16028, Aug 2016. ISSN 2057-3960. doi: 10.1038/npjcompumats.2016.28. URL https://doi.org/10.1038/npjcompumats.2016.28.

Ward, L., Dunn, A., Faghaninia, A., Zimmermann, N. E., Bajaj, S., Wang, Q., Montoya, J., Chen, J., Bystrom, K., Dylla, M., Chard, K., Asta, M., Persson, K. A., Snyder, G. J., Foster, I., and Jain, A. Matminer: An open source toolkit for materials data mining. *Computational Materials Science*, 152:60–69, 2018. ISSN 0927-0256. doi: https://doi.org/10.1016/j.commatsci.2018.05.018. URL https://www.sciencedirect.com/science/article/pii/S0927025618303252.

Xie, T. and Grossman, J. C. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.*, 120:145301, Apr 2018. doi: 10.1103/PhysRevLett.120.145301.

Xie, T., Fu, X., Ganea, O.-E., Barzilay, R., and Jaakkola, T. S. Crystal diffusion variational autoencoder for periodic material generation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=03RLpj-tc_.

Xie, Y., Xu, Z., Zhang, J., Wang, Z., and Ji, S. Self-supervised learning of graph neural networks: A unified review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2412–2429, 2023. doi: 10.1109/TPAMI.2022.3170559.

Zaidi, S., Schaarschmidt, M., Martens, J., Kim, H., Teh, Y. W., Sanchez-Gonzalez, A., Battaglia, P., Pascanu, R., and Godwin, J. Pre-training via denoising for molecular property prediction. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tYIMtogyee.

Zeni, C., Pinsler, R., Zügner, D., Fowler, A., Horton, M., Fu, X., Shysheya, S., Crabbé, J., Sun, L., Smith, J., Nguyen, B., Schulz, H., Lewis, S., Huang, C.-W., Lu, Z., Zhou, Y., Yang, H., Hao, H., Li, J., Tomioka, R., and Xie, T. Mattergen: a generative model for inorganic materials design, 2024.

Zhao, Y., Siriwardane, E. M. D., Wu, Z., Fu, N., Al-Fahdi, M., Hu, M., and Hu, J. Physics guided deep learning for generative design of crystal materials with symmetry constraints. *npj Computational Materials*, 9 (1):38, Mar 2023. ISSN 2057-3960. doi: 10.1038/s41524-023-00987-9. URL https://doi.org/10.1038/s41524-023-00987-9.

A. Supplemental data

Hyperparameter	Value
Perturbation scale σ	0.5
hidden_ layer_ sizes_input	$(128, 64)$
hidden_layer_sizes_conv	$(128, 128, 64)$
hidden_layer_sizes_output	$(64, 32)$
dim_node_embedding	16
dim_edge_embedding	100
dim_state_embedding	2
n_blocks	3
nlayers_set2set	1
niters_set2set	2
Optimizer	Adam
Activation	SoftPlus2
Minibatch size	256
Number of epochs	4096

Table 2. Hyperparameters used when pretraining models. See the MatGL (Ko et al., 2021) documentation for details on what MEGNet-specific hyperparameters mean.

Hyperparameter	Value
hidden_ layer_ sizes_input	$(64, 32)$
hidden_layer_sizes_conv	$(64, 64, 32)$
hidden_layer_sizes_output	$(32, 16)$
dim_node_embedding	16
dim_edge_embedding	100
dim_state_embedding	2
n_blocks	3
nlayers_set2set	1
niters_set2set	2

Table 3. Hyperparameters used for the MEGNet model trained from scratch. See the MatGL (Ko et al., 2021) documentation for details on what MEGNet-specific hyperparameters mean.

Hyperparameter	Value
Optimizer	Adam
Activation	SoftPlus2
Number of epochs	96
Learning rate	$1e-3$
Minibatch size	128

Table 4. Hyperparameters used both by the MEGNet model trained from scratch and the model finetuned after pretraining.

Parameter	Setting
Cutoff distance for edge construction	5
# of centers in Gaussian radial expansion	100
Width of Gaussian functions	0.5

Table 5. Parameters used when constructing multigraph representations of material structures.

Dataset	Property	Property description
boltztrap_mp	s_n	n-type Seebeck coefficient
dielectric_constant	poly_total	average of eigenvalues of total contributions to the dielectric tensor
jarvis_dft_2d	exfoliation_en	exfoliation energy
matbench_log_grvh	g_vrh	Voigt-Reuss-Hill average shear modulus
matbench_log_kvrh	k_vrh	Voigt-Reuss-Hill average bulk modulus
matbench_perovskites	e_form	Heat of formation
matbench_phonons	last_phdos_peak	Frequency of the highest frequency optical phonon mode peak

Table 6. The target property for each dataset used in our studies. More details are available at MatMiner (Ward et al., 2018) (https://hackingmaterials.lbl.gov/matminer/dataset_summary.html).