Deep Imbalanced Regression

Yuzhe Yang[1] Kaiwen Zha[1] Ying-Cong Chen[1] Hao Wang[2]
Dina Katabi[1]

[1]MIT Computer Science & Artificial Intelligence Laboratory
[2]Department of Computer Science, Rutgers University

ICML 2021

Presenter: Gianmarco Midena

26 November 2024

Overview

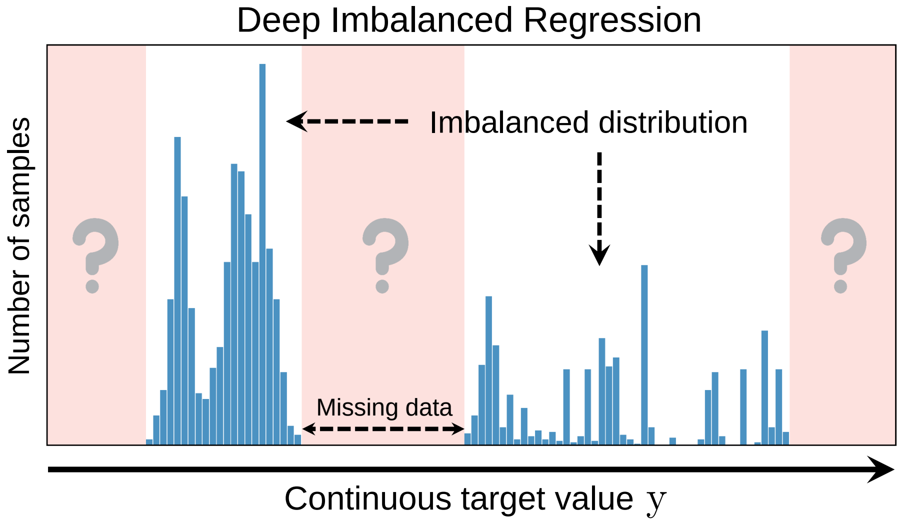


Image credit: *Yang et al. (2021)*

Problem Settings

- $\mathcal{Y} \subset \mathbb{R}$: continuous label space

Problem Settings

- $\mathcal{Y} \subset \mathbb{R}$: continuous label space
- $\mathcal{B} = \{1, \ldots, M\} \subset \mathbb{Z}^+$: index space
  - divides $\mathcal{Y}$ into M groups (bins) with equal intervals $[y_j, y_{j+1})$
  - $\{[y_0, y_1), \ldots, [y_{M-1}, y_M)\}$: discrete label space
  - $y_k \in \mathcal{Y}$
  - minimum resolution
    - e.g., $\delta y \triangleq y_{j+1} - y_j = 1$ in age estimation

Evaluation

Evaluation

- Divide target space into disjoint regions (bins)
  - *Many-shot*: > 100 training examples
  - *Medium-shot*: 20-100 training examples
  - *Few-shot*: < 20 training examples
  - *Zero-shot*: 0 training examples
  - Inspired by Liu et al. 2019

Evaluation

- Divide target space into disjoint regions (bins)
  - *Many-shot*: > 100 training examples
  - *Medium-shot*: 20-100 training examples
  - *Few-shot*: < 20 training examples
  - *Zero-shot*: 0 training examples
  - Inspired by Liu et al. 2019

- (main) Metrics
  - Mean Absolute Error (MAE)
  - Mean Squared Error (MSE)
  - Pearson Correlation (PCC)
  - Geometric Mean Error (GM)

$$GM = \sqrt[n]{\prod_{i=1}^{n} |y_i - \hat{y}_i|}$$

★ Pros: $+$ fairness (uniformity) in prediction

Datasets - Overview

Dataset	Input type	Target type	Target range	Bin size	Max bin density	Min bin density
IMDB-WIKI-DIR	Face image	Age	0 - 186	1	7,149	1
AgeDB-DIR	Face image	Age	0 - 101	1	353	1
STS-B-DIR	Sentence pairs	Text similarity score	0 - 5	0.1	428	1
NYUD2-DIR	Indoor scene image	Depth map	0.7 - 10 [m]	0.1	1.46×10^8	1.13×10^6
SHHS-DIR	PSG signals	Health condition score	0 - 100	1	275	0

Dataset	# Training set	# Val. set	# Test set
IMDB-WIKI-DIR	191,509	11,022	11,022
AgeDB-DIR	12,208	2,140	2,140
STS-B-DIR	5,249	1,000	1,000
NYUD2-DIR	50,688 (3.51×10^9)	–	654 (8.70×10^5)
SHHS-DIR	1,892	369	369

Table credit: *Yang et al. (2021)*

Datasets - Overview

Dataset	Input type	Target type	Target range	Bin size	Max bin density	Min bin density
IMDB-WIKI-DIR	Face image	Age	0 - 186	1	7,149	1
AgeDB-DIR	Face image	Age	0 - 101	1	353	1
STS-B-DIR	Sentence pairs	Text similarity score	0 - 5	0.1	428	1
NYUD2-DIR	Indoor scene image	Depth map	0.7 - 10 [m]	0.1	1.46×10^8	1.13×10^6
SHHS-DIR	PSG signals	Health condition score	0 - 100	1	275	0

Dataset	# Training set	# Val. set	# Test set
IMDB-WIKI-DIR	191,509	11,022	11,022
AgeDB-DIR	12,208	2,140	2,140
STS-B-DIR	5,249	1,000	1,000
NYUD2-DIR	50,688 (3.51×10^9)	–	654 (8.70×10^5)
SHHS-DIR	1,892	369	369

Table credit: *Yang et al. (2021)*

- Full-night Polysomnography (PSG) signals: Electroencephalography (EEG), Electrocardiography (ECG), and breathing signals (airflow, abdomen, and thorax).
- An indoor scene image (left) and its depth map (right):

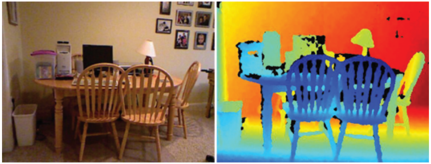
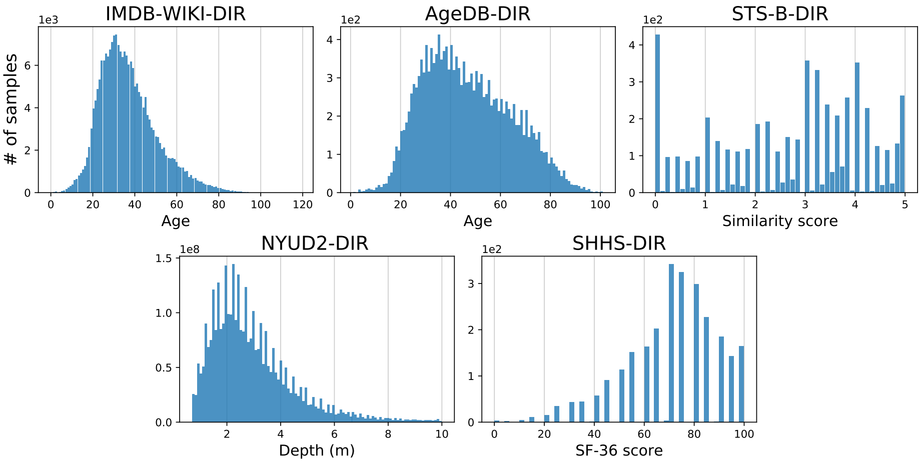


Image credit: *https://cs.nyu.edu/~fergus/datasets/nyu_depth_v2.html*

Label Distributions

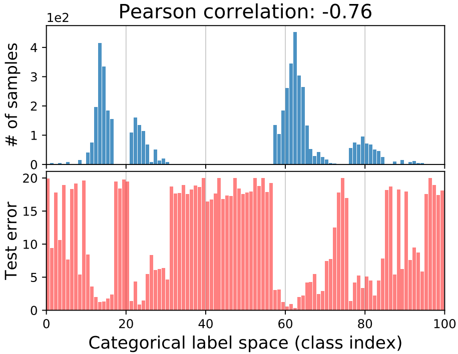


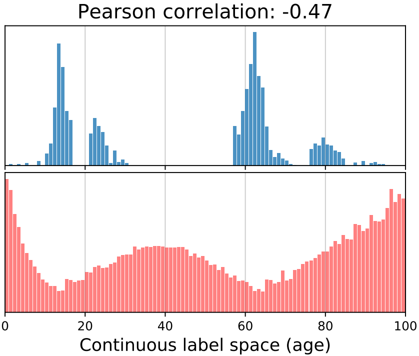
- Typically for training only

Image credit: *Yang et al. (2021)*

Label Distribution Smoothing (LDS)

Imbalanced Categorical vs. Continuous Label Space (1/3)





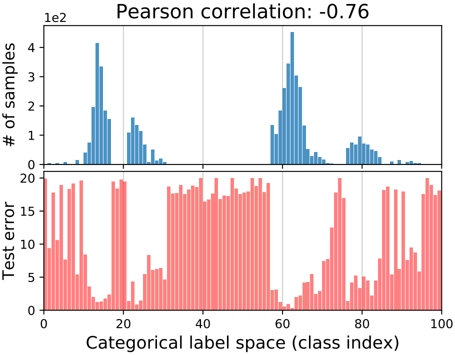
- task: picture ⟶ class
- data souce: CIFAR-100

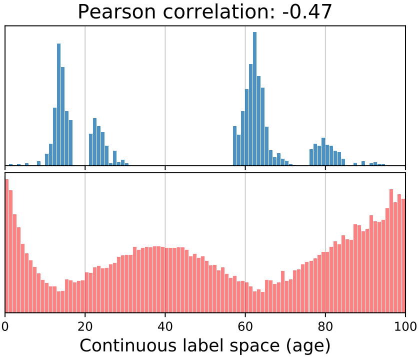
- task:
  person's picture ⟶ person's age
- age subrange: 0-99
- data souce: IMDB-WIKI

- Simulated label imbalance
- Label density distributions forced to be equal
- Balanced test sets

Image credit: *Yang et al. (2021)*

Imbalanced Categorical vs. Continuous Label Space (2/3)


Pearson correlation: -0.76

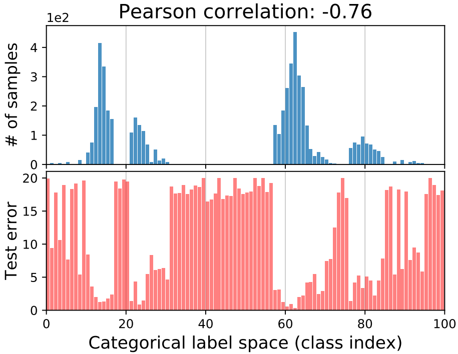

Pearson correlation: -0.47

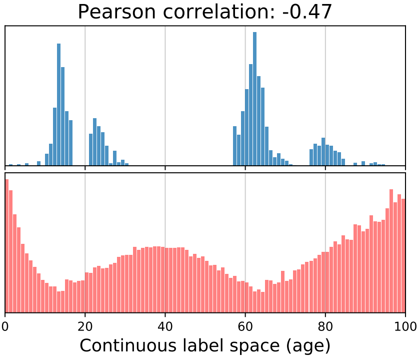
- error distribution *correlates* with label density distribution

- error distribution DOES NOT *correlate* well with label density distribution

Image credit: *Yang et al. (2021)*

Imbalanced Categorical vs. Continuous Label Space (2/3)



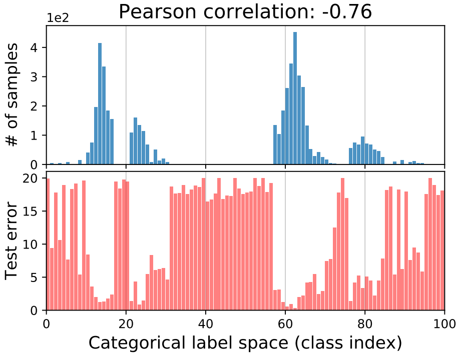


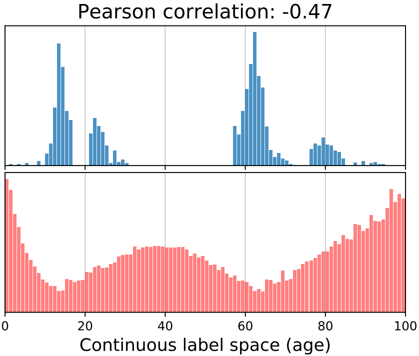
- error distribution *correlates* with label density distribution
- majority classes with more examples are better learned than minority classes

- error distribution DOES NOT *correlate* well with label density distribution
- smoother error distribution

Image credit: *Yang et al. (2021)*

Imbalanced Categorical vs. Continuous Label Space (3/3)



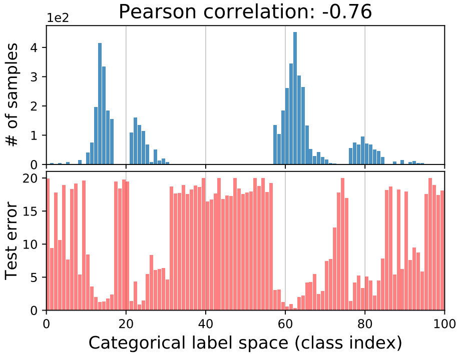


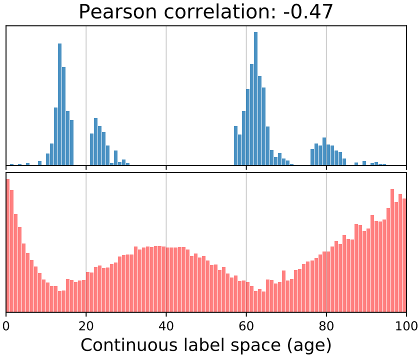
- Compensating for imbalance in empirical label density distribution WORKS WELL.

- Compensating for imbalance in empirical label density distribution is INACCURATE.
- Empirical density does not accurately reflect imbalance as seen by model.

Image credit: *Yang et al. (2021)*

Imbalanced Categorical vs. Continuous Label Space (3/3)





- Compensating for imbalance in empirical label density distribution WORKS WELL.

- Compensating for imbalance in empirical label density distribution is INACCURATE.
- Empirical density does not accurately reflect imbalance as seen by model.
- Intuition: dependence between features at nearby labels.
- Proposed solution: **Label Distribution Smoothing (LDS)**

Image credit: *Yang et al. (2021)*

Label Distribution Smoothing (LDS) - Overview

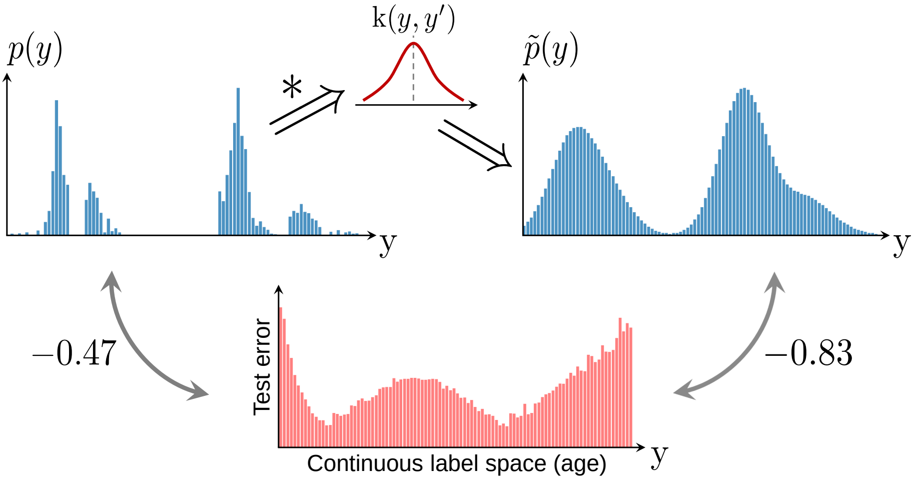


Image credit: *Yang et al. (2021)*

Label Distribution Smoothing (LDS)

- Starting points
  - Dependence between features at nearby continuous labels.
  - Expected density estimation
    - ★ Significant literature in statistics (Parzen 1962)
    - ★ Kernel density estimation

Label Distribution Smoothing (LDS)

- Starting points
  - Dependence between features at nearby continuous labels.
  - Expected density estimation
    - ★ Significant literature in statistics (Parzen 1962)
    - ★ Kernel density estimation
- Functioning
  - Convolves symmetric kernel with empirical label density distribution.
  - Extracts kernel-smoothed label density accounting for feature overlap of neighbouring labels.

Label Distribution Smoothing (LDS)

- Starting points
  - Dependence between features at nearby continuous labels.
  - Expected density estimation
    - Significant literature in statistics (Parzen 1962)
    - Kernel density estimation
- Functioning
  - Convolves symmetric kernel with empirical label density distribution.
  - Extracts kernel-smoothed label density accounting for feature overlap of neighbouring labels.
- Symmetric kernel
  - E.g., Gaussian, Laplacian, triangular kernel.
  - Similarity between target values w.r.t. their distance in target space.

Label Distribution Smoothing (LDS)

- Starting points
  - Dependence between features at nearby continuous labels.
  - Expected density estimation
    - Significant literature in statistics (Parzen 1962)
    - Kernel density estimation
- Functioning
  - Convolves symmetric kernel with empirical label density distribution.
  - Extracts kernel-smoothed label density accounting for feature overlap of neighbouring labels.
- Symmetric kernel
  - E.g., Gaussian, Laplacian, triangular kernel.
  - Similarity between target values w.r.t. their distance in target space.
- *Effective label density distribution*

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

where

- $p(y)$: nr. occurrences of label y in training data.

Label Distribution Smoothing (LDS)

- Starting points
  - Dependence between features at nearby continuous labels.
  - Expected density estimation
    - Significant literature in statistics (Parzen 1962)
    - Kernel density estimation
- Functioning
  - Convolves symmetric kernel with empirical label density distribution.
  - Extracts kernel-smoothed label density accounting for feature overlap of neighbouring labels.
- Symmetric kernel
  - E.g., Gaussian, Laplacian, triangular kernel.
  - Similarity between target values w.r.t. their distance in target space.
- *Effective label density distribution*

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

where

  - $p(y)$: nr. occurrences of label y in training data.
- Usage
  - Possible direct adaptation of class imbalance techniques.
  - E.g., loss weighted by inverse effective label density.

Feature Distribution Smoothing (FDS)

Feature Distribution Smoothing (FDS) - Preliminaries

- Starting points
  1. Continuity in **target** space $\longleftrightarrow$ Continuity in **feature** space
  2. Data balance $\Longrightarrow$ close feature statistics of nearby targets

Feature Distribution Smoothing (FDS) - Preliminaries

- Starting points
  1. Continuity in **target** space $\longleftrightarrow$ Continuity in **feature** space
  2. Data balance $\Longrightarrow$ close feature statistics of nearby targets

- Feature statistics: mean and variance (or covariance) w.r.t. each bin

$$\{\boldsymbol{\mu}_b, \boldsymbol{\sigma}_b\}_{b=1}^{B}$$

Feature Distribution Smoothing (FDS) - Preliminaries

- Starting points
  1. Continuity in **target** space $\longleftrightarrow$ Continuity in **feature** space
  2. Data balance $\Longrightarrow$ close feature statistics of nearby targets

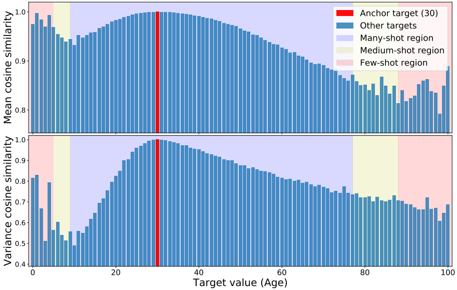
- Feature statistics: mean and variance (or covariance) w.r.t. each bin

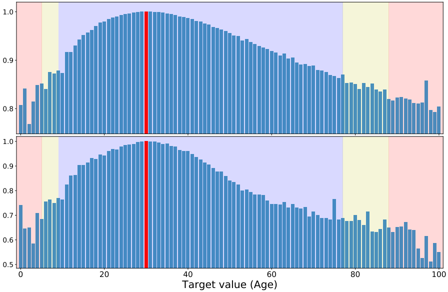
$$\{\boldsymbol{\mu}_b, \boldsymbol{\sigma}_b\}_{b=1}^{B}$$

- (next slides) Feature statistics similarity: cosine similarity of feature statistics between one anchor bin b_0 and all other bins
  - $b_0 = 0, 30, 60, 90$ (age): chosen anchor bins
  - different target densities:
    many ($>$100), medium (20-100), few ($<$20) examples
  - task: person's picture $\longrightarrow$ person's age
  - data source: IMDB-WIKI

Feature statistics similarity (1/4)

Anchor age 30





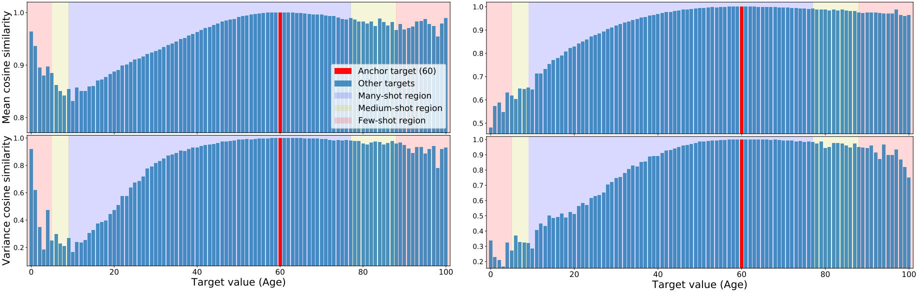
- High similarity in neighbourhood
- High similarities with further regions
- Lower similarities with some closer regions

- Improved feature statistics calibration:
  - High similarity only in neighbourhood
  - “The further the region the lower the similarity”
  - More gradual similarity change

Image credit: *Yang et al. (2021)*

Feature statistics similarity (2/4)

Anchor age 60



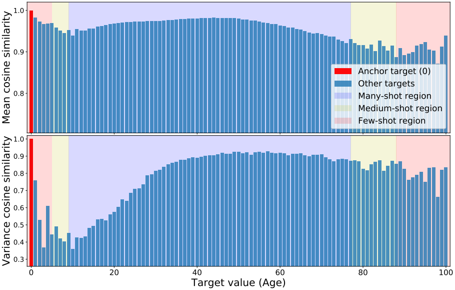
- High similarity in neighbourhood
- High similarities with further regions
- Lower similarities with some closer regions

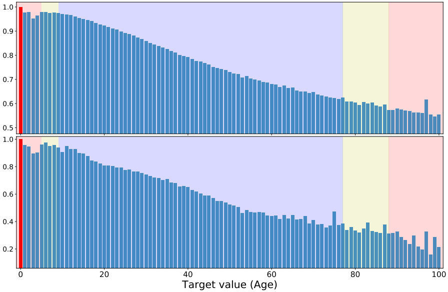
- Improved feature statistics calibration:
  - High similarity only in neighbourhood
  - “The further the region the lower the similarity”
  - More gradual similarity change

Image credit: *Yang et al. (2021)*

Feature statistics similarity (3/4)

Anchor age 0





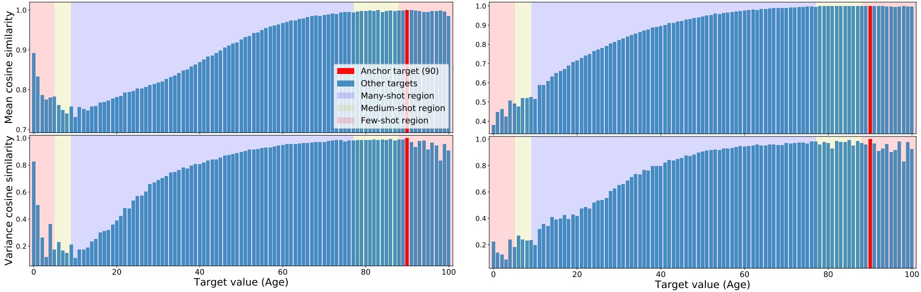
- High similarity in neighbourhood for mean
- High similarities with further regions
- Lower similarities with some closer regions, e.g., variance neighbourhood

- Improved feature statistics calibration:
  - High similarity only in neighbourhood
  - "The further the region the lower the similarity"
  - More gradual similarity change

Image credit: *Yang et al. (2021)*

Feature statistics similarity (4/4)

Anchor age 90

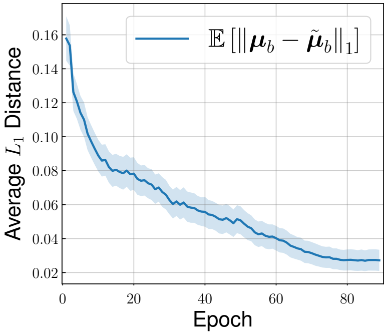


- High similarity in neighbourhood, esp. for mean
- High similarities with further regions
- Lower similarities with some closer regions

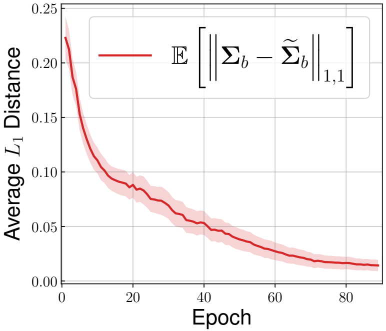
- Improved feature statistics calibration:
  - High similarity only in neighbourhood
  - "The further the region the lower the similarity"
  - More gradual similarity change

Image credit: *Yang et al. (2021)*

Change of feature statistics w.r.t. epoch



(a) Mean

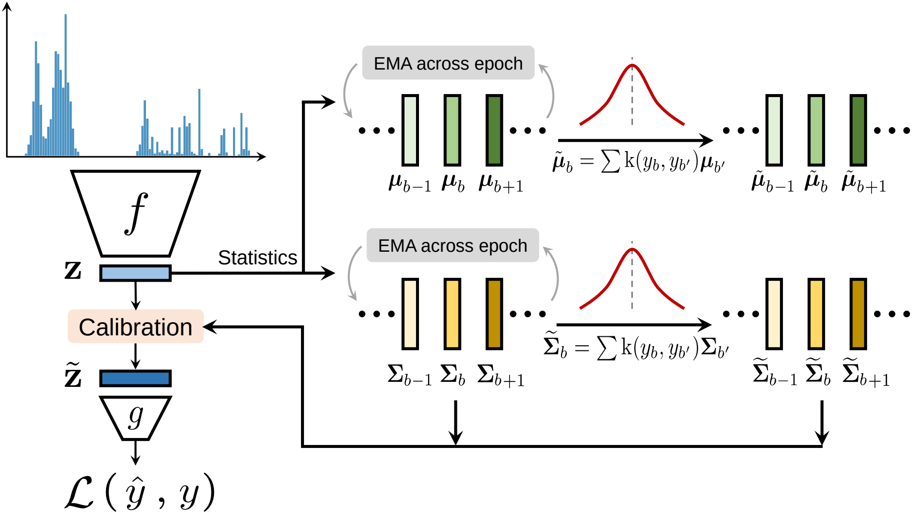


(b) Covariance

- $\boldsymbol{\mu}, \boldsymbol{\Sigma}$: Running mean and covariance
- $\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}$: Smoothed mean and covariance

Image credit: *Yang et al. (2021)*

Feature Distribution Smoothing (FDS) - Overview



f
$\mathbf{z}$
Statistics
Calibration
$\tilde{\mathbf{z}}$
g
$\mathcal{L}(\hat{y}, y)$
EMA across epoch
$\boldsymbol{\mu}_{b-1}$ $\boldsymbol{\mu}_b$ $\boldsymbol{\mu}_{b+1}$
$\tilde{\boldsymbol{\mu}}_b = \sum \mathrm{k}(y_b, y_{b'})\boldsymbol{\mu}_{b'}$
$\tilde{\boldsymbol{\mu}}_{b-1}$ $\tilde{\boldsymbol{\mu}}_b$ $\tilde{\boldsymbol{\mu}}_{b+1}$
EMA across epoch
$\boldsymbol{\Sigma}_{b-1}$ $\boldsymbol{\Sigma}_b$ $\boldsymbol{\Sigma}_{b+1}$
$\widetilde{\boldsymbol{\Sigma}}_b = \sum \mathrm{k}(y_b, y_{b'})\boldsymbol{\Sigma}_{b'}$
$\widetilde{\boldsymbol{\Sigma}}_{b-1}$ $\widetilde{\boldsymbol{\Sigma}}_b$ $\widetilde{\boldsymbol{\Sigma}}_{b+1}$

Image credit: *Yang et al. (2021)*

Feature Distribution Smoothing (FDS)

- Transfers feature statistics between nearby bins.
- Aims to calibrate potentially biased estimates of feature distribution, esp. for underrepresented target values in training data.

Feature Distribution Smoothing (FDS)

- Transfers feature statistics between nearby bins.
- Aims to calibrate potentially biased estimates of feature distribution, esp. for underrepresented target values in training data.
- General functioning
  - Estimates mean $\boldsymbol{\mu}_b$ and covariance $\boldsymbol{\Sigma}_b$ feature statistics by each target bin.
  - Smooths feature statistics over target bins $\mathcal{B}$ by symmetric kernel $k(y_b, y_b')$. Obtains smoothed mean $\tilde{\boldsymbol{\mu}}_b$ and covariance $\tilde{\boldsymbol{\Sigma}}_b$ feature statistics.
  - Whitens features (Sun et al. 2016): $\boldsymbol{z}^w = \boldsymbol{\Sigma}_b^{-\frac{1}{2}}(\boldsymbol{z} - \boldsymbol{\mu}_b)$
  - Re-colors whitened features (Sun et al. 2016): $\boldsymbol{z}^r = \tilde{\boldsymbol{\Sigma}}_b^{\frac{1}{2}}\boldsymbol{z}^w + \tilde{\boldsymbol{\mu}}_b$

Feature Distribution Smoothing (FDS)

- Transfers feature statistics between nearby bins.

- Aims to calibrate potentially biased estimates of feature distribution, esp. for underrepresented target values in training data.

- General functioning
  - Estimates mean $\boldsymbol{\mu}_b$ and covariance $\boldsymbol{\Sigma}_b$ feature statistics by each target bin.
  - Smooths feature statistics over target bins $\mathcal{B}$ by symmetric kernel $k(y_b, y_b')$. Obtains smoothed mean $\tilde{\boldsymbol{\mu}}_b$ and covariance $\tilde{\boldsymbol{\Sigma}}_b$ feature statistics.
  - Whitens features (Sun et al. 2016): $\boldsymbol{z}^w = \boldsymbol{\Sigma}_b^{-\frac{1}{2}}(\boldsymbol{z} - \boldsymbol{\mu}_b)$
  - Re-colors whitened features (Sun et al. 2016): $\boldsymbol{z}^r = \tilde{\boldsymbol{\Sigma}}_b^{\frac{1}{2}}\boldsymbol{z}^w + \tilde{\boldsymbol{\mu}}_b$

- Integration into deep learning
  - Feature calibration layer after final feature map.
  - Momentum update running statistics $\{\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\}$ across each epoch.
    - ★ Exponential Moving Average (EMA)
  - Smoothed statistics $\{\tilde{\boldsymbol{\Sigma}}_b, \tilde{\boldsymbol{\mu}}_b\}$ updated across different but fixed within each training epoch.

Experimental settings

Baselines (1/2)

1. Vanilla: neglects data imbalance

Baselines (1/2)

1. Vanilla: neglects data imbalance
   - + LDS: loss re-weighting by inverse density

Baselines (1/2)

1. Vanilla: neglects data imbalance
   - + LDS: loss re-weighting by inverse density
2. Synthetic samples
   - SMOTER (Torgo et al. 2013)
     1. Defines frequent and rare regions using label density.
     2. Creates synthetic samples for pre-defined rare regions by linearly interpolating both inputs and labels.
   - SMOGN (Branco et al. 2017): augments SMOTER with Gaussian noise

Baselines (1/2)

1. Vanilla: neglects data imbalance

   + LDS: loss re-weighting by inverse density

2. Synthetic samples
   - SMOTER (Torgo et al. 2013)
     1. Defines frequent and rare regions using label density.

        + LDS: density estimation
     2. Creates synthetic samples for pre-defined rare regions by linearly interpolating both inputs and labels.
   - SMOGN (Branco et al. 2017): augments SMOTER with Gaussian noise

Baselines (1/2)

1. Vanilla: neglects data imbalance
   - + LDS: loss re-weighting by inverse density
2. Synthetic samples
   - SMOTER (Torgo et al. 2013)
     1. Defines frequent and rare regions using label density.
        - + LDS: density estimation
     2. Creates synthetic samples for pre-defined rare regions by linearly interpolating both inputs and labels.
   - SMOGN (Branco et al. 2017): augments SMOTER with Gaussian noise
3. Focal-R

$$\frac{1}{n}\sum_{i=1}^{n}\sigma(|\beta e_i|)^{\gamma} e_i$$

   - Error-aware loss
   - Maps absolute error into $[0, 1]$.
   - e_i: L_1 error for i-th sample
   - β, γ: hyper-parameters
   - Inspired by Focal Loss (Lin 2017) for classification

Baselines (1/2)

1. Vanilla: neglects data imbalance
    + LDS: loss re-weighting by inverse density

2. Synthetic samples
    - SMOTER (Torgo et al. 2013)
        1. Defines frequent and rare regions using label density.
        + LDS: density estimation
        2. Creates synthetic samples for pre-defined rare regions by linearly interpolating both inputs and labels.
    - SMOGN (Branco et al. 2017): augments SMOTER with Gaussian noise

3. Focal-R

$$\frac{1}{n}\sum_{i=1}^{n}\sigma(|\beta e_i|)^{\gamma}e_i$$

    - Error-aware loss
    - Maps absolute error into $[0, 1]$.
    - e_i: L_1 error for i-th sample
    - β, γ: hyper-parameters
    - Inspired by Focal Loss (Lin 2017) for classification
    + LDS: loss re-weighting by inverse density

Baselines (2/2)

4. Regressor re-training (RRT)
    - Two-stage training
        1. Train encoder
        2. Re-train regressor with inverse re-weighting and frozen encoder.
    - Inspired by Kang et al. 2019

Baselines (2/2)

4. Regressor re-training (RRT)
   - Two-stage training
     1. Train encoder
     2. Re-train regressor with inverse re-weighting and frozen encoder.

     + LDS: loss re-weighting by inverse density
   - Inspired by Kang et al. 2019

Baselines (2/2)

4. Regressor re-training (RRT)
    - Two-stage training
        1. Train encoder
        2. Re-train regressor with inverse re-weighting and frozen encoder.
        + LDS: loss re-weighting by inverse density
    - Inspired by Kang et al. 2019

5. Cost-sensitive re-weighting: re-weighting schemes based on label distribution
    - Inverse-frequency weighting (INV)
    - Square-root weighting variant (SQINV)

Baselines (2/2)

4. Regressor re-training (RRT)
    - Two-stage training
        1. Train encoder
        2. Re-train regressor with inverse re-weighting and frozen encoder.

        + LDS: loss re-weighting by inverse density
    - Inspired by Kang et al. 2019

5. Cost-sensitive re-weighting: re-weighting schemes based on label distribution
    - Inverse-frequency weighting (INV)
    - Square-root weighting variant (SQINV)

    + LDS: density estimation

Network Architectures

Architecture	Data
ResNet-50 (He et al. 2016)	IMDB-WIKI, AgeDB
BiLSTM + GloVe word embeddings (A. Wang et al. 2018)	STS-B
ResNet-50-based Encoder-Decoder (Hu et al. 2019)	NYUD2
CNN-RNN + ResNet (H. Wang et al. 2019)	SHHS

Results

Inferring Age from Images

IMDB-WIKI

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	**4.30**	9.05	19.46
SMOGN (Branco et al. 2017)	8.03	**7.30**	14.02	25.93	4.63	**4.30**	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**	**7.97**	7.38	**13.22**	**22.95**	**4.59**	4.39	**7.84**	**14.94**
Focal-R	7.97	7.12	15.14	26.96	4.49	4.10	10.37	21.20
Focal-R + **LDS**	7.90	**7.10**	14.72	25.84	**4.47**	**4.09**	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**	**7.88**	**7.10**	**14.08**	**25.75**	**4.47**	4.11	**9.32**	**18.67**
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	**4.02**	8.72	16.92
RRT + **FDS**	**7.65**	**7.02**	12.68	23.85	**4.31**	4.03	7.58	16.28
RRT + **LDS** + **FDS**	**7.65**	7.06	**12.41**	**23.51**	**4.31**	4.07	**7.17**	**15.44**
SQInv	7.87	7.24	12.44	22.76	4.47	4.22	7.25	15.10
SQInv + **LDS**	7.83	7.31	**12.43**	22.51	4.42	4.19	7.00	13.94
SQInv + **FDS**	7.83	7.23	12.60	22.37	4.42	4.20	**6.93**	13.48
SQInv + **LDS** + **FDS**	**7.78**	**7.20**	12.61	**22.19**	**4.37**	**4.12**	7.39	**12.61**
Ours (best) vs. Vanilla	**+0.41**	**+0.21**	**+2.71**	**+4.14**	**+0.26**	**+0.15**	**+3.66**	**+7.85**

- Either LDS, FDS, or both leads to performance gains.
- LDS + FDS often achieves best results:
  - maintains or improves performance overall and on many-shot regions,
  - boosts performance for medium-shot and few-shot regions.

Table credit: *Yang et al. (2021)*

Inferring Age from Images

AgeDB

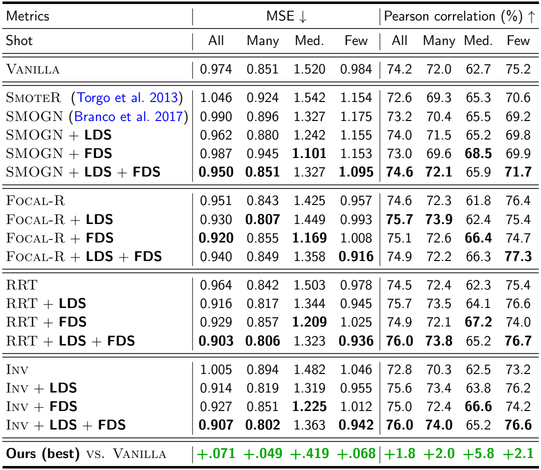
Metrics	MAE ↓				GM ↓			
Shot	All	Many	Med.	Few	All	Many	Med.	Few
Vanilla	7.77	6.62	9.55	13.67	5.05	4.23	7.01	10.75
SmoteR (Torgo et al. 2013)	8.16	7.39	8.65	12.28	5.21	4.65	5.69	8.49
SMOGN (Branco et al. 2017)	8.26	7.64	9.01	12.09	5.36	4.90	6.19	8.44
SMOGN + **LDS**	7.96	7.44	8.64	11.77	5.03	4.68	5.69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

- Either LDS, FDS, or both leads to performance gains.
- LDS + FDS often achieves best results:
  - maintains or improves performance overall and on many-shot regions,
  - boosts performance for medium-shot and few-shot regions.

Table credit: *Yang et al. (2021)*

Inferring Text Similarity Score

STS-B



Metrics	MSE ↓				Pearson correlation (%) ↑			
Shot	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	**1.101**	1.153	73.0	69.6	**68.5**	69.9
SMOGN + **LDS** + **FDS**	**0.950**	**0.851**	1.327	**1.095**	**74.6**	**72.1**	65.9	**71.7**
Focal-R	0.951	0.843	1.425	0.957	74.6	72.3	61.8	76.4
Focal-R + **LDS**	0.930	**0.807**	1.449	0.993	**75.7**	**73.9**	62.4	75.4
Focal-R + **FDS**	**0.920**	0.855	**1.169**	1.008	75.1	72.6	**66.4**	74.7
Focal-R + **LDS** + **FDS**	0.940	0.849	1.358	**0.916**	74.9	72.2	66.3	**77.3**
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	**1.209**	1.025	74.9	72.1	**67.2**	74.0
RRT + **LDS** + **FDS**	**0.903**	**0.806**	1.323	**0.936**	**76.0**	**73.8**	65.2	**76.7**
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	**1.225**	1.012	75.0	72.4	**66.6**	74.2
Inv + **LDS** + **FDS**	**0.907**	**0.802**	1.363	**0.942**	**76.0**	**74.0**	65.2	**76.6**
Ours (best) vs. Vanilla	+.071	+.049	+.419	+.068	+1.8	+2.0	+5.8	+2.1

- Both LDS and FDS improve results for various methods, esp. medium- and few-shot regions.

Table credit: *Yang et al. (2021)*

Inferring Depth

NYUD2

Metrics	RMSE ↓				δ_1 ↑			
Shot	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	**0.615**	0.940	2.059	0.681	**0.760**	0.695	0.596
Vanilla + **LDS** + **FDS**	**1.338**	0.670	**0.851**	**1.880**	**0.705**	0.730	**0.764**	**0.655**
Ours (best) vs. Vanilla	**+.139**	**-.024**	**+.101**	**+.243**	**+.028**	**-.017**	**+.071**	**+.085**

FDS and LDS

- alleviates overfitting on many-shot regions,
- generalizes better to all regions,
- slightly degrades many-shot region,
- boosts other regions.

Table credit: *Yang et al. (2021)*

Inferring Health Score

SHHS-DIR

Metrics	MAE ↓				GM ↓			
Shot	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	**11.06**	13.56	15.99	9.45	**6.95**	8.81	11.13
Focal-R + **LDS** + **FDS**	**14.02**	11.08	**12.24**	**15.49**	**9.32**	7.18	**8.10**	**10.39**
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**	**14.33**	**11.96**	**12.47**	**15.92**	**9.63**	**7.35**	**8.74**	**11.17**
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	**11.12**	12.29	15.53	8.94	**6.91**	7.79	10.65
Inv + **LDS** + **FDS**	**13.76**	**11.12**	**12.18**	**15.07**	**8.70**	6.94	**7.60**	**10.18**
Ours (best) vs. Vanilla	**+1.60**	**+1.41**	**+1.80**	**+1.87**	**+1.93**	**+1.13**	**+1.99**	**+2.02**

- Both FDS and LDS are effective.
- FDS + LDS often get highest gains over all tested regions.
- Note: SMOTER and SMOGN not directly applicable.

Table credit: *Yang et al. (2021)*

Could LDS + FDS help when label distribution is skewed with one or more Gaussian peaks?

- Experimental setup
  - Curated skewed label distributions with 1-4 Gaussian peaks on IMDB-WIKI-DIR
  - Compared to vanilla model

Skewed label distribution with one Gaussian peak

IMDB-WIKI

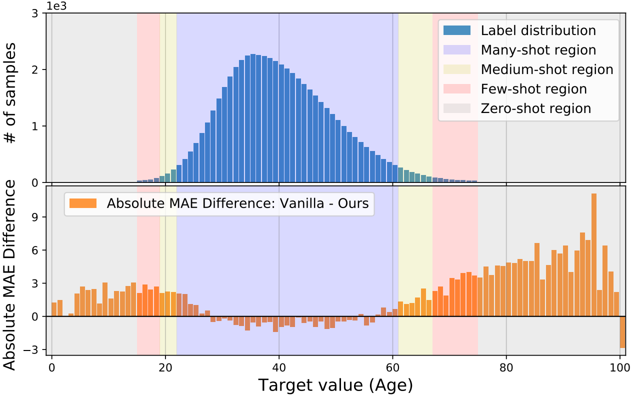


Figure: MAE gains of LDS + FDS over vanilla model.

- Performance gains, esp. for extrapolation & interpolation

Image credit: *Yang et al. (2021)*

Skewed label distribution with two Gaussian peaks

IMDB-WIKI

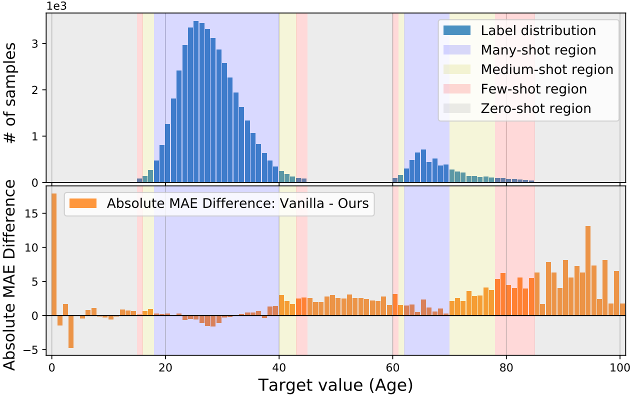


Figure: MAE gains of LDS + FDS over vanilla model.

- Performance gains, esp. for extrapolation & interpolation

Image credit: *Yang et al. (2021)*

Skewed label distribution with three Gaussian peaks

IMDB-WIKI

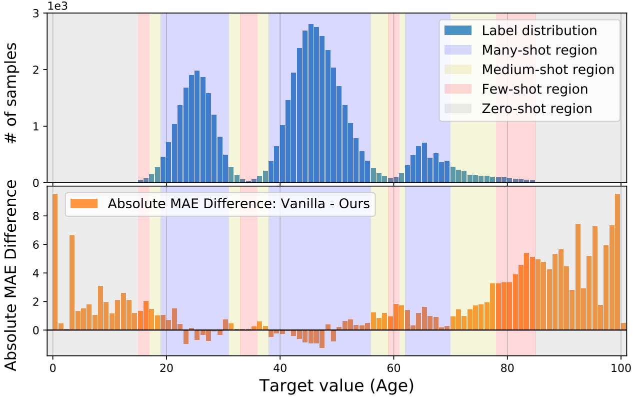


Figure: MAE gains of LDS + FDS over vanilla model.

- Performance gains, esp. for extrapolation & interpolation

Image credit: *Yang et al. (2021)*

Skewed label distribution with four Gaussian peaks

IMDB-WIKI

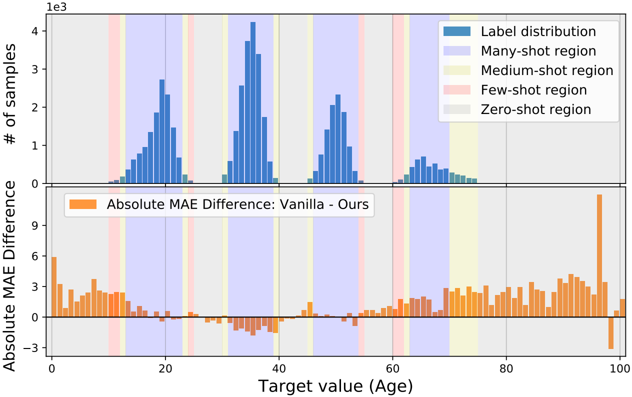


Figure: MAE gains of LDS + FDS over vanilla model.

- Performance gains, esp. for extrapolation & interpolation

Image credit: *Yang et al. (2021)*

Could LDS + FDS help when label distribution is skewed with one or more Gaussian peaks?

- Experimental setup
  - Curated skewed label distributions with 1-4 Gaussian peaks on IMDB-WIKI-DIR
  - Compared to vanilla model
- Findings
  - Robustness to distribution change
  - Brings improvement

Different skewed label distributions

IMDB-WIKI

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

- Typically best overall results by LDS + FDS
- Typically best many-shot results by LDS
- LDS + FDS can degrade many-shot region

Table credit: *Yang et al. (2021)*

Balanced vs. Imbalanced Test Label Distribution

IMDB-WIKI

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**

- Skewed label distribution for training set
- Case: balanced label distribution for test set.
  - LDS and FDS can improve performance of all regions.
- Case: skewed label distribution for test set, same label distribution for training set.
  - Minor degradation in many-shot region.
  - Boosts in medium-shot and few-shot regions.
  - Note: overall performance dominated by many-shot region, potentially biased and undesired evaluation.

Table credit: *Yang et al. (2021)*

Ablation: loss function

STS-B

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:																
MAE (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 (L2)	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 (sL1)	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:																
MAE (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 (L2)	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 (sL1)	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

- Similar results for all losses
- Robust to different losses

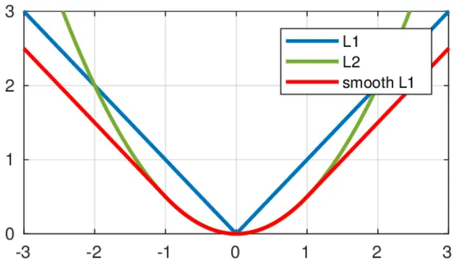


Table credit: *Yang et al. (2021)*, Image credit: `https://medium.com/artificialis/loss-functions-361b2ad439a`

Ablation: kernel type

IMDB-WIKI

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

- All kernel types lead to gains
- Often best results with Gaussian kernel

Table credit: *Yang et al. (2021)*

Ablation: kernel type

STS-B

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

- All kernel types lead to gains
- Often best results with Gaussian kernel

Table credit: *Yang et al. (2021)*

Ablation: Gaussian kernel hyper-parameters

IMDB-WIKI

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

- Gaussian kernel size $l \in \{5, 9, 15\}$ and standard deviation $\sigma \in \{1, 2, 3\}$
- LDS
  - Smaller σ usually leads to slightly better results over all regions.
  - Larger gains w.r.t. performance in medium-shot and few-shot regions.
  - Minor degradation in many-shot regions.
- FDS
  - Smaller l often obtains slightly higher improvements over all regions.
  - Equally boosts all regions, with slightly smaller improvements in medium-shot and few-shot regions.
- 3.3-6.2% overall MSE gain
- Best results with $l = 5$ and $\sigma = 2$
- Robust to different hyper-parameters

Table credit: *Yang et al. (2021)*

Ablation: Gaussian kernel hyper-parameters

STS-B

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

- Gaussian kernel size $l \in \{5, 9, 15\}$ and standard deviation $\sigma \in \{1, 2, 3\}$
- 3.3-6.2% overall MSE gain
- Best results with $l = 5$ and $\sigma = 2$
- Robust to different hyper-parameters

Table credit: *Yang et al. (2021)*

Comparison to imbalanced classification methods

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. 2019)	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**

- Imbalanced regression methods outperform classification ones.
- Can reduce error up to 50-60% in few-shot regions
- Imbalanced classification methods can perform worse than vanilla regression.
- Main finding: imbalance regression requires something different than just imbalance classification methods, which
  - can ignore similarity between nearby targets,
  - can ignore similarity between features linked to nearby targets,
  - cannot interpolate & extrapolate in continuous label space, so cannot deal with zero-shot label regions.

Table credit: *Yang et al. (2021)*

Summary

- Learning from imbalanced data with continuous targets.
- Generalization to the entire target range.
- Exploiting similarity between nearby targets in both label and feature spaces.
- Results on five curated large-scale real-world benchmarks in computer vision, natural language processing, and healthcare.

References I

- Branco, Paula, Luís Torgo, and Rita P Ribeiro (2017). “SMOGN: a pre-processing approach for imbalanced regression”. In: *First international workshop on learning with imbalanced domains: Theory and applications*. PMLR, pp. 36–50.
- Cui, Yin et al. (2019). “Class-balanced loss based on effective number of samples”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277.
- He, Kaiming et al. (2016). “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.
- Hu, Junjie et al. (2019). “Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries”. In: *2019 IEEE winter conference on applications of computer vision (WACV)*. IEEE, pp. 1043–1051.
- Kang, Bingyi et al. (2019). “Decoupling representation and classifier for long-tailed recognition”. In: *arXiv preprint arXiv:1910.09217*.

References II

Lin, T (2017). “Focal Loss for Dense Object Detection”. In: *arXiv preprint arXiv:1708.02002*.

Liu, Ziwei et al. (2019). “Large-scale long-tailed recognition in an open world”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2537–2546.

Parzen, Emanuel (1962). “On estimation of a probability density function and mode”. In: *The annals of mathematical statistics* 33.3, pp. 1065–1076.

Sun, Baochen, Jiashi Feng, and Kate Saenko (2016). “Return of frustratingly easy domain adaptation”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1.

Torgo, Luís et al. (2013). “Smote for regression”. In: *Portuguese conference on artificial intelligence*. Springer, pp. 378–389.

References III

- Wang, Alex et al. (2018). “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”. English (US). In: *EMNLP 2018 - 2018 EMNLP Workshop BlackboxNLP*. EMNLP 2018 - 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Proceedings of the 1st Workshop. 1st Workshop on BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, co-located with the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018 ; Conference date: 01-11-2018. Association for Computational Linguistics (ACL), pp. 353–355.
- Wang, Hao et al. (2019). “Bidirectional inference networks: A class of deep bayesian networks for health profiling”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 766–773.
- Yang, Yuzhe et al. (2021). “Delving into deep imbalanced regression”. In: *International conference on machine learning*. PMLR, pp. 11842–11851.