

# Delving into Deep Imbalanced Regression

Yuzhe Yang<sup>1</sup> Kaiwen Zha<sup>1</sup> Ying-Cong Chen<sup>1</sup> Hao Wang<sup>2</sup> Dina Katabi<sup>1</sup>

## Abstract

Real-world data often exhibit imbalanced distributions, where certain target values have significantly fewer observations. Existing techniques for dealing with imbalanced data focus on targets with categorical indices, i.e., different classes. However, many tasks involve continuous targets, where hard boundaries between classes do not exist. We define Deep Imbalanced Regression (DIR) as learning from such imbalanced data with continuous targets, dealing with potential missing data for certain target values, and generalizing to the entire target range. Motivated by the intrinsic difference between categorical and continuous label space, we propose distribution smoothing for both labels and features, which explicitly acknowledges the effects of nearby targets, and calibrates both label and learned feature distributions. We curate and benchmark large-scale DIR datasets from common real-world tasks in computer vision, natural language processing, and healthcare domains. Extensive experiments verify the superior performance of our strategies. Our work fills the gap in benchmarks and techniques for practical imbalanced regression problems. Code and data are available at: <https://github.com/YyzHarry/imbalanced-regression>.

## 1. Introduction

Data imbalance is ubiquitous and inherent in the real world. Rather than preserving an ideal uniform distribution over each category, the data often exhibit skewed distributions with a long tail (Buda et al., 2018; Liu et al., 2019), where certain target values have significantly fewer observations. This phenomenon poses great challenges for deep recognition models, and has motivated many prior techniques for addressing data imbalance (Cao et al., 2019; Cui et al., 2019; Huang et al., 2019; Liu et al., 2019; Tang et al., 2020).

<sup>1</sup>MIT Computer Science & Artificial Intelligence Laboratory

<sup>2</sup>Department of Computer Science, Rutgers University. Correspondence to: Yuzhe Yang <yuzhe@mit.edu>.

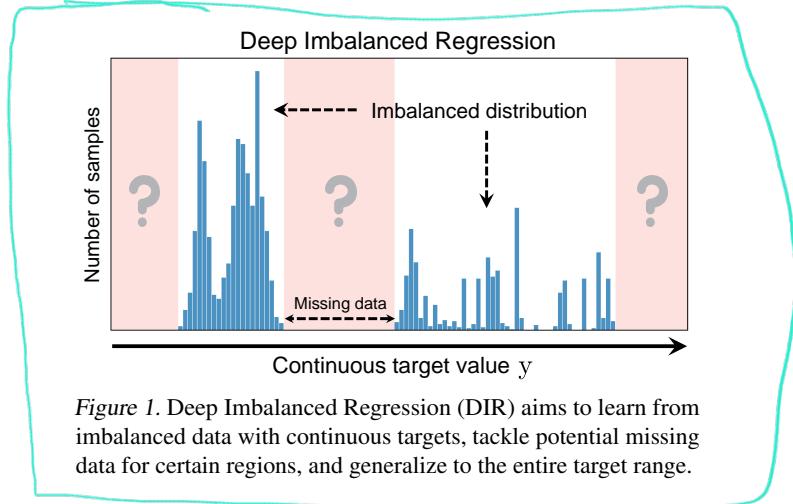


Figure 1. Deep Imbalanced Regression (DIR) aims to learn from imbalanced data with continuous targets, tackle potential missing data for certain regions, and generalize to the entire target range.

Existing solutions for learning from imbalanced data, however, focus on targets with categorical indices, i.e., the targets are different classes. However, many real-world tasks involve continuous and even infinite target values. For example, in vision applications, one needs to infer the age of different people based on their visual appearances, where age is a continuous target and can be highly imbalanced. Treating different ages as distinct classes is unlikely to yield the best results because it does not take advantage of the similarity between people with nearby ages. Similar issues happen in medical applications since many health metrics including heart rate, blood pressure, and oxygen saturation, are continuous and often have skewed distributions across patient populations.

In this work, we systematically investigate Deep Imbalanced Regression (DIR) arising in real-world settings (see Fig. 1). We define DIR as learning continuous targets from natural imbalanced data, dealing with potentially missing data for certain target values, and generalizing to a test set that is balanced over the entire range of continuous target values. This definition is analogous to the class imbalance problem (Liu et al., 2019), but focuses on the continuous setting.

DIR brings new challenges distinct from its classification counterpart. First, given continuous (potentially infinite) target values, the hard boundaries between classes no longer exist, causing ambiguity when directly applying traditional imbalanced classification methods such as re-sampling and re-weighting. Moreover, continuous labels inherently possess a meaningful distance between targets, which has im-

plication for how we should interpret data imbalance. For example, say two target labels  $t_1$  and  $t_2$  have a small number of observations in training data. However,  $t_1$  is in a highly represented neighborhood (i.e., there are many samples in the range  $[t_1 - \Delta, t_1 + \Delta]$ ), while  $t_2$  is in a weakly represented neighborhood. In this case,  $t_1$  does not suffer from the same level of imbalance as  $t_2$ . Finally, unlike classification, certain target values may have no data at all, which motivates the need for target extrapolation & interpolation.

In this paper, we propose two simple yet effective methods for addressing DIR: label distribution smoothing (LDS) and feature distribution smoothing (FDS). A key idea underlying both approaches is to leverage the similarity between nearby targets by employing a kernel distribution to perform explicit distribution smoothing in the label and feature spaces. Both techniques can be easily embedded into existing deep networks and allow optimization in an end-to-end fashion. We verify that our techniques not only successfully calibrate for the intrinsic underlying imbalance, but also provide large and consistent gains when combined with other methods.

To support practical evaluation of imbalanced regression, we curate and benchmark large-scale DIR datasets for common real-world tasks in computer vision, natural language processing, and healthcare. They range from single-value prediction such as age, text similarity score, health condition score, to dense-value prediction such as depth. We further set up benchmarks for proper DIR performance evaluation.

Our contributions are as follows:

- We formally define the DIR task as learning from imbalanced data with continuous targets, and generalizing to the entire target range. DIR provides thorough and unbiased evaluation of learning algorithms in practical settings.
- We develop two simple, effective, and interpretable algorithms for DIR, LDS and FDS, which exploit the similarity between nearby targets in both label and feature space.
- We curate benchmark DIR datasets in different domains: computer vision, natural language processing, and healthcare. We set up strong baselines as well as benchmarks for proper DIR performance evaluation.
- Extensive experiments on large-scale DIR datasets verify the consistent and superior performance of our strategies.

## 2. Related Work

**Imbalanced Classification.** Much prior work has focused on the imbalanced classification problem (also referred to as long-tailed recognition (Liu et al., 2019)). Past solutions can be divided into data-based and model-based solutions: Data-based solutions either over-sample the minority class or under-sample the majority (Chawla et al., 2002; García & Herrera, 2009; He et al., 2008). For example, SMOTE generates synthetic samples for minority classes by linearly

interpolating samples in the same class (Chawla et al., 2002).

Model-based solutions include re-weighting or adjusting the loss function to compensate for class imbalance (Cao et al., 2019; Cui et al., 2019; Dong et al., 2019; Huang et al., 2016; 2019), and leveraging relevant learning paradigms, including transfer learning (Yin et al., 2019), metric learning (Zhang et al., 2017), meta-learning (Shu et al., 2019), and two-stage training (Kang et al., 2020). Recent studies have also discovered that semi-supervised learning and self-supervised learning lead to better imbalanced classification results (Yang & Xu, 2020). In contrast to these past work, we identify the limitations of applying class imbalance methods to regression problems, and introduce new techniques particularly suitable for learning continuous target values.

**Imbalanced Regression.** Regression over imbalanced data is not as well explored. Most of the work on this topic is a direct adaptation of the SMOTE algorithm to regression scenarios (Branco et al., 2017; 2018; Torgo et al., 2013). Synthetic samples are created for pre-defined rare target regions by either directly interpolating both inputs and targets (Torgo et al., 2013), or using Gaussian noise augmentation (Branco et al., 2017). A bagging-based ensemble method that incorporates multiple data pre-processing steps has also been introduced (Branco et al., 2018). However, there exist several intrinsic drawbacks for these methods. First, they fail to take the distance between targets into account, and rather heuristically divide the dataset into rare and frequent sets, then plug in classification-based methods. Moreover, modern data is of extremely high dimension (e.g., images and physiological signals); linear interpolation of two samples of such data does not lead to meaningful new synthetic samples. Our methods are intrinsically different from past work in their approach. They can be combined with existing methods to improve their performance, as we show in Sec. 4. Further, our approaches are tested on large-scale real-world datasets in computer vision, NLP, and healthcare.

## 3. Methods

**Problem Setting.** Let  $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$  be a training set, where  $\mathbf{x}_i \in \mathbb{R}^d$  denotes the input and  $y_i \in \mathbb{R}$  is the label, which is a continuous target. We introduce an additional structure for the label space  $\mathcal{Y}$ , where we divide  $\mathcal{Y}$  into  $B$  groups (bins) with equal intervals, i.e.,  $[y_0, y_1], [y_1, y_2], \dots, [y_{B-1}, y_B]$ .

Throughout the paper, we use  $b \in \mathcal{B}$  to denote the group index of the target value, where  $\mathcal{B} = \{1, \dots, B\} \subset \mathbb{Z}^+$  is the index space. In practice, the defined bins reflect a minimum resolution we care for grouping data in a regression task. For instance, in age estimation, we could define  $\delta y \triangleq y_{b+1} - y_b = 1$ , showing a minimum age difference of 1 is of interest. Finally, we denote  $\mathbf{z} = f(\mathbf{x}; \theta)$  as the feature for  $\mathbf{x}$ , where  $f(\mathbf{x}; \theta)$  is parameterized by a deep neural network model with parameter  $\theta$ . The final prediction  $\hat{y}$  is given by a regression function  $g(\cdot)$  that operates over  $\mathbf{z}$ .

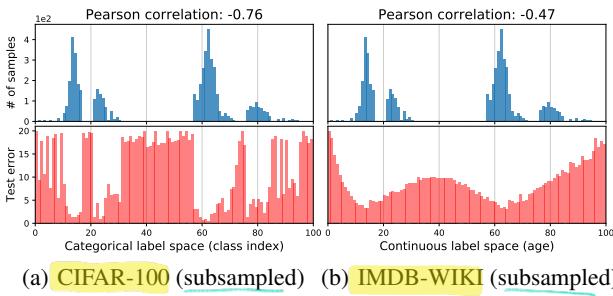


Figure 2. Comparison on the test error distribution (bottom) using same training label distribution (top) on two different datasets: (a) CIFAR-100, a classification task with categorical label space. (b) IMDB-WIKI, a regression task with continuous label space.

### 3.1. Label Distribution Smoothing

We start by showing an example to demonstrate the difference between classification and regression when imbalance comes into the picture.

**Motivating Example.** We employ two datasets: (1) CIFAR-100 (Krizhevsky et al., 2009), which is a 100-class classification dataset, and (2) the IMDB-WIKI dataset (Rothe et al., 2018), which is a large-scale image dataset for age estimation from visual appearance. The two datasets have intrinsically different label space: CIFAR-100 exhibits *categorical label space* where the target is class index, while IMDB-WIKI has a *continuous label space* where the target is age. We limit the age range to  $0 \sim 99$  so that the two datasets have the same label range, and *subsample them to simulate data imbalance*, while ensuring they have exactly the same label density distribution (Fig. 2). We make both test sets balanced. We then train a plain ResNet-50 model on the two datasets, and plot their test error distributions.

We observe from Fig. 2(a) that the error distribution *correlates with label density distribution*. Specifically, the test error as a function of class index has a high negative Pearson correlation with the label density distribution (i.e.,  $-0.76$ ) in the categorical label space. The phenomenon is expected, as majority classes with more samples are better learned than minority classes. Interestingly however, as Fig. 2(b) shows, the error distribution is very different for IMDB-WIKI with continuous label space, even when the label density distribution is the same as CIFAR-100. In particular, the error distribution is much smoother and no longer correlates well with the label density distribution ( $-0.47$ ).

The reason why this example is interesting is that all imbalanced learning methods, directly or indirectly, operate by compensating for the imbalance in the *empirical label density distribution*. This works well for class imbalance, but for continuous labels the empirical density does not accurately reflect the imbalance as seen by the neural network. Hence, compensating for data imbalance based on empirical label density is inaccurate for the continuous label space.

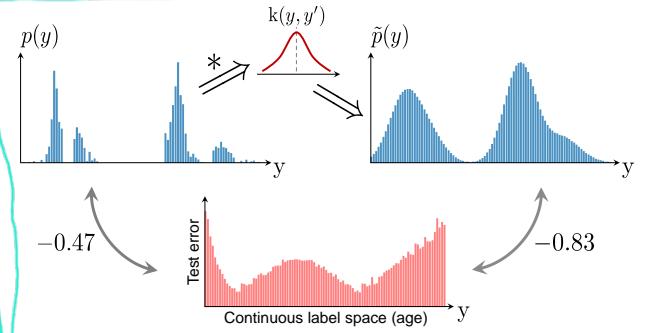


Figure 3. Label distribution smoothing (LDS) convolves a symmetric kernel with the empirical label density to estimate the effective label density distribution that accounts for the continuity of labels.

**LDS for Imbalanced Data Density Estimation.** The above example shows that, in the continuous case, the empirical label distribution does not reflect the real label density distribution. This is because of the dependence between data samples at nearby labels (e.g., images of close ages). In fact, there is a significant literature in statistics on how to estimate the expected density in such cases (Parzen, 1962). Thus, Label Distribution Smoothing (LDS) advocates the use of kernel density estimation to learn the effective imbalance in datasets that corresponds to continuous targets.

LDS convolves a symmetric kernel with the empirical density distribution to extract a kernel-smoothed version that accounts for the overlap in information of data samples of nearby labels. A symmetric kernel is any kernel that satisfies:  $k(y, y') = k(y', y)$  and  $\nabla_y k(y, y') + \nabla_{y'} k(y', y) = 0$ ,  $\forall y, y' \in \mathcal{Y}$ . Note that a Gaussian or a Laplacian kernel is a symmetric kernel, while  $k(y, y') = yy'$  is not. The symmetric kernel characterizes the similarity between target values  $y'$  and any  $y$  w.r.t. their distance in the target space. Thus, LDS computes the *effective label density distribution* as:

$$\tilde{p}(y') \triangleq \int_{\mathcal{Y}} k(y, y') p(y) dy, \quad (1)$$

where  $p(y)$  is the number of appearances of label of  $y$  in the training data, and  $\tilde{p}(y')$  is the effective density of label  $y'$ .

Fig. 3 illustrates LDS and how it smooths the label density distribution. Further, it shows that the resulting label density computed by LDS correlates well with the error distribution ( $-0.83$ ). This demonstrates that LDS captures the real imbalance that affects regression problems.

Now that the effective label density is available, techniques for addressing class imbalance problems can be directly adapted to the DIR context. For example, a straightforward adaptation can be the cost-sensitive re-weighting method, where we re-weight the loss function by multiplying it by the inverse of the LDS estimated label density for each target.

We show in Sec. 4 that LDS can be seamlessly incorporated with a wide range of techniques to boost DIR performance.

### 3.2. Feature Distribution Smoothing

We are motivated by the intuition that continuity in the target space should create a corresponding continuity in the feature space. That is, if the model works properly and the data is balanced, one expects the feature statistics corresponding to nearby targets to be close to each other.

**Motivating Example.** We use an illustrative example to highlight the impact of data imbalance on feature statistics in DIR. Again, we use a plain model trained on the images in the IMDB-WIKI dataset to infer a person's age from visual appearance. We focus on the learned feature space, i.e.,  $\mathbf{z}$ . We use a minimum bin size of 1, i.e.,  $y_{b+1} - y_b = 1$ , and group features with the same target value in the same bin. We then compute the feature statistics (i.e., mean and variance) with respect to the data in each bin, which we denote as  $\{\mu_b, \sigma_b\}_{b=1}^B$ . To visualize the similarity between feature statistics, we select an anchor bin  $b_0$ , and calculate the cosine similarity of the feature statistics between  $b_0$  and all other bins. The results are summarized in Fig. 4 for  $b_0 = 30$ . The figure also shows the regions with different data densities using the colors purple, yellow, and pink.

Fig. 4 shows that the feature statistics around  $b_0 = 30$  are highly similar to their values at  $b_0 = 30$ . Specifically, the cosine similarity of the feature mean and feature variance for all bins between age 25 and 35 are within a few percent from their values at age 30 (the anchor age). Further, the similarity gets higher for tighter ranges around the anchor. Note that bin 30 falls in the high shot region. In fact, it is among the few bins that have the most samples. So, the figure confirms the intuition that when there is enough data, and for continuous targets, the feature statistics are similar to nearby bins. Interestingly, the figure also shows the problem with regions that have very few data samples, like the age range 0 to 6 years (shown in pink). Note that the mean and variance in this range show unexpectedly high similarity to age 30. In fact, it is shocking that the feature statistics at age 30 are more similar to age 1 than age 17. This unjustified similarity is due to data imbalance. Specifically, since there are not enough images for ages 0 to 6, this range thus inherits its priors from the range with the maximum amount of data, which is the range around age 30.

**FDS Algorithm.** Inspired by these observations, we propose feature distribution smoothing (FDS), which performs distribution smoothing on the feature space, i.e., transfers the feature statistics between nearby target bins. This procedure aims to calibrate the potentially biased estimates of feature distribution, especially for underrepresented target values (e.g., medium- and few-shot groups) in training data.

FDS is performed by first estimating the statistics of each bin. Without loss of generality, we substitute variance with covariance to reflect also the relationship between the vari-

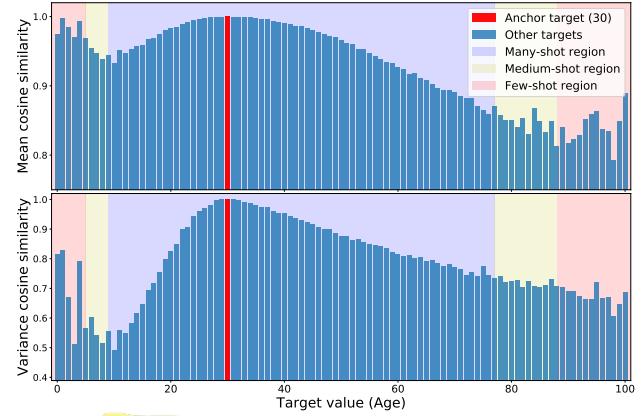


Figure 4. Feature statistics similarity for age 30. **Top:** Cosine similarity of the feature mean at a particular age w.r.t. its value at the anchor age. **Bottom:** Cosine similarity of the feature variance at a particular age w.r.t. its value at the anchor age. The color of the background refers to the data density in a particular target range. The figure shows that nearby ages have close similarities; However, it also shows that there is unjustified similarity between images at ages 0 to 6 and age 30, due to data imbalance.

ous feature elements within  $\mathbf{z}$ :

$$\boldsymbol{\mu}_b = \frac{1}{N_b} \sum_{i=1}^{N_b} \mathbf{z}_i, \quad (2)$$

$$\boldsymbol{\Sigma}_b = \frac{1}{N_b - 1} \sum_{i=1}^{N_b} (\mathbf{z}_i - \boldsymbol{\mu}_b)(\mathbf{z}_i - \boldsymbol{\mu}_b)^\top, \quad (3)$$

where  $N_b$  is the total number of samples in  $b$ -th bin. Given the feature statistics, we employ again a symmetric kernel  $k(y_b, y_{b'})$  to smooth the distribution of the feature mean and covariance over the target bins  $\mathcal{B}$ . This results in a smoothed version of the statistics:

$$\tilde{\boldsymbol{\mu}}_b = \sum_{b' \in \mathcal{B}} k(y_b, y_{b'}) \boldsymbol{\mu}_{b'}, \quad (4)$$

$$\tilde{\boldsymbol{\Sigma}}_b = \sum_{b' \in \mathcal{B}} k(y_b, y_{b'}) \boldsymbol{\Sigma}_{b'}. \quad (5)$$

With both  $\{\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\}$  and  $\{\tilde{\boldsymbol{\mu}}_b, \tilde{\boldsymbol{\Sigma}}_b\}$ , we then follow the standard whitening and re-coloring procedure (Sun et al., 2016) to calibrate the feature representation for each input sample:

$$\tilde{\mathbf{z}} = \tilde{\boldsymbol{\Sigma}}_b^{-\frac{1}{2}} (\mathbf{z} - \boldsymbol{\mu}_b) + \tilde{\boldsymbol{\mu}}_b. \quad (6)$$

We integrate FDS into deep networks by inserting a feature calibration layer after the final feature map. To train the model, we employ a *momentum update* of the running statistics  $\{\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\}$  across each epoch. Correspondingly, the smoothed statistics  $\{\tilde{\boldsymbol{\mu}}_b, \tilde{\boldsymbol{\Sigma}}_b\}$  are updated across different epochs but fixed within each training epoch. The momentum update, which performs an exponential moving average (EMA) of running statistics, results in more stable and accurate estimations of the feature statistics during training. The

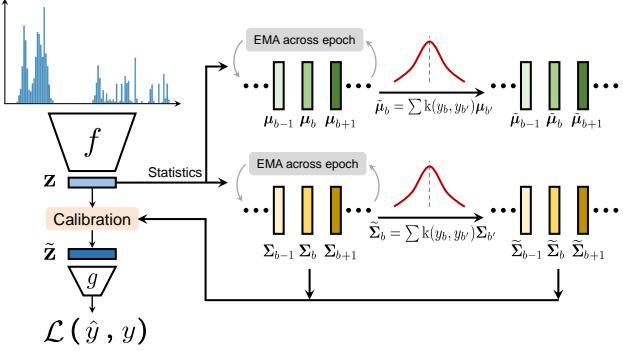


Figure 5. Feature distribution smoothing (FDS) introduces a feature calibration layer that uses kernel smoothing to smooth the distributions of feature mean and covariance over the target space.

calibrated features  $\tilde{z}$  are then passed to the final regression function and used to compute the loss.

We note that FDS can be integrated with any neural network model, as well as any past work on improving label imbalance. In Sec. 4, we integrate FDS with a variety of prior techniques for addressing data imbalance, and demonstrate that it consistently improves performance.

## 4. Benchmarking DIR

**Datasets.** We curate five DIR benchmarks that span computer vision, natural language processing, and healthcare. Fig. 6 shows the label density distribution of these datasets, and their level of imbalance.

- **IMDB-WIKI-DIR (age):** We construct IMDB-WIKI-DIR using the IMDB-WIKI dataset (Rothe et al., 2018), which contains 523.0K face images and the corresponding ages. We filter out unqualified images, and manually construct balanced validation and test set over the supported ages. The length of each bin is 1 year, with a minimum age of 0 and a maximum age of 186. The number of images per bin varies between 1 and 7149, exhibiting significant data imbalance. Overall, the curated dataset has 191.5K images for training, 11.0K images for validation and testing.
- **AgeDB-DIR (age):** AgeDB-DIR is constructed in a similar manner from the AgeDB dataset (Moschoglou et al., 2017). It contains 12.2K images for training, with a minimum age of 0 and a maximum age of 101, and maximum bin density of 353 images and minimum bin density of 1. The validation and test set are balanced with 2.1K images.
- **STS-B-DIR (text similarity score):** We construct STS-B-DIR from the Semantic Textual Similarity Benchmark (Cer et al., 2017; Wang et al., 2018), which is a collection of sentence pairs drawn from news headlines, video and image captions, and natural language inference data. Each pair is annotated by multiple annotators with an averaged

continuous similarity score from 0 to 5. From the original training set of 7.2K pairs, we create a training set with 5.2K pairs, and balanced validation set and test set of 1K pairs each. The length of each bin is 0.1.

- **NYUD2-DIR (depth):** We create NYUD2-DIR based on the NYU Depth Dataset V2 (Nathan Silberman & Fergus, 2012), which provides images and depth maps for different indoor scenes. The depth maps have an upper bound of 10 meters and we set the bin length as 0.1 meter. Following standard practices (Bhat et al., 2020; Hu et al., 2019), we use 50K images for training and 654 images for testing. We randomly select 9357 test pixels for each bin to make the test set balanced.

- **SHHS-DIR (health condition score):** We create SHHS-DIR based on the SHHS dataset (Quan et al., 1997), which contains full-night Polysomnography (PSG) from 2651 subjects. Available PSG signals include Electroencephalography (EEG), Electrocardiography (ECG), and breathing signals (airflow, abdomen, and thorax), which are used as inputs. The dataset also includes the 36-Item Short Form Health Survey (SF-36) (Ware Jr & Sherbourne, 1992) for each subject, where a General Health score is extracted. The score is used as the target value with a minimum score of 0 and maximum of 100.

**Network Architectures.** We employ ResNet-50 (He et al., 2016) as our backbone network for IMDB-WIKI-DIR and AgeDB-DIR. Following (Wang et al., 2018), we adopt the same BiLSTM + GloVe word embeddings baseline for STS-B-DIR. For NYUD2-DIR, we use ResNet-50-based encoder-decoder architecture introduced in (Hu et al., 2019). Finally, for SHHS-DIR, we use the same CNN-RNN architecture with ResNet block for PSG signals as in (Wang et al., 2019).

**Baselines.** Since the literature has only a few proposals for DIR, in addition to past work on imbalanced regression (Branco et al., 2017; Torgo et al., 2013), we adapt a few imbalanced classification methods for regression, and propose a strong set of baselines. Below, we describe the baselines, and how we can combine LDS with each method. For FDS, it can be directly integrated with any baseline as a calibration layer, as described in Sec. 3.2.

- **Vanilla model:** We use term **VANILLA** to denote a model that does not include any technique for dealing with imbalanced data. To combine the vanilla model with LDS, we re-weight the loss function by multiplying it by the inverse of the LDS estimated density for each target bin.
- **Synthetic samples:** We choose existing methods for imbalanced regression, including **SMOTER** (Torgo et al., 2013) and **SMOGN** (Branco et al., 2017). SMOTER first defines frequent and rare regions using the original label density, and creates synthetic samples for pre-defined rare

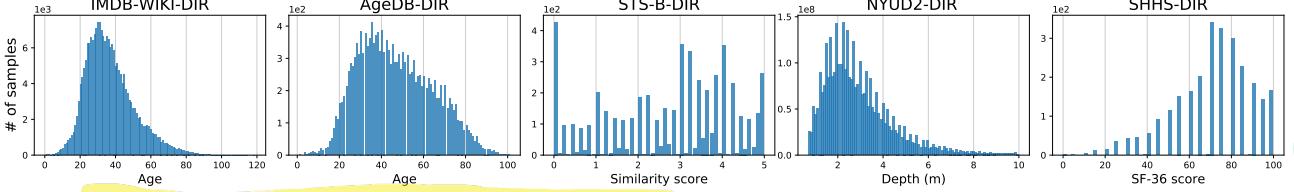


Figure 6. Overview of training set label distribution for five DIR datasets. They range from single-value prediction such as age, textual similarity score, and health condition score, to dense-value prediction such as depth estimation. More details are provided in Appendix B.

regions by linearly interpolating both inputs and targets. SMOGN further adds Gaussian noise to SMOTER. We note that LDS can be directly used for a better estimation of label density when dividing the target space.

- **Error-aware loss:** Inspired by the Focal loss (Lin et al., 2017) for classification, we propose a regression version called **Focal-R**, where the scaling factor is replaced by a continuous function that maps the absolute error into  $[0, 1]$ . Precisely, Focal-R loss based on  $L_1$  distance can be written as  $\frac{1}{n} \sum_{i=1}^n \sigma(|\beta e_i|)^\gamma e_i$ , where  $e_i$  is the  $L_1$  error for  $i$ -th sample,  $\sigma(\cdot)$  is the Sigmoid function, and  $\beta, \gamma$  are hyper-parameters. To combine Focal-R with LDS, we multiply the loss with the inverse frequency of the estimated label density.
- **Two-stage training:** Following (Kang et al., 2020) where feature and classifier are decoupled and trained in two stages, we propose a regression version called regressor re-training (**RRT**), where in the first stage we train the encoder normally, and in the second stage freeze the encoder and re-train the regressor  $g(\cdot)$  with inverse re-weighting. When adding LDS, the re-weighting in the second stage is based on the label density estimated through LDS.
- **Cost-sensitive re-weighting:** Since we divide the target space into finite bins, classic re-weighting methods can be directly plugged in. We adopt two re-weighting schemes based on the label distribution: inverse-frequency weighting (**INV**) and its square-root weighting variant (**SQINV**). When combining with LDS, instead of using the original label density, we use the LDS estimated target density.

**Evaluation Process and Metrics.** Following (Liu et al., 2019), we divide the target space into three disjoint subsets: *many-shot region* (bins with over 100 training samples), *medium-shot region* (bins with 20~100 training samples), and *few-shot region* (bins with under 20 training samples), and report results on these subsets, as well as overall performance. We also refer to regions with no training samples as *zero-shot*, and investigate the ability of our techniques to generalize to zero-shot regions in Sec. 4.2. For metrics, we use common metrics for regression, such as the mean-average-error (MAE), mean-squared-error (MSE), and Pearson correlation. We further propose another metric, called error Geometric Mean (GM), and is defined as  $(\prod_{i=1}^n e_i)^{\frac{1}{n}}$  for better prediction fairness.

Table 1. Benchmarking results on IMDB-WIKI-DIR.

Metrics	MAE ↓				GM ↓				
	Shot	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA		8.06	7.23	15.12	26.33	4.57	4.17	10.59	20.46
SMOTER (Torgo et al., 2013)		8.14	7.42	14.15	25.28	4.64	<b>4.30</b>	9.05	19.46
SMOGN (Branco et al., 2017)		8.03	<b>7.30</b>	14.02	25.93	4.63	<b>4.30</b>	8.74	20.12
SMOGN + LDS		8.02	7.39	13.71	23.22	4.63	4.39	8.71	15.80
SMOGN + FDS		8.03	7.35	14.06	23.44	4.65	4.33	8.87	16.00
SMOGN + LDS + FDS		<b>7.97</b>	7.38	<b>13.22</b>	<b>22.95</b>	<b>4.59</b>	4.39	<b>7.84</b>	<b>14.94</b>
FOCAL-R		7.97	7.12	15.14	26.96	4.49	4.10	10.37	21.20
FOCAL-R + LDS		7.90	<b>7.10</b>	14.72	25.84	<b>4.47</b>	<b>4.09</b>	10.11	19.14
FOCAL-R + FDS		7.96	7.14	14.71	26.06	4.51	4.12	10.16	19.56
FOCAL-R + LDS + FDS		<b>7.88</b>	<b>7.10</b>	<b>14.08</b>	<b>25.75</b>	<b>4.47</b>	4.11	<b>9.32</b>	<b>18.67</b>
RRT		7.81	7.07	14.06	25.13	4.35	4.03	8.91	16.96
RRT + LDS		7.79	7.08	13.76	24.64	4.34	<b>4.02</b>	8.72	16.92
RRT + FDS		<b>7.65</b>	<b>7.02</b>	12.68	23.85	<b>4.31</b>	4.03	7.58	16.28
RRT + LDS + FDS		<b>7.65</b>	7.06	<b>12.41</b>	<b>23.51</b>	<b>4.31</b>	4.07	<b>7.17</b>	<b>15.44</b>
SQINV		7.87	7.24	12.44	22.76	4.47	4.22	7.25	15.10
SQINV + LDS		7.83	7.31	<b>12.43</b>	22.51	4.42	4.19	7.00	13.94
SQINV + FDS		7.83	7.23	12.60	22.37	4.42	4.20	<b>6.93</b>	13.48
SQINV + LDS + FDS		<b>7.78</b>	<b>7.20</b>	12.61	<b>22.19</b>	<b>4.37</b>	<b>4.12</b>	7.39	<b>12.61</b>
OURS (BEST) VS. VANILLA		+0.41	+0.21	+2.71	+4.14	+0.26	+0.15	+3.66	+7.85

#### 4.1. Main Results

We report the main results in this section for all DIR datasets. All training details, hyper-parameter settings, and additional results are provided in Appendix C and D.

**Inferring Age from Images: IMDB-WIKI-DIR & AgeDB-DIR.** We report the performance of different methods in Table 1 and 2, respectively. For each dataset, we group the baselines into four sections to reflect their different strategies. First, as both tables indicate, when applied to modern high-dimensional data like images, SMOTER and SMOGN can actually degrade the performance in comparison to the vanilla model. Moreover, within each group, adding either LDS, FDS, or both leads to performance gains, while LDS + FDS often achieves the best results. Finally, when compared to the vanilla model, using our LDS and FDS maintains or slightly improves the performance overall and on the many-shot regions, while substantially boosting the performance for the medium-shot and few-shot regions.

**Inferring Text Similarity Score: STS-B-DIR.** Table 3 shows the results, where similar observations can be made on STS-B-DIR. Again, both SMOTER and SMOGN perform worse than the vanilla model. In contrast, both LDS and FDS consistently and substantially improve the results for various methods, especially in medium- and few-shot re-

Table 2. Benchmarking results on AgeDB-DIR.

Metrics	MAE ↓				GM ↓			
	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	7.77	6.62	9.55	13.67	5.05	4.23	7.01	10.75
SMOTER (Torgo et al., 2013)	8.16	7.39	8.65	12.28	5.21	4.65	5.69	8.49
SMOGN (Branco et al., 2017)	8.26	7.64	9.01	12.09	5.36	4.90	6.19	8.44
SMOGN + LDS	7.96	7.44	8.64	11.77	5.03	4.68	5.69	7.98
SMOGN + FDS	8.06	7.52	8.75	11.89	5.02	4.66	5.63	8.02
SMOGN + LDS + FDS	<b>7.90</b>	<b>7.32</b>	<b>8.51</b>	<b>11.19</b>	<b>4.98</b>	<b>4.64</b>	<b>5.41</b>	<b>7.35</b>
FOCAL-R	7.64	6.68	9.22	13.00	4.90	4.26	6.39	9.52
FOCAL-R + LDS	7.56	<b>6.67</b>	8.82	12.40	4.82	4.27	5.87	8.83
FOCAL-R + FDS	7.65	6.89	8.70	<b>11.92</b>	4.83	4.32	5.89	<b>8.04</b>
FOCAL-R + LDS + FDS	<b>7.47</b>	6.69	<b>8.30</b>	12.55	<b>4.71</b>	<b>4.25</b>	<b>5.36</b>	8.59
RRT	7.74	6.98	8.79	11.99	5.00	4.50	5.88	8.63
RRT + LDS	7.72	7.00	8.75	11.62	4.98	4.54	5.71	8.27
RRT + FDS	7.70	<b>6.95</b>	8.76	11.86	4.82	<b>4.32</b>	5.83	8.08
RRT + LDS + FDS	<b>7.66</b>	6.99	<b>8.60</b>	<b>11.32</b>	<b>4.80</b>	4.42	<b>5.53</b>	<b>6.99</b>
SQINV	7.81	7.16	8.80	11.20	4.99	4.57	5.73	7.77
SQINV + LDS	7.67	<b>6.98</b>	8.86	10.89	4.85	4.39	5.80	7.45
SQINV + FDS	7.69	7.10	8.86	<b>9.98</b>	4.83	4.41	5.97	<b>6.29</b>
SQINV + LDS + FDS	<b>7.55</b>	7.01	<b>8.24</b>	10.79	<b>4.72</b>	<b>4.36</b>	<b>5.45</b>	6.79
OURS (BEST) vs. VANILLA	+0.30	-0.05	+1.31	+3.69	+0.34	-0.02	+1.65	+4.46

Table 3. Benchmarking results on STS-B-DIR.

Metrics	MSE ↓				Pearson correlation (%) ↑			
	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	0.974	0.851	1.520	0.984	74.2	72.0	62.7	75.2
SMOTER (Torgo et al., 2013)	1.046	0.924	1.542	1.154	72.6	69.3	65.3	70.6
SMOGN (Branco et al., 2017)	0.990	0.896	1.327	1.175	73.2	70.4	65.5	69.2
SMOGN + LDS	0.962	0.880	1.242	1.155	74.0	71.5	65.2	69.8
SMOGN + FDS	0.987	0.945	<b>1.101</b>	1.153	73.0	69.6	<b>68.5</b>	69.9
SMOGN + LDS + FDS	<b>0.950</b>	<b>0.851</b>	1.327	<b>1.095</b>	<b>74.6</b>	<b>72.1</b>	65.9	<b>71.7</b>
FOCAL-R	0.951	0.843	1.425	0.957	74.6	72.3	61.8	76.4
FOCAL-R + LDS	0.930	<b>0.807</b>	1.449	0.993	<b>75.7</b>	<b>73.9</b>	62.4	75.4
FOCAL-R + FDS	<b>0.920</b>	0.855	<b>1.169</b>	1.008	75.1	72.6	<b>66.4</b>	74.7
FOCAL-R + LDS + FDS	0.940	0.849	1.358	<b>0.916</b>	74.9	72.2	66.3	<b>77.3</b>
RRT	0.964	0.842	1.503	0.978	74.5	72.4	62.3	75.4
RRT + LDS	0.916	0.817	1.344	0.945	75.7	73.5	64.1	76.6
RRT + FDS	0.929	0.857	<b>1.209</b>	1.025	74.9	72.1	<b>67.2</b>	74.0
RRT + LDS + FDS	<b>0.903</b>	<b>0.806</b>	1.323	<b>0.936</b>	<b>76.0</b>	<b>73.8</b>	65.2	<b>76.7</b>
INV	1.005	0.894	1.482	1.046	72.8	70.3	62.5	73.2
INV + LDS	0.914	0.819	1.319	0.955	75.6	73.4	63.8	76.2
INV + FDS	0.927	0.851	<b>1.225</b>	1.012	75.0	72.4	<b>66.6</b>	74.2
INV + LDS + FDS	<b>0.907</b>	<b>0.802</b>	1.363	<b>0.942</b>	<b>76.0</b>	<b>74.0</b>	65.2	<b>76.6</b>
OURS (BEST) vs. VANILLA	+0.071	+0.049	+0.419	+0.068	+1.8	+2.0	+5.8	+2.1

gions. The advantage is even more profound under *Pearson correlation*, which is commonly used for this NLP task.

**Inferring Depth: NYUD2-DIR.** For NYUD2-DIR, which is a dense regression task, we verify from Table 4 that adding LDS and FDS significantly improves the results. We note that the vanilla model can inevitably overfit to the many-shot regions during training. FDS and LDS help alleviate this effect, and generalize better to all regions, with minor degradation in the many-shot region but significant boosts for other regions.

**Inferring Health Score: SHHS-DIR.** Table 5 reports the results on SHHS-DIR. Since SMOTER and SMOGN are not directly applicable to this medical data, we skip them for

Table 4. Benchmarking results on NYUD2-DIR.

Metrics	RMSE ↓				$\delta_1 \uparrow$			
	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	1.477	0.591	0.952	2.123	0.677	0.777	0.693	0.570
VANILLA + LDS	1.387	0.671	0.913	1.954	0.672	0.701	0.706	0.630
VANILLA + FDS	1.442	<b>0.615</b>	0.940	2.059	0.681	<b>0.760</b>	0.695	0.596
VANILLA + LDS + FDS	<b>1.338</b>	0.670	<b>0.851</b>	<b>1.880</b>	<b>0.705</b>	0.730	<b>0.764</b>	<b>0.655</b>
OURS (BEST) vs. VANILLA	+0.139	-0.024	+0.101	+0.243	+0.028	-0.017	+0.071	+0.085

Table 5. Benchmarking results on SHHS-DIR.

Metrics	MAE ↓				GM ↓			
	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	15.36	12.47	13.98	16.94	10.63	8.04	9.59	12.20
FOCAL-R	14.67	11.70	13.69	17.06	9.98	7.93	8.85	11.95
FOCAL-R + LDS	14.49	12.01	12.43	16.57	9.98	7.89	8.59	11.40
FOCAL-R + FDS	14.18	<b>11.06</b>	13.58	15.99	9.45	<b>6.95</b>	8.81	11.13
FOCAL-R + LDS + FDS	<b>14.02</b>	11.08	<b>12.24</b>	<b>15.49</b>	<b>9.32</b>	7.18	<b>8.10</b>	<b>10.39</b>
RRT	14.78	12.43	14.01	16.48	10.12	8.05	9.71	11.96
RRT + LDS	14.56	12.08	13.44	16.45	9.89	7.85	9.18	11.82
RRT + FDS	14.36	11.97	13.33	16.08	9.74	7.54	9.20	11.31
RRT + LDS + FDS	<b>14.33</b>	<b>11.96</b>	<b>12.47</b>	<b>15.92</b>	<b>9.63</b>	<b>7.35</b>	<b>8.74</b>	<b>11.17</b>
INV	14.39	11.84	13.12	16.02	9.34	7.73	8.49	11.20
INV + LDS	14.14	11.66	12.77	16.05	9.26	7.64	8.18	11.32
INV + FDS	13.91	<b>11.12</b>	12.29	15.53	8.94	<b>6.91</b>	7.79	10.65
INV + LDS + FDS	<b>13.76</b>	<b>11.12</b>	<b>12.18</b>	<b>15.07</b>	<b>8.70</b>	6.94	<b>7.60</b>	<b>10.18</b>
OURS (BEST) vs. VANILLA	+1.60	+1.41	+1.80	+1.87	+1.93	+1.13	+1.99	+2.02

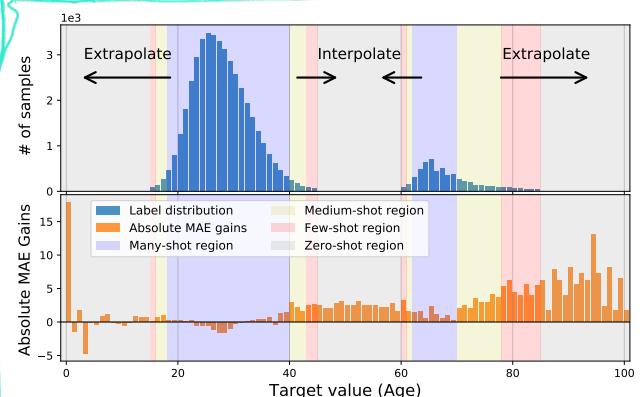
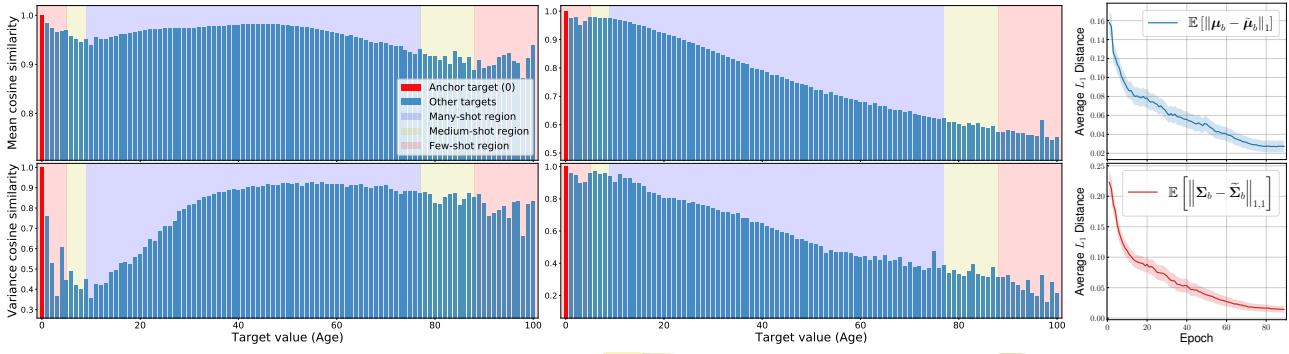


Figure 7. The absolute MAE gains of LDS + FDS over the vanilla model, on a curated subset of IMDB-WIKI-DIR with certain target values having no training data. We establish notable performance gains w.r.t. all regions, especially for extrapolation & interpolation.

## 4.2. Further Analysis

**Extrapolation & Interpolation.** In real-world DIR tasks, certain target values can have no data at all (e.g., see SHHS-DIR and STS-B-DIR in Fig. 6). This motivates the need for target extrapolation and interpolation. We curate a subset from the training set of IMDB-WIKI-DIR, which has no



(a) Feature statistics similarity for age 0, without FDS (b) Feature statistics similarity for age 0, with FDS (c) Statistics change

Figure 8. Analysis on how FDS works. (a) & (b) Feature statistics similarity for anchor age 0, using model trained without and with FDS. (c)  $L_1$  distance between the running statistics  $\{\mu_b, \Sigma_b\}$  and the smoothed statistics  $\{\tilde{\mu}_b, \tilde{\Sigma}_b\}$  during training.

**Table 6.** Interpolation & extrapolation results on the curated subset of IMDB-WIKI-DIR. Using LDS and FDS, the generalization results on zero-shot regions can be consistently improved.

Metrics	MAE $\downarrow$				GM $\downarrow$			
	All	w/ data	Interp.	Extrap.	All	w/ data	Interp.	Extrap.
VANILLA	11.72	9.32	16.13	18.19	7.44	5.33	14.41	16.74
VANILLA + LDS	10.54	8.31	14.14	17.38	6.50	4.67	12.13	15.36
VANILLA + FDS	11.40	8.97	15.83	18.01	7.18	5.12	14.02	16.48
VANILLA + LDS + FDS	<b>10.27</b>	<b>8.11</b>	<b>13.71</b>	<b>17.02</b>	<b>6.33</b>	<b>4.55</b>	<b>11.71</b>	<b>15.13</b>
<b>OURS (BEST) VS. VANILLA</b>	<b>+1.45</b>	<b>+1.21</b>	<b>+2.42</b>	<b>+1.17</b>	<b>+1.11</b>	<b>+0.78</b>	<b>+2.70</b>	<b>+1.61</b>

training data in certain regions (Fig. 7), but evaluate on the original testset for zero-shot generalization analysis.

As Table 6 shows, compared to the vanilla model, LDS and FDS can both improve the results not only on regions that have data, but also achieve larger gains on those without data. Specifically, substantial improvements are established for both target interpolation and extrapolation, where interpolation enjoys larger boosts.

We further visualize the absolute MAE gains of our method over vanilla model in Fig. 7. Our method provides a comprehensive treatment to the many, medium, few, as well as zero-shot regions, achieving remarkable performance gains.

**Understanding FDS.** We investigate how FDS influences the feature statistics. In Fig. 8(a) and 8(b) we plot the similarity of the feature statistics for anchor age 0, using model trained without and with FDS. As the figure indicates, since age 0 lies in the few-shot region, the feature statistics can have a large bias, i.e., age 0 shares large similarity with region  $40 \sim 80$  as in Fig. 8(a). In contrast, when FDS is added, the statistics are better calibrated, resulting in a high similarity only in its neighborhood, and a gradually decreasing similarity score as target value becomes larger. We further visualize the  $L_1$  distance between the running statistics  $\{\mu_b, \Sigma_b\}$  and the smoothed statistics  $\{\tilde{\mu}_b, \tilde{\Sigma}_b\}$  during training in Fig. 8(c). Interestingly, the average  $L_1$  distance becomes smaller and gradually diminishes as the

training evolves, indicating that the model learns to generate features that are more accurate even without smoothing, and finally the smoothing module can be removed during inference. We provide more results for different anchor ages in Appendix E.7, where similar effects can be observed.

#### Ablation: Kernel type for LDS & FDS (Appendix E.1).

We study the effects of different kernel types for LDS and FDS when applying distribution smoothing. We select three different kernel types, i.e., Gaussian, Laplacian, and Triangular kernel, and evaluate their influences on both LDS and FDS. In general, all kernel types lead to notable gains (e.g., 3.7%  $\sim$  6.2% relative MSE gains on STS-B-DIR), with the Gaussian kernel often delivering the best results.

**Ablation: Different regression loss functions (Appendix E.2).** We investigate the influence of different training loss functions on LDS and FDS. We select three common losses used for regression tasks, i.e.,  $L_1$  loss, MSE loss, and the Huber loss (also referred to as smoothed  $L_1$  loss). We find that similar results are obtained for all losses, indicating that both LDS and FDS are robust to different loss functions.

**Ablation: Hyper-parameter for LDS & FDS (Appendix E.3).** We investigate the effects of hyper-parameters on both LDS and FDS. As we mainly employ the Gaussian kernel for distribution smoothing, we extensively study different choices of the kernel size  $l$  and standard deviation  $\sigma$ . Interestingly, we find LDS and FDS are surprisingly robust to different hyper-parameters in a given range, and obtain similar gains. For example, on STS-B-DIR with  $l \in \{5, 9, 15\}$  and  $\sigma \in \{1, 2, 3\}$ , overall MSE gains range from 3.3% to 6.2%, with  $l = 5$  and  $\sigma = 2$  exhibiting the best results.

**Ablation: Robustness to diverse skewed label densities (Appendix E.4).** We curate different imbalanced distributions for IMDB-WIKI-DIR by combining different number of disjoint skewed Gaussian distributions over the target space, with potential missing data in certain target regions, and evaluate the robustness of FDS and LDS to the distribution change. We verify that even under different imbalanced

label distributions, LDS and FDS consistently boost the performance across all regions compared to the vanilla model, with relative MAE gains ranging from 8.8% to 12.4%.

**Comparisons to imbalanced classification methods (Appendix E.6).** Finally, to gain more insights on the intrinsic difference between imbalanced classification & imbalanced regression problems, we directly apply existing imbalanced classification schemes on several appropriate DIR datasets, and show empirical comparisons with imbalanced regression approaches. We demonstrate in Appendix E.6 that LDS and FDS outperform imbalanced classification schemes by a large margin, where the errors for few-shot regions can be reduced by up to 50% to 60%. Interestingly, the results also show that imbalanced classification schemes often perform worse than even the vanilla regression model, which confirms that regression requires different approaches for data imbalance than simply applying classification methods. We note that imbalanced classification methods could fail on regression problems for several reasons. First, they ignore the similarity between data samples that are close w.r.t. the continuous target. Moreover, classification cannot extrapolate or interpolate in the continuous label space, therefore unable to deal with missing data in certain target regions.

## 5. Conclusion

We introduce the DIR task that learns from natural imbalanced data with continuous targets, and generalizes to the entire target range. We propose two simple and effective algorithms for DIR that exploit the similarity between nearby targets in both label and feature spaces. Extensive results on five curated large-scale real-world DIR benchmarks confirm the superior performance of our methods. Our work fills the gap in benchmarks and techniques for practical DIR tasks.

## References

- Bhat, S. F., Alhashim, I., and Wonka, P. Adabins: Depth estimation using adaptive bins. *arXiv preprint arXiv:2011.14141*, 2020.
- Branco, P., Torgo, L., and Ribeiro, R. P. Smogn: a pre-processing approach for imbalanced regression. In *First international workshop on learning with imbalanced domains: Theory and applications*, pp. 36–50. PMLR, 2017.
- Branco, P., Torgo, L., and Ribeiro, R. P. Rebagg: Resampled bagging for imbalanced regression. In *Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*, pp. 67–81. PMLR, 2018.
- Buda, M., Maki, A., and Mazurowski, M. A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.
- Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019.
- Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, pp. 1–14, 2017.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. Class-balanced loss based on effective number of samples. In *CVPR*, 2019.
- Dong, Q., Gong, S., and Zhu, X. Imbalanced deep learning by minority class incremental rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(6):1367–1381, Jun 2019.
- Eigen, D., Puhrsch, C., and Fergus, R. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, 2014.
- García, S. and Herrera, F. Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evolutionary computation*, 17(3):275–306, 2009.
- Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N. F., Peters, M., Schmitz, M., and Zettlemoyer, L. S. AllenNLP: A deep semantic natural language processing platform. 2017.
- He, H., Bai, Y., Garcia, E. A., and Li, S. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE international joint conference on neural networks*, pp. 1322–1328, 2008.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.
- Hu, J., Ozay, M., Zhang, Y., and Okatani, T. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *WACV*, 2019.
- Huang, C., Li, Y., Change Loy, C., and Tang, X. Learning deep representation for imbalanced classification. In *CVPR*, 2016.
- Huang, C., Li, Y., Chen, C. L., and Tang, X. Deep imbalanced learning for face recognition and attribute prediction. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

- Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., and Kalantidis, Y. Decoupling representation and classifier for long-tailed recognition. *ICLR*, 2020.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.
- Lei, T., Zhang, Y., Wang, S. I., Dai, H., and Artzi, Y. Simple recurrent units for highly parallelizable recurrence. In *EMNLP*, pp. 4470–4481, 2018.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *ICCV*, pp. 2980–2988, 2017.
- Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., and Yu, S. X. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019.
- Loper, E. and Bird, S. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.
- Moschoglou, S., Papaioannou, A., Sagonas, C., Deng, J., Kotsia, I., and Zafeiriou, S. Agedb: The first manually collected, in-the-wild age database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, volume 2, pp. 5, 2017.
- Nathan Silberman, Derek Hoiem, P. K. and Fergus, R. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.
- Parzen, E. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3): 1065–1076, 1962.
- Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.
- Quan, S. F., Howard, B. V., Iber, C., Kiley, J. P., Nieto, F. J., O’Connor, G. T., Rapoport, D. M., Redline, S., Robbins, J., Samet, J. M., et al. The sleep heart health study: design, rationale, and methods. *Sleep*, 20(12):1077–1085, 1997.
- Rothe, R., Timofte, R., and Gool, L. V. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, 126(2-4):144–157, 2018.
- Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. Meta-weight-net: Learning an explicit mapping for sample weighting. *arXiv preprint arXiv:1902.07379*, 2019.
- Sun, B., Feng, J., and Saenko, K. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.
- Tang, K., Huang, J., and Zhang, H. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020.
- Torgo, L., Ribeiro, R. P., Pfahringer, B., and Branco, P. Smote for regression. In *Portuguese conference on artificial intelligence*, pp. 378–389. Springer, 2013.
- Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., and Bengio, Y. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, 2019.
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. *EMNLP 2018*, pp. 353, 2018.
- Wang, H., Mao, C., He, H., Zhao, M., Jaakkola, T. S., and Katabi, D. Bidirectional inference networks: A class of deep bayesian networks for health profiling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 766–773, 2019.
- Ware Jr, J. E. and Sherbourne, C. D. The mos 36-item short-form health survey (sf-36): I. conceptual framework and item selection. *Medical care*, pp. 473–483, 1992.
- Yang, Y. and Xu, Z. Rethinking the value of labels for improving class-imbalanced learning. In *NeurIPS*, 2020.
- Yin, X., Yu, X., Sohn, K., Liu, X., and Chandraker, M. Feature transfer learning for face recognition with under-represented data. In *In Proceeding of IEEE Computer Vision and Pattern Recognition*, Long Beach, CA, June 2019.
- Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.
- Zhang, X., Fang, Z., Wen, Y., Li, Z., and Qiao, Y. Range loss for deep face recognition with long-tailed training data. In *ICCV*, 2017.

# Supplementary Material

## A. Pseudo Code for LDS & FDS

We provide the pseudo code of the proposed LDS and FDS algorithms in Algorithm 1 and 2, respectively. For LDS, we provide an illustrative example which combines LDS with the loss inverse re-weighting scheme.

---

**Algorithm 1** Label Distribution Smoothing (LDS)

---

**Input:** Training set  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ , bin size  $\Delta b$ , symmetric kernel distribution  $k(y, y')$   
 Calculate the empirical label density distribution  $p(y)$  based on  $\Delta b$  and  $\mathcal{D}$   
 Calculate the effective label density distribution  $\tilde{p}(y') \triangleq \int_{\mathcal{Y}} k(y, y') p(y) dy$

```
/* Example: Combine LDS with loss inverse re-weighting */
for all  $(\mathbf{x}_i, y_i) \in \mathcal{D}$  do
    Compute weight for each sample as  $w_i = \frac{c}{\tilde{p}(y_i)} \propto \frac{1}{\tilde{p}(y_i)}$  (constant  $c$  as scaling factor)
end for
for number of training iterations do
    Sample a mini-batch  $\{(\mathbf{x}_i, y_i, w_i)\}_{i=1}^m$  from  $\mathcal{D}$ 
    Forward  $\{\mathbf{x}_i\}_{i=1}^m$  and get corresponding predictions  $\{\hat{y}_i\}_{i=1}^m$ 
    Do one training step using the weighted loss  $\frac{1}{m} \sum_{i=1}^m w_i \mathcal{L}(\hat{y}_i, y_i)$ 
end for
```

---

**Algorithm 2** Feature Distribution Smoothing (FDS)

---

**Input:** Training set  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ , bin index space  $\mathcal{B}$ , symmetric kernel distribution  $k(y, y')$ , encoder  $f$ , regression function  $g$ , total training epochs  $E$ , FDS momentum  $\alpha$   
**for all**  $b \in \mathcal{B}$  **do**
 Initialize the running statistics  $\{\boldsymbol{\mu}_b^{(0)}, \boldsymbol{\Sigma}_b^{(0)}\}$  and the smoothed statistics  $\{\tilde{\boldsymbol{\mu}}_b^{(0)}, \tilde{\boldsymbol{\Sigma}}_b^{(0)}\}$ 
**end for**
**for**  $e = 0$  **to**  $E$  **do**
**repeat**
 Sample a mini-batch  $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$  from  $\mathcal{D}$ 
**for**  $i = 1$  **to**  $m$  (in parallel) **do**
 $\mathbf{z}_i = f(\mathbf{x}_i)$ 
 $\tilde{\mathbf{z}}_i = \left(\tilde{\boldsymbol{\Sigma}}_b^{(e)}\right)^{\frac{1}{2}} \left(\boldsymbol{\Sigma}_b^{(e)}\right)^{-\frac{1}{2}} (\mathbf{z}_i - \boldsymbol{\mu}_b^{(e)}) + \tilde{\boldsymbol{\mu}}_b^{(e)} \quad /*$  Feature statistics calibration \*/
  $\hat{y}_i = g(\tilde{\mathbf{z}}_i)$ 
**end for**
 Do one training step with loss  $\frac{1}{m} \sum_{i=1}^m \mathcal{L}(\hat{y}_i, y_i)$ 
**until** iterate over all training samples at current epoch  $e$ 
**/\* Update running statistics & smoothed statistics \*/**
**for all**  $b \in \mathcal{B}$  **do**
 Estimate current running statistics of  $b$ -th bin  $\{\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\}$  using Eqn. (2) and (3)
  $\boldsymbol{\mu}_b^{(e+1)} \leftarrow \alpha \times \boldsymbol{\mu}_b^{(e)} + (1 - \alpha) \times \boldsymbol{\mu}_b$ 
 $\boldsymbol{\Sigma}_b^{(e+1)} \leftarrow \alpha \times \boldsymbol{\Sigma}_b^{(e)} + (1 - \alpha) \times \boldsymbol{\Sigma}_b$ 
**end for**
 Update smoothed statistics  $\{\tilde{\boldsymbol{\mu}}_b^{(e+1)}, \tilde{\boldsymbol{\Sigma}}_b^{(e+1)}\}_{b \in \mathcal{B}}$  based on  $\{\boldsymbol{\mu}_b^{(e+1)}, \boldsymbol{\Sigma}_b^{(e+1)}\}_{b \in \mathcal{B}}$  using Eqn. (4) and (5)
**end for**


---

## B. Details of DIR Datasets

In this section, we provide the detailed information of the five curated DIR datasets we used in our experiments. Table 7 provides an overview of the five datasets.

Table 7. Overview of the five curated DIR datasets used in our experiments.

Dataset	Target type	Target range	Bin size	Max bin density	Min bin density	# Training set	# Val. set	# Test set
IMDB-WIKI-DIR	Age	0 ~ 186	1	7,149	1	191,509	11,022	11,022
AgeDB-DIR	Age	0 ~ 101	1	353	1	12,208	2,140	2,140
STS-B-DIR	Text similarity score	0 ~ 5	0.1	428	1	5,249	1,000	1,000
NYUD2-DIR	Depth	0.7 ~ 10	0.1	$1.46 \times 10^8$	$1.13 \times 10^6$	50,688 ( $3.51 \times 10^9$ )	—	$654 (8.70 \times 10^5)$
SHHS-DIR	Health condition score	0 ~ 100	1	275	0	1,892	369	369

### B.1. IMDB-WIKI-DIR

The original IMDB-WIKI dataset (Rothe et al., 2018) is a large-scale face image dataset for age estimation from single input image. The original version contains 523.0K face images and the corresponding ages, where 460.7K face images are collected from the IMDB website and 62.3K images from the Wikipedia website. We construct IMDB-WIKI-DIR by first filtering out unqualified images with low face scores (Rothe et al., 2018), and then manually creating balanced validation and test set over the supported ages. Overall, the curated dataset has 191.5K images for training, and 11.0K images for validation and testing, respectively. We make the length of each bin to be 1 year, with a minimum age of 0 and a maximum age of 186. The number of images per bin varies between 1 and 7,149, exhibiting significant data imbalance.

As for the data pre-processing, the images are first resized to  $224 \times 224$ . During training, we follow the standard data augmentation scheme (He et al., 2016) to do zero-padding with 16 pixels on each side, and then random crop back to the original image size. We then randomly flip the images horizontally and normalize them into  $[0, 1]$ .

### B.2. AgeDB-DIR

The original AgeDB dataset (Moschoglou et al., 2017) is a manually collected in-the-wild age database with accurate and noise-free labels. Similar to IMDB-WIKI, the task is also to estimate age from visual appearance. The original dataset contains 16,488 images in total. We construct AgeDB-DIR in a similar manner as IMDB-WIKI-DIR, where the training set contains 12,208 images, with a minimum age of 0 and a maximum age of 101, and maximum bin density of 353 images and minimum bin density of 1. The validation set and test set are made balanced with 2,140 images. Similarly, the images in AgeDB are resized to  $224 \times 224$ , and go through the same data pre-processing schedule as in the IMDB-WIKI-DIR dataset.

### B.3. STS-B-DIR

The original Semantic Textual Similarity Benchmark (STS-B) (Cer et al., 2017), also included in the GLUE benchmark (Wang et al., 2018), is a collection of sentence pairs drawn from news headlines, video and image captions, and natural language inference data. Each pair is human-annotated by multiple annotators with an averaged continuous similarity score from 0 to 5. The task is to predict these scores from the sentence pairs. From the original training set of 7.2K pairs, we create a training set with 5.2K pairs, and balanced validation set and test set of 1K pairs each for STS-B-DIR. We make the length of each bin to be 0.1, and the number of training pairs per bin varies between 1 and 428.

As for the data pre-processing, the sentences are first tokenized using NLTK toolkit (Loper & Bird, 2002) with a maximum length of 40. We then count the frequencies of all words (tokens) of all splits, build the word vocabulary based on the word frequency, and finally use the 300D GloVe word embeddings (840B Common Crawl version) (Pennington et al., 2014) to embed words in the vocabulary into 300-dimensional vectors. Following (Wang et al., 2018), we use AllenNLP (Gardner et al., 2017) open source library to facilitate the data processing, as well as model training and evaluation.

### B.4. NYUD2-DIR

We create NYUD2-DIR based on the NYU Depth Dataset V2 (Nathan Silberman & Fergus, 2012), which provides images and depth maps for different indoor scenes. Our task is to predict the depth maps from the RGB scene images. The depth maps have an upper bound of 10 meters and a lower bound of 0.7 meters. Following standard practices (Bhat et al., 2020; Hu et al., 2019), we use 50K images for training and 654 images for testing. We set the bin length to 0.1 meter and the number of pixels per bin varies between  $1.13 \times 10^6$  and  $1.46 \times 10^8$ . Besides, we randomly select 9,357 test pixels (the minimum number of bin pixels in the test set) for each bin from 654 test images to make the test set balanced, with a total of  $8.70 \times 10^5$  test pixels in the NYUD2-DIR test set, as indicated in Table 7.

Following (Hu et al., 2019), for both training and evaluation phases, we first downsample images (both RGB and depth) from original size  $640 \times 480$  to  $320 \times 240$  using bilinear interpolation, then conduct center crop to obtain images of size  $304 \times 228$ , and finally normalize them into  $[0, 1]$ . Note that our pixel statistics are calculated and selected based on this resolution. For training, we further downsample the depth maps to  $114 \times 152$  to fit the size of outputs. Additionally, we also employ the following data argumentation methods during training: (1) Flip: randomly flip both RGB and depth images horizontally with probability of 0.5; (2) Rotation: rotate both RGB and depth images by a random degree from -5 to 5; (3) Color Jitter: randomly scale the brightness, contrast, and saturation values of the RGB images by  $c \in [0.6, 1.4]$ .

## B.5. SHHS-DIR

We create SHHS-DIR based on the SHHS dataset (Quan et al., 1997), which contains full-night Polysomnography (PSG) signals from 2,651 subjects. The signal length for each subject varies from 7,278 seconds to 45,448 seconds. Available PSG signals include Electroencephalography (EEG), Electrocardiography (ECG), and breathing signals (airflow, abdomen, and thorax). In the experiments, we consider all of these PSG signals as high-dimensional information, and use them as inputs. Specifically, we first preprocess both EEG and ECG signals to transform them from time domain to the frequency domain using the short-time Fourier transform (STFT), and get the dense EEG spectrograms  $\mathbf{x}_e \in \mathbb{R}^{64 \times l_i}$  and ECG spectrograms  $\mathbf{x}_c \in \mathbb{R}^{22 \times l_i}$ , where  $l_i \in [7278, 45448]$  is the signal length for the  $i$ -th subject. For the breathing signals, we use the original time series with a sampling rate of 10Hz, resulting in the high-dimensional input as  $\mathbf{x}_b \in \mathbb{R}^{3 \times 10l_i}$ , where the three different breathing sources are concatenated as different channels.

The dataset also includes the 36-Item Short Form Health Survey (SF-36) (Ware Jr & Sherbourne, 1992) for each subject, where a General Health score is extracted. We use the score as the target value, and formulate the task as predicting the General Health score for different subjects from their PSG signals (i.e.,  $\mathbf{x}_e, \mathbf{x}_c, \mathbf{x}_b$ ). The training set of SHHS-DIR contains 1,892 samples (subjects), and the validation set and test set are made balanced over the health score with 369 samples each. We set the length of each bin to be 1, with a minimum score of 0 and a maximum score of 100. The number of samples per bin varies between 0 and 275, indicating the missing data issue in certain target bins.

## C. Experimental Settings

### C.1. Implementation Details

**IMDB-WIKI-DIR & AgeDB-DIR.** We use ResNet-50 model (He et al., 2016) for all IMDB-WIKI-DIR and AgeDB-DIR experiments. We train all models for 90 epochs using the Adam optimizer (Kingma & Ba, 2014), with an initial learning rate of  $10^{-3}$  and then decayed by 0.1 at the 60-th and 80-th epoch, respectively. We mainly employ the  $L_1$  loss throughout the experiments, and fix the batch size as 256.

For both LDS and FDS, we use the Gaussian kernel for distribution smoothing, with the kernel size  $l = 5$  and the standard deviation  $\sigma = 2$ . We study different choices of kernel types, training losses, and hyper-parameter values in Sec. E.1, E.2, and E.3. For the implementation of FDS, we simply use the feature variance instead of covariance for better computational efficiency. The momentum of FDS is fixed as 0.9. As for the baseline methods, we set  $\beta = 0.2$  and  $\gamma = 1$  for FOCAL-R. For RRT, in the second training stage, we employ an initial learning rate of  $10^{-4}$  with total training epochs of 30. For SMOTER and SMOGN, we divide the target range based on a manually defined relevance method, under-sample majority regions, and over-sample minority regions by either interpolating with selected nearest neighbors (Torgo et al., 2013) or also adding Gaussian noise perturbation (Branco et al., 2017). We use pixel-wise Euclidean distance to define the image distance, which is further used to determine nearest neighbors, and set Gaussian perturbation ratio as 0.1 for SMOGN.

**STS-B-DIR.** Following (Wang et al., 2018), we use 300D GloVe word embeddings (840B Common Crawl version) (Pennington et al., 2014) and a two-layer, 1500D (per direction) BiLSTM with max pooling to encode the paired sentences into independent vectors  $u$  and  $v$ , and then pass  $[u; v; |u - v|; uv]$  to a regressor. We train all models using the Adam optimizer with a fixed learning rate  $10^{-4}$ . We validate the model every 10 epochs, use MSE as the validation metric, and stop training when performance does not improve, i.e., validation error does not decrease, after 10 validation checks. We employ the MSE loss throughout the experiments and fix the batch size as 128.

We use the same hyper-parameter settings for both LDS and FDS as in the IMDB-WIKI-DIR experiments. For the baselines, we employ MSE-based FOCAL-R and set  $\beta = 20$  and  $\gamma = 1$ . For RRT, the hyper-parameter settings remain the same between the first and the second training stage. For SMOTER and SMOGN, we use the Euclidean distance between the word embeddings to measure the sentence distance and do interpolation or Gaussian noise argumentation based on the word

embeddings. We set Gaussian perturbation ratio as 0.1 and the number of neighbors  $k = 7$ . For STS-B-DIR, we define *many-shot region* as bins with over 100 training samples, *medium-shot region* with 30~100 training samples, and *few-shot region* with under 30 training samples.

**NYUD2-DIR.** We use ResNet-50-based encoder-decoder architecture proposed by (Hu et al., 2019) for all NYUD2-DIR experiments, which consists of an encoder, a decoder, a multi-scale feature fusion module, and a refinement module. We train all models for 20 epochs using Adam optimizer with an initial learning rate of  $10^{-4}$  and then decayed by 0.1 every 5 epochs. To better evaluate the performance of our methods, we simply use the MSE loss as the depth loss without adding the gradient and surface normal losses as in (Hu et al., 2019). We fix the batch size as 32 for all experiments. We use the same hyper-parameter settings for both LDS and FDS as in the IMDB-WIKI-DIR experiments. For NYUD2-DIR, *many-shot region* is defined as bins with over  $2.6 \times 10^7$  training pixels, *medium-shot region* as bins with  $1.0 \times 10^7 \sim 2.6 \times 10^7$  training pixels, and *few-shot region* as bins with under  $1.0 \times 10^7$  training pixels.

**SHHS-DIR.** Following (Wang et al., 2019), we use a CNN-RNN network architecture for SHHS-DIR experiments. The network first employs three encoders with the same architecture to encode the high-dimensional EEG  $\mathbf{x}_e$ , ECG  $\mathbf{x}_c$ , and breathing signals  $\mathbf{x}_b$  into fixed-length vectors (each with 256 dimensions). The encodings are then concatenated and sent to a 3-layer MLP regression network to produce the output value. Each of the encoder uses the ResNet block (He et al., 2016) with 1D convolution as the CNN components, and employs the simple recurrent units (SRU) (Lei et al., 2018) as the RNN components. We train all models for 80 epochs using the Adam optimizer with a learning rate of  $10^{-3}$ , and remain all other hyper-parameters the same as (Wang et al., 2019). We use the same hyper-parameter settings for both LDS and FDS, as well as other baseline methods as in the IMDB-WIKI-DIR experiments.

## C.2. Evaluation Metrics

We describe in detail all the evaluation metrics we used in our experiments.

**MAE.** The mean absolute error (MAE) is defined as  $\frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$ , which represents the averaged absolute difference between the ground truth and predicted values over all samples.

**MSE & RMSE.** The mean squared error (MSE) is defined as  $\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$ , which represents the averaged squared difference between the ground truth and predicted values over all samples. The root mean squared error (RMSE) is computed by simply taking the square root of MSE.

**GM.** We propose another evaluation metric for regression, called error Geometric Mean (GM), and is defined as  $(\prod_{i=1}^N e_i)^{\frac{1}{N}}$ , where  $e_i \triangleq |y_i - \hat{y}_i|$  represents the  $L_1$  error of each sample. GM aims to characterize the fairness (uniformity) of model predictions using the geometric mean instead of the arithmetic mean over the prediction errors.

**Pearson correlation & Spearman correlation.** Following the common evaluation practice as in the STS-B (Cer et al., 2017) and the GLUE benchmark (Wang et al., 2018), we employ Pearson correlation as well as Spearman correlation for performance evaluation on STS-B-DIR, where Pearson correlation evaluates the linear relationship between predictions and corresponding ground truth values, and Spearman correlation evaluates the monotonic rank-order relationship.

**Mean  $\log_{10}$  error & Threshold accuracy.** For NYUD2-DIR, we further use several standard depth estimation evaluation metrics proposed by (Eigen et al., 2014): Mean  $\log_{10}$  error ( $\log_{10}$ ), which is expressed as  $\frac{1}{N} \sum_{i=1}^N |\log_{10} d_i - \log_{10} g_i|$ ; Threshold accuracy ( $\delta_i$ ), which is defined as the percentage of  $d_i$  such that  $\max\left(\frac{d_i}{g_i}, \frac{g_i}{d_i}\right) = \delta_i < 1.25^i$  ( $i = 1, 2, 3$ ). Here,  $g_i$  denotes the value of a pixel in the ground truth depth image,  $d_i$  represents the value of its corresponding pixel in the predicted depth image, and  $N$  is the total number of evaluation pixels.

## D. Additional Results

We provide complete evaluation results on the five DIR datasets, where more baselines and evaluation metrics are included in addition to the reported results in the main paper.

### D.1. Complete Results on IMDB-WIKI-DIR

We include more baseline methods for comparison on IMDB-WIKI-DIR. Specifically, the following two baselines are added for comparison in the group of *Synthetic samples* strategies:

Table 8. Complete evaluation results on IMDB-WIKI-DIR.

Metrics	MSE ↓				MAE ↓				GM ↓			
	Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.
VANILLA	138.06	108.70	366.09	964.92	8.06	7.23	15.12	26.33	4.57	4.17	10.59	20.46
VANILLA + LDS	131.65	109.04	<b>298.98</b>	829.35	7.83	7.31	<b>12.43</b>	22.51	4.42	4.19	<b>7.00</b>	13.94
VANILLA + FDS	133.81	107.51	332.90	916.18	7.85	<b>7.18</b>	13.35	24.12	4.47	4.18	8.18	15.18
VANILLA + LDS + FDS	<b>129.35</b>	<b>106.52</b>	311.49	<b>811.82</b>	<b>7.78</b>	7.20	12.61	<b>22.19</b>	<b>4.37</b>	<b>4.12</b>	7.39	<b>12.61</b>
MIXUP (Zhang et al., 2018)	141.11	109.13	389.95	1037.98	8.22	<b>7.29</b>	16.23	28.11	4.68	<b>4.22</b>	12.28	23.55
M-MIXUP (Verma et al., 2019)	137.45	<b>108.33</b>	363.72	957.53	8.22	7.39	15.24	26.70	4.80	4.39	10.85	21.86
SMOTER (Torgo et al., 2013)	138.75	111.55	346.09	935.89	8.14	7.42	14.15	25.28	4.64	4.30	9.05	19.46
SMOGN (Branco et al., 2017)	136.09	109.15	339.09	944.20	8.03	7.30	14.02	25.93	4.63	4.30	8.74	20.12
SMOGN + LDS	137.31	111.79	333.15	823.07	8.02	7.39	13.71	23.22	4.63	4.39	8.71	15.80
SMOGN + FDS	137.82	109.42	340.65	847.96	8.03	7.35	14.06	23.44	4.65	4.33	8.87	16.00
SMOGN + LDS + FDS	<b>135.26</b>	110.91	<b>326.52</b>	<b>808.45</b>	<b>7.97</b>	7.38	<b>13.22</b>	<b>22.95</b>	<b>4.59</b>	4.39	<b>7.84</b>	<b>14.94</b>
FOCAL-R	136.98	106.87	368.60	1002.90	7.97	7.12	15.14	26.96	4.49	4.10	10.37	21.20
FOCAL-R + LDS	132.81	105.62	354.37	949.03	<b>7.90</b>	<b>7.10</b>	14.72	25.84	<b>4.47</b>	<b>4.09</b>	10.11	19.14
FOCAL-R + FDS	133.74	105.35	351.00	958.91	7.96	7.14	14.71	26.06	4.51	4.12	10.16	19.56
FOCAL-R + LDS + FDS	<b>132.58</b>	<b>105.33</b>	<b>338.65</b>	<b>944.92</b>	<b>7.88</b>	<b>7.10</b>	<b>14.08</b>	<b>25.75</b>	<b>4.47</b>	4.11	<b>9.32</b>	<b>18.67</b>
RRT	132.99	105.73	341.36	928.26	7.81	7.07	14.06	25.13	4.35	4.03	8.91	16.96
RRT + LDS	132.91	105.97	338.98	916.98	7.79	7.08	13.76	24.64	4.34	<b>4.02</b>	8.72	16.92
RRT + FDS	129.88	<b>104.63</b>	310.69	890.04	<b>7.65</b>	<b>7.02</b>	12.68	23.85	<b>4.31</b>	4.03	7.58	16.28
RRT + LDS + FDS	<b>129.14</b>	105.92	<b>306.69</b>	<b>880.13</b>	<b>7.65</b>	7.06	<b>12.41</b>	<b>23.51</b>	<b>4.31</b>	4.07	<b>7.17</b>	<b>15.44</b>
INV	139.48	116.72	305.19	869.50	8.17	7.64	12.46	22.83	4.70	4.51	6.94	13.78
SQINV	134.36	111.23	308.63	834.08	7.87	7.24	12.44	22.76	4.47	4.22	7.25	15.10
SQINV + LDS	131.65	109.04	<b>298.98</b>	829.35	7.83	7.31	<b>12.43</b>	22.51	4.42	4.19	7.00	13.94
SQINV + FDS	132.64	109.28	311.35	851.06	7.83	7.23	12.60	22.37	4.42	4.20	<b>6.93</b>	13.48
SQINV + LDS + FDS	<b>129.35</b>	<b>106.52</b>	311.49	<b>811.82</b>	<b>7.78</b>	<b>7.20</b>	12.61	<b>22.19</b>	<b>4.37</b>	<b>4.12</b>	7.39	<b>12.61</b>
<b>OURS (BEST) VS. VANILLA</b>	<b>+8.92</b>	<b>+4.07</b>	<b>+67.11</b>	<b>+156.47</b>	<b>+0.41</b>	<b>+0.21</b>	<b>+2.71</b>	<b>+4.14</b>	<b>+0.26</b>	<b>+0.15</b>	<b>+3.66</b>	<b>+7.85</b>

- **Mixup** (Zhang et al., 2018): MIXUP trains a deep model using samples created by the convex combinations of pairs of inputs and corresponding labels. It has shown promising results on improving the generalization of deep models as a regularization technique.
- **Manifold-Mixup** (M-MIXUP) (Verma et al., 2019): M-MIXUP extends the idea of MIXUP from input space to the hidden representation space, where the linear interpolations are performed in (multiple) deep hidden layers.

We note that both MIXUP and M-MIXUP are not tailored for imbalanced regression problems, but share similarities with SMOTER and SMOGN as synthetic samples are constructed. The differences lie in the fact that MIXUP and M-MIXUP create virtual samples (either in input space or feature space) on the fly during network training, while SMOTER and SMOGN operate on a newly generated and fixed dataset for training. We set  $\alpha = 0.2$  for MIXUP in implementation, and set  $\alpha = 0.2$  as well and eligible layers  $\mathcal{S} = \{0, 1, 2, 3\}$  for M-MIXUP. In addition, for INV which re-weights the loss based on the inverse frequency in the empirical label distribution, we further clip the maximum weight to be at most  $200\times$  larger than the minimum weight to avoid extreme loss values.

We show the complete results in Table 8. As the table illustrates, both MIXUP and M-MIXUP can improve the performance in the many-shot region, but lead to negligible improvements in the medium-shot and few-shot regions. In contrast, adding both FDS and LDS can substantially improve the results, especially for the underrepresented regions. Finally, FDS and LDS lead to remarkable improvements when compared to the VANILLA model across all evaluation metrics.

## D.2. Complete Results on AgeDB-DIR

We provide complete evaluation results for AgeDB-DIR in Table 9. Similar to IMDB-WIKI-DIR, within each group of techniques, adding either LDS, FDS, or both can lead to performance gains, while LDS + FDS often achieves the best results. Overall, for different groups of strategies, both FDS and LDS consistently boost the performance, where the larger gains come from the medium-shot and few-shot regions.

Table 9. Complete evaluation results on AgeDB-DIR.

Metrics	MSE ↓				MAE ↓				GM ↓			
	Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.
VANILLA	101.60	78.40	138.52	253.74	7.77	6.62	9.55	13.67	5.05	4.23	7.01	10.75
VANILLA + LDS	102.22	83.62	128.73	<b>204.64</b>	7.67	6.98	8.86	10.89	4.85	4.39	5.80	7.45
VANILLA + FDS	<b>98.55</b>	<b>75.06</b>	123.58	235.70	<b>7.55</b>	<b>6.50</b>	8.97	13.01	4.75	<b>4.03</b>	6.42	9.93
VANILLA + LDS + FDS	99.46	84.10	<b>112.20</b>	209.27	<b>7.55</b>	7.01	<b>8.24</b>	<b>10.79</b>	<b>4.72</b>	4.36	<b>5.45</b>	<b>6.79</b>
SMOTER (Torgo et al., 2013)	114.34	93.35	129.89	244.57	8.16	7.39	8.65	12.28	5.21	4.65	5.69	8.49
SMOGN (Branco et al., 2017)	117.29	101.36	133.86	232.90	8.26	7.64	9.01	12.09	5.36	4.90	6.19	8.44
SMOGN + LDS	110.43	93.73	124.19	229.35	7.96	7.44	8.64	11.77	5.03	4.68	5.69	7.98
SMOGN + FDS	112.42	97.68	131.37	233.30	8.06	7.52	8.75	11.89	5.02	4.66	5.63	8.02
SMOGN + LDS + FDS	<b>108.41</b>	<b>91.58</b>	<b>120.28</b>	<b>218.59</b>	<b>7.90</b>	<b>7.32</b>	<b>8.51</b>	<b>11.19</b>	<b>4.98</b>	<b>4.64</b>	<b>5.41</b>	<b>7.35</b>
FOCAL-R	101.26	77.03	131.81	252.47	7.64	6.68	9.22	13.00	4.90	4.26	6.39	9.52
FOCAL-R + LDS	98.80	77.14	125.53	229.36	<b>7.56</b>	<b>6.67</b>	8.82	12.40	4.82	4.27	5.87	8.83
FOCAL-R + FDS	100.14	80.97	121.84	<b>221.15</b>	7.65	6.89	8.70	<b>11.92</b>	4.83	4.32	5.89	<b>8.04</b>
FOCAL-R + LDS + FDS	<b>96.70</b>	<b>76.11</b>	<b>115.86</b>	238.25	<b>7.47</b>	6.69	<b>8.30</b>	12.55	<b>4.71</b>	<b>4.25</b>	<b>5.36</b>	8.59
RRT	102.89	83.37	125.66	224.27	7.74	6.98	8.79	11.99	5.00	4.50	5.88	8.63
RRT + LDS	102.63	83.93	126.01	214.66	7.72	7.00	8.75	11.62	4.98	4.54	5.71	8.27
RRT + FDS	102.09	84.49	122.89	224.05	7.70	<b>6.95</b>	8.76	11.86	4.82	<b>4.32</b>	5.83	8.08
RRT + LDS + FDS	<b>101.74</b>	<b>83.12</b>	<b>121.08</b>	<b>210.78</b>	<b>7.66</b>	6.99	<b>8.60</b>	<b>11.32</b>	<b>4.80</b>	4.42	<b>5.53</b>	<b>6.99</b>
INV	110.24	91.93	130.68	211.92	7.97	7.31	8.81	11.62	5.05	4.64	5.75	8.20
SQINV	105.14	87.21	127.66	212.30	7.81	7.16	8.80	11.20	4.99	4.57	5.73	7.77
SQINV + LDS	102.22	<b>83.62</b>	128.73	204.64	7.67	<b>6.98</b>	8.86	10.89	4.85	4.39	5.80	7.45
SQINV + FDS	101.67	86.49	129.61	<b>167.75</b>	7.69	7.10	8.86	<b>9.98</b>	4.83	4.41	5.97	<b>6.29</b>
SQINV + LDS + FDS	<b>99.46</b>	84.10	<b>112.20</b>	209.27	<b>7.55</b>	7.01	<b>8.24</b>	10.79	<b>4.72</b>	<b>4.36</b>	<b>5.45</b>	6.79
OURS (BEST) vs. VANILLA	<b>+4.90</b>	<b>+3.34</b>	<b>+26.32</b>	<b>+85.99</b>	<b>+0.30</b>	<b>+0.12</b>	<b>+1.31</b>	<b>+3.69</b>	<b>+0.34</b>	<b>+0.20</b>	<b>+1.65</b>	<b>+4.46</b>

Table 10. Complete evaluation results on STS-B-DIR.

Metrics	MSE ↓				MAE ↓				Pearson correlation (%) ↑				Spearman correlation (%) ↑			
	Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.
VANILLA	0.974	0.851	1.520	0.984	0.794	0.740	1.043	0.771	74.2	72.0	62.7	75.2	74.4	68.8	50.5	<b>75.0</b>
VANILLA + LDS	0.914	0.819	1.319	0.955	0.773	0.729	0.970	0.772	75.6	73.4	63.8	76.2	76.1	70.4	<b>55.6</b>	74.3
VANILLA + FDS	0.916	0.875	<b>1.027</b>	1.086	0.767	0.746	<b>0.840</b>	0.811	75.5	73.0	<b>67.0</b>	72.8	75.8	69.9	54.4	72.0
VANILLA + LDS + FDS	<b>0.907</b>	<b>0.802</b>	1.363	<b>0.942</b>	<b>0.766</b>	<b>0.718</b>	0.986	<b>0.755</b>	<b>76.0</b>	<b>74.0</b>	65.2	<b>76.6</b>	<b>76.4</b>	<b>70.7</b>	54.9	74.9
SMOTER (Torgo et al., 2013)	1.046	0.924	1.542	1.154	0.834	0.782	1.052	0.861	72.6	69.3	65.3	70.6	72.6	65.6	<b>55.6</b>	69.1
SMOGN (Branco et al., 2017)	0.990	0.896	1.327	1.175	0.798	0.755	0.967	0.848	73.2	70.4	65.5	69.2	73.2	67.0	55.1	67.0
SMOGN + LDS	0.962	0.880	1.242	1.155	0.787	0.748	0.944	0.837	74.0	71.5	65.2	69.8	74.3	68.5	53.6	67.1
SMOGN + FDS	0.987	0.945	<b>1.101</b>	1.153	0.796	0.776	<b>0.864</b>	0.838	73.0	69.6	<b>68.5</b>	69.9	72.9	66.0	54.3	68.0
SMOGN + LDS + FDS	<b>0.950</b>	<b>0.851</b>	1.327	<b>1.095</b>	<b>0.785</b>	<b>0.738</b>	0.987	<b>0.799</b>	<b>74.6</b>	<b>72.1</b>	65.9	<b>71.7</b>	<b>75.0</b>	<b>68.9</b>	54.4	<b>70.3</b>
FOCAL-R	0.951	0.843	1.425	0.957	0.790	0.739	1.028	0.759	74.6	72.3	61.8	76.4	75.0	69.4	51.9	75.5
FOCAL-R + LDS	0.930	<b>0.807</b>	1.449	0.993	0.781	<b>0.723</b>	1.031	0.801	<b>75.7</b>	<b>73.9</b>	62.4	75.4	<b>76.2</b>	<b>71.2</b>	50.7	74.7
FOCAL-R + FDS	<b>0.920</b>	0.855	<b>1.169</b>	1.008	<b>0.775</b>	0.743	<b>0.903</b>	0.804	75.1	72.6	<b>66.4</b>	74.7	75.4	69.4	<b>52.7</b>	75.4
FOCAL-R + LDS + FDS	0.940	0.849	1.358	<b>0.916</b>	0.785	0.737	0.984	<b>0.732</b>	74.9	72.2	66.3	<b>77.3</b>	75.1	69.2	52.5	<b>76.4</b>
RRT	0.964	0.842	1.503	0.978	0.793	0.739	1.044	0.768	74.5	72.4	62.3	75.4	74.7	69.2	51.3	<b>74.7</b>
RRT + LDS	0.916	0.817	1.344	0.945	0.772	0.727	0.980	<b>0.756</b>	75.7	73.5	64.1	76.6	76.1	70.4	53.2	74.2
RRT + FDS	0.929	0.857	<b>1.209</b>	1.025	0.769	0.736	<b>0.905</b>	0.795	74.9	72.1	<b>67.2</b>	74.0	75.0	69.1	52.8	74.6
RRT + LDS + FDS	<b>0.903</b>	<b>0.806</b>	1.323	<b>0.936</b>	<b>0.764</b>	<b>0.719</b>	0.965	0.760	<b>76.0</b>	<b>73.8</b>	65.2	<b>76.7</b>	<b>76.4</b>	<b>70.8</b>	<b>54.7</b>	<b>74.7</b>
INV	1.005	0.894	1.482	1.046	0.805	0.761	1.016	0.780	72.8	70.3	62.5	73.2	73.1	67.2	54.1	71.4
INV + LDS	0.914	0.819	1.319	0.955	0.773	0.729	0.970	0.772	75.6	73.4	63.8	76.2	76.1	70.4	<b>55.6</b>	74.3
INV + FDS	0.927	0.851	<b>1.225</b>	1.012	0.771	0.740	<b>0.914</b>	0.756	75.0	72.4	<b>66.6</b>	74.2	75.2	69.2	55.2	74.8
INV + LDS + FDS	<b>0.907</b>	<b>0.802</b>	1.363	<b>0.942</b>	<b>0.766</b>	<b>0.718</b>	0.986	<b>0.755</b>	<b>76.0</b>	<b>74.0</b>	65.2	<b>76.6</b>	<b>76.4</b>	<b>70.7</b>	54.9	<b>74.9</b>
OURS (BEST) vs. VANILLA	<b>+.071</b>	<b>.049</b>	<b>.419</b>	<b>.068</b>	<b>+.030</b>	<b>.022</b>	<b>.203</b>	<b>.039</b>	<b>+1.8</b>	<b>+2.0</b>	<b>+5.8</b>	<b>+2.1</b>	<b>+2.0</b>	<b>+2.4</b>	<b>+5.1</b>	<b>+1.4</b>

### D.3. Complete Results on STS-B-DIR

We present complete results on STS-B-DIR in Table 10, where more metrics, such as MAE and Spearman correlation are added for further evaluation. In summary, across all the metrics used, by adding LDS and FDS we can substantially improve the results, particularly for the medium-shot and few-shot regions. The advantage is even more profound under *Pearson correlation*, which is commonly used for this task.

Table 11. Complete evaluation results on NYUD2-DIR.

Metrics	RMSE ↓				$\log_{10} \downarrow$				$\delta_1 \uparrow$				$\delta_2 \uparrow$				$\delta_3 \uparrow$			
	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
Shot																				
VANILLA	1.477	0.591	0.952	2.123	0.086	0.066	0.082	0.107	0.677	0.777	0.693	0.570	0.899	0.956	0.906	0.840	0.969	0.990	0.975	0.946
VANILLA + LDS	1.387	0.671	0.913	1.954	0.086	0.079	0.079	0.097	0.672	0.701	0.706	0.630	0.907	0.932	0.929	0.875	0.976	0.984	0.982	0.964
VANILLA + FDS	1.442	<b>0.615</b>	0.940	2.059	0.084	<b>0.069</b>	0.080	0.101	0.681	<b>0.760</b>	0.695	0.596	0.903	<b>0.952</b>	0.918	0.849	0.975	<b>0.989</b>	0.976	0.960
VANILLA + LDS + FDS	1.338	0.670	<b>0.851</b>	<b>1.880</b>	<b>0.080</b>	0.074	<b>0.070</b>	<b>0.090</b>	<b>0.705</b>	0.730	<b>0.764</b>	<b>0.655</b>	<b>0.916</b>	0.939	<b>0.941</b>	<b>0.884</b>	<b>0.979</b>	0.984	<b>0.983</b>	<b>0.971</b>
OURS (BEST) VS. VANILLA	+1.19	-0.24	+.101	+.243	+.006	-0.03	+.012	+.017	+.028	-0.017	+.071	+.085	+.017	-0.004	+.035	+.044	+.010	-0.001	+.008	+.025

Table 12. Complete evaluation results on SHHS-DIR.

Metrics	MSE ↓				MAE ↓				GM ↓			
	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
Shot												
VANILLA	369.18	269.37	311.45	417.31	15.36	12.47	13.98	16.94	10.63	8.04	9.59	12.20
VANILLA + LDS	309.19	220.87	252.53	394.91	14.14	11.66	12.77	16.05	9.26	7.64	8.18	11.32
VANILLA + FDS	303.82	214.63	267.08	386.75	13.84	11.13	12.72	15.95	8.89	<b>6.93</b>	8.05	11.19
VANILLA + LDS + FDS	<b>292.18</b>	<b>211.89</b>	<b>247.48</b>	<b>346.01</b>	<b>13.76</b>	<b>11.12</b>	<b>12.18</b>	<b>15.07</b>	<b>8.70</b>	6.94	<b>7.60</b>	<b>10.18</b>
FOCAL-R	345.44	219.75	309.01	430.26	14.67	11.70	13.69	17.06	9.98	7.93	8.85	11.95
FOCAL-R + LDS	317.39	242.18	270.04	411.73	14.49	12.01	12.43	16.57	9.98	7.89	8.59	11.40
FOCAL-R + FDS	310.94	<b>185.16</b>	303.90	391.22	14.18	<b>11.06</b>	13.56	15.99	9.45	<b>6.95</b>	8.81	11.13
FOCAL-R + LDS + FDS	<b>297.85</b>	193.42	<b>259.33</b>	<b>375.16</b>	<b>14.02</b>	11.08	<b>12.24</b>	<b>15.49</b>	<b>9.32</b>	7.18	<b>8.10</b>	<b>10.39</b>
RRT	354.75	274.01	308.83	408.47	14.78	12.43	14.01	16.48	10.12	8.05	9.71	11.96
RRT + LDS	344.18	245.39	304.32	402.56	14.56	12.08	13.44	16.45	9.89	7.85	9.18	11.82
RRT + FDS	328.66	239.83	298.71	397.25	14.36	11.97	13.33	16.08	9.74	7.54	9.20	11.31
RRT + LDS + FDS	<b>313.58</b>	<b>238.07</b>	<b>276.50</b>	<b>380.64</b>	<b>14.33</b>	<b>11.96</b>	<b>12.47</b>	<b>15.92</b>	<b>9.63</b>	<b>7.35</b>	<b>8.74</b>	<b>11.17</b>
INV	322.17	231.68	293.43	387.48	14.39	11.84	13.12	16.02	9.34	7.73	8.49	11.20
INV + LDS	309.19	220.87	252.53	394.91	14.14	11.66	12.77	16.05	9.26	7.64	8.18	11.32
INV + FDS	307.95	219.36	247.55	361.29	13.91	<b>11.12</b>	12.29	15.53	8.94	<b>6.91</b>	7.79	10.65
INV + LDS + FDS	<b>292.18</b>	<b>211.89</b>	<b>247.48</b>	<b>346.01</b>	<b>13.76</b>	<b>11.12</b>	<b>12.18</b>	<b>15.07</b>	<b>8.70</b>	6.94	<b>7.60</b>	<b>10.18</b>
OURS (BEST) VS. VANILLA	<b>+77.00</b>	<b>+84.21</b>	<b>+63.97</b>	<b>+71.30</b>	<b>+1.60</b>	<b>+1.41</b>	<b>+1.80</b>	<b>+1.87</b>	<b>+1.93</b>	<b>+1.13</b>	<b>+1.99</b>	<b>+2.02</b>

#### D.4. Complete Results on NYUD2-DIR

Table 11 shows the complete evaluation results on NYUD2-DIR. As described before, we further add common metrics for depth estimation evaluation, including  $\log_{10}$ ,  $\delta_1$ ,  $\delta_2$ , and  $\delta_3$ . The table reveals the following results. First, either FDS or LDS alone can improve the overall depth regression results, where LDS is more effective for improving performance in the few-shot region. Furthermore, when combined together, LDS & FDS can alleviate the overfitting phenomenon to many-shot regions of the vanilla model, and generalize better to all regions.

#### D.5. Complete Results on SHHS-DIR

We report the complete results on SHHS-DIR in Table 12. The results again confirm the effectiveness of both LDS and FDS beyond the success on typical image data and text data, as superior performance is demonstrated when applied for real-world imbalanced regression tasks with healthcare data as inputs (i.e., PSG signals). We verify that by combining LDS and FDS, the highest performance gains are established over all tested regions.

### E. Further Analysis and Ablation Studies

#### E.1. Kernel Type for LDS & FDS

We study the effects of different kernel types for LDS and FDS when applying distribution smoothing, in addition to the default setting where Gaussian kernels are employed. We select three different kernel types, i.e., Gaussian, Laplacian, and Triangular kernel, and evaluate their effects on both LDS and FDS. We remain other hyper-parameters unchanged as in Sec. C.1, and report results on IMDB-WIKI-DIR in Table 13 and results on STS-B-DIR in Table 14. In general, as both tables indicate, all kernel types can lead to notable gains compared to the vanilla model. Moreover, Gaussian kernel often delivers the best results among all kernel types, which is consistent for both LDS and FDS.

Table 13. Ablation study of different kernel types for LDS &amp; FDS on IMDB-WIKI-DIR.

Metrics	MSE ↓				MAE ↓				GM ↓			
	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	138.06	108.70	366.09	964.92	8.06	7.23	15.12	26.33	4.57	4.17	10.59	20.46
<b>LDS:</b>												
GAUSSIAN KERNEL	131.65	109.04	298.98	834.08	7.83	7.31	12.43	22.51	4.42	4.19	7.00	13.94
TRIANGULAR KERNEL	133.77	110.24	309.70	850.74	7.89	7.30	12.72	22.80	4.50	4.24	7.75	14.91
LAPLACIAN KERNEL	132.87	109.27	312.10	829.83	7.87	7.29	12.68	22.38	4.50	4.26	7.29	13.71
<b>FDS:</b>												
GAUSSIAN KERNEL	133.81	107.51	332.90	916.18	7.85	7.18	13.35	24.12	4.47	4.18	8.18	15.18
TRIANGULAR KERNEL	134.09	110.49	301.18	927.99	7.97	7.41	12.20	23.99	4.64	4.41	7.06	14.28
LAPLACIAN KERNEL	133.00	104.26	352.95	968.62	8.05	7.25	14.78	26.16	4.71	4.33	10.19	19.09

Table 14. Ablation study of different kernel types for LDS &amp; FDS on STS-B-DIR.

Metrics	MSE ↓				MAE ↓				Pearson correlation (%) ↑				Spearman correlation (%) ↑			
	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	0.974	0.851	1.520	0.984	0.794	0.740	1.043	0.771	74.2	72.0	62.7	75.2	74.4	68.8	50.5	75.0
<b>LDS:</b>																
GAUSSIAN KERNEL	0.914	0.819	1.319	0.955	0.773	0.729	0.970	0.772	75.6	73.4	63.8	76.2	76.1	70.4	55.6	74.3
TRIANGULAR KERNEL	0.938	0.870	1.193	1.039	0.786	0.754	0.929	0.784	74.8	72.4	64.1	74.0	75.2	69.3	54.1	73.9
LAPLACIAN KERNEL	0.938	0.829	1.413	0.962	0.782	0.731	1.014	0.773	75.7	73.0	65.8	76.5	76.0	70.0	52.3	75.2
<b>FDS:</b>																
GAUSSIAN KERNEL	0.916	0.875	1.027	1.086	0.767	0.746	0.840	0.811	75.5	73.0	67.0	72.8	75.8	69.9	54.4	72.0
TRIANGULAR KERNEL	0.935	0.863	1.239	0.966	0.762	0.725	0.912	0.788	74.6	72.4	64.8	75.9	74.4	69.1	48.4	75.4
LAPLACIAN KERNEL	0.925	0.843	1.247	1.020	0.771	0.733	0.929	0.800	75.0	72.6	64.7	74.2	75.4	70.1	53.5	73.5

## E.2. Training Loss for LDS & FDS

In the main paper, we fix the training loss function used for each dataset (e.g., MSE loss is used for experiments on STS-B-DIR). In this section, we investigate the influence of different training loss functions on LDS & FDS. We select three common losses used for regression tasks, i.e.,  $L_1$  loss, MSE loss, and the Huber loss (also referred to as smoothed  $L_1$  loss). We show the results on STS-B-DIR in Table 15, where similar results are obtained for all the losses, with no significant performance differences observed between loss functions, indicating that FDS & LDS are robust to different loss functions.

Table 15. Ablation study of different loss functions used during training for LDS &amp; FDS on STS-B-DIR.

Metrics	MSE ↓				MAE ↓				Pearson correlation (%) ↑				Spearman correlation (%) ↑			
	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
<b>LDS:</b>																
L1	0.893	0.808	1.241	0.964	0.765	0.727	0.938	0.758	76.3	73.9	66.0	75.9	76.7	71.1	54.5	75.6
MSE	0.914	0.819	1.319	0.955	0.773	0.729	0.970	0.772	75.6	73.4	63.8	76.2	76.1	70.4	55.6	74.3
HUBER LOSS	0.902	0.811	1.276	0.978	0.761	0.718	0.954	0.751	76.1	74.2	64.7	75.5	76.5	71.6	52.9	74.3
<b>FDS:</b>																
L1	0.918	0.860	1.105	1.082	0.762	0.733	0.859	0.833	75.5	73.7	65.3	72.3	75.6	70.9	52.1	71.5
MSE	0.916	0.875	1.027	1.086	0.767	0.746	0.840	0.811	75.5	73.0	67.0	72.8	75.8	69.9	54.4	72.0
HUBER LOSS	0.920	0.867	1.097	1.052	0.765	0.741	0.858	0.800	75.3	72.9	66.6	73.6	75.3	69.7	52.3	73.6

## E.3. Hyper-parameters for LDS & FDS

In this section, we study the effects of different hyper-parameters on both LDS and FDS. As we mainly employ the Gaussian kernel for distribution smoothing, we extensively study different choices of the kernel size  $l$  and the standard deviation  $\sigma$ . Specifically, we conduct controlled experiments on IMDB-WIKI-DIR and STS-B-DIR, where we vary the choices of these hyper-parameters as  $l \in \{5, 9, 15\}$  and  $\sigma \in \{1, 2, 3\}$ , and leave other training hyper-parameters unchanged.

Table 16. Hyper-parameter study on kernel size  $l$  and standard deviation  $\sigma$  for LDS & FDS on IMDB-WIKI-DIR.

Metrics		MSE ↓				MAE ↓				GM ↓			
Shot		All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA		138.06	108.70	366.09	964.92	8.06	7.23	15.12	26.33	4.57	4.17	10.59	20.46
$l$	$\sigma$												
<b>LDS:</b>													
5	1	132.08	108.53	309.03	843.53	7.80	7.22	12.61	22.33	4.42	4.19	7.16	12.54
9	1	135.04	112.32	307.90	803.15	7.97	7.39	12.74	22.19	4.55	4.30	7.53	14.11
15	1	134.06	110.49	308.83	864.30	7.84	7.28	12.35	22.81	4.44	4.22	6.95	14.22
5	2	131.65	109.04	298.98	834.08	7.83	7.31	12.43	22.51	4.42	4.19	7.00	13.94
9	2	136.78	112.41	322.65	850.47	8.02	7.41	13.00	23.23	4.55	4.29	7.55	15.65
15	2	135.66	111.68	319.20	833.02	7.98	7.40	12.74	22.27	4.60	4.37	7.30	12.92
5	3	137.56	113.50	322.47	831.38	8.07	7.47	13.06	22.85	4.63	4.36	7.87	15.11
9	3	138.91	114.89	319.40	863.16	8.18	7.57	13.19	23.33	4.71	4.44	8.09	15.17
15	3	138.86	114.25	326.97	856.27	8.18	7.54	13.53	23.17	4.77	4.47	8.52	15.25
<b>FDS:</b>													
5	1	133.63	104.80	354.24	972.54	7.87	7.06	14.71	25.96	4.42	4.04	9.95	18.47
9	1	134.34	105.97	356.54	919.16	7.95	7.18	14.58	24.80	4.54	4.20	9.56	15.13
15	1	136.32	107.47	355.84	948.71	7.97	7.23	14.81	25.59	4.60	4.23	9.99	17.60
5	2	133.81	107.51	332.90	916.18	7.85	7.18	13.35	24.12	4.47	4.18	8.18	15.18
9	2	133.99	105.01	357.31	963.79	7.94	7.11	14.95	25.97	4.48	4.09	10.49	18.19
15	2	136.61	107.93	361.08	973.56	7.98	7.23	14.68	25.21	4.61	4.24	10.14	17.91
5	3	136.81	107.76	359.08	953.16	7.98	7.18	14.85	24.94	4.53	4.15	10.27	17.33
9	3	133.48	104.14	359.80	972.29	7.94	7.09	15.04	25.87	4.48	4.09	10.40	16.85
15	3	132.55	103.08	360.39	970.43	8.03	7.22	14.86	25.40	4.67	4.33	10.04	13.86

 Table 17. Hyper-parameter study on kernel size  $l$  and standard deviation  $\sigma$  for LDS & FDS on STS-B-DIR.

Metrics		MSE ↓				MAE ↓				Pearson correlation (%) ↑				Spearman correlation (%) ↑			
Shot		All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA		0.974	0.851	1.520	0.984	0.794	0.740	1.043	0.771	74.2	72.0	62.7	75.2	74.4	68.8	50.5	75.0
$l$	$\sigma$																
<b>LDS:</b>																	
5	1	0.942	0.825	1.431	1.023	0.781	0.726	1.016	0.809	75.1	73.2	61.8	74.5	75.3	70.2	52.2	72.5
9	1	0.931	0.840	1.323	0.962	0.785	0.744	0.972	0.773	75.0	72.7	63.3	75.8	75.6	70.1	53.6	74.8
15	1	0.941	0.833	1.413	0.953	0.781	0.728	1.014	0.776	75.0	72.8	62.6	76.3	75.5	70.2	52.0	74.6
5	2	0.914	0.819	1.319	0.955	0.773	0.729	0.970	0.772	75.6	73.4	63.8	76.2	76.1	70.4	55.6	74.3
9	2	0.926	0.823	1.379	0.944	0.782	0.733	1.003	0.764	75.5	73.4	63.6	76.8	76.0	70.5	53.5	76.2
15	2	0.949	0.831	1.452	1.005	0.788	0.735	1.023	0.782	74.9	72.9	63.0	74.7	75.4	70.1	52.5	73.6
5	3	0.928	0.845	1.250	1.041	0.775	0.733	0.951	0.798	75.1	73.3	63.2	73.8	75.3	70.4	51.4	72.6
9	3	0.939	0.816	1.462	1.000	0.786	0.732	1.030	0.783	75.3	73.5	62.6	74.7	75.9	70.9	53.0	73.7
15	3	0.927	0.824	1.348	1.010	0.774	0.726	0.982	0.780	75.2	73.4	62.2	74.6	75.7	70.7	53.0	72.3
<b>FDS:</b>																	
5	1	0.943	0.869	1.217	1.066	0.776	0.742	0.914	0.799	74.4	71.7	65.6	72.5	74.2	68.4	51.1	71.2
9	1	0.927	0.851	1.193	1.096	0.770	0.736	0.896	0.822	74.9	72.8	65.8	71.6	74.8	69.7	52.3	68.3
15	1	0.926	0.854	1.202	1.029	0.776	0.743	0.914	0.800	74.9	72.6	66.1	74.0	75.1	69.8	49.5	73.6
5	2	0.916	0.875	1.027	1.086	0.767	0.746	0.840	0.811	75.5	73.0	67.0	72.8	75.8	69.9	54.4	72.0
9	2	0.933	0.888	1.068	1.081	0.776	0.752	0.855	0.839	74.8	72.0	67.9	72.2	74.9	68.9	53.3	72.0
15	2	0.944	0.890	1.125	1.078	0.783	0.761	0.864	0.822	74.4	71.8	65.8	72.2	74.5	68.9	53.1	70.9
5	3	0.924	0.860	1.190	0.964	0.771	0.740	0.897	0.790	75.0	72.7	64.4	76.1	75.1	69.4	53.8	76.5
9	3	0.932	0.878	1.149	0.982	0.770	0.746	0.876	0.780	74.8	72.5	63.8	75.3	74.8	69.3	50.2	75.6
15	3	0.956	0.915	1.110	1.016	0.784	0.767	0.855	0.803	74.4	72.1	63.7	75.5	74.3	68.7	50.0	74.6

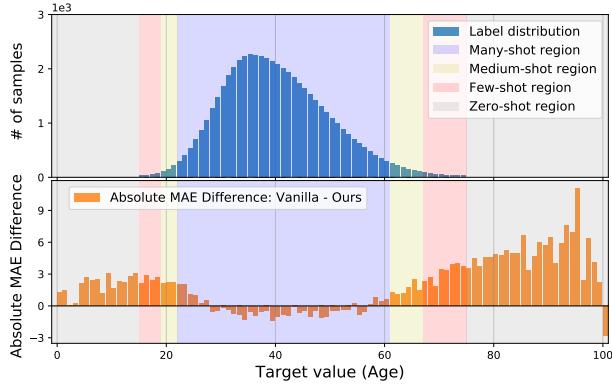
**IMDB-WIKI-DIR.** We first report the results on IMDB-WIKI-DIR in Table 16. The table reveals the following observations. First, both LDS and FDS are robust to different hyper-parameters within the given range, where similar performance gains are obtained across different choices of  $\{l, \sigma\}$ . Specifically, for LDS, the relative MAE improvements in the few-shot regions range from 11.4% to 15.7%, where a smaller  $\sigma$  usually leads to slightly better results over all regions. As for FDS, similar conclusion can be made, while a smaller  $l$  often obtains slightly higher improvements. Interestingly, we can also observe

that LDS leads to larger gains w.r.t. the performance in medium-shot and few-shot regions, while with minor degradation in many-shot regions. In contrast, FDS equally boosts all the regions, with slightly smaller improvements in medium-shot and few-shot regions compared to LDS. Finally, for both LDS and FDS, setting  $l = 5$  and  $\sigma = 2$  exhibits the best results.

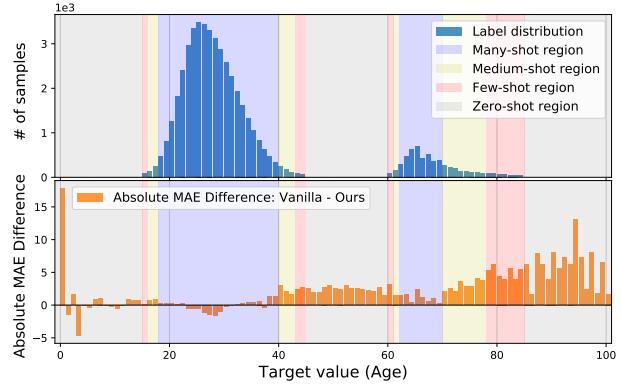
**STS-B-DIR.** Further, we show the results of different hyper-parameters on STS-B-DIR in Table 17. Similar to the results on IMDB-WIKI-DIR, we observe that both LDS and FDS are robust to the hyper-parameter changes, where the performance gaps between  $\{l, \sigma\}$  pairs become smaller. In summary, the overall MSE gains range from 3.3% to 6.2% compared to the vanilla model, with  $l = 5$  and  $\sigma = 2$  exhibiting the best results for both LDS and FDS.

#### E.4. Robustness to Diverse Skewed Label Distributions

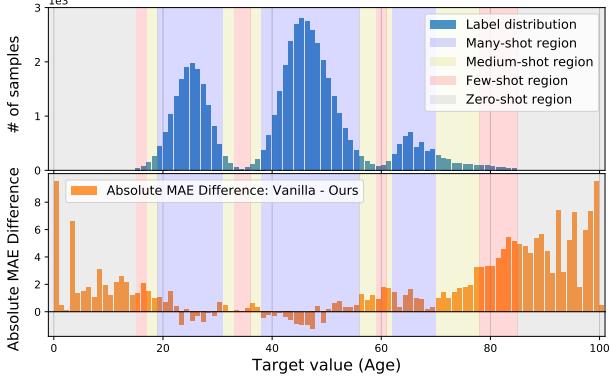
We analyze the effects of different skewed label distributions on our techniques for DIR tasks. We curate different imbalanced label distributions for IMDB-WIKI-DIR by combining different number of skewed Gaussians over the target space. Precisely, as shown in Fig. 9, we create new training sets with  $\{1, 2, 3, 4\}$  disjoint skewed Gaussian distributions over the label space, with potential missing data in certain target regions, and evaluate the robustness of LDS and FDS to the distribution change.



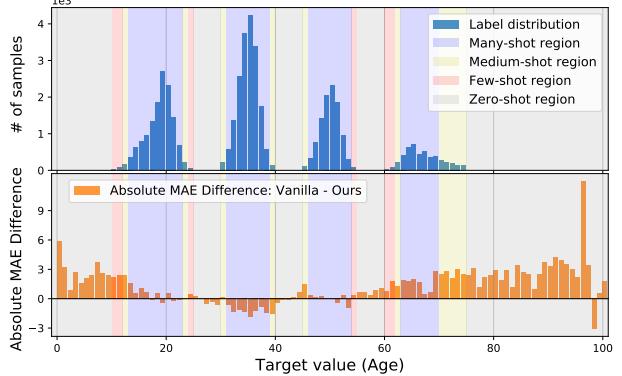
(a) Curated skewed label distribution, with 1 Gaussian peak



(b) Curated skewed label distribution, with 2 Gaussian peaks



(c) Curated skewed label distribution, with 3 Gaussian peaks



(d) Curated skewed label distribution, with 4 Gaussian peaks

Figure 9. The absolute MAE gains of LDS + FDS over the vanilla model under different skewed label distributions. We curate different imbalanced label distributions on IMDB-WIKI-DIR using different number of skewed Gaussians over the target space. We confirm that LDS and FDS are robust to distribution change, and can consistently bring improvements under different imbalanced label distributions.

We verify in Table 18 that even under different imbalanced label distributions, LDS and FDS consistently bring improvements compared to the vanilla model. Substantial improvements are established not only on regions that have data, but more prominent on those without data, i.e., zero-shot regions that require target interpolation or extrapolation. We further visualize the absolute MAE gains of our methods over the vanilla model for the curated skewed distributions in Fig. 9. Our methods provide a comprehensive treatment to the many, medium, few, as well as zero-shot regions, where remarkable performance gains are achieved across all skewed distributions, confirming the robustness of LDS and FDS under distribution change.

Table 18. Ablation study on different skewed label distributions on IMDB-WIKI-DIR.

Metrics	MAE ↓							GM ↓						
	Shot	All	Many	Med.	Few	Zero	Interp.	Extrap.	All	Many	Med.	Few	Zero	Interp.
<b>1 peak:</b>														
VANILLA	11.20	6.05	11.43	14.76	22.67	—	22.67	7.02	<b>3.84</b>	8.67	12.26	21.07	—	21.07
VANILLA + LDS	10.09	6.26	9.91	12.12	19.37	—	19.37	6.14	3.92	6.50	8.30	16.35	—	16.35
VANILLA + FDS	11.04	<b>5.97</b>	11.19	14.54	22.35	—	22.35	6.96	<b>3.84</b>	8.54	12.08	20.71	—	20.71
VANILLA + LDS + FDS	<b>10.00</b>	6.28	<b>9.66</b>	<b>11.83</b>	<b>19.21</b>	—	<b>19.21</b>	<b>6.09</b>	3.96	<b>6.26</b>	<b>8.14</b>	<b>15.89</b>	—	<b>15.89</b>
<b>2 peaks:</b>														
VANILLA	11.72	6.83	11.78	15.35	16.86	16.13	18.19	7.44	3.61	8.06	12.94	15.21	14.41	16.74
VANILLA + LDS	10.54	6.72	9.65	12.60	15.30	14.14	17.38	6.50	3.65	<b>5.65</b>	9.30	13.20	12.13	15.36
VANILLA + FDS	11.40	6.69	11.02	14.85	16.61	15.83	18.01	7.18	<b>3.50</b>	7.49	12.73	14.86	14.02	16.48
VANILLA + LDS + FDS	<b>10.27</b>	<b>6.61</b>	<b>9.46</b>	<b>11.96</b>	<b>14.89</b>	<b>13.71</b>	<b>17.02</b>	<b>6.33</b>	3.54	5.68	<b>8.80</b>	<b>12.83</b>	<b>11.71</b>	<b>15.13</b>
<b>3 peaks:</b>														
VANILLA	9.83	7.01	9.81	11.93	20.11	—	20.11	6.04	3.93	6.94	9.84	17.77	—	17.77
VANILLA + LDS	9.08	<b>6.77</b>	8.82	10.48	18.43	—	18.43	<b>5.35</b>	<b>3.78</b>	5.63	7.49	15.46	—	15.46
VANILLA + FDS	9.65	6.88	9.58	11.75	19.80	—	19.80	5.86	3.83	6.68	9.48	17.43	—	17.43
VANILLA + LDS + FDS	<b>8.96</b>	6.88	<b>8.62</b>	<b>10.08</b>	<b>17.76</b>	—	<b>17.76</b>	5.38	3.90	<b>5.61</b>	<b>7.36</b>	<b>14.65</b>	—	<b>14.65</b>
<b>4 peaks:</b>														
VANILLA	9.49	7.23	9.73	10.85	12.16	8.23	18.78	5.68	3.45	6.95	8.20	9.43	6.89	16.02
VANILLA + LDS	8.80	<b>6.98</b>	8.26	10.07	11.26	8.31	<b>16.22</b>	5.10	<b>3.33</b>	<b>5.07</b>	7.08	8.47	6.66	<b>12.74</b>
VANILLA + FDS	9.28	7.11	9.16	10.88	11.95	8.30	18.11	5.49	3.36	6.35	8.15	9.21	6.82	15.30
VANILLA + LDS + FDS	<b>8.76</b>	7.07	<b>8.23</b>	<b>9.54</b>	<b>11.13</b>	<b>8.05</b>	16.32	<b>5.05</b>	3.36	<b>5.07</b>	<b>6.56</b>	<b>8.30</b>	<b>6.34</b>	13.10

## E.5. Additional Study on Test Set Label Distributions

We define the evaluation of DIR as generalizing to a testset that is balanced over the entire target range, which is also aligned with the evaluation in the class imbalance setting (Liu et al., 2019). In this section, we further investigate the performance under different test set label distributions. Specifically, we consider the test set to have exactly the same label distribution as the training set, i.e., the test set also exhibits skewed label distribution (see IMDB-WIKI-DIR in Fig. 6). We show the results in Table 19. As the table indicates, in the balanced testset case, using LDS and FDS can consistently improve the performance of all the regions, demonstrating that our approaches provide a comprehensive and unbiased treatment to all the target values, achieving substantial improvements. Moreover, when the testset has the same label distribution as the training set, we observe that adding LDS and FDS leads to minor degradation in the many-shot region, but drastically boosts the performance in medium-shot and few-shot regions. Note that when testset also exhibits skewed label distribution, the overall performance is dominated by the many-shot region, which can result in biased and undesired evaluation for DIR tasks.

Table 19. Additional study of performance on different test set label distributions on IMDB-WIKI-DIR.

Metrics	MSE ↓				MAE ↓				GM ↓			
	Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.
<b>Balanced:</b>												
VANILLA	138.06	108.70	366.09	964.92	8.06	7.23	15.12	26.33	4.57	4.17	10.59	20.46
VANILLA + LDS + FDS	<b>129.35</b>	<b>106.52</b>	<b>311.49</b>	<b>811.82</b>	<b>7.78</b>	<b>7.20</b>	<b>12.61</b>	<b>22.19</b>	<b>4.37</b>	<b>4.12</b>	<b>7.39</b>	<b>12.61</b>
<b>Same as training set:</b>												
VANILLA	<b>68.44</b>	<b>62.10</b>	320.52	1350.01	<b>5.84</b>	<b>5.72</b>	15.11	30.54	<b>3.44</b>	<b>3.40</b>	11.76	24.06
VANILLA + LDS + FDS	69.86	63.43	<b>161.97</b>	<b>1067.89</b>	5.90	5.77	<b>9.94</b>	<b>25.17</b>	3.48	3.44	<b>7.03</b>	<b>15.95</b>

## E.6. Further Comparisons to Imbalanced Classification Methods

We provide additional study on comparisons to imbalanced classification methods. For DIR tasks that are appropriate (e.g., limited target value ranges), imbalanced classification methods can also be plugged in by discretizing the continuous label space. To gain more insights on the intrinsic difference between imbalanced classification and imbalanced regression

problems, we directly apply existing imbalanced classification schemes on several appropriate DIR datasets, and show empirical comparisons with imbalanced regression approaches. Specifically, we select the subsampled IMDB-WIKI-DIR (see Fig. 2), STS-B-DIR, and NYUD2-DIR for comparison. We compare with CB (Cui et al., 2019) and cRT (Kang et al., 2020), which are the state-of-the-art methods for imbalanced classification. We also denote the vanilla classification method as CLS-VANILLA. For fair comparison, the classes are set to the same bins used in LDS and FDS. Table 20 confirms that LDS and FDS outperform imbalanced classification schemes by a large margin across all DIR datasets, where the errors for few-shot regions can be reduced by up to 50% to 60%. Interestingly, the results also show that imbalanced classification schemes often perform *worse* than even the vanilla regression model (i.e., REG-VANILLA), which confirms that regression requires different approaches for data imbalance than simply applying classification methods.

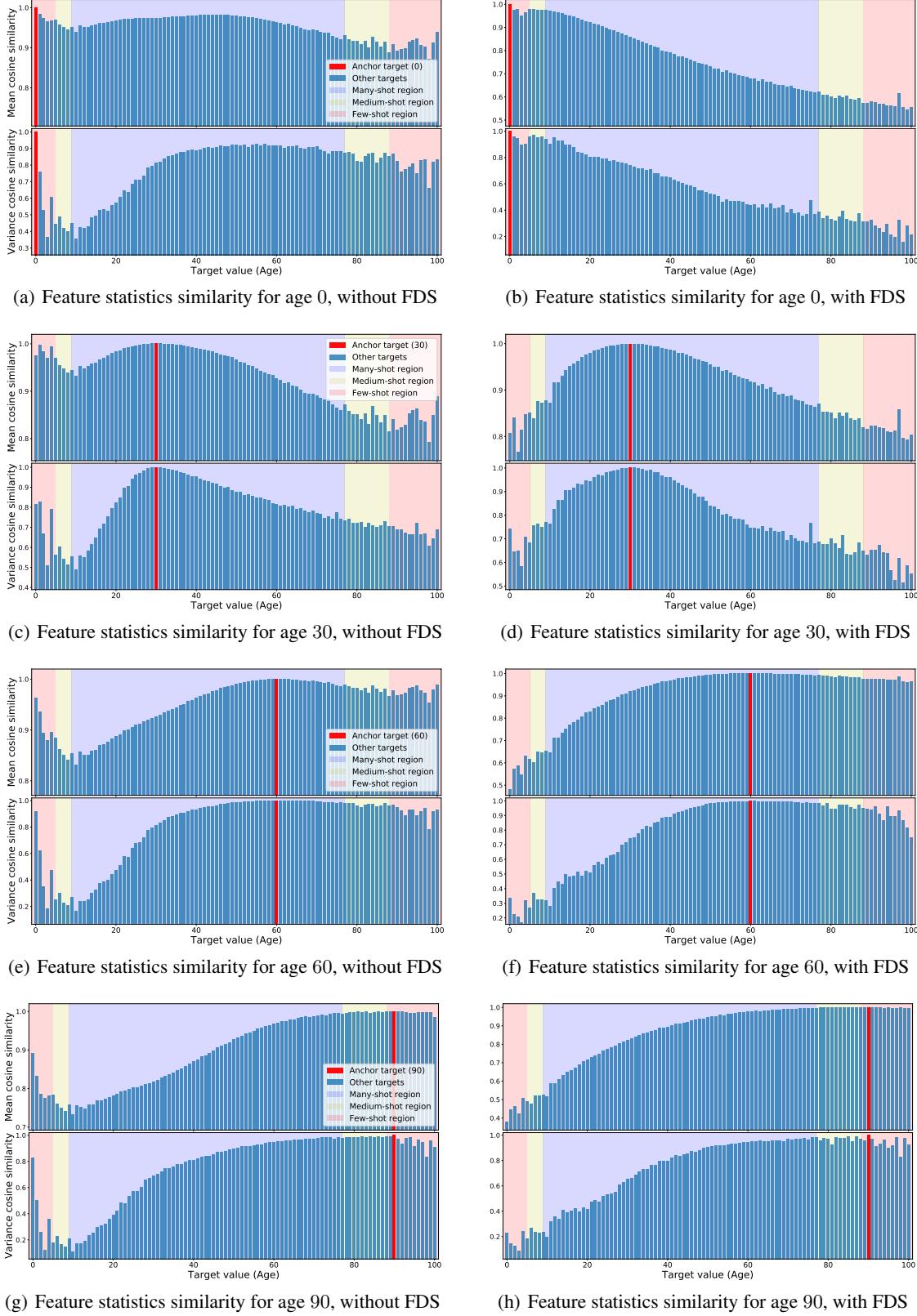
We note that imbalanced classification methods could fail on regression problems for several reasons. First, they ignore the similarity between data samples that are close w.r.t. the continuous target; Treating different target values as distinct classes is unlikely to yield the best results because it does not take advantage of the similarity between nearby targets. Moreover, classification methods cannot extrapolate or interpolate in the continuous label space, therefore unable to deal with missing data in certain target regions.

Table 20. Additional study on comparisons to imbalanced classification methods across several appropriate DIR datasets.

Dataset	IMDB-WIKI-DIR (subsampled)				STS-B-DIR				NYUD2-DIR			
Metric	MAE $\downarrow$				MSE $\downarrow$				RMSE $\downarrow$			
Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
<b>Imbalanced Classification:</b>												
CLS-VANILLA	15.94	15.64	18.95	30.21	1.926	1.906	2.022	1.907	1.576	0.596	1.011	2.275
CB (Cui et al., 2019)	22.41	22.32	22.05	32.90	2.159	2.194	2.028	2.107	1.664	0.592	1.044	2.415
CRT (Kang et al., 2020)	15.65	15.33	17.52	29.54	1.891	1.906	1.930	1.650	1.488	0.659	1.032	2.107
<b>Imbalanced Regression:</b>												
REG-VANILLA	14.64	13.98	17.47	30.29	0.974	0.851	1.520	0.984	1.477	<b>0.591</b>	0.952	2.123
LDS	14.03	13.72	15.93	26.71	0.914	0.819	1.319	0.955	1.387	0.671	0.913	1.954
FDS	13.97	13.55	16.42	24.64	0.916	0.875	<b>1.027</b>	1.086	1.442	0.615	0.940	2.059
LDS + FDS	<b>13.32</b>	<b>13.14</b>	<b>15.06</b>	<b>23.87</b>	<b>0.907</b>	<b>0.802</b>	1.363	<b>0.942</b>	<b>1.338</b>	0.670	<b>0.851</b>	<b>1.880</b>

## E.7. Complete Visualization for Feature Statistics Similarity

We provide additional results for understanding FDS, i.e., how FDS influences the feature statistics. In Fig. 10, we plot the similarity of the feature statistics for different anchor ages in  $\{0, 30, 60, 90\}$ , using models trained without and with FDS. As the figure indicates, for the vanilla model (i.e., Fig. 10(a), 10(c), 10(e), and 10(g)), there exists unexpected high similarities between the anchor ages and the regions that have very few data samples. For example, in Fig. 10(a) where the anchor age is 0, the highest similarity is obtained with age range between 40 and 80, rather than its nearby ages. Moreover, for anchor ages that lie in the many-shot regions (e.g., Fig. 10(c), 10(e), and 10(g)), they also exhibit unjustified feature statistics similarity with samples from age range 0 to 6, which is due to data imbalance. In contrast, by adding FDS (i.e., Fig. 10(b), 10(d), 10(f), and 10(h)), the statistics are better calibrated for all anchor ages, leading to a high similarity only in the neighborhood, and a gradually decreasing similarity score as target value becomes smaller or larger.



**Figure 10.** Analysis on how FDS works. **First column:** Feature statistics similarity for anchor ages {0, 30, 60, 90}, using model trained without FDS. **Second column:** Feature statistics similarity for anchor ages {0, 30, 60, 90}, using model trained with FDS. We show that using FDS, the statistics are better calibrated for all anchor ages, leading to a high similarity only in the neighborhood, and a gradually decreasing similarity score as target value becomes smaller or larger.