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

/* 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 $\{\tilde{\boldsymbol{\mu}}_b^{(0)}, \widetilde{\boldsymbol{\Sigma}}_b^{(0)}\}$
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
            $\mathbf{z}_i = f(\mathbf{x}_i)$
            $\tilde{\mathbf{z}}_i = \left(\widetilde{\boldsymbol{\Sigma}}_b^{(e)}\right)^{\frac{1}{2}} \left(\boldsymbol{\Sigma}_b^{(e)}\right)^{-\frac{1}{2}} (\mathbf{z}_i - \boldsymbol{\mu}_b^{(e)}) + \tilde{\boldsymbol{\mu}}_b^{(e)}$      /* 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 $\{\tilde{\boldsymbol{\mu}}_b^{(e+1)}, \widetilde{\boldsymbol{\Sigma}}_b^{(e+1)}\}_{b \in \mathcal{B}}$ 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	$0 \sim 186$	1	7,149	1	191,509	11,022	11,022
AgeDB-DIR	Age	$0 \sim 101$	1	353	1	12,208	2,140	2,140
STS-B-DIR	Text similarity score	$0 \sim 5$	0.1	428	1	5,249	1,000	1,000
NYUD2-DIR	Depth	$0.7 \sim 10$	0.1	1.46×10^8	1.13×10^6	50,688 (3.51×10^9)	–	654 (8.70×10^5)
SHHS-DIR	Health condition score	$0 \sim 100$	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 224×224. 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 224×224, 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 1.13×10^6 and 1.46×10^8. 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 8.70×10^5 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 640×480 to 320×240 using bilinear interpolation, then conduct center crop to obtain images of size 304×228, 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 114×152 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 $\mathbf{x}_b \in \mathbb{R}^{3 \times 10 l_i}$, 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$\sim$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 2.6×10^7 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 1.0×10^7 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.	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
Vanilla + **LDS**	131.65	109.04	**298.98**	829.35	7.83	7.31	**12.43**	22.51	4.42	4.19	**7.00**	13.94
Vanilla + **FDS**	133.81	107.51	332.90	916.18	7.85	**7.18**	13.35	24.12	4.47	4.18	8.18	15.18
Vanilla + **LDS** + **FDS**	**129.35**	**106.52**	311.49	**811.82**	**7.78**	7.20	12.61	**22.19**	**4.37**	**4.12**	7.39	**12.61**
Mixup (Zhang et al., 2018)	141.11	109.13	389.95	1037.98	8.22	**7.29**	16.23	28.11	4.68	**4.22**	12.28	23.55
M-Mixup (Verma et al., 2019)	137.45	**108.33**	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**	**135.26**	110.91	**326.52**	**808.45**	**7.97**	7.38	**13.22**	**22.95**	**4.59**	4.39	**7.84**	**14.94**
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	7.90	**7.10**	14.72	25.84	**4.47**	**4.09**	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**	**132.58**	**105.33**	**338.65**	**944.92**	**7.88**	**7.10**	**14.08**	**25.75**	**4.47**	4.11	**9.32**	**18.67**
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	**4.02**	8.72	16.92
RRT + **FDS**	129.88	**104.63**	310.69	890.04	**7.65**	**7.02**	12.68	23.85	**4.31**	4.03	7.58	16.28
RRT + **LDS** + **FDS**	**129.14**	105.92	**306.69**	**880.13**	**7.65**	7.06	**12.41**	**23.51**	**4.31**	4.07	**7.17**	**15.44**
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	**298.98**	829.35	7.83	7.31	**12.43**	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	**6.93**	13.48
SQInv + **LDS** + **FDS**	**129.35**	**106.52**	311.49	**811.82**	**7.78**	**7.20**	12.61	**22.19**	**4.37**	**4.12**	7.39	**12.61**
Ours (best) vs. Vanilla	**+8.92**	**+4.07**	**+67.11**	**+156.47**	**+0.41**	**+0.21**	**+2.71**	**+4.14**	**+0.26**	**+0.15**	**+3.66**	**+7.85**

- **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.	Few
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	**204.64**	7.67	6.98	8.86	10.89	4.85	4.39	5.80	7.45
Vanilla + **FDS**	**98.55**	**75.06**	123.58	235.70	**7.55**	**6.50**	8.97	13.01	4.75	**4.03**	6.42	9.93
Vanilla + **LDS** + **FDS**	99.46	84.10	**112.20**	209.27	**7.55**	7.01	**8.24**	**10.79**	**4.72**	4.36	**5.45**	**6.79**
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**	**108.41**	**91.58**	**120.28**	**218.59**	**7.90**	**7.32**	**8.51**	**11.19**	**4.98**	**4.64**	**5.41**	**7.35**
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	7.56	**6.67**	8.82	12.40	4.82	4.27	5.87	8.83
Focal-R + **FDS**	100.14	80.97	121.84	**221.15**	7.65	6.89	8.70	**11.92**	4.83	4.32	5.89	**8.04**
Focal-R + **LDS** + **FDS**	**96.70**	**76.11**	**115.86**	238.25	**7.47**	6.69	**8.30**	12.55	**4.71**	**4.25**	**5.36**	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	**6.95**	8.76	11.86	4.82	**4.32**	5.83	8.08
RRT + **LDS** + **FDS**	**101.74**	**83.12**	**121.08**	**210.78**	**7.66**	6.99	**8.60**	**11.32**	**4.80**	4.42	**5.53**	**6.99**
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	**83.62**	128.73	204.64	7.67	**6.98**	8.86	10.89	4.85	4.39	5.80	7.45
SQInv + **FDS**	101.67	86.49	129.61	**167.75**	7.69	7.10	8.86	**9.98**	4.83	4.41	5.97	**6.29**
SQInv + **LDS** + **FDS**	**99.46**	84.10	**112.20**	209.27	**7.55**	7.01	**8.24**	10.79	**4.72**	**4.36**	**5.45**	6.79
Ours (best) vs. Vanilla	**+4.90**	**+3.34**	**+26.32**	**+85.99**	**+0.30**	**+0.12**	**+1.31**	**+3.69**	**+0.34**	**+0.20**	**+1.65**	**+4.46**

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.	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**
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	**55.6**	74.3
Vanilla + **FDS**	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
Vanilla + **LDS** + **FDS**	**0.907**	**0.802**	1.363	**0.942**	**0.766**	**0.718**	0.986	**0.755**	**76.0**	**74.0**	65.2	**76.6**	**76.4**	**70.7**	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	**55.6**	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	**1.101**	1.153	0.796	0.776	**0.864**	0.838	73.0	69.6	**68.5**	69.9	72.9	66.0	54.3	68.0
SMOGN + **LDS** + **FDS**	**0.950**	**0.851**	1.327	**1.095**	**0.785**	**0.738**	0.987	**0.799**	**74.6**	**72.1**	65.9	**71.7**	**75.0**	**68.9**	54.4	**70.3**
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	**0.807**	1.449	0.993	0.781	**0.723**	1.031	0.801	**75.7**	**73.9**	62.4	75.4	**76.2**	**71.2**	50.7	74.7
Focal-R + **FDS**	**0.920**	0.855	**1.169**	1.008	**0.775**	0.743	**0.903**	0.804	75.1	72.6	**66.4**	74.7	75.4	69.4	**52.7**	75.4
Focal-R + **LDS** + **FDS**	0.940	0.849	1.358	**0.916**	0.785	0.737	0.984	**0.732**	74.9	72.2	66.3	**77.3**	75.1	69.2	52.5	**76.4**
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	**74.7**
RRT + **LDS**	0.916	0.817	1.344	0.945	0.772	0.727	0.980	**0.756**	75.7	73.5	64.1	76.6	76.1	70.4	53.2	74.2
RRT + **FDS**	0.929	0.857	**1.209**	1.025	0.769	0.736	**0.905**	0.795	74.9	72.1	**67.2**	74.0	75.0	69.1	52.8	74.6
RRT + **LDS** + **FDS**	**0.903**	**0.806**	1.323	**0.936**	**0.764**	**0.719**	0.965	0.760	**76.0**	**73.8**	65.2	**76.7**	**76.4**	**70.8**	**54.7**	**74.7**
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	**55.6**	74.3
Inv + **FDS**	0.927	0.851	**1.225**	1.012	0.771	0.740	**0.914**	0.756	75.0	72.4	**66.6**	74.2	75.2	69.2	55.2	74.8
Inv + **LDS** + **FDS**	**0.907**	**0.802**	1.363	**0.942**	**0.766**	**0.718**	0.986	**0.755**	**76.0**	**74.0**	65.2	**76.6**	**76.4**	**70.7**	54.9	**74.9**
Ours (best) vs. Vanilla	**+.071**	**+.049**	**+.419**	**+.068**	**+.030**	**+.022**	**+.203**	**+.039**	**+1.8**	**+2.0**	**+5.8**	**+2.1**	**+2.0**	**+2.4**	**+5.1**	**+1.4**

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 ↓				$\log_{10}\downarrow$				$\delta_1\uparrow$				$\delta_2\uparrow$				$\delta_3\uparrow$			
Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
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	**0.615**	0.940	2.059	0.084	**0.069**	0.080	0.101	0.681	**0.760**	0.695	0.596	0.903	**0.952**	0.918	0.849	0.975	**0.989**	0.976	0.960
VANILLA + **LDS** + **FDS**	**1.338**	0.670	**0.851**	**1.880**	**0.080**	0.074	**0.070**	**0.090**	**0.705**	0.730	**0.764**	**0.655**	**0.916**	0.939	**0.941**	**0.884**	**0.979**	0.984	**0.983**	**0.971**
OURS (BEST) VS. VANILLA	**+.139**	**-.024**	**+.101**	**+.243**	**+.006**	**-.003**	**+.012**	**+.017**	**+.028**	**-.017**	**+.071**	**+.085**	**+.017**	**-.004**	**+.035**	**+.044**	**+.010**	**-.001**	**+.008**	**+.025**

Table 12. Complete evaluation results on SHHS-DIR.

Metrics	MSE ↓				MAE ↓				GM ↓			
Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
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	**6.93**	8.05	11.19
VANILLA + **LDS** + **FDS**	**292.18**	**211.89**	**247.48**	**346.01**	**13.76**	**11.12**	**12.18**	**15.07**	**8.70**	6.94	**7.60**	**10.18**
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	**185.16**	303.90	391.22	14.18	**11.06**	13.56	15.99	9.45	**6.95**	8.81	11.13
FOCAL-R + **LDS** + **FDS**	**297.85**	193.42	**259.33**	**375.16**	**14.02**	11.08	**12.24**	**15.49**	**9.32**	7.18	**8.10**	**10.39**
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**	**313.58**	**238.07**	**276.50**	**380.64**	**14.33**	**11.96**	**12.47**	**15.92**	**9.63**	**7.35**	**8.74**	**11.17**
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	**11.12**	12.29	15.53	8.94	**6.91**	7.79	10.65
INV + **LDS** + **FDS**	**292.18**	**211.89**	**247.48**	**346.01**	**13.76**	**11.12**	**12.18**	**15.07**	**8.70**	6.94	**7.60**	**10.18**
OURS (BEST) VS. VANILLA	**+77.00**	**+84.21**	**+63.97**	**+71.30**	**+1.60**	**+1.41**	**+1.80**	**+1.87**	**+1.93**	**+1.13**	**+1.99**	**+2.02**

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 $\log_{10}$, δ_1, δ_2, and δ_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 & 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
LDS:												
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
FDS:												
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 & 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
LDS:																
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
FDS:																
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 & 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
LDS:																
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
FDS:																
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 σ. Specifically, we conduct controlled experiments on IMDB-WIKI-DIR and STS-B-DIR, where we vary the choices of these hyper-parameters as $l \in \{\texttt{5}, \texttt{9}, \texttt{15}\}$ and $\sigma \in \{\texttt{1}, \texttt{2}, \texttt{3}\}$, and leave other training hyper-parameters unchanged.

Table 16. Hyper-parameter study on kernel size l and standard deviation σ 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	σ												
LDS:													
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
FDS:													
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 σ for LDS & FDS on STS-B-DIR.

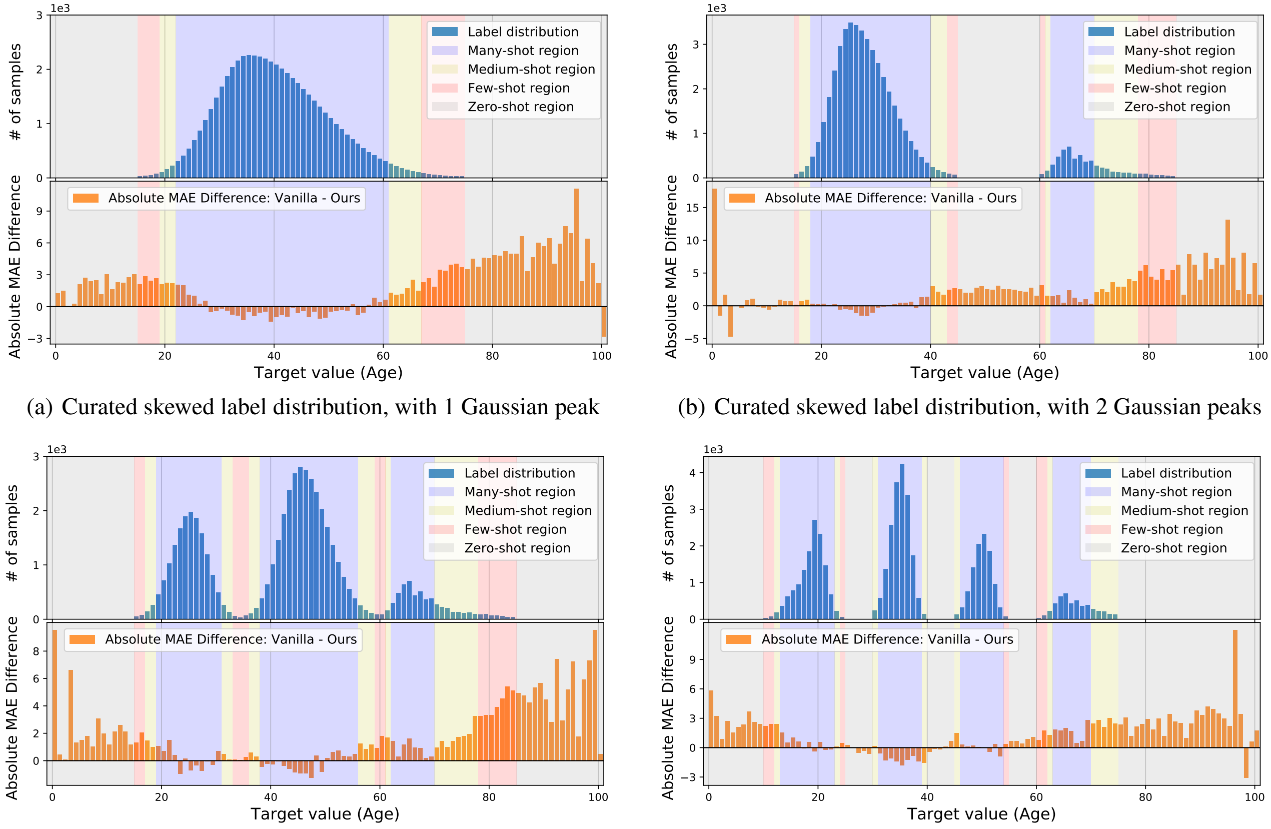
Metrics		MSE ↓				MAE ↓				Pearson correlation (%) ↑				Spearman correlation (%) ↑			
Shot		All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA		0.974	0.851	1.520	0.984	0.794	0.740	1.043	0.771	74.2	72.0	62.7	75.2	74.4	68.8	50.5	75.0
l	σ																
LDS:																	
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
FDS:																	
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 σ 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 $\{\texttt{1,2,3,4}\}$ 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.	Extrap.
1 peak:														
Vanilla	11.20	6.05	11.43	14.76	22.67	–	22.67	7.02	**3.84**	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	**5.97**	11.19	14.54	22.35	–	22.35	6.96	**3.84**	8.54	12.08	20.71	–	20.71
Vanilla + **LDS** + **FDS**	**10.00**	6.28	**9.66**	**11.83**	**19.21**	–	**19.21**	**6.09**	3.96	**6.26**	**8.14**	**15.89**	–	**15.89**
2 peaks:														
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	**5.65**	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	**3.50**	7.49	12.73	14.86	14.02	16.48
Vanilla + **LDS** + **FDS**	**10.27**	**6.61**	**9.46**	**11.96**	**14.89**	**13.71**	**17.02**	**6.33**	3.54	5.68	**8.80**	**12.83**	**11.71**	**15.13**
3 peaks:														
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	**6.77**	8.82	10.48	18.43	–	18.43	**5.35**	**3.78**	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**	**8.96**	6.88	**8.62**	**10.08**	**17.76**	–	**17.76**	5.38	3.90	**5.61**	**7.36**	**14.65**	–	**14.65**
4 peaks:														
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	**6.98**	8.26	10.07	11.26	8.31	**16.22**	5.10	**3.33**	**5.07**	7.08	8.47	6.66	**12.74**
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**	**8.76**	7.07	**8.23**	**9.54**	**11.13**	**8.05**	16.32	**5.05**	3.36	**5.07**	**6.56**	**8.30**	**6.34**	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.	Few
Balanced:												
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**	**129.35**	**106.52**	**311.49**	**811.82**	**7.78**	**7.20**	**12.61**	**22.19**	**4.37**	**4.12**	**7.39**	**12.61**
Same as training set:												
Vanilla	**68.44**	**62.10**	320.52	1350.01	**5.84**	**5.72**	15.11	30.54	**3.44**	**3.40**	11.76	24.06
Vanilla + **LDS** + **FDS**	69.86	63.43	**161.97**	**1067.89**	5.90	5.77	**9.94**	**25.17**	3.48	3.44	**7.03**	**15.95**

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.

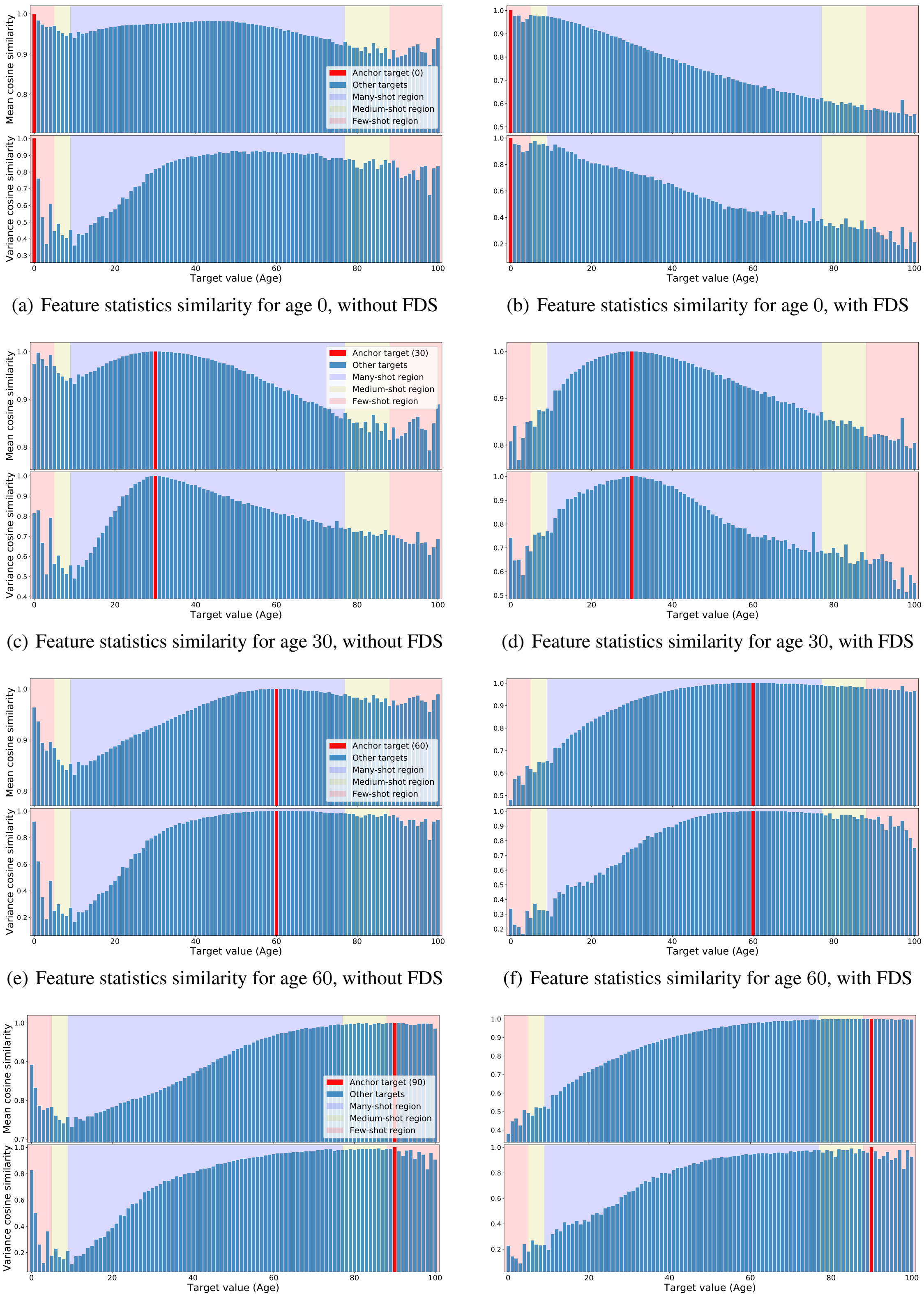
We note that imbalanced classification methods could fail on regression problems for several reasons. First, they ignore the similarity between data samples that are close w.r.t. the continuous target; Treating different target values as distinct classes is unlikely to yield the best results because 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 ↓				MSE ↓				RMSE ↓			
Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
Imbalanced Classification:												
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
Imbalanced Regression:												
REG-VANILLA	14.64	13.98	17.47	30.29	0.974	0.851	1.520	0.984	1.477	**0.591**	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	**1.027**	1.086	1.442	0.615	0.940	2.059
LDS + FDS	**13.32**	**13.14**	**15.06**	**23.87**	**0.907**	**0.802**	1.363	**0.942**	**1.338**	0.670	**0.851**	**1.880**

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.



(a) Feature statistics similarity for age 0, without FDS

(b) Feature statistics similarity for age 0, with FDS

(c) Feature statistics similarity for age 30, without FDS

(d) Feature statistics similarity for age 30, with FDS

(e) Feature statistics similarity for age 60, without FDS

(f) Feature statistics similarity for age 60, with FDS

(g) Feature statistics similarity for age 90, without FDS

(h) Feature statistics similarity for age 90, with FDS

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.