Robust Self-training Strategy for Various Molecular Biology Prediction Tasks*

Hehuan Ma
University of Texas at Arlington
Arlington, Texas, United States
hehuan.ma@mavs.uta.edu

Feng Jiang
University of Texas at Arlington
Arlington, Texas, United States
fxj8843@mavs.uta.edu

Yu Rong
Tencent AI Lab
Shenzhen, China
yu.rong@hotmail.com

Yuzhi Guo
University of Texas at Arlington
Arlington, Texas, United States
yuzhi.guo@mavs.uta.edu

Junzhou Huang†
University of Texas at Arlington
Arlington, Texas, United States
jzhuang@uta.edu

ABSTRACT

Molecular biology prediction tasks suffer the limited labeled data problem since it normally demands a series of professional experiments to label the target molecule. Self-training is one of the semi-supervised learning paradigms that utilizes both labeled and unlabeled data. It trains a teacher model on labeled data, and uses it to generate pseudo labels for unlabeled data. The labeled and pseudo-labeled data are then combined to train a student model. However, the pseudo labels generated from the teacher model are not sufficiently accurate. Thus, we propose a robust self-training strategy by exploring robust loss function to handle such noisy labels, which is model and task agnostic, and can be easily embedded with any prediction tasks. We have conducted molecular biology prediction tasks to gradually evaluate the performance of proposed robust self-training strategy. The results demonstrate that the proposed method consistently boosts the prediction performance, especially for molecular regression tasks, which have gained a 41.5% average improvement.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; **Supervised learning**; **Unsupervised learning**; **Semi-supervised learning settings**; • **Applied computing** → **Computational biology**; **Molecular structural biology**; **Molecular sequence analysis**.

KEYWORDS

molecular biology, prediction tasks, neural network, self-training, semi-supervised learning, bioinformatics

ACM Reference Format:

Hehuan Ma, Feng Jiang, Yu Rong, Yuzhi Guo, and Junzhou Huang. 2022. Robust Self-training Strategy for Various Molecular Biology Prediction Tasks.

*This work was partially supported by the NSF CAREER grant IIS-1553687 and Cancer Prevention and Research Institute of Texas (CPRIT) award (RP190107).

†Corresponding author.

BCB '22, August 7–10, 2022, Northbrook, IL, USA

ACM ISBN 978-1-4503-9386-7/22/08.
https://doi.org/10.1145/3535508.3545998

In *13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '22), August 7–10, 2022, Northbrook, IL, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3535508.3545998

1 INTRODUCTION

Molecular biology prediction is one crucial and fundamental task for bioinformatics areas such as drug discovery [3, 19, 30]. It includes various molecule-relevant tasks, such as molecular property prediction and protein secondary or tertiary structure prediction. With the development of deep learning techniques, more and more research tackle these tasks with various deep learning models [7–9, 12, 15, 25, 27, 29, 31]. The prediction task is well-known as a supervised problem, which takes the labeled data as input and employs computational models to predict the corresponding labels. Many existing studies target at such problems in this manner [4, 6, 16, 17, 24]. However, one of the ongoing problems in molecular biology is that labeled data is limited and also difficult to obtain. It usually requires a series of professional experiments, which is time-consuming and costly. Therefore, more paradigms have been developed to utilize unlabeled data to help promote supervised learning, such as semi-supervised learning [11, 21, 33]. Within this field, a simple yet effective paradigm that exploits both unlabeled and labeled data, called self-training, is rarely explored for molecular biology prediction tasks.

In general, self-training is established in four steps: 1) a teacher model is trained on labeled data; 2) the trained teacher model is employed to generate pseudo labels for unlabeled data; 3) the labeled data and the pseudo-labeled data are combined to train a student model; 4) the student model then becomes the teacher model to repeat steps 2-3 until the training is converged. In this fashion, more data is included in the training process, and the student model is able to inherit from the teacher. This paradigm is easy to implement and powerful to boost the training process. Self-training has been widely used in other areas and obtained promising performance, e.g., Computer Vision (CV) [1, 28, 34], and Nature Language Processing (NLP) [2, 10, 14]. One primary reason is that not only the unlabeled data is enormous, the size of labeled data is also quite large, so are the training models. Thus, the teacher model can sufficiently learn from the labeled data, and achieve favorable performance. Then the student is able to learn better. However, for most molecular biology prediction tasks, the size of the labeled dataset is only a few thousands, and the corresponding prediction performance is not as

high as image classification whose accuracy may achieve 95%. Such scenarios lead to a problem: the generated pseudo labels may not be accurate. Such noisy labels may further bias student learning. Therefore, how to handle the label noise is the major concern when establishing self-training strategy in molecular biology area.

One straightforward way to encourage the model to learn from the noisy labels, is to design a loss with regularization to leverage the neural network learning. Mean absolute error (MAE) and cross-entropy (CE) loss are two commonly used loss functions in prediction tasks, where the former is utilized in regression tasks and the latter is used for classification. MAE has been theoretically proved to be robust to label noise during the training, while the CE is not [5]. Recently, robust loss functions have been studied to tackle the noisy label problem in classification tasks by generalizing MAE and CE, and have achieved impressive performance when solving the image classification problem [5, 18, 26, 32].

In this paper, we propose to integrate robust loss function and self-training to form a robust self-training framework for molecular biology prediction tasks. Extensive experiments have been conducted over molecular regression and classification tasks to gradually evaluate the effectiveness of proposed robust self-training strategy. Our contributions can be summarized as 1) we are the first to propose a robust self-training paradigm that utilizes robust loss to constrain the student training; 2) the proposed framework is straightforward, and easy to fit into any prediction tasks, which is a simple yet practical strategy to promote the molecular biology prediction tasks; 3) extensive experiments on molecular biology prediction tasks demonstrate that self-training can improve the prediction performance by involving more unlabeled data, and the robust loss can further boost the performance by leveraging the label noise.

2 METHODS

2.1 Problem Definition

Molecular biology prediction problems can be further referred to as regression problems or classification problems. Given a molecule $\mathcal{M}$, the label needs to be predicted is denoted as y, where $y \in \mathbb{R}$ for a regression problem, and $y \in \{0, 1, \ldots, K-1\}$ for a K-class classification problem. The input molecule $\mathcal{M}$, can be any format according to the task specifics, e.g., protein sequence for protein secondary structure prediction, or molecular graph structure for molecular property prediction. In this study, we conduct two types of experiments to gradually demonstrate the effectiveness of proposed robust self-training strategy: molecular property regression, and molecular property classification.

2.2 Robust Self-training Overview

Our proposed robust self-training strategy is implemented on top of the self-training framework. Figure 1 illustrates the overall architecture, which can be viewed as two parts, train teacher and train student. First of all, a teacher model is trained on the labeled dataset $\mathcal{D}_l$, and a trained teacher model T is obtained. After that, the student training process begins. It starts with generating the pseudo labels for the unlabeled dataset $\mathcal{D}_u$ to construct a pseudo-labeled dataset $\mathcal{D}_p$. Then the student model is initialized with the teacher model, and trained on shuffled $\mathcal{D}_l + \mathcal{D}_p$. After training for several epochs, we consider that as one iteration, the best model during $i\text{-}th$ iteration is selected as the best student model S_i, then regard it as the new teacher model to repeat the previous steps. This process is repeated for i iterations until the student model is converged.

2.3 Molecular Biology Prediction Tasks

Molecular biology prediction task can be considered as two parts in the view of deep learning, which are molecular encoder model and prediction model. Molecular encoder model generates a vector that represents the input molecule, and the prediction model takes the vector to make a prediction. The input molecule can be represented as any format, e.g., sequence or graph structure. We take molecular property regression and classification tasks as examples to evaluate the performance of our proposed strategy. It is noteworthy that any prediction tasks can be adapted with proposed robust self-training since our method generates pseudo labels by training teacher model from the labeled data, such as protein secondary and tertiary structure prediction [8, 13].

In our experiments, we utilize the molecular graph structure and employ two representative graph-based models, EGNN [23] and GIN [29], as the backbone models to predict molecular regression and classification properties. We give a universal definition here for the graph-based encoder and the prediction model.

Molecule $\mathcal{M}$ can be naturally represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $|\mathcal{V}| = p$ refers to the set of p atoms and $|\mathcal{E}| = q$ refers to a set of q bonds in the molecule. The features of atom v is referred as $\mathbf{a}_v \in \mathbb{R}^{d_a}$, and the features of bond (v, u) is referred as $\mathbf{b}_{vu} \in \mathbb{R}^{d_b}$, where $\mathbb{R}^{d_a}$ and $\mathbb{R}^{d_b}$ represent the feature dimension of atom and bond respectively. $\mathcal{N}(v)$ represents the neighbor atoms of atom v, which is identified by the connected bonds. GNN-based models generally perform a message passing and state update protocol for updating atom/bond features. Then, the states of all the atoms are captured to generate a vector representation $\mathbf{h}_{\mathcal{G}}$ through a readout mechanism.

After going through the graph encoder model, the graph representation $\mathbf{h}_{\mathcal{G}}$ is then fed into the prediction model to make a prediction of the property. The prediction model is generally a simple neural network such as multi-layer perceptron (MLP): $\hat{y} = MLP(\mathbf{h}_{\mathcal{G}})$, where $\hat{y}$ is the output of the prediction model, which refers to the predicted probability for the classification tasks or the actual predicted property value for the regression tasks.

Next, each backbone model is introduced along with the employed robust loss respectively.

2.3.1 Regression task. As we have mentioned earlier, MAE has been proved to be robust for label noise. Therefore, we first conduct experiments on molecular regression tasks with MAE as the loss function. EGNN [23] is one most recent work to address such problems. Other than the commonly used message passing process based on the graph structure and features, EGNN further explores the geometric information by considering the atom coordinates $\mathbf{x}^d = \left\{\mathbf{x}_0^d, \ldots, \mathbf{x}_{p-1}^d\right\}$. The message update for layer d is defined as:

$$\mathbf{m}_{vu}^d = \phi_e\left(\mathbf{h}_v^d, \mathbf{h}_u^d, \left\|\mathbf{x}_v^d - \mathbf{x}_u^d\right\|^2, e_{vu}\right), \tag{1}$$

$$\mathbf{x}_v^{d+1} = \mathbf{x}_v^d + C\sum_{u \neq v}\left(\mathbf{x}_v^d - \mathbf{x}_u^d\right)\phi_x\left(\mathbf{m}_{vu}^d\right), \tag{2}$$

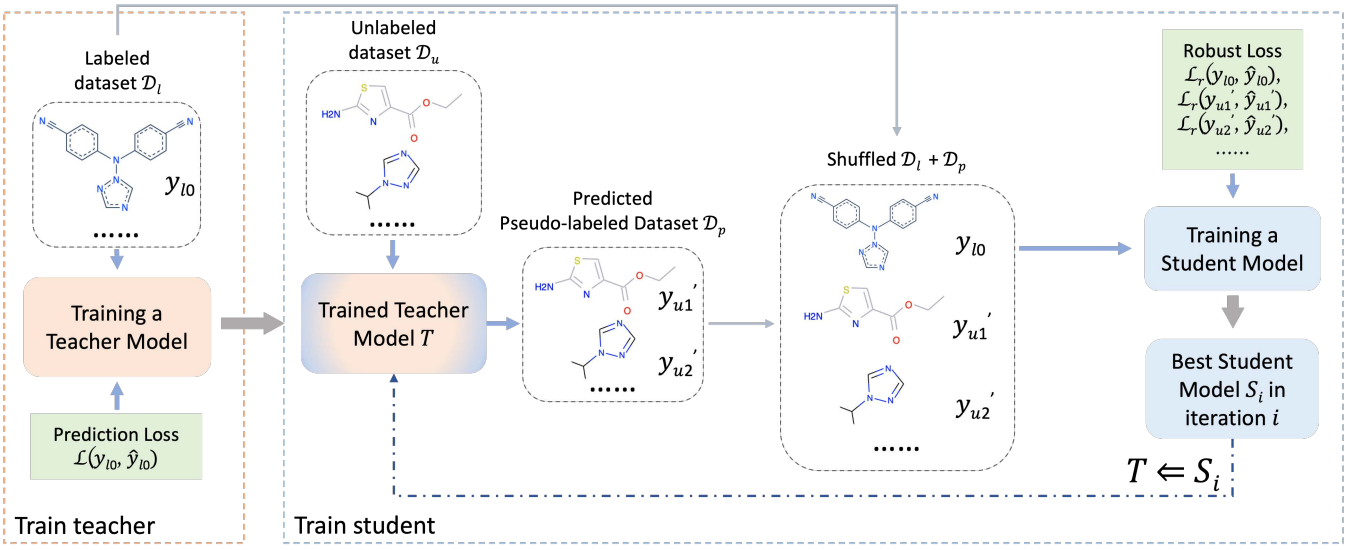

Figure 1: A overview illustration of the robust self-training architecture. More details are described in Section 2.2.

$$\mathbf{h}_{\mathcal{N}(v)}^{d+1} = \text{AGGREGATE}\left(\left\{\mathbf{m}_{vu}^{d}, \forall u \in \mathcal{N}(v)\right\}\right), \tag{3}$$

where x_v^d and x_u^d are the coordinates of atom v and its neighbor atom u at d-th step, vu represents the bond between them, e_{vu} denotes the bond features, ϕ_e and ϕ_x are two output operations, and C equals $1/(p-1)$.

MAE is used as the robust loss function to constrain the network training with regards to noisy labels, which is defined as:

$$\mathcal{L}_{MAE} = \frac{1}{M}\sum_{i=1}^{M}|y_i - \hat{y}_i|, \tag{4}$$

where M is the size of the dataset.

2.3.2 Classification task. Classification tasks are dominating for molecular biology prediction problems as well. We then conduct experiments over molecular property classification tasks to evaluate the effectiveness of the self-training paradigm. However, the commonly used cross-entropy (CE) loss is not robust, so we employ the generalized cross-entropy (GCE) loss [32] to boost the self-training. The backbone model utilized for this task is GIN [29]. GIN is theoretically proved as one of the most powerful GNN models. It utilizes multi-layer perceptron (MLP) for state update, and employs a concatenate operation over all passing steps during the readout phase. The updated rule can be summarized as:

$$\mathbf{h}_v^{d+1} = \text{MLP}^{d+1}\left(\left(1+\epsilon^{d+1}\right)\cdot\mathbf{h}_v^d + \sum_{u\in\mathcal{N}(v)}\mathbf{h}_u^d\right), \tag{5}$$

$$\mathbf{h}_{\mathcal{G}} = \text{CONCAT}\left(\text{READOUT}\left(\left\{\mathbf{h}_v^{d+1} \mid v \in \mathcal{V}\right\}\right)\right), \tag{6}$$

where ϵ is a fixed scalar or a learnable parameter.

GCE loss is a generalized version of CE and MAE. The CE loss is defined by:

$$\mathcal{L}_{CE} = -\sum_{k=1}^{K} y^k \log \hat{y}^k, \tag{7}$$

for a K-class classification problem (K=2 for binary classification), where y^k is the one-hot encoding label, and $\hat{y}^k$ denotes the probability output from the prediction network. Allow $f_k(x) = \hat{y}^k$, GCE loss is designed by:

$$\mathcal{L}_{GCE} = \frac{(1-f_k(x)^q)}{q}, \text{ where } q \in (0,1]. \tag{8}$$

GCE loss is reduced to CE loss and MAE loss when $q \to 0$ and $q = 1$, respectively. Detailed proofs can be found in [32].

3 EXPERIMENTS

Extensive experiments are conducted gradually to evaluate the performance of proposed robust self-training strategy. Since MAE is theoretically proved to be robust to label noise, we first implement self-training on molecular regression task, and utilize MAE loss as the robust loss function to demonstrate the superiority of proposed method. Then we explore GCE loss on the molecular classification task to further confirm the effectiveness of integrating robust loss function with self-training.

3.1 Datasets Description and Setup

QM9 [20] is a standard benchmark for molecular property regression problem. It is a subset of GDB-17 database [22], which contains 134k molecules. It comprehensively provides 12 quantum chemical properties for each molecule, including geometric, energetic, electronic, thermodynamic, etc. **HIV** is introduced by the Drug Therapeutics Program (DTP) AIDS Antiviral Screen. It contains the test result of 41,127 molecule compounds with the ability for inhibiting HIV replication. The widely used version provided by MoleculeNet contains inactive labels and activa labels, which makes it a binary classification task [27].

For all tasks, we randomly select 50% of the data as the unlabeled dataset, and the rest is used as the the labeled dataset with a 3:1:1 training/validation/test ratio. We do not use an external unlabeled

Table 1: Mean Absolute Error (MAE) for each molecular property regression benchmark on QM9 dataset. Lower is better, best score is marked in bold, and green color indicates our proposed method. The last two rows illustrate the improvement percentage by our method compared with others. Avg demonstrates the average score over the row, which denotes the MAE average for all 12 tasks for the first three rows, and the average improvement for the last two rows. Details about each property can be found in [20].

Task	α	$\Delta\varepsilon$	$\varepsilon_{\text{HOMO}}$	$\varepsilon_{\text{LUMO}}$	μ	C_ν	G	H	R^2	U	U_0	ZPVE	Avg
Unit	bohr^3	meV	meV	meV	D	cal/mol K	meV	meV	bohr^3	meV	meV	meV	
EGNN-labeled	0.118	0.070	0.044	0.041	0.057	0.044	0.020	0.021	0.151	0.020	0.019	2.111	0.226
EGNN-self-training (Ours)	**0.067**	**0.048**	**0.028**	**0.025**	**0.028**	**0.031**	**0.010**	**0.010**	**0.083**	**0.011**	**0.010**	**1.521**	**0.156**
EGNN-all	0.071	**0.048**	0.029	**0.025**	0.029	**0.031**	0.012	0.012	0.106	0.012	0.011	1.55	0.161
Ours v.s. EGNN-labeled ↑	+43.2%	+31.6%	+36.1%	+38.4%	+50.6%	+30.0%	+50.7%	+52.8%	+45.2%	+43.6%	+49.2%	+28.0%	+41.5%
Ours v.s. EGNN-all ↑	+5.6%	0.00%	+3.5%	+0.00%	+3.5%	+0.00%	+16.7%	+16.7%	+21.7%	+8.3%	+9.1%	+1.9%	+7.2%

Note that for molecular regression task, the commonly used MAE is provably robust, so "EGNN-self-training" represents our work.
The scores of EGNN-all are obtained from the original EGNN paper [23].

dataset here since most molecules may not express target property at all, which may lead to a biased comparison.

3.2 Experimental Details

3.2.1 Baselines. For all the experiments, we consider training solely on the labeled datasets as the fundamental baselines, which is denoted as "-labeled". Then, we establish our vanilla implementation by running experiments with self-training paradigm on both labeled dataset and unlabeled dataset, denoted as "-self-training". Last, we integrate robust loss with our vanilla self-training benchmark to demonstrate the superiority of our robust self-training, denoted as "-robust". Since the unlabeled dataset is formed by randomly selecting 50% from the original labeled dataset, we also compare the performance when using the original backbone model without self-training on all the data with labels, denoted as "-all".

3.2.2 Configurations. We follow the original implementation and settings of the backbone models, and implement robust self-training on top of them. All the hyper-parameters of the backbone models remain the same to ensure a fair comparison. For the settings of robust self-training, we perform three iterations for the student training, and tune the hyper-parameter q when employing GCE loss. For molecular classification task, we run the experiments three times to alleviate the randomness since HIV dataset is much smaller than other datasets, leading to relatively unstable performance. We take the average and standard deviation of the evaluation scores as the final results. For molecular regression task, we follow the original configurations and evaluations to run the experiments one time. The results do not vary much since the training data is sufficiently large and the converged stage is stable.

3.2.3 Training strategy. We follow the same procedure for all three tasks. First, we train a teacher model on the labeled data, and use it to generate pseudo labels for the unlabeled dataset. Next, for the vanilla self-training, we train the student model which takes the teacher model as the initialization on the combined labeled and pseudo-labeled dataset. Note that the pseudo-labeled dataset is only merged into the training dataset along with the labeled training dataset. The validation and test datasets remain the same from the teacher model training. Furthermore, we choose the best student model in the current iteration as the new teacher model to generate a new pseudo-labeled dataset and initialize the student model for the next iteration. We run the student training for three iterations, and take the best validation model to evaluate the test dataset performance. For robust self-training, the procedure is the same as vanilla self-training, except robust loss function is employed.

3.3 Experimental Results

Our first experiment is to employ the self-training paradigm directly on molecular regression tasks, since MAE is theoretically robust to noisy labels. The comparison results for each property are shown in Table 1. As we can observe, the performance of the self-training strategy outperforms EGNN-label consistently by a 41.5% average improvement. Moreover, the performance is competitive against the supervised training on the all-labeled dataset. Our implementation achieves the best performance on 9/12 tasks compared with the original EGNN-all on all 134k labeled data, which gains the average MAE boost by 7.2%. The experiments on regression tasks with MAE sufficiently demonstrate that robust loss function is a perfect fit for self-training strategy by dealing with the generated pseudo labels.

Table 2: ROC-AUC score for molecular property classification benchmark on HIV dataset. Higher is better, best score is marked in bold, and green color indicates our proposed method.

	GIN-labeled	GIN-self-training	GIN-robust	GIN-all
HIV	$0.786_{\pm 0.008}$	$0.798_{\pm 0.005}$	$\mathbf{0.822}_{\pm 0.005}$	$0.820_{\pm 0.015}$

We then conduct experiments on the HIV dataset to evaluate how robust self-training performs on classification task. As shown in Table 2, the improvement of directly implementing self-training is limited, which is reasonable since CE loss is not theoretically robust [5]. Therefore, we explore robust loss function GCE and integrate it with self-training to form the robust self-training paradigm, which further boosting the ROC-AUC to 0.822. Moreover, our method is competitive with the original GIN implementation on the all-labeled dataset with a 0.002 improvement. Note that in our self-training experiments, 50% of the dataset is treated as unlabeled, while GIN-all is trained on 100% labeled dataset.

Extensive experiments empirically demonstrate that our proposed robust self-training strategy is capable of efficiently exploring both labeled and unlabeled data as well as handling the noisy pseudo labels. Moreover, we roughly conduct experiments on protein secondary structure prediction task by adopting GCE loss, and has achieved promising results, which further proves the effectiveness of our proposed strategy. Next, we will explore more details regarding the robust loss type with different prediction tasks.

4 CONCLUSION

In this study, we propose a robust self-training paradigm for various molecular biology prediction tasks by exploring robust loss function to constrain the self-training process. We first train a teacher model on labeled dataset, then use the teacher model to generate pseudo labels for the unlabeled dataset. Next, the student model is trained on the combination of the labeled dataset and the pseudo-labeled dataset. This process is iterated by regarding the student as the new teacher and re-generating the pseudo-labeled dataset until the training is eventually converged. Since the pseudo labels are not the ground-truth labels which means noises exist, we then utilize robust loss function to restrain the student training. Extensive experiments have demonstrated that self-training accompanied with robust loss can boost the prediction performance by taking advantage of both labeled and unlabeled data. Moreover, our proposed robust self-training is model and task agnostic, which can be easily inserted into any molecular biology prediction tasks, and benefits the general computational molecular biology society.

REFERENCES

[1] Yauhen Babakhin, Artsiom Sanakoyeu, and Hirotoshi Kitamura. 2019. Semi-supervised segmentation of salt bodies in seismic images using an ensemble of convolutional neural networks. In *German Conference on Pattern Recognition*. Springer, 218–231.

[2] Yong Cheng. 2019. Semi-supervised learning for neural machine translation. In *Joint training for neural machine translation*. Springer, 25–40.

[3] Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. 2018. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface* 15, 141 (2018), 20170387.

[4] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. *arXiv preprint arXiv:1509.09292* (2015).

[5] Aritra Ghosh, Himanshu Kumar, and PS Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[6] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*. PMLR, 1263–1272.

[7] Yuzhi Guo, Jiaxiang Wu, Hehuan Ma, Sheng Wang, and Junzhou Huang. 2020. Bagging msa learning: Enhancing low-quality pssm with deep learning for accurate protein structure property prediction. In *International Conference on Research in Computational Molecular Biology*. Springer, 88–103.

[8] Yuzhi Guo, Jiaxiang Wu, Hehuan Ma, Sheng Wang, and Junzhou Huang. 2020. Protein Ensemble Learning with Atrous Spatial Pyramid Networks for Secondary Structure Prediction. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 17–22.

[9] Yuzhi Guo, Jiaxiang Wu, Hehuan Ma, Sheng Wang, and Junzhou Huang. 2021. EPTool: a new enhancing PSSM tool for protein secondary structure prediction. *Journal of Computational Biology* 28, 4 (2021), 362–364.

[10] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. *arXiv preprint arXiv:1909.13788* (2019).

[11] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. 2019. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265* (2019).

[12] David T Jones. 1999. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology* 292, 2 (1999), 195–202.

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

[14] Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. 2019. Learning to self-train for semi-supervised few-shot classification. *Advances in Neural Information Processing Systems* 32 (2019), 10276–10286.

[15] Hehuan Ma, Yatao Bian, Yu Rong, Wenbing Huang, Tingyang Xu, Weiyang Xie, Geyan Ye, and Junzhou Huang. 2022. Cross-dependent graph neural networks for molecular property prediction. *Bioinformatics* 38, 7 (2022), 2003–2009.

[16] Hehuan Ma, Yu Rong, Boyang Liu, Yuzhi Guo, Chaochao Yan, and Junzhou Huang. 2021. Gradient-Norm Based Attentive Loss for Molecular Property Prediction. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 497–502.

[17] Hehuan Ma, Chaochao Yan, Yuzhi Guo, Sheng Wang, Yuhong Wang, Hongmao Sun, and Junzhou Huang. 2020. Improving molecular property prediction on limited data with deep multi-label learning. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2779–2784.

[18] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. 2020. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*. PMLR, 6543–6553.

[19] Steven M Paul, Daniel S Mytelka, Christopher T Dunwiddie, Charles C Persinger, Bernard H Munos, Stacy R Lindborg, and Aaron L Schacht. 2010. How to improve R&D productivity: the pharmaceutical industry's grand challenge. *Nature reviews Drug discovery* 9, 3 (2010), 203.

[20] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. 2014. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data* 1, 1 (2014), 1–7.

[21] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. 2020. Self-supervised graph transformer on large-scale molecular data. *arXiv preprint arXiv:2007.02835* (2020).

[22] Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. 2012. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *Journal of chemical information and modeling* 52, 11 (2012), 2864–2875.

[23] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. 2021. E (n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844* (2021).

[24] Søren Kaae Sønderby and Ole Winther. 2014. Protein secondary structure prediction with long short term memory networks. *arXiv preprint arXiv:1412.7828* (2014).

[25] Sheng Wang, Yuzhi Guo, Yuhong Wang, Hongmao Sun, and Junzhou Huang. 2019. SMILES-BERT: large scale unsupervised pre-training for molecular property prediction. In *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*. 429–436.

[26] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. 2019. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 322–330.

[27] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science* 9, 2 (2018), 513–530.

[28] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10687–10698.

[29] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[30] Chaochao Yan, Qianggang Ding, Peilin Zhao, Shuangjia Zheng, Jinyu Yang, Yang Yu, and Junzhou Huang. 2020. Retroxpert: Decompose retrosynthesis prediction like a chemist. *Advances in Neural Information Processing Systems* 33 (2020), 11248–11258.

[31] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. 2019. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling* 59, 8 (2019), 3370–3388.

[32] Zhilu Zhang and Mert R Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *32nd Conference on Neural Information Processing Systems (NeurIPS)*.

[33] Nan Zhao, Jing Ginger Han, Chi-Ren Shyu, and Dmitry Korkin. 2014. Determining effects of non-synonymous SNPs on protein-protein interactions using supervised and semi-supervised learning. *PLoS computational biology* 10, 5 (2014), e1003592.

[34] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. *Advances in neural information processing systems* 33 (2020), 3833–3845.