Semi-Supervised Graph Imbalanced Regression

Gang Liu
University of Notre Dame, USA
gliu7@nd.edu

Tong Zhao
Snap Inc., USA
tzhao@snap.com

Eric Inae
University of Notre Dame, USA
einae@nd.edu

Tengfei Luo
University of Notre Dame, USA
tluo@nd.edu

Meng Jiang
University of Notre Dame, USA
mjiang2@nd.edu

ABSTRACT

Data imbalance is easily found in annotated data when the observations of certain continuous label values are difficult to collect for regression tasks. When they come to molecule and polymer property predictions, the annotated graph datasets are often small because labeling them requires expensive equipment and effort. To address the lack of examples of rare label values in graph regression tasks, we propose a semi-supervised framework to progressively balance training data and reduce model bias via self-training. The training data balance is achieved by (1) pseudo-labeling more graphs for under-represented labels with a novel regression confidence measurement and (2) augmenting graph examples in latent space for remaining rare labels after data balancing with pseudo-labels. The former is to identify quality examples from unlabeled data whose labels are confidently predicted and sample a subset of them with a reverse distribution from the imbalanced annotated data. The latter collaborates with the former to target a perfect balance using a novel label-anchored mixup algorithm. We perform experiments in seven regression tasks on graph datasets. Results demonstrate that the proposed framework significantly reduces the error of predicted graph properties, especially in under-represented label areas.

CCS CONCEPTS

• **Computing methodologies** → **Semi-supervised learning settings**; • **Applied computing** → **Chemistry**.

KEYWORDS

Graph Property Regression; Imbalanced Regression; Semi-supervised learning; Self-training

ACM Reference Format:
Gang Liu, Tong Zhao, Eric Inae, Tengfei Luo, and Meng Jiang. 2023. Semi-Supervised Graph Imbalanced Regression. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3580305.3599497

1 INTRODUCTION

Predicting graphs properties has attracted great attention from drug discovery [33, 48] and material design [26, 53], because molecules and polymers are naturally graphs. Properties such as density, melting temperature, and oxygen permeability are often in continuous value spaces [33, 48, 53]. Graph regression tasks are important and challenging. It is hard to observe label values in certain rare areas since the annotated data usually concentrate on small yet popular areas in the property spaces. Graph regression datasets are ubiquitously imbalanced. Previous attempts that address data imbalance mostly focused on categorical properties and classification tasks, however, *imbalanced regression tasks on graphs are under-explored.*

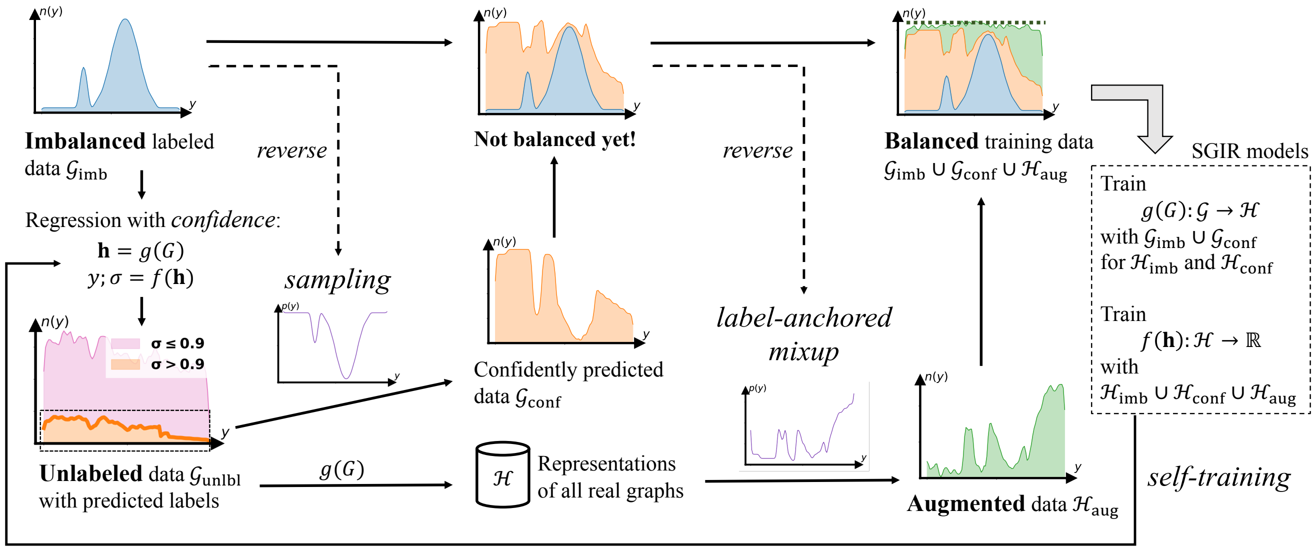
Besides data imbalance, the annotated graph regression data are often small in real world. For example, measuring the property of a molecule or polymer often needs expensive experiments or simulations. It has taken nearly 70 years to collect *only around 600* polymers with experimentally measured oxygen permeability in the Polymer Gas Separation Membrane Database [41]. On the other side, we have *hundreds of thousands* of unlabeled graphs.

Pseudo-labeling unlabeled graphs may enrich and balance training data, however, there are two challenges. First, if one directly trained a model on the imbalanced labeled data and used it to do pseudo-labeling, it would not be reliable to generate accurate and balanced labels. Second, quite a number of unlabeled graphs might not follow the distribution of labeled data. Massive label noise is inevitable in pseudo-labeling. Therefore, selection is necessary to expand the set of data examples for training. Moreover, the selected pseudo-labels without noise cannot alleviate the label imbalance problem. Because the biased model tends to generate more pseudo-labels in the label ranges where most data concentrate. In this situation, the selected pseudo-labels may aggravate the model bias and lead the model to have even worse performance on the label ranges where we lack enough data. Even though the pseudo-labeling had involved quality selection and the unlabeled set had been fully used to address label imbalance, the label distribution of annotated and pseudo-labeled examples might still be far from a perfect balance. This is because there might not be a sufficient number of pseudo-labeled examples to fill the gap in the under-represented label ranges.

Figure 1 illustrates our ideas to overcome the above challenges. First, we want to progressively reduce the model bias by gradually improving training data from the labeled and unlabeled sets. The performance of pseudo-labeling models and the quality of the expanded training data can mutually enhance each other through iterations. Second, we relate the regression confidence to the prediction variance under perturbations. Higher confidence indicates

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599497



$n(y)$
y
Imbalanced labeled data $\mathcal{G}_{\text{imb}}$
reverse
sampling
$p(y)$
Regression with *confidence*:
$\mathbf{h} = g(G)$
$y; \sigma = f(\mathbf{h})$
$\sigma \leq 0.9$
$\sigma > 0.9$
Unlabeled data $\mathcal{G}_{\text{unlbl}}$ with predicted labels
$g(G)$
Not balanced yet!
Confidently predicted data $\mathcal{G}_{\text{conf}}$
$\mathcal{H}$
Representations of all real graphs
reverse
label-anchored mixup
Balanced training data $\mathcal{G}_{\text{imb}} \cup \mathcal{G}_{\text{conf}} \cup \mathcal{H}_{\text{aug}}$
Augmented data $\mathcal{H}_{\text{aug}}$
SGIR models
Train $g(G)\colon \mathcal{G} \to \mathcal{H}$ with $\mathcal{G}_{\text{imb}} \cup \mathcal{G}_{\text{conf}}$ for $\mathcal{H}_{\text{imb}}$ and $\mathcal{H}_{\text{conf}}$
Train $f(\mathbf{h})\colon \mathcal{H} \to \mathbb{R}$ with $\mathcal{H}_{\text{imb}} \cup \mathcal{H}_{\text{conf}} \cup \mathcal{H}_{\text{aug}}$
self-training

Figure 1: An overview of our SGIR framework to train effective graph regression models with imbalanced labeled data. To balance the data properly, SGIR selects highly confident examples from predicted labels of unlabeled data and augments label areas that seriously lack data (even after added the confidently predicted data) by a novel label-anchored mixup algorithm.

a lower prediction variance under different perturbation environments. Therefore, we define and use *regression confidence* score to avoid pseudo-label noise and select quality examples in regression tasks. To fully exploit the quality pseudo-labels to compensate for the data imbalance in different label ranges, we use a reversed distribution of the imbalanced annotated data to reveal label ranges that need to be more or less selected for label balancing. Third, we attempt to achieve the perfect balance of training data by creating graph examples of any given label value in the remaining under-represented ranges.

In this paper, we propose SGIR, a novel Semi-supervised Framework for Graph Imbalanced Regression. This framework has three novel designs to implement our ideas. First, SGIR is a self-training framework with multiple iterations for model learning and balanced training data generation. Our second design is to sample more quality pseudo-labels for the less represented label ranges. We define a new measurement of regression confidence from recent studies on graph rationalization methods which provide perturbations for predictions at training and inference. After applying the confidence to filter out pseudo-label noise, we adopt *reverse sampling* to find optimal sampling rates at each label value that maximize the possibility of data balance. Intuitively, if a label value is less frequent in the annotated data, the sampling rate at this value is bigger and more pseudo-labeled examples are selected for model training. Third, we design a novel *label-anchored mixup* algorithm to augment graph examples by mixing up a virtual data point and a real graph example in latent space. Each virtual point is anchored at a certain label value that is still rare in the expanded labeled data. The mixed-up graph representations continue complementing the label ranges where we seriously lack data examples.

To empirically demonstrate the advantage of SGIR, we conduct experiments on seven graph property regression tasks from three different domains. Results show that SGIR significantly reduces the prediction error on all the tasks and in both under-/well-represented label ranges. For example, on the smallest dataset Mol-FreeSolv that has only 276 annotated graphs, SGIR reduces the mean absolute error from 1.114 to 0.777 (relatively 30% improvement) in the most under-represented label range and reduces the error from 0.642 to 0.563 (12% improvement) in the entire label space compared to state-of-the-art graph regression methods. To summarize:

- We address a new problem of graph imbalance regression with a novel semi-supervised framework SGIR.
- SGIR is a novel self-training framework creating balanced and enriched training data from pseudo-labels and augmented examples with three collaborated components: regression confidence, reverse sampling, and label-anchored mixup.
- SGIR is theoretically motivated and empirically validated on seven graph regression tasks. It outperforms other semi-supervised learning and imbalanced regression methods in both well-represented and under-represented label ranges.

2 RELATED WORK

2.1 Imbalanced Learning

Data resampling is known as under-sampling majority classes or over-sampling minority classes. SMOTE [9] created synthetic data for minority classes using linear interpolations on labeled data. Cost-sensitive techniques [11, 22] assigned higher weights to the loss of minority classes. And posterior re-calibration [7, 29, 42] encouraged larger margins for the prediction logits of minority classes. Imbalanced regression tasks have unique challenges due to continuous label values [51]. Some of the methods from imbalanced classifications were extended to imbalanced regression tasks. For example, SMOGN [6] adopted the idea and method of SMOTE for regression; Recently, Yang et al. [51] used regression focal loss and cost-sensitive reweighting; and BMSE [34] used logit re-calibration to predict numerical labels. LDS [51] smoothed label distribution using kernel density estimation. RankSim [13] regularized the latent space by approximating the distance of data points in the label space. Although these methods would improve performance on under-represented labels, they come at the expense of decreased performance on well-represented labels, particularly when annotated data is limited. SGIR avoids this by using unlabeled graphs to create more labels in the under-represented label ranges.

2.2 Semi-supervised Learning

To exploit unlabeled data, semi-supervised image classifiers such as FixMatch [36] and MixMatch [4] used pseudo-labeling and consistency regularization. Their performance relies on weak and strong data augmentation techniques, which are under-explored for regression tasks and graph property prediction tasks. At the same time, semi-supervised learners suffer from the model bias caused by the unlabeled imbalance. Therefore, after pseudo-labeling unlabeled data, DARP [19] and DASO [31] refined the biased pseudo-labels by aligning their distribution with an approximated true class distribution of unlabeled data. CADR [16] adjusted the threshold for pseudo-label assignments. CReST [46] selected more pseudo-labels for minority classes in self-training. To the best of our knowledge, there was no work that leveraged unlabeled data for regression tasks on imbalanced graph data, although SSDKL [17] performed semi-supervised regression for non-graph data without considering label imbalance. SGIR makes the first attempt to solve the imbalanced regression problem using semi-supervised learning.

2.3 Graph Property Prediction

Graph neural network models (GNN) [14, 20, 43, 50] have demonstrated their power for regression tasks in the fields of biology, chemistry, and material science [16, 24]. Data augmentation [23, 57] is an effective way to exploit limited labeled data. The node- and link-level augmentations [35, 58, 59] modified graph structure to improve the accuracy of node classification and link prediction. On the graph level, augmentation methods were mostly designed for classification [15, 45]. Recently, GREA [24] delivered promising results for predicting polymer properties. But the model bias caused by imbalanced continuous labels was not addressed. InfoGraph [39] exploited unlabeled graphs, however, the data imbalance issue was not addressed either. Our work aims to achieve balanced training data for graph regression in real practice where we have a small set of imbalanced labeled graphs and a large set of unlabeled data.

3 PROBLEM DEFINITION

To predict the property $y \in \mathbb{R}$ of a graph $G \in \mathcal{G}$, a graph regression model usually consists of an encoder $g : G \longrightarrow \mathbf{h} \in \mathbb{R}^d$ and a decoder $f : \mathbf{h} \longrightarrow \hat{y} \in \mathbb{R}$. The encoder $g(\cdot)$ is often a graph neural network (GNN) that outputs the d-dimensional representation vector $\mathbf{h}$ of graph G, and the decoder $f(\cdot)$ is often a multi-layer perceptron (MLP) that makes the label prediction $\hat{y}$ given $\mathbf{h}$.

Let $\mathcal{G}_{\text{imb}} = \{(G_i, y_i)\}_{i=1}^{n_{\text{imb}}}$ denote the labeled training data for graph regression models, where n_{imb} is the number of training graphs in the imbalanced labeled dataset. It often concentrates on certain areas in the continuous label space. To reveal it, we first divide the label space into C intervals and use them to fully cover the range of continuous label values. These intervals are $[b_0, b_1), [b_1, b_2), \ldots, [b_{C-1}, b_C)$. Then, we assign the labeled examples into C intervals and count them in each interval to construct the frequency set $\{\mu_i\}_{i=1}^{C}$. We could find that $\frac{\max\{\mu_i\}}{\min\{\mu_i\}} \gg 1$ (*i.e.*, label imbalance) often exists, instead of $\mu_1 = \mu_2 = \cdots = \mu_C$ (*i.e.*, label balance) that is assumed by most existing models. The existing models may be biased to small areas in the label space that are dominated by the majority of labeled data and lack a good generalization to areas that are equally important but have much fewer examples.

Labeling continuous graph properties is difficult [53], limiting the size of labeled data. Fortunately, a large number of unlabeled graphs are often available though ignored in most existing studies. In this work, we aim to use the unlabeled examples to alleviate the label imbalance issue in graph regression tasks. That is, let $\mathcal{G}_{\text{unlbl}} = \{G_j\}_{j=n_{\text{imb}}+1}^{n_{\text{imb}}+n_{\text{unlbl}}}$ denote the n_{unlbl} available unlabeled graphs. We want to train $g(\cdot)$ and $f(\cdot)$ to deliver good performance through the whole continuous label space by utilizing both $\mathcal{G}_{\text{imb}}$ and $\mathcal{G}_{\text{unlbl}}$.

4 PROPOSED FRAMEWORK

To progressively reduce label imbalance bias, we propose a novel framework named SGIR that iteratively creates reliable labeled examples in the areas of label space where annotations were not frequent. In Figure 1, SGIR uses a graph regression model to create the labels and uses the gradually balanced data to train the regression model. To let data balancing and model construction mutually enhance each other, SGIR is a self-training framework that trains the encoder $g(\cdot)$ and decoder $f(\cdot)$ using two strategies through multiple iterations. The first strategy is to use pseudo-labels based on confident predictions and reverse sampling, leveraging unlabeled data (see Section 4.2). Because the unlabeled graph set still may not contain real examples of rare label values, the second strategy is to augment the graph representation examples for the rare areas using a novel label-anchored mixup algorithm (see Section 4.3).

4.1 A Self-Training Framework for Iteratively Balancing Scalar Label Data

A classic self-training framework is expected to be a virtuous circle exploiting the unlabeled data in label-balanced classification/regression tasks [27, 49]. It first trains a classifier/regressor that iteratively assigns pseudo-labels to the set of unlabeled training examples $\mathcal{G}_{\text{unlbl}}$ with a margin greater than a certain threshold. The pseudo-labeled examples are then used to enrich the labeled training set. And the classifier continues training with the updated training set. For a virtuous circle of model training with imbalanced labeled set $\mathcal{G}_{\text{imb}}$, the most confident predictions on $\mathcal{G}_{\text{unlbl}}$ should be selected to compensate for the under-represented labels, as well as to enrich the dataset $\mathcal{G}_{\text{imb}}$. In each iteration, the model becomes less biased to the majority of labels. And the less biased model can make predictions of higher accuracy and confidence on the unlabeled data. Therefore, we hypothesize that model training and data balancing can mutually enhance each other.

SGIR is a self-training framework targeting to generalize the model performance everywhere in the continuous label space with particularly designed balanced training data from the labeled graph data $\mathcal{G}_{\text{imb}}$, confidently selected graph data $\mathcal{G}_{\text{conf}}$, and augmented representation data $\mathcal{H}_{\text{aug}}$. For the next round of model training, the gradually balanced training data reduce the label imbalance bias carried by the graph encoder $g(\cdot)$ and decoder $f(\cdot)$. Then the less biased graph encoder and decoder are applied to generate balanced training data of higher quality. Through these iterations, the model bias from the imbalanced or low-quality balanced data would be progressively reduced because of the gradually enhanced quality of balanced training data.

4.2 Balancing with Confidently Predicted Labels

At each iteration, SGIR enriches and balances training data with pseudo-labels of good quality. The unlabeled data examples in $\mathcal{G}_{\text{unlbl}}$ are firstly exploited by reliable and confident predictions. Then the reverse sampling from the imbalanced label distribution of original training data $\mathcal{G}_{\text{imb}}$ is used to select more pseudo-labels for under-represented label ranges.

4.2.1 Graph regression with confidence. A standard regression model outputs a scalar without a certain definition of confidence of its prediction. The confidence is often measured by how much the predicted probability is close to 1 in classifications. The lack of confidence measurements in graph regression tasks may introduce noise to the self-training framework that aims at label balancing. It would be more severe when the domain gap exists between labeled and unlabeled data [5]. Recent studies [24, 47] have proposed two concepts that help us define a good measurement: rationale subgraph and environment subgraph. A rationale subgraph is supposed to best support and explain the prediction at property inference. Its counterpart environment subgraph is the complementary subgraph in the example, which perturbs the prediction from the rationale subgraph if used. Our idea is to measure the confidence of graph property prediction based on the reliability of the identified rationale subgraphs. Specifically, we use the variance of predicted label values from graphs that consist of a specific rationale subgraph and one of many possible environment subgraphs.

We use an existing supervised graph regression model that can identify rationale and environment subgraphs in any graph example to predict its property. We denote G_i as the i-th graph in a batch of size B. The model separates G_i into $G_i^{(r)}$ and $G_i^{(e)}$. For the j-th graph G_j in the same batch, we have a combined example $G_{(i,j)} = G_i^{(r)} \cup G_j^{(e)}$ that has the rationale of G_i and environment subgraph of G_j. So it is expected to have the same label of G_i. By enumerating $j \in \{1, 2, \ldots, B\}$, the encoder $g(\cdot)$ and decoder $f(\cdot)$ are trained to predict the label value of any $G_{(i,j)}$. We define the confidence of predicting the label of G_i as:

$$\sigma_i = \frac{1}{\mathrm{Var}\left(\{f(g(G_{(i,j)}))\}_{j=1,2,\ldots,B} \right)}. \tag{1}$$

It is the reciprocal of prediction variance. In implementation, we choose GREA [24] as the model. Considering efficiency, GREA creates $G_{(i,j)}$ in the latent space without decoding its graph structure. That is, it directly gets the representation of $G_{(i,j)}$ as the sum of the representation vectors $\mathbf{h}_i^{(r)}$ of $G_i^{(r)}$ and $\mathbf{h}_j^{(e)}$ of $G_j^{(e)}$. So we have

$$\sigma_i = \frac{1}{\mathrm{Var}\left(\{f(\mathbf{h}_i^{(r)} + \mathbf{h}_j^{(e)})\}_{j=1,2,\ldots,B} \right)}. \tag{2}$$

Now we have predicted labels and confidence values for graph examples in the large unlabeled dataset $\mathcal{G}_{\text{unlbl}}$. Examples with low confidences will bring noise to the training data if we use them all. So we only consider a data example G_i to be of good quality if its confidence σ_i is not smaller than a threshold τ. We name this confidence measurement based on graph rationalization as GRation. GRation is tailored for graph regression tasks by considering the environment subgraphs as perturbations. We will compare its effect on quality graph selection against other graph-irrelevant methods such as Dropout [12], Certi [40], DER (Deep Evidential Regression) [2], and Simple (no confidence) in experiments.

After leveraging the unlabeled data, the label distribution of quality examples may still be biased to the majority of labels. So we further apply reverse sampling on these examples from $\mathcal{G}_{\text{unlbl}}$ to balance the distribution of training data.

4.2.2 Reverse sampling. The reverse sampling in SGIR helps reduce the model bias to label imbalance. Specifically, we want to selectively add unlabeled examples predicted in the under-represented label ranges. Suppose we have the frequency set $\{\mu_i\}_{i=1}^C$ of C intervals. We denote p_i as the sampling rate at the i-th interval and follow Wei et al. [46] to calculate it. Basically, to perform reverse sampling, we want $p_i < p_j$ if $\mu_i > \mu_j$. We define a new frequency set $\{\mu'_i\}_{i=1}^C$ in which μ'_i equals the i-th smallest in $\{\mu\}$ if μ_i is the i-th biggest in $\{\mu\}$. Then the sampling rate is

$$p_i = \frac{\mu'_i}{\max\{\mu_1, \mu_2, \ldots, \mu_C\}}. \tag{3}$$

To this end, we have the confidently labeled and reversed sampled data $\mathcal{G}_{\text{conf}}$. In each self-training iteration, we combine it with the original training set $\mathcal{G}_{\text{imb}}$.

4.3 Balancing with Augmentation via Label-Anchored Mixup

Although $\mathcal{G}_{\text{imb}} \cup \mathcal{G}_{\text{conf}}$ is more balanced than $\mathcal{G}_{\text{imb}}$, we observe that $\mathcal{G}_{\text{imb}} \cup \mathcal{G}_{\text{conf}}$ is usually far from a *perfect balance*, even if $\mathcal{G}_{\text{unlbl}}$ could be hundreds of times bigger than $\mathcal{G}_{\text{imb}}$. To create graph examples targeting the remaining under-represented label ranges, we design a novel label-anchored mixup algorithm for graph imbalanced regression. Compared to existing mixup algorithms [44, 45] for classifications without awareness of imbalance, our new algorithm can augment training data with additional examples for target ranges of continuous label value.

A mixup operation in the label-anchored mixup is to mix up two things in a latent space: (1) a virtual data point representing an interval of targeted label and (2) a real graph example. Specifically, we first calculate the representation of a target label interval by averaging the representation vectors of graphs in the interval from the labeled dataset $\mathcal{G}_{\text{imb}}$. Let $\mathbf{M} \in \{0,1\}^{C \times n_{\text{imb}}}$ be an indicator matrix, where $M_{i,j} = 1$ means that the label of $G_j \in \mathcal{G}_{\text{imb}}$ belongs to the i-th interval. We denote $\mathbf{H} \in \mathbb{R}^{n_{\text{imb}} \times d}$ as the matrix of graph representations from the GNN encoder $g(\cdot)$ for $\mathcal{G}_{\text{imb}}$. The representation matrix $\mathbf{Z} \in \mathbb{R}^{C \times d}$ of all intervals is calculated

$$\mathbf{Z} = \mathrm{norm}(\mathbf{M}) \cdot \mathbf{H}, \tag{4}$$

where norm$(\cdot)$ is the row-wise normalization and a_i is the center value of the i-th interval. We have the representation-label pairs of all the label intervals $\{(\mathbf{z}_i, a_i)\}_{i=1}^{C}$, where $\mathbf{z}_i$ is the i-th row of $\mathbf{Z}$.

Now we can use each interval center a_i as a label anchor to augment graph data examples in a latent space. We select $n_i \propto p_i$ real graphs from $\mathcal{G}_{\text{imb}} \cup \mathcal{G}_{\text{conf}}$ whose labels are closest to a_i, where p_i is calculated by Eq. (3). The more real graphs are selected, the more graph representations are augmented. n_i is likely to be big when the label anchor a_i remains under-represented after $\mathcal{G}_{\text{conf}}$ is added to training set. Note that the labels were annotated if the graphs were in $\mathcal{G}_{\text{imb}}$ and predicted if they were in $\mathcal{G}_{\text{unlbl}}$. For $j \in \{1, 2, \dots, n_i\}$, we mix up the interval $(\mathbf{z}_i, a_i)$ and a real graph $(\mathbf{h}_j, y_j)$, where $\mathbf{h}_j$ and y_i are the representation vector and the annotated or predicted label of the j-th graph, respectively. Then the mixup operation is defined as

$$\begin{cases} \tilde{\mathbf{h}}_{(i,j)} &= \lambda \cdot \mathbf{z}_i + (1-\lambda) \cdot \mathbf{h}_j, \\ \tilde{y}_{(i,j)} &= \lambda \cdot a_i + (1-\lambda) \cdot y_j, \end{cases} \tag{5}$$

where $\tilde{\mathbf{h}}_{(i,j)}$ and $\tilde{y}_{(i,j)}$ are the representation vector and label of the augmented graph, respectively. $\lambda = \max(\lambda', 1-\lambda')$, $\lambda' \sim \text{Beta}(1, \beta)$, and β is a hyperparameter. λ is often closer to 1 because we want $\tilde{y}_{(i,j)}$ to be closer to the label anchor a_i. Let $\mathcal{H}_{\text{aug}}$ denote the set of representation vectors of all the augmented graphs. Combined with $\mathcal{G}_{\text{imb}}$ and $\mathcal{G}_{\text{conf}}$, we end up with a label-balanced training set for the next round of self-training.

4.4 Optimization

In each iteration of self-training, we jointly optimize the parameters of graph encoder $g(\cdot)$ and label predictor $f(\cdot)$ with a gradually balanced training set $\mathcal{G}_{\text{imb}} \cup \mathcal{G}_{\text{conf}} \cup \mathcal{H}_{\text{aug}}$.

We use the mean absolute error (MAE) as the regression loss. Specifically, for each $(G, y) \in \mathcal{G}_{\text{imb}} \cup \mathcal{G}_{\text{conf}}$, the loss is $\ell_{\text{imb+conf}} = \text{MAE}(f(g(G)), y)$. Given $(\mathbf{h}, y) \in \mathcal{H}_{\text{aug}}$, the loss is $\ell_{\text{aug}} = \text{MAE}(f(\mathbf{h}), y)$. So the total loss for SGIR is

$$\mathcal{L} = \sum_{(G,y) \in \mathcal{G}_{\text{imb}} \cup \mathcal{G}_{\text{conf}}} \ell_{\text{imb+conf}}(G, y) + \sum_{(\mathbf{h},y) \in \mathcal{H}_{\text{aug}}} \ell_{\text{aug}}(\mathbf{h}, y).$$

Our framework is flexible with any graph encoder-decoder models. To be consistent and given the promising results in graph regression tasks, we use the design of graph encoder and decoder in GREA [24] which is also used for measuring prediction confidence in Eq. (2).

4.5 Theoretical Motivations

There is a lack of theoretical principle for imbalanced regression. Our theoretical motivation extends the generalization error bound from classification [7] to regression. The original bound enforces bigger margins for minority classes, which potentially hurt the model performance for well-represented classes [42, 55]. Our result provides a more safe way to reduce the error bound by utilizing unlabeled graphs with self-training in graph regression tasks.

As we divide the label distribution into C intervals, every graph example can be assigned into an interval (as the ground-truth interval) according to the distance between the interval center and the ground-truth label value. Besides, we use $S_{[b_i,b_{i+1})}(G)$ to denote the reciprocal of the distance between the predicted label of the graph G and the i-th interval $[b_i, b_{i+1})$, where $i \in \{1, 2, \dots, C\}$. In this way, we could define $f(\cdot)$ as a regression function that outputs a continuous predicted label. Then $S_{[b_i,b_{i+1})}(G)$ consists of $f(\cdot)$ and outputs the logits to classify the graph to the i-the interval.

We consider all training examples to follow the same distribution. We assume that conditional on label intervals, the distributions of graph sampling are the same at training and testing stages. So, the standard 0-1 test error on the balanced test distribution is

$$\mathcal{E}_{\text{bal}}[f] = \Pr_{(G,[b_i,b_{i+1})) \sim \mathcal{P}_{\text{bal}}} \left[S_{[b_i,b_{i+1})}(G) < \max_{j \neq i} S_{[b_j,b_{j+1})}(G) \right], \tag{6}$$

where $\mathcal{P}_{\text{bal}}$ denotes the balanced test distribution. It first samples a label interval uniformly and then samples graphs conditionally on the interval. The error for the i-th interval $[b_i, b_{i+1})$ is defined as

$$\mathcal{E}_{[b_i,b_{i+1})}[f] = \Pr_{G \sim \mathcal{P}_{[b_i,b_{i+1})}} \left[S_{[b_i,b_{i+1})}(G) < \max_{j \neq i} S_{[b_j,b_{j+1})}(G) \right], \tag{7}$$

where $\mathcal{P}_{[b_i,b_{i+1})}$ denotes the distribution for the interval $[b_i, b_{i+1})$. We define $\gamma(G, [b_i, b_{i+1})) = S_{[b_i,b_{i+1})}(G) - \max_{j \neq i} S_{[b_j,b_{j+1})}(G)$ as the margin of an example G assigned to the interval $[b_i, b_{i+1})$. To define the training margin $\gamma_{[b_i,b_{i+1})}$ for the interval $[b_i, b_{i+1})$, we calculate the minimal margin across all examples assigned to that interval:

$$\gamma_{[b_i,b_{i+1})} = \min_{G_j \in [b_i,b_{i+1})} \gamma\left(G_j, [b_i, b_{i+1})\right). \tag{8}$$

We assume that the MAE regression loss is small enough to correctly assign all training examples to the corresponding intervals. Given the hypothesis class $\mathcal{F}$, $\mathrm{C}(\mathcal{F})$ is assumed to be a proper complexity measure of $\mathcal{F}$. We assume there are $n_{[b_i,b_{i+1})}$ examples i.i.d sampled from the conditional distribution $\mathcal{P}_{[b_i,b_{i+1})}$ for the interval $[b_i, b_{i+1})$. So, we apply the standard margin-based generalization bound to obtain the following theorem [7, 18, 56]:

Theorem 4.1. *With probability $(1-\delta)$ over the randomness of the training data, the error $\mathcal{E}_{[b_i,b_{i+1})}$ for interval $[b_i, b_{i+1})$ is bounded by*

$$\begin{aligned} \mathcal{E}_{[b_i,b_{i+1})}[f] \lessapprox & \frac{1}{\gamma_{[b_i,b_{i+1})}} \sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_{[b_i,b_{i+1})}}} \\ & + \sqrt{\frac{\log\log_2(1/\gamma_{[b_i,b_{i+1})}) + \log(1/\delta)}{n_{[b_i,b_{i+1})}}}, \end{aligned} \tag{9}$$

where $\lessapprox$ hides constant terms. Taking union bound over all intervals, we have $\mathcal{E}_{\text{bal}}[f] \lessapprox \frac{1}{C}\sum_{i=1}^{C} \mathcal{E}_{[b_i,b_{i+1})}[f]$.

Proofs are in appendix A. The bound decreases as the increase of the examples in corresponding label ranges. The SGIR is motivated to reduce and balance the bound for different intervals by manipulating $n_{[b_i,b_{i+1})}$ with pseudo-labels and augmented examples. Particularly, we discuss in appendix A.3 that the augmented examples do not break our assumption for the theorem and future directions of imbalanced regression theories without intervals.

5 EXPERIMENTS

We conduct experiments to demonstrate the effectiveness of SGIR and answer the research question: how it performs on graph regression tasks and at different label ranges (RQ1). We also make a few ablation studies to investigate the effect of model design: where the effectiveness comes from (RQ2).

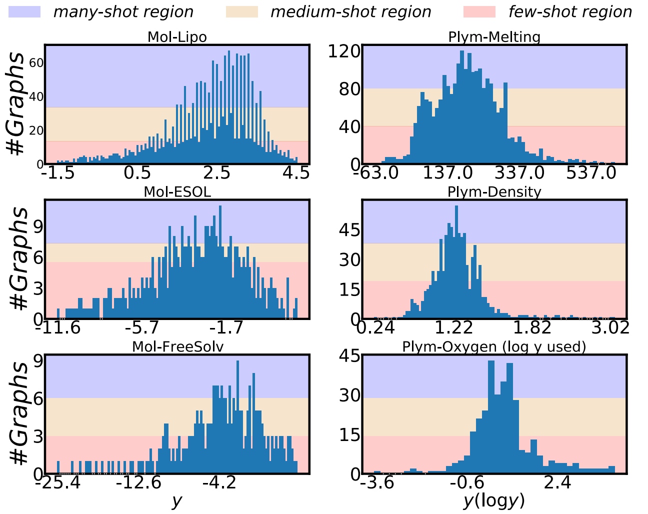


Figure 2: Imbalanced training distributions $\mathcal{G}_{\text{imb}}$ of annotated molecule and polymers.

5.1 Experimental Settings

5.1.1 Datasets. We include three **molecule** [48] (Mol-) and three **polymer** [24] datasets (Plym-). Besides labeled graphs, we combine a database of 133,015 molecules in QM9 [33] and an integration of four sets of 13,114 polymers in total [24] to create a set of **146,129** unlabeled graphs to set up semi-supervised graph regression. We note that the unlabeled graphs may be slightly less than 146,129 for a polymer task on Plym-Melting, Plym-Density or Plym-Oxygen. Because we remove the overlapping graph examples for the current polymer task with the polymer unlabeled data. We follow [51] to split the datasets to characterize imbalanced training distributions and balanced test distributions. We also include an **age regression dataset**, Superpixel-Age, which is constructed from image superpixels using the algorithms from [21] on the image dataset *AgeDB-DIR* from [30, 51]. Details are in Figure 3 and appendix B.1.

5.1.2 Evaluation metrics. We report model performance on three different sub-ranges following the work in [13, 34, 51], besides the *entire range* of label space. The three sub-ranges are the *many-shot region*, *medium-shot region*, and *few-shot region*. The sub-ranges are defined by the number of training graphs in each label value interval. Details for each dataset are presented in Figure 2 and Figure 3. To evaluate the regression performance, we use mean absolute error (MAE) and geometric mean (GM) [51]. Lower values (↓) of MAE or GM indicate better performance.

5.1.3 Baselines and Implementations. Besides the GNN model, we broadly consider baselines from the fields of imbalanced regression and semi-supervised graph learning. Specifically, imbalanced regression baselines include LDS [51], BMSE [34], and RankSim [13]. The semi-supervised graph learning baseline is InfoGraph [39] and the graph learning baseline is GREA [24]. To implement SGIR and the baselines, the GNN encoder is GIN [50] and the decoder is a three-layer MLP to output property values. The threshold τ for selecting confident predictions is determined by the value at a certain percentile of the confidence score distribution. For all the methods, we reports the results on the test sets using the mean (standard deviation) over 10 runs with parameters that are randomly initialized. More Implementation details are in Appendix.

5.2 RQ1: Effectiveness on Property Prediction

5.2.1 Effectiveness on Molecule and Polymer Prediction. Table 1 presents results of SGIR and baseline methods on six graph regression tasks. We have three observations.

Overall effectiveness in the entire label range: SGIR performs consistently better than competitive baselines on all tasks. Columns "All" in Table 1 show that SGIR reduces MAE over the best baselines (whose MAEs are underlined in the table) relatively by 9.1%, 8.1%, and 12.3% on the three molecule datasets, respectively. Specifically, on Mol-FreeSolv, the MAE was reduced from 0.642 to 0.563 with no change on the standard deviation. This is because SGIR enrich and *balance* the training data with confidently predicted pseudo-labels and augments for data examples on all the possible label ranges, whereas all the baseline models suffer from the bias caused by imbalanced annotations.

Effectiveness in few-shot label ranges: The performance improvements of SGIR on graph regression tasks are simultaneously from three different label ranges: *many-shot region*, *medium-shot region*, and *few-shot region*. By looking at the results of baselines, we find that the best performance at a particular range would sacrifice the performance at a different label range. For example, on the Mol-Lipo and Mol-FreeSolv datasets, while GREA is the second best and best baseline, respectively, in the *many-shot region*, its performance in the *few-shot region* is worse than the basic GNN models. Similarly, on the Mol-FreeSolv dataset, LDS reduces the MAE from GNN relatively by +3.5% in the *few-shot region* with a trade-off of a -29% performance decrease in the *many-shot region*. Compared to baselines, the improvements from SGIR in the under-represented label ranges are theoretically guaranteed without sacrificing the performance in the well-represented label range. And our experimental observations support the theoretical guarantee, even in more challenging scenarios, *i.e.,* predictions in the label ranges of fewer training shots on smaller datasets. Specifically, SGIR reduces MAE relatively by 30.3% and 9.0% in the *few-shot region* on Mol-FreeSolv and Plym-Oxygen. Because SGIR leverages the mutual enhancement of model construction and data balancing: the gradually balanced training data reduce model bias to popular labels; the less biased model improves the quality of pseudo-labels and augmented examples in the *few-shot region*.

Effectiveness on different graph regression tasks: We observe that the improvements on molecule regression tasks are more significant than those on polymer regression tasks. We hypothesize the reasons to be (1) the quality of unlabeled source data and (2) the size of the label space. First, our unlabeled graphs consist of more than a hundred thousand unlabeled small molecule graphs from QM9 [33] and around ten thousand polymers (macromolecules) from [24]. The massive quantity of unlabeled molecules make it easier to have good quality pseudo-labels and augmented examples for the three small molecule regression tasks on Mol-Lipo, Mol-ESOL, and Mol-FreeSolv [33]. Because the majority of unlabeled molecule graphs have a big domain gap with the polymer regression tasks, the quality of expanded training data in polymer

Table 1: Results of Mean(Std) on six molecule/polymer datasets. The best mean is bolded. The best baseline is underlined.

		MAE ↓				GM ↓			
		All	Many-shot	Med.-shot	Few-shot	All	Many-shot	Med.-shot	Few-shot
Mol-Lipo	GNN	0.485(0.010)	0.421(0.030)	0.462(0.013)	0.566(0.032)	0.297(0.012)	0.252(0.022)	0.294(0.016)	0.348(0.030)
	RankSim	0.475(0.018)	0.388(0.017)	0.438(0.007)	0.587(0.043)	0.297(0.015)	0.249(0.017)	0.274(0.006)	0.380(0.044)
	BMSE	0.494(0.007)	0.409(0.019)	0.450(0.007)	0.614(0.033)	0.304(0.008)	0.260(0.014)	0.279(0.015)	0.382(0.038)
	LDS	0.468(0.009)	0.394(0.012)	0.449(0.012)	0.551(0.026)	0.294(0.010)	0.251(0.009)	0.281(0.010)	0.356(0.033)
	InfoGraph	0.499(0.008)	0.421(0.024)	0.471(0.013)	0.596(0.026)	0.314(0.011)	0.269(0.018)	0.300(0.006)	0.376(0.029)
	GREA	0.487(0.002)	0.391(0.015)	0.434(0.008)	0.626(0.018)	0.294(0.010)	0.251(0.009)	0.281(0.010)	0.356(0.033)
	SGIR	**0.432**(0.012)	**0.357**(0.019)	**0.413**(0.017)	**0.515**(0.020)	**0.264**(0.013)	**0.224**(0.016)	**0.256**(0.017)	**0.314**(0.015)
Mol-ESOL	GNN	0.508(0.015)	0.398(0.018)	0.448(0.012)	0.696(0.025)	0.299(0.017)	0.231(0.017)	0.279(0.014)	0.425(0.035)
	RankSim	0.501(0.014)	0.389(0.021)	0.443(0.019)	0.689(0.025)	0.293(0.021)	0.227(0.028)	0.258(0.020)	0.449(0.030)
	BMSE	0.533(0.023)	0.400(0.027)	0.449(0.015)	0.777(0.069)	0.308(0.018)	0.245(0.036)	0.266(0.009)	0.473(0.035)
	LDS	0.517(0.016)	0.423(0.012)	0.474(0.029)	0.668(0.010)	0.304(0.010)	0.261(0.007)	0.283(0.025)	0.393(0.009)
	InfoGraph	0.561(0.025)	0.475(0.034)	0.466(0.036)	0.776(0.036)	0.336(0.014)	0.306(0.022)	0.276(0.013)	0.484(0.029)
	GREA	0.497(0.031)	0.396(0.040)	0.456(0.033)	0.652(0.045)	0.289(0.032)	**0.226**(0.038)	0.270(0.025)	0.404(0.051)
	SGIR	**0.457**(0.015)	**0.370**(0.022)	**0.411**(0.011)	**0.604**(0.024)	**0.263**(0.016)	**0.226**(0.021)	**0.240**(0.015)	**0.347**(0.030)
Mol-FreeSolv	GNN	0.726(0.039)	0.617(0.061)	0.695(0.055)	1.154(0.082)	0.363(0.025)	0.317(0.027)	0.360(0.029)	0.556(0.073)
	RankSim	0.779(0.109)	0.764(0.225)	0.674(0.072)	1.220(0.146)	0.367(0.026)	0.396(0.052)	0.315(0.030)	0.537(0.082)
	BMSE	0.856(0.071)	0.809(0.117)	0.820(0.064)	1.122(0.076)	0.456(0.042)	0.426(0.029)	0.457(0.054)	0.552(0.062)
	LDS	0.809(0.071)	0.796(0.071)	0.737(0.088)	1.114(0.141)	0.443(0.045)	0.489(0.036)	0.387(0.052)	0.580(0.146)
	InfoGraph	0.933(0.042)	0.830(0.081)	0.913(0.030)	1.308(0.171)	0.542(0.048)	0.505(0.107)	0.528(0.038)	0.789(0.183)
	GREA	0.642(0.026)	0.541(0.064)	0.570(0.008)	1.202(0.023)	0.321(0.038)	0.294(0.064)	0.301(0.024)	0.537(0.049)
	SGIR	**0.563**(0.026)	**0.535**(0.038)	**0.528**(0.046)	**0.777**(0.061)	**0.264**(0.029)	**0.286**(0.013)	**0.244**(0.046)	**0.304**(0.078)
Plym-Melting	GNN	41.8(1.2)	35.5(1.2)	33.0(0.7)	54.7(2.2)	23.2(1.0)	21.3(1.1)	16.2(1.0)	33.4(2.5)
	RankSim	41.1(0.9)	34.1(0.5)	33.6(1.1)	53.5(1.2)	22.6(1.1)	20.5(0.5)	16.8(1.0)	31.4(2.8)
	BMSE	42.1(0.7)	35.8(1.4)	34.1(1.3)	54.4(1.5)	23.7(1.2)	21.5(1.0)	18.1(0.5)	32.4(3.0)
	LDS	41.6(0.3)	35.3(0.9)	34.5(1.1)	53.2(0.8)	23.2(0.2)	20.5(1.2)	18.3(0.5)	31.4(1.1)
	InfoGraph	43.6(2.8)	35.3(2.3)	35.0(2.3)	58.3(4.1)	24.6(1.9)	21.3(1.5)	18.4(1.5)	35.4(4.1)
	GREA	41.2(0.8)	33.3(0.5)	32.7(0.7)	55.3(3.0)	23.4(0.6)	20.0(0.6)	17.3(0.7)	34.3(2.9)
	SGIR	**38.9**(0.7)	**31.7**(0.3)	**31.5**(1.1)	**51.4**(1.6)	**21.1**(1.2)	**18.5**(0.5)	**15.9**(1.4)	**30.2**(1.9)
Plym-Density (scaled:×1, 000)	GNN	61.2(5.4)	63.4(18.9)	46.6(1.6)	72.0(2.8)	29.3(0.6)	29.6(3.3)	23.5(0.9)	35.5(2.0)
	RankSim	57.5(1.8)	55.1(2.2)	46.3(1.8)	69.4(3.3)	29.3(1.6)	29.9(2.8)	23.1(2.1)	35.4(2.5)
	BMSE	61.8(2.0)	59.1(8.6)	48.2(2.0)	75.9(3.5)	31.9(1.3)	31.8(4.2)	26.3(2.2)	38.2(3.2)
	LDS	60.1(2.4)	60.4(6.2)	47.0(1.3)	71.3(2.5)	31.5(2.0)	33.2(3.5)	24.4(3.0)	38.0(2.4)
	InfoGraph	54.9(1.7)	46.8(1.0)	43.0(1.9)	72.3(3.2)	29.3(1.8)	27.3(1.4)	**22.6**(1.2)	39.2(4.3)
	GREA	60.3(1.9)	49.0(4.4)	48.1(2.5)	80.7(4.2)	32.3(1.6)	26.7(2.7)	27.2(2.3)	44.7(6.1)
	SGIR	**53.0**(0.5)	**45.4**(1.7)	**42.5**(2.8)	**68.6**(2.6)	**26.6**(0.4)	**24.0**(2.2)	23.0(1.3)	**33.4**(3.0)
Plym-Oxygen	GNN	183.5(33.4)	6.3(3.2)	14.6(6.6)	464.0(85.3)	7.0(1.8)	2.4(0.7)	3.9(1.1)	29.9(7.2)
	RankSim	165.7(27.4)	3.9(1.4)	13.0(2.0)	420.7(69.7)	5.9(1.4)	**1.8**(0.3)	3.6(1.7)	26.6(6.7)
	BMSE	190.4(33.4)	26.4(21.6)	27.0(16.4)	454.3(88.9)	25.7(14.8)	14.9(11.7)	15.9(9.6)	63.2(23.5)
	LDS	180.0(23.0)	6.6(4.0)	**11.8**(2.0)	456.3(60.2)	7.6(1.6)	2.4(0.6)	4.7(1.4)	33.6(9.2)
	InfoGraph	199.5(31.5)	7.5(7.2)	13.0(1.8)	505.5(78.2)	7.8(1.9)	2.3(0.5)	5.1(2.2)	34.8(8.5)
	GREA	182.5(30.0)	9.0(8.6)	14.4(4.9)	458.8(79.2)	7.1(1.3)	2.1(0.5)	4.4(1.3)	31.7(5.0)
	SGIR	**150.9**(17.8)	**3.8**(1.1)	12.2(0.6)	**382.8**(46.9)	**5.8**(0.4)	2.1(0.7)	**3.3**(0.8)	**24.4**(6.8)

regression tasks would be relatively worse than the quality of those in molecule regression. This inspires us to collect more polymer data in the future, even if their properties could not be annotated. Second, Figure 2 has shown that the label ranges in the polymer regression tasks are usually much wider than the ranges in the molecule regression tasks. This poses a great challenge for accurate predictions, especially when we train with a small dataset.

5.2.2 Effectiveness on Age Prediction. Besides molecules and polymers, Table 2 presents more results by comparing different methods on the Superpixel-Age dataset. SGIR consistently improves the model performance compared to the best baselines in different label ranges. In the entire label range, SGIR reduces the MAE (GM) relatively by +4.7% (+3.6%). The advantages mainly stem from the enhancements in the *few-shot region*, as demonstrated in Table 2, which shows an improvement of +4.3% and +3.1% on the MAE and GM metrics, respectively. Different from LDS, SGIR improves the model performance for the under-represented and well-represented label ranges at the same time. Table 2 showcases that the empirical advantages of SGIR could generalize across different domains.

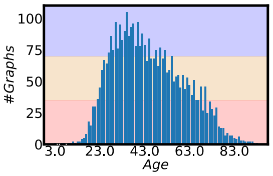


Figure 3: Imbalanced training distributions $\mathcal{G}_{\text{imb}}$ in the Superpixel-Age dataset.

	MAE ↓				GM ↓			
	All	Many-shot	Med.-shot	Few-shot	All	Many-shot	Med.-shot	Few-shot
GNN	14.583(0.413)	10.524(0.994)	11.698(0.404)	22.127(0.780)	9.996(0.386)	7.265(0.858)	7.910(0.492)	18.404(0.673)
RankSim	<u>14.464</u>(0.401)	10.468(0.759)	11.610(0.774)	21.910(0.700)	<u>9.606</u>(0.303)	<u>6.936</u>(0.598)	7.721(0.660)	17.534(1.768)
BMSE	15.179(0.594)	10.639(2.303)	12.201(0.900)	23.321(2.525)	10.419(0.393)	7.249(1.526)	8.659(0.827)	19.719(4.318)
LDS	14.674(0.191)	10.972(0.495)	11.985(0.627)	<u>21.623</u>(0.926)	9.867(0.291)	7.317(0.672)	7.997(0.633)	<u>17.298</u>(0.957)
InfoGraph	14.515(0.605)	10.610(1.063)	<u>11.150</u>(0.158)	22.476(1.147)	9.879(0.524)	7.391(0.995)	<u>7.377</u>(0.333)	18.969(1.873)
GREA	14.682(0.300)	<u>10.283</u>(0.503)	11.999(0.585)	22.329(0.570)	10.037(0.438)	7.051(0.455)	8.273(0.565)	18.142(1.276)
SGIR	**13.787**(0.123)	**10.171**(0.4156)	**11.066**(0.389)	**20.687**(0.839)	**9.261**(0.221)	**6.928**(0.355)	**7.247**(0.593)	**16.769**(1.418)

Table 2: Results of Mean(Std) on the age prediction using graphs from image superpixels. The best mean is bold. The best baseline is <u>underlined</u>.

Table 3: A comprehensive ablation study on molecule regression datasets with the metric MAE (↓). σ is the confidence score in Section 4.2.1. p is the reverse sampling in Section 4.2.2. $(\tilde{\mathbf{h}}, \tilde{y})$ is the label-anchored mixup in Section 4.3.

	σ	p	$(\tilde{\mathbf{h}}, \tilde{y})$	All	Many-shot	Med.-shot	Few-shot
Mol-Lipo	w/o $\mathcal{G}_{\text{unlbl}}$			0.477(0.014)	0.378(0.030)	0.440(0.011)	0.600(0.006)
	✓	✗	✗	0.448(0.006)	0.371(0.004)	0.421(0.012)	0.543(0.016)
	✗	✓	✗	0.446(0.008)	**0.356**(0.003)	**0.407**(0.011)	0.564(0.016)
	✓	✓	✗	0.442(0.012)	0.372(0.007)	0.415(0.004)	0.533(0.026)
	✗	✗	✓	0.456(0.007)	0.372(0.014)	0.436(0.010)	0.549(0.005)
	✓	✓	✓	**0.432**(0.012)	0.357(0.019)	0.413(0.017)	**0.515**(0.020)
Mol-ESOL	w/o $\mathcal{G}_{\text{unlbl}}$			0.477(0.027)	0.375(0.014)	0.432(0.042)	0.637(0.042)
	✓	✗	✗	0.475(0.014)	0.369(0.014)	0.446(0.017)	0.618(0.039)
	✗	✓	✗	0.480(0.017)	0.380(0.035)	0.440(0.017)	0.630(0.020)
	✓	✓	✗	0.468(0.007)	0.379(0.012)	0.425(0.013)	0.612(0.028)
	✗	✗	✓	0.474(0.010)	**0.353**(0.018)	0.450(0.009)	0.623(0.027)
	✓	✓	✓	**0.457**(0.015)	0.370(0.022)	**0.411**(0.011)	**0.604**(0.024)
Mol-FreeSolv	w/o $\mathcal{G}_{\text{unlbl}}$			0.619(0.019)	**0.525**(0.022)	0.590(0.035)	1.000(0.072)
	✓	✗	✗	0.604(0.020)	0.557(0.037)	0.560(0.029)	0.903(0.055)
	✗	✓	✗	0.660(0.028)	0.574(0.015)	0.650(0.036)	0.941(0.066)
	✓	✓	✗	0.568(0.029)	0.538(0.020)	**0.520**(0.045)	0.831(0.132)
	✗	✗	✓	0.593(0.045)	0.536(0.033)	0.542(0.067)	0.947(0.062)
	✓	✓	✓	**0.563**(0.026)	0.535(0.038)	0.528(0.046)	**0.777**(0.061)

Table 4: Investigation on choices of regression confidence with the metric MAE (↓). We disable all other SGIR components except the regression confidence score. Our confidence score (GRation) in Eq. (1) removes noise more effectively than others in graph regression tasks.

	Choice of σ	All	Many-shot	Med.-shot	Few-shot
Mol-Lipo	Simple	0.481(0.010)	0.389(0.007)	0.440(0.013)	0.603(0.023)
	Dropout	0.450(0.026)	**0.365**(0.031)	**0.420**(0.022)	0.555(0.037)
	Certi	0.452(0.011)	0.384(0.018)	0.433(0.013)	**0.532**(0.010)
	DER	1.026(0.033)	0.604(0.035)	0.760(0.016)	1.672(0.111)
	GRation	**0.448**(0.006)	0.371(0.004)	0.421(0.012)	0.543(0.016)
Mol-ESOL	Simple	0.499(0.016)	0.397(0.023)	0.457(0.018)	0.656(0.033)
	Dropout	0.483(0.011)	0.381(0.027)	**0.443**(0.018)	0.636(0.027)
	Certi	0.487(0.030)	0.389(0.039)	0.439(0.024)	0.647(0.043)
	DER	0.918(0.135)	0.776(0.086)	0.826(0.098)	1.182(0.245)
	GRation	**0.475**(0.014)	**0.369**(0.014)	0.446(0.017)	**0.618**(0.039)
Mol-FreeSolv	Simple	0.697(0.056)	0.616(0.025)	0.663(0.033)	1.054(0.260)
	Dropout	0.639(0.013)	0.578(0.060)	0.589(0.017)	1.005(0.140)
	Certi	0.654(0.049)	0.589(0.046)	0.611(0.053)	0.999(0.130)
	DER	1.483(0.174)	1.180(0.162)	1.450(0.188)	2.480(0.373)
	GRation	**0.604**(0.020)	**0.557**(0.037)	**0.560**(0.029)	**0.903**(0.055)

5.3 RQ2: Ablation Studies on Framework Design

We conduct five studies and analyze the results below. Four ablation studies are (1) $\mathcal{G}_{\text{conf}}$ and $\mathcal{H}_{\text{aug}}$ for data balancing; (2) mutually enhanced iterative process; (3) choices of confidence score; and (4) quality and diversity of the label-anchored mixup. (5) The sensitivity analysis for the label interval number C.

5.3.1 *Effect of balancing data with different components in $\mathcal{G}_{conf}$ and $\mathcal{H}_{aug}$.*

Studies on molecule regression tasks in Table 3 present how SGIR improves the initial supervised performance to the most advanced semi-supervised performance step by step. In the first line for each dataset, we use only imbalanced training data $\mathcal{G}_{\text{conf}}$ to train the regression model and observe that the model performs badly in the *few-shot region*. The fourth line for each dataset combines the use of regression confidence σ and the reverse sampling p to produce $\mathcal{G}_{\text{conf}}$. It improves the MAE performance in the *few-shot region* relatively by +11.2%, +3.2%, and +15.9% on the Mol-Lipo, Mol-ESOL, and Mol-FreeSolv datasets, respectively. The label-anchored mixup algorithm produces the augmented graph representations $\mathcal{H}_{\text{aug}}$ for the under-represented label ranges. By applying $\mathcal{H}_{\text{aug}}$ with $\mathcal{G}_{\text{conf}}$, the last line continues improving the MAE performance in the *few-shot region* (compared to the third line) relatively by +3.3%, +1.3%, and +6.5% on the Mol-Lipo, Mol-ESOL, and Mol-FreeSolv datasets, respectively. Because the use of $\mathcal{H}_{\text{aug}}$ provides a chance to lead the label distributions of training data closer to a perfect balance. Specifically, the effect of semi-supervised pseudo-labeling, or $\mathcal{G}_{\text{conf}}$, comes from the regression confidence σ and reverse sampling rate p. Results on Mol-ESOL and Mol-FreeSolv show that without the confidence σ (the second line), reverse sampling was useless due to heavy label noise. Results on all molecule datasets indicate that without the reverse sampling rate p (the first line), the improvement to *few-shot region* by pseudo-labels was limited.

5.3.2 *Effect of iterative self-training.*

Figure 4 confirms that model learning and balanced training data mutually enhance each other in SGIR. Because we find that the model performance gradually

Table 5: Nine options on the implementation of the label-anchored mixup in Eq. (5). Except for the imbalanced labeled graphs $\mathcal{G}_{\text{imb}}$, the additional source of the interval representation $\mathbf{z}_i$ and the real graph representation $\mathbf{h}_j$ could be $\mathcal{G}_{\text{conf}}$ or $\mathcal{G}_{\text{unlbl}}$. We find that source $\mathbf{z}_i$ from $\mathcal{G}_{\text{imb}}$ and source $\mathbf{h}_j$ from $\mathcal{G}_{\text{unlbl}}$ are usually the best.

Additional Source		Plym-Oxygen			
$\mathbf{z}_i$	$\mathbf{h}_j$	All	Many-shot	Med.-shot	Few-shot
None	None	165.5(12.2)	4.7(1.7)	16.5(7.2)	417.4(31.1)
None	$\mathcal{G}_{\text{conf}}$	158.1(17.0)	4.1(0.7)	**11.3**(0.7)	401.9(45.1)
None	$\mathcal{G}_{\text{unlbl}}$	**150.9**(17.8)	**3.8**(1.1)	12.2(0.6)	**382.8**(46.9)
$\mathcal{G}_{\text{conf}}$	None	166.0(18.2)	11.9(11.3)	12.6(0.9)	414.0(52.6)
$\mathcal{G}_{\text{conf}}$	$\mathcal{G}_{\text{conf}}$	158.8(8.4)	7.7(8.9)	15.4(7.8)	397.5(15.4)
$\mathcal{G}_{\text{conf}}$	$\mathcal{G}_{\text{unlbl}}$	169.5(56.1)	4.5(1.2)	12.7(1.8)	430.4(145.0)
$\mathcal{G}_{\text{unlbl}}$	None	173.1(30.3)	3.7(0.4)	13.5(1.4)	440.0(79.3)
$\mathcal{G}_{\text{unlbl}}$	$\mathcal{G}_{\text{conf}}$	174.5(9.3)	8.1(3.3)	11.9(0.9)	440.4(25.5)
$\mathcal{G}_{\text{unlbl}}$	$\mathcal{G}_{\text{unlbl}}$	156.3(20.5)	8.2(2.9)	12.9(0.9)	392.3(50.6)

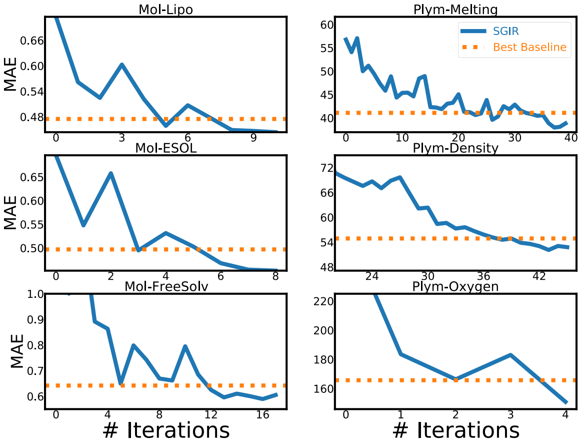
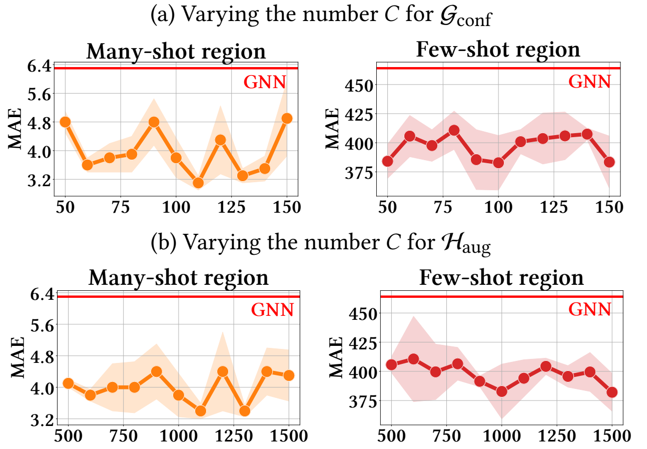


Figure 4: SGIR test performance (entire label range) through multiple self-training iterations. MAE for Plym-Density is scaled by ×1, 000. The iterative self-training algorithm is effective for gradually improving the quality of training data.

approximates and outperforms the best baseline in the entire label range. It also indicates that the quality of the training data is steadily improved over iterations.

5.3.3 Effect of regression confidence measurements. Table 4 shows that compared to existing methods that could define regression confidence, the measurement we define and use, GRation, is the best option for evaluating the quality of pseudo-labels in graph regression tasks. Because GRation uses various environments subgraphs, which provide diverse perturbations for robust graph learning [24]. We also observe that Dropout can be a good alternative of GRation. Dropout has extensive assessments [12] and makes it possible for SGIR to be extended to regression tasks for other data modalities such as images and texts.

5.3.4 Effect of label-anchored mixup augmentation. We implement $\mathbf{z}_i$ using $\mathcal{G}_{\text{imb}}$ to improve the augmentation quality and $\mathcal{G}_{\text{imb}} \cup \mathcal{G}_{\text{unlbl}}$ to improve the diversity. Table 5 presents empirical studies to support our idea. It shows that when many noisy representation vectors from unlabeled graphs are included in the interval center $\mathbf{z}_i$, the quality of augmented examples is relatively low, which degrades the model performance in different label ranges. On the other hand, the representations of unlabeled graphs improve the diversity of the augmented examples when we assign low mixup weights to them as in Eq. (5). Considering both quality and diversity, the effectiveness of the algorithm is further demonstrated in Table 3 by significantly reducing the errors for rare labels. From the fifth line of each dataset in Table 3, we find that it is also promising to directly use the label-anchored mixup augmentation (as $\mathcal{G}_{\text{imb}} \cup \mathcal{H}_{\text{aug}}$) for data balancing. Although its performance may be inferior to the performance using $\mathcal{G}_{\text{imb}} \cup \mathcal{G}_{\text{conf}}$ (as the third line of each dataset in Table 3), the potential of the label-anchored mixup algorithm could be further enhanced by improving the quality of the augmented examples to close the gap with real molecular graphs.


(a) Varying the number C for $\mathcal{G}_{\text{conf}}$

(b) Varying the number C for $\mathcal{H}_{\text{aug}}$

Figure 5: Sensitivity analysis on the number of label intervals (C) for pseudo-labeling selection $\mathcal{G}_{\text{conf}}$ (top) and label-anchored mixup algorithm $\mathcal{H}_{\text{aug}}$ (bottom). Results are drawn on the Plym-Oxygen.

5.3.5 Sensitivity analysis for the label interval number C. We find the best values of C in main experiments using the validation set for pseudo-labeling and label-anchored mixup. We suggest setting the number C to approximately 100 for pseudo-labeling and around 1,000 for label-anchored mixup. Specifically, sensitivity analysis is conducted on the Plym-Oxygen dataset to analyze the effect of the number C. Results are presented in Figure 5. We observe that SGIR is robust to a wide range of choices for the number of intervals.

6 CONCLUSIONS

In this work, we explored a novel graph imbalanced regression task and improved semi-supervised learning on it. We proposed a self-training framework to gradually reduce the model bias of data imbalance through multiple iterations. In each iteration, we selected more high-quality pseudo-labels for rare label values and continued augmenting training data to approximate the perfectly balanced label distribution. Experiments demonstrated the effectiveness and reasonable design of the proposed framework, especially on material science.

ACKNOWLEDGMENTS

This work was supported in part by NSF IIS-2142827, IIS-2146761, CBET-2102592, and ONR N00014-22-1-2507.

REFERENCES

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence* 34, 11 (2012), 2274–2282.

[2] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. 2020. Deep evidential regression. *Advances in Neural Information Processing Systems* 33 (2020), 14927–14937.

[3] Peter L Bartlett and Shahar Mendelson. 2002. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research* 3, Nov (2002), 463–482.

[4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems* 32 (2019).

[5] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alex Kurakin. 2022. Adamatch: A unified approach to semi-supervised learning and domain adaptation. *International Conference on Learning Representations* (2022).

[6] Paula Branco, Luís Torgo, and Rita P Ribeiro. 2017. SMOGN: a pre-processing approach for imbalanced regression. In *First international workshop on learning with imbalanced domains: Theory and applications*. PMLR, 36–50.

[7] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems* 32 (2019).

[8] Luigi Carratino, Moustapha Cissé, Rodolphe Jenatton, and Jean-Philippe Vert. 2020. On mixup regularization. *arXiv preprint arXiv:2006.06049* (2020).

[9] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[10] Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. 2008. Sample selection bias correction theory. In *Algorithmic Learning Theory: 19th International Conference, ALT 2008, Budapest, Hungary, October 13-16, 2008. Proceedings 19*. Springer, 38–53.

[11] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9268–9277.

[12] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. PMLR, 1050–1059.

[13] Yu Gong, Greg Mori, and Frederick Tung. 2022. RankSim: Ranking Similarity Regularization for Deep Imbalanced Regression. *International Conference on Machine Learning* (2022).

[14] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 1025–1035.

[15] Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. 2022. G-Mixup: Graph Data Augmentation for Graph Classification. *arXiv preprint arXiv:2202.07179* (2022).

[16] Xinting Hu, Yulei Niu, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. 2022. On Non-Random Missing Labels in Semi-Supervised Learning. In *International Conference on Learning Representations*.

[17] Neal Jean, Sang Michael Xie, and Stefano Ermon. 2018. Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance. *Advances in Neural Information Processing Systems* 31 (2018).

[18] Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. 2008. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. *Advances in neural information processing systems* 21 (2008).

[19] Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. 2020. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. *Advances in Neural Information Processing Systems* 33 (2020), 14567–14579.

[20] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

[21] Boris Knyazev, Graham W Taylor, and Mohamed Amer. 2019. Understanding attention and generalization in graph neural networks. *Advances in neural information processing systems* 32 (2019).

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[23] Gang Liu, Eric Inae, Tong Zhao, Jiaxin Xu, Tengfei Luo, and Meng Jiang. 2023. Data-Centric Learning from Unlabeled Graphs with Diffusion Model. *arXiv preprint arXiv:2303.10108* (2023).

[24] Gang Liu, Tong Zhao, Jiaxin Xu, Tengfei Luo, and Meng Jiang. 2022. Graph Rationalization with Environment-based Augmentations. In *Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[25] Zixuan Liu, Ziqiao Wang, Hongyu Guo, and Yongyi Mao. 2023. Over-Training with Mixup May Hurt Generalization. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=JmkjrlVE-DG

[26] Ruimin Ma and Tengfei Luo. 2020. PI1M: a benchmark database for polymer informatics. *Journal of Chemical Information and Modeling* 60, 10 (2020), 4684–4690.

[27] Geoffrey J McLachlan. 1975. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *J. Amer. Statist. Assoc.* 70, 350 (1975), 365–369.

[28] David Mendez, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F Mosquera, Prudence Mutowo, Michał Nowotka, et al. 2019. ChEMBL: towards direct deposition of bioassay data. *Nucleic acids research* 47, D1 (2019), D930–D940.

[29] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. 2021. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*. https://openreview.net/forum?id=37nvvqkCo5

[30] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. 2017. Agedb: the first manually collected, in-the-wild age database. In *proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 51–59.

[31] Youngtaek Oh, Dong-Jin Kim, and In So Kweon. 2022. Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[32] Shingo Otsuka, Isao Kuwajima, Junko Hosoya, Yibin Xu, and Masayoshi Yamazaki. 2011. PoLyInfo: Polymer database for polymeric materials design. In *2011 International Conference on Emerging Intelligent Data and Web Technologies*. IEEE, 22–29.

[33] 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.

[34] Jiawei Ren, Mingyuan Zhang, Cunjun Yu, and Ziwei Liu. 2022. Balanced MSE for Imbalanced Visual Regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[35] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2019. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. *arXiv preprint arXiv:1907.10903* (2019).

[36] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems* 33 (2020), 596–608.

[37] Masashi Sugiyama and Amos J Storkey. 2006. Mixture regression for covariate shift. *Advances in neural information processing systems* 19 (2006).

[38] Baochen Sun, Jiashi Feng, and Kate Saenko. 2016. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[39] Fan-Yun Sun, Jordon Hoffman, Vikas Verma, and Jian Tang. 2020. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *International Conference on Learning Representations*.

[40] Natasa Tagasovska and David Lopez-Paz. 2019. Single-model uncertainties for deep learning. *Advances in Neural Information Processing Systems* 32 (2019).

[41] A Thornton, L Robeson, B Freeman, and D Uhlmann. 2012. Polymer Gas Separation Membrane Database.

[42] Junjiao Tian, Yen-Cheng Liu, Nathaniel Glaser, Yen-Chang Hsu, and Zsolt Kira. 2020. Posterior re-calibration for imbalanced datasets. *Advances in Neural Information Processing Systems* 33 (2020), 8101–8113.

[43] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[44] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*. PMLR, 6438–6447.

[45] Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. 2021. Mixup for node and graph classification. In *Proceedings of the Web Conference 2021*. 3663–3674.

[46] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. 2021. CREST: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10857–10866.

[47] Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat seng Chua. 2022. Discovering Invariant Rationales for Graph Neural Networks. In *ICLR*.

[48] 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.

[49] 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.

[50] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *International Conference on Learning Representations*. https://openreview.net/forum?id=ryGs6iA5Km

[51] Yuzhe Yang, Kaiwen Zha, Yingcong Chen, Hao Wang, and Dina Katabi. 2021. Delving into deep imbalanced regression. In *International Conference on Machine Learning*. PMLR, 11842–11851.

[52] Huaxiu Yao, Yiping Wang, Linjun Zhang, James Y Zou, and Chelsea Finn. 2022. C-mixup: Improving generalization in regression. *Advances in Neural Information Processing Systems* 35 (2022), 3361–3376.

[53] Qi Yuan, Mariagiulia Longo, Aaron W Thornton, Neil B McKeown, Bibiana Comesana-Gandara, Johannes C Jansen, and Kim E Jelfs. 2021. Imputation of missing gas permeability data for polymer membranes using machine learning. *Journal of Membrane Science* 627 (2021), 119207.

[54] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. 2021. How Does Mixup Help With Robustness and Generalization?. In *International Conference on Learning Representations*. https://openreview.net/forum?id=8yKEo06dKNo

[55] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. 2023. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[56] Tong Zhao, Tianwen Jiang, Neil Shah, and Meng Jiang. 2021. A synergistic approach for graph anomaly detection with pattern mining and feature learning. *IEEE Transactions on Neural Networks and Learning Systems* 33, 6 (2021), 2393–2405.

[57] Tong Zhao, Wei Jin, Yozen Liu, Yingheng Wang, Gang Liu, Stephan Günneman, Neil Shah, and Meng Jiang. 2022. Graph Data Augmentation for Graph Machine Learning: A Survey. *arXiv preprint arXiv:2202.08871* (2022).

[58] Tong Zhao, Gang Liu, Daheng Wang, Wenhao Yu, and Meng Jiang. 2022. Learning from Counterfactual Links for Link Prediction. In *International Conference on Machine Learning*. PMLR, 26911–26926.

[59] Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. 2021. Data Augmentation for Graph Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 11015–11023.

A PROOFS OF THEORETICAL MOTIVATIONS

We rely on two theorems to derive theorem 4.1.

A.1 Existing Theorems

Given a classifier f from the function class $\mathcal{F}$, an input example x from the feature space $\mathcal{X}$ and its label y.

Theorem A.1 (from [3, 18]). *Assume the expected loss on examples is $\mathcal{E}[f]$ and the corresponding empirical loss $\hat{\mathcal{E}}[f]$. Assume the loss is Lipschitz with Lipschitz constant L_e. And it is bounded by c_0. For any $\delta > 0$ and with probability at least $1 - \delta$ simultaneously for all $f \in \mathcal{F}$ we have that*

$$\mathcal{E}[f] \leq \hat{\mathcal{E}}[f] + 2L_e\mathcal{R}_n(\mathcal{F}) + c_0\sqrt{\frac{\log(1/\delta)}{2n}}, \tag{10}$$

where n is the number of example and $\hat{\mathcal{R}}_n(\mathcal{F})$ is the Rademacher complexity measurement of the hypothesis class $\mathcal{F}$.

Theorem A.2 (from [18]). *Applying theorem A.1 and considering the fraction of data having γ-margin mistakes, or $K_\gamma[f] := \frac{|i:y_i f(x_i)<\gamma|}{n}$. Assume $\forall f \in \mathcal{F}$ we have $\sup_{x\in\mathcal{X}} |f(x)| \leq c_1$. Then, with probability at least $1 - \delta$ over the example, for all margins $\gamma > 0$ and all $f \in \mathcal{F}$ we have,*

$$\mathcal{E}[f] \leq K_\gamma[f] + 4\frac{\mathcal{R}_n(\mathcal{F})}{\gamma} + \sqrt{\frac{2\log\left(\log_2(4c_1/\gamma)\right) + \log(1/\delta)}{2n}}, \tag{11}$$

$$\leq K_\gamma[f] + 4\frac{\mathcal{R}_n(\mathcal{F})}{\gamma} + \sqrt{\frac{\log\left(\log_2\frac{4c_1}{\gamma}\right)}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}}. \tag{12}$$

A.2 Proof of theorem 4.1

In our work, we use the regression function f to predict the label value. We calculate the reciprocal of the distance between the predicted label and interval centers as unnormalized probabilities of the graph $S_{[b_i,b_{i+1})}(G)$ being assigned to the interval $[b_i, b_{i+1})$, $i \in \{1, 2, \ldots, C\}$. Given a hard margin γ, we use $\mathcal{E}_{\gamma,[b_i,b_{i+1})}[f]$ to denote the hard margin loss for examples in the interval $[b_i, b_{i+1})$:

$$\mathcal{E}_{\gamma,[b_i,b_{i+1})}[f] = \Pr_{G\sim\mathcal{P}_{[b_i,b_{i+1})}}\left[S_{[b_i,b_{i+1})}(G) < \max_{j\neq i} S_{[b_j,b_{j+1})}(G) + \gamma\right]. \tag{13}$$

We assume its empirical variant is $\hat{\mathcal{E}}_{\gamma,[b_i,b_{i+1})}[f]$. The empirical Rademacher complexity $\hat{\mathcal{R}}_{(b_i,b_{i+1}]}(\mathcal{F})$ is used as the complexity measurement $\mathrm{C}(\mathcal{F})$ for the hypothesis class $\mathcal{F}$. With a vector σ of i.i.d. uniform $\{-1, +1\}$ bits, we have

$$\hat{\mathcal{R}}_{(b_i,b_{i+1}]}(\mathcal{F}) = \tag{14}$$

$$\frac{1}{n_{(b_i,b_{i+1}]}}\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\sum_{G_i\in[b_i,b_{i+1})}\sigma_i\left[S_{[b_i,b_{i+1})}(G_i) - \max_{j\neq i} S_{[b_j,b_{j+1})}(G_i)\right]\right] \tag{15}$$

As any G_i in the interval $(b_i, b_{i+1}]$ is an i.i.d. sample from the distribution $\mathcal{P}_{[b_i,b_{i+1})}$, we directly apply the standard margin-based generalization bound theorem A.2 [18]: with probability $1 - \delta$, for all choices of $\gamma_{[b_i,b_{i+1})} > 0$ and $f \in \mathcal{F}$,

$$\mathcal{E}_{[b_i,b_{i+1})} \leq \hat{\mathcal{E}}_{\gamma,[b_i,b_{i+1})}[f] + 4\frac{\hat{\mathcal{R}}_{(b_i,b_{i+1}]}(\mathcal{F})}{\gamma_{[b_i,b_{i+1})}} \tag{16}$$

$$+ \sqrt{\frac{2\log\left(\log_2\left(\frac{4c_1}{\gamma_{[b_i,b_{i+1})}}\right)\right) + \log(1/\delta)}{2n_{[b_i,b_{i+1})}}},$$

$$\leq \hat{\mathcal{E}}_{\gamma,[b_i,b_{i+1})}[f] + \frac{1}{\gamma_{[b_i,b_{i+1})}}\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_{[b_i,b_{i+1})}}} \tag{17}$$

$$+ \sqrt{\frac{2\log\left(\log_2\left(\frac{4c_1}{\gamma_{[b_i,b_{i+1})}}\right)\right)\log(1/\delta)}{2n_{[b_i,b_{i+1})}}},$$

$$\lessapprox \frac{1}{\gamma_{[b_i,b_{i+1})}}\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_{[b_i,b_{i+1})}}} + \sqrt{\frac{\log\log_2(1/\gamma_{[b_i,b_{i+1})}) + \log(1/\delta)}{2n_{[b_i,b_{i+1})}}}. \tag{18}$$

We derive Eq. (17) from Eq. (16) because the Rademacher complexity $\hat{\mathcal{R}}_{(b_i,b_{i+1}]}(\mathcal{F})$ typically scales as $\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_{(b_i,b_{i+1}]}}}$ for some complexity measurement $\mathrm{C}(\mathcal{F})$ [7]. We derive Eq. (18) from Eq. (17) by ignoring constant factors [7]. Since the overall performance $\mathcal{E}_{\text{bal}}[f]$ is calculated over all intervals, we get it as $\mathcal{E}_{\text{bal}}[f] = \frac{1}{C}\sum_{i=1}^{C}\mathcal{E}_{[b_i,b_{i+1})}$.

A.3 Discussions

Existing work on the theoretical analysis of mixup [8, 25, 54] mainly focused on image classification and the augmented graph examples are not i.i.d sampled from the conditional distribution $\mathcal{P}_{[b_i,b_{i+1}]}$ for a specific interval $[b_i, b_{i+1}]$. While a recent work proposed the C-mixup [52] to sample closer pairs of examples with higher probability for regression tasks, it did not fit our theoretical motivation to address the label imbalance issue: with C-mixup, the pairs in the over-represented label ranges have a higher probability to be sampled than the under-represented ones. Compared to these theories and methods for the mixup algorithm, our label-anchored mixup allows direct application to imbalanced regression tasks without compromising the assumption in our theoretical motivation. This is because we use the augmented virtual examples $\mathcal{H}_{\text{aug}}$ based on the label anchor within intervals $[b_i, b_{i+1}]$. Augmented examples are independently created with Eq. (5). Since the interval centers could be mixed with any other real graphs from $\mathcal{G}_{\text{imb}} \cup \mathcal{G}_{\text{conf}}$, any value in the interval space could be sampled. Besides, it is reasonable to use the distribution of the entire label space (from $\mathcal{G}_{\text{imb}} \cup \mathcal{G}_{\text{conf}}$) to approximate the distribution within the interval and assume that the conditional distribution $\mathcal{P}_{[b_i,b_{i+1}]}$ does not change.

We build the theoretical principle for imbalanced regression with intervals to connect with existing theoretical principles for classification. Future theoretical work on imbalanced regression can leverage the advantages of using mixture regressor models [37], which have been used to address covariate shift problems in regression tasks. Additionally, exploring the promising connection between domain adaptation theories and sample selection bias [10, 38] holds potential for further advancements in this field.

Table 6: Statistics of six tasks for graph property regression.

Dataset	# Graphs (Train/Valid/Test)	# Nodes (Avg./Max)	# Edges (Avg./Max)
Mol-Lipo	2,048 / 1,076 / 1,076	27.0 / 115	59.0 / 236
Mol-ESOL	446 / 341 / 341	13.3 / 55	27.4 / 125
Mol-FreeSolv	276 / 183 / 183	8.7 / 24	16.8 / 50
Plym-Melting	2,419 / 616 / 616	26.9 / 102	55.4 / 212
Plym-Density	844 / 425 / 425	27.3 / 93	57.6 / 210
Plym-Oxygen	339 / 128 / 128	37.3 / 103	82.1 / 234
Superpixel-Age	3619 / 628 / 628	67.9 / 75.0	265.6 / 300

B EXPERIMENTS

B.1 Dataset Details

We give a comprehensive introduction to our datasets used for regression tasks and splitting idea from [13, 51].

Mol-Lipo predicts the property of lipophilicity consisting of 4200 molecules. The lipophilicity is important for solubility and membrane permeability in drug molecules. This dataset originates from ChEMBL [28]. The property is from experimental results for the octanol/water distribution coefficient (log D at pH 7.4). *Mol-ESOL* predicts the water solubility (log solubility in mols per litre) from chemical structures consisting of 1128 small organic molecules. *Mol-FreeSolv* predicts the hydration free energy of molecules in water consisting of 642 molecules. The property is experimentally measured or calculated. *Plym-Melting* predicts the property of melting temperature (°C). It is collected from PolyInfo, a web-based polymer database [32]. *Plym-Density* predicts the property of polymer density (g/cm^3). It is collected from PolyInfo, a web-based polymer database [32]. *Plym-Oxygen* predicts the property of oxygen permeability (Barrer). It is created from the Membrane Society of Australasia portal consisting of experimentally measured gas permeability data [41]. *Unlabeled Data for Molecules and Polymers* The total number of unlabeled graphs for molecule and polymers is 146,129, consisting of 133,015 molecules from QM9 [33] and 13,114 monomers (the repeated units of polymers) from [24]. QM9 is a molecule dataset for stable small organic molecules consisting of atoms C, H, O, N, and F. We use it as a source of unlabeled data. We integrate four polymer regression datasets including Plym-Melting, Plym-Density, Plym-Oxygen and another one from [24] for the glass transition temperature as the other source of unlabeled data. We note that the unlabeled graphs may be slightly less than 146,129 for a polymer task on Plym-Melting, Plym-Density or Plym-Oxygen. It is because we remove the overlapping graphs for the current polymer task with the polymer unlabeled data.

Data splitting for Molecules and Polymers We split the datasets based on the approach in previous works [13, 51] motivated for two reasons. First, we want the training sets to well characterize the imbalanced label distribution as presented in the original datasets. Second, we want relatively balanced valid and test sets to fairly evaluate the model performance in different ranges of label values.

Superpixel-Age The details of the age regression dataset are presented in Table 6 (Superpixel-Age) and Figure 3. The graph dataset Superpixel-Age is constructed from image superpixels using the algorithms from [21] on the image dataset *AgeDB-DIR* from [30, 51]. Each face image in *AgeDB-DIR* has an age label from 0 to 101. We fisrt compute the SLIC superpixels for each image without losing the label-specific information [1, 21]. Then we use the superpixels as nodes and calculate the spatial distance between superpixels to build edges for each image [21]. Binary edges are constructed between superpixel nodes by applying a threshold on the top-5% of the smallest spatial distances. After building a graph for each image, we follow the data splitting in [51] to study the imbalanced regression problem. We randomly remove 70% labels in the training/validation/test data and use them as unlabeled graphs. Finally, the graph dataset Superpixel-Age consists of 3,619 graphs for training, 628 graphs for validation, 628 graphs for testing, and 11,613 unlabeled graphs for semi-supervised learning.

B.2 Implementation Details

We use the Graph Isomorphism Network (GIN) [50] as the GNN encoder for f_θ to get the graph representation and three layers of Multilayer perceptron (MLP) as the decoder to predict graph properties. The threshold τ for selecting confident predictions is determined by the value at a certain percentile of the confidence score distribution. To implement it, we set it up as a hyperparameter τ_{pct} determining the percentile value of the prediction variance (*i.e.,* the reciprocal of confidence) of the labeled training data. All methods are implemented on a RTX 2080Ti card (11 GB RAM). We reports the results on the test sets using the mean (standard deviation) over 10 runs. Note that the underlying design of the graph learning model used in SGIR is GREA with a learning objective as follows. Given $(G, y) \in \mathcal{G}_{\text{imb}} \cup \mathcal{G}_{\text{conf}}$, GREA [24] will output a vector $\mathbf{m} \in \mathbb{R}^K$ that indicates the probability of K nodes in a graph being in the rationale subgraph. So, we could get $\mathbf{h}^{(r)} = \mathbf{1}_K^\top \cdot (\mathbf{m} \times \mathbf{H})$ and $\mathbf{h}^{(e)} = \mathbf{1}_K^\top \cdot ((\mathbf{1}_K - \mathbf{m}) \times \mathbf{H})$, where $\mathbf{H} \in \mathbb{R}^{K \times d}$ is the node representation matrix. By this, the optimization objectives of a graph consist of

$$\begin{cases} \ell_{\text{imb+conf}} = \text{MAE}(f(\mathbf{h}^{(r)}), y) + \mathbb{E}_{G'} \left[\text{MAE}(f(\mathbf{h} + \mathbf{h}'), y) \right] \\ \qquad\qquad + \text{Var}_{G'} \left(\{ \text{MAE}(f(\mathbf{h} + \mathbf{h}'), y) \} \right), \\ \\ \ell_{\text{regu}} = \frac{1}{K} \sum_{k=1}^{K} |\mathbf{m}_k| - \gamma \end{cases}$$

ℓ_{regu} regularizes the vector $\mathbf{m}$ and $\gamma \in [0, 1]$ is a hyperparameter to control the expected size of $G^{(r)}$. G' is the possible graph in the same batch that provides environment subgraphs and $\mathbf{h}'$ is the representation vector of the environment subgraph. When combining the rationale-environment pairs to create new graph examples, the original GREA creates the same number of examples for the under-represented rationale and the well/over-represented rationale. We observe that it may make the training examples more imbalanced. Therefore, we use the reweighting technique to penalize more for the expectation term ($\mathbb{E}_{G'} \left[\text{MAE}(f(\mathbf{h} + \mathbf{h}'), y) \right]$) and variance term ($\text{Var}_{G'} \left(\{ \text{MAE}(f(\mathbf{h} + \mathbf{h}'), y) \} \right)$) in $\ell_{\text{imb+conf}}$ when the label is from the under-represented ranges. The weight of the expectation and variance terms for a graph with label y is

$$w = \frac{\exp(\sum_{b=1}^{B} |y - y_b| / t)}{\exp(\sum_{j=1}^{B} \sum_{b=1}^{B} |y - y_b| / t)},$$

where B is the batch size and t is the temperature hyper-parameter.