Balanced MSE for Imbalanced Visual Regression
– Supplementary Material –

Jiawei Ren[1], Mingyuan Zhang[1], Cunjun Yu[2], Ziwei Liu[1] ✉
[1]S-Lab, Nanyang Technological University
[2]School of Computing, National University of Singapore
{jiawei011, mingyuan001}@e.ntu.edu.sg, cunjun.yu@comp.nus.edu.sg, ziwei.liu@ntu.edu.sg

1. Proofs and Derivations

1.1. Proof for Theorem 1

By Bayes Rule, we have:

$$p_{\text{train}}(\boldsymbol{y}|\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y}) \cdot p_{\text{train}}(\boldsymbol{y})/p_{\text{train}}(\boldsymbol{x}) \tag{1.1}$$

$$p_{\text{bal}}(\boldsymbol{y}|\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y}) \cdot p_{\text{bal}}(\boldsymbol{y})/p_{\text{bal}}(\boldsymbol{x}) \tag{1.2}$$

By change of variables, we have:

$$p_{\text{train}}(\boldsymbol{y}|\boldsymbol{x}) = p_{\text{bal}}(\boldsymbol{y}|\boldsymbol{x}) \cdot \frac{p_{\text{train}}(\boldsymbol{y})}{p_{\text{bal}}(\boldsymbol{y})} \cdot \frac{p_{\text{bal}}(\boldsymbol{x})}{p_{\text{train}}(\boldsymbol{x})} \tag{1.3}$$

The evidence ratio $\frac{p_{\text{bal}}(\boldsymbol{x})}{p_{\text{train}}(\boldsymbol{x})}$ in Eq. 1.3 is unknown. To bypass the unknown ratio, we use the definition that the integral of $p_{\text{train}}(\boldsymbol{y}|\boldsymbol{x})$ over space Y should be equal to 1. Using the simple fact, we have:

$$p_{\text{train}}(\boldsymbol{y}|\boldsymbol{x}) = \frac{p_{\text{train}}(\boldsymbol{y}|\boldsymbol{x})}{\int_Y p_{\text{train}}(\boldsymbol{y}'|\boldsymbol{x})d\boldsymbol{y}'}. \tag{1.4}$$

Bring Eq. 1.3 into Eq. 1.4, we have:

$$\begin{aligned}
p_{\text{train}}(\boldsymbol{y}|\boldsymbol{x}) &= \frac{p_{\text{bal}}(\boldsymbol{y}|\boldsymbol{x}) \cdot \frac{p_{\text{train}}(\boldsymbol{y})}{p_{\text{bal}}(\boldsymbol{y})} \cdot \frac{p_{\text{bal}}(\boldsymbol{x})}{p_{\text{train}}(\boldsymbol{x})}}{\int_Y p_{\text{bal}}(\boldsymbol{y}'|\boldsymbol{x}) \cdot \frac{p_{\text{train}}(\boldsymbol{y}')}{p_{\text{bal}}(\boldsymbol{y}')} \cdot \frac{p_{\text{bal}}(\boldsymbol{x})}{p_{\text{train}}(\boldsymbol{x})} d\boldsymbol{y}'} && (1.5)\\
&= \frac{p_{\text{bal}}(\boldsymbol{y}|\boldsymbol{x}) \cdot \frac{p_{\text{train}}(\boldsymbol{y})}{p_{\text{bal}}(\boldsymbol{y})}}{\int_Y p_{\text{bal}}(\boldsymbol{y}'|\boldsymbol{x}) \cdot \frac{p_{\text{train}}(\boldsymbol{y}')}{p_{\text{bal}}(\boldsymbol{y}')} d\boldsymbol{y}'} && (1.6)\\
&= \frac{p_{\text{bal}}(\boldsymbol{y}|\boldsymbol{x}) \cdot p_{\text{train}}(\boldsymbol{y})}{\int_Y p_{\text{bal}}(\boldsymbol{y}'|\boldsymbol{x}) \cdot p_{\text{train}}(\boldsymbol{y}') d\boldsymbol{y}'} && (1.7)
\end{aligned}$$

1.2. MSE as a Special Case of Balanced MSE

We show that MSE is a special case of Balanced MSE. When $p_{\text{train}}(\boldsymbol{y})$ is uniform on Y,

$$\begin{aligned}
&\log \int_Y \mathcal{N}(\boldsymbol{y}; \boldsymbol{y}_{\text{pred}}, \sigma^2_{\text{noise}}\mathrm{I}) \cdot p_{\text{train}}(\boldsymbol{y}) d\boldsymbol{y}\\
&= \log \int_Y \mathcal{N}(\boldsymbol{y}; \boldsymbol{y}_{\text{pred}}, \sigma^2_{\text{noise}}\mathrm{I}) \cdot C d\boldsymbol{y}\\
&= \log \int_Y \mathcal{N}(\boldsymbol{y}; \boldsymbol{y}_{\text{pred}}, \sigma^2_{\text{noise}}\mathrm{I}) d\boldsymbol{y} + \log C\\
&= \log 1 + \log C = \log C,
\end{aligned} \tag{1.8}$$

where C is some constant. Then, the Balanced MSE loss becomes $-\log \mathcal{N}(\boldsymbol{y}; \boldsymbol{y}_{\text{pred}}, \sigma^2_{\text{noise}}\mathrm{I}) + \log C$ and is equivalent to the standard MSE loss.

1.3. GAI Loss Derivation

We continue our derivation from Eq 3.11. The integral of a Gaussian is trivial to solve:

$$\sum_{i=1}^{K} \phi_i S_i \int_Y \mathcal{N}(\boldsymbol{y}; \tilde{\boldsymbol{\mu}}_i, \tilde{\boldsymbol{\Sigma}}_i) d\boldsymbol{y} = \sum_{i=1}^{K} \phi_i S_i \tag{1.9}$$

Therefore, the closed-form loss of Balanced MSE is:

$$\begin{aligned}
L = &-\log \mathcal{N}(\boldsymbol{y}; \boldsymbol{y}_{\text{pred}}, \sigma^2_{\text{noise}}\mathrm{I})\\
&+ \log \int_Y \mathcal{N}(\boldsymbol{y}'; \boldsymbol{y}_{\text{pred}}, \sigma^2_{\text{noise}}\mathrm{I}) \cdot p_{\text{train}}(\boldsymbol{y}') d\boldsymbol{y}'\\
&= -\log \mathcal{N}(\boldsymbol{y}; \boldsymbol{y}_{\text{pred}}, \sigma^2_{\text{noise}}\mathrm{I}) + \log \sum_{i=1}^{K} \phi_i S_i
\end{aligned} \tag{1.10}$$

Recall that S_i is the norm of the product of two Gaussians. S_i itself is also a Gaussian:

$$S_i = \mathcal{N}(\boldsymbol{y}_{\text{pred}}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i + \sigma^2_{\text{noise}}\mathrm{I}) \tag{1.11}$$

Bring Eq. 1.11 back to Eq. 1.10, we have:

$$\begin{aligned} L = &-\log \mathcal{N}(\boldsymbol{y}_{\text{target}}; \boldsymbol{y}_{\text{pred}}, \sigma^2_{\text{noise}}\mathrm{I}) \\ &+ \log \sum_{i=1}^{K} \phi_i \cdot \mathcal{N}(\boldsymbol{y}_{\text{pred}}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i + \sigma^2_{\text{noise}}\mathrm{I}) \end{aligned} \tag{1.12}$$

2. Implementation Details

2.1. Synthetic Benchmark

2.1.1 Dataset Construction

For the training set, we first randomly sample 1024 labels $\boldsymbol{y}$ from a predefined label distribution $p_{\text{train}}(\boldsymbol{y})$, *e.g.*, a normal distribution. Then, we minus a random noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathrm{I})$ from the lables, to obtain the true labels $\tilde{\boldsymbol{y}}$ so that $\boldsymbol{y} = \tilde{\boldsymbol{y}} + \boldsymbol{\epsilon}$. For an invertible mapping function $f : X \rightarrow Y$, *e.g.*, a linear function, we find its inverse function f^{-1}, and generate inputs $\boldsymbol{x}$ from the true labels $\tilde{\boldsymbol{y}}$ using f^{-1}. After that, we have:

$$\boldsymbol{y} = \tilde{\boldsymbol{y}} + \boldsymbol{\epsilon} = f(\boldsymbol{x}) + \boldsymbol{\epsilon} \tag{2.1}$$

To this end, $(\boldsymbol{x}, \boldsymbol{y})$ is a standard regression dataset and $\boldsymbol{y}$ has a predefined imbalanced distribution. We call f the oracle relation and our goal is to estimate f from $(\boldsymbol{x}, \boldsymbol{y})$.

For the test set, we repeat the above procedure except that we use a uniform label distribution and do not apply the random noise.

2.1.2 Training Details

In training, we use a batch size 256. For one-dimensional linear regression, we train the models for 2K epochs. We use SGD optimzer with momentum 0.9. We set the learning rate to 1e-3. For non-linear regressions and two-dimensional linear regressions, we train the models for 10K epochs. We use Adam [7] optimizer and set the learning rate to 0.2.

2.2. IMDB-WIKI-DIR

We follow the RRT setting in [14]. Concretely, we use ResNet-50 [2] model as the backbone. We train the vanilla model for 90 epochs using Adam optimizer [7]. We decay the learning rate from 10^{-3} by 0.1 at 60-th epoch and 80-th epoch. We then freeze the backbone, re-initialize and train the last linear layer. For the retraining, we train the last linear layer for 30 epochs with a constant learning rate at 10^{-4}. We use a GMM with 2 components.

2.3. NYUD2-DIR

We follow the settings in [14]. We use a ResNet-50-based encoder-decoder architecture proposed by [3]. We train the model for 20 epochs using Adam optimizer with an initial learning rate at 10^{-4}. The learning rate decays by 0.1 every 5 epochs. Only direct supervision on depth is used in training. We use a GMM with 16 components.

2.4. IHMR

We use a pretrained SPIN [8] model as the feature extractor, and re-train the linear regressor for 20 epochs. We follow SPIN to train on the following 3D datasets: Human3.6M [4], MPI-INF-3DHP [11]; and following 2D datasets: LSP [6]; LSP-extended [8], MPII [1], COCO [10]. We test on 3DPW [13]. Static fits are used to provide supervision on the 2D datasets. We use a constant learning rate at 10^{-4}. We use a GMM with 16 components.

2.5. Noise Scale Learning

We set σ_{noise} as a learnable variable that requires gradient, and add it into the optimizer so that σ_{noise} can be optimized together with model parameters. There are no additional network or architecture modifications for the noise scale learning.

3. Experiment on random seeds

We compare least square, reweighting, and Balanced MSE under different random seeds in the one-dimensional linear regression. A visualization of results is shown in Fig. 1. We observe that reweighting is sensitive to random seeds. Reweighting's performance varies drastically when random seed changes. This may attribute to the fact that reweighting signifies rare labels' noise and the zero mean noise assumption no longer holds. In comparison, Balanced MSE is robust to different noise sampling results.

4. Quantitative results for the synthetic benchmark

We show the quantitative results for the synthetic benchmarks. There are three settings in the quantitative results. **Normal**: one-dimensional linear regression where the label distribution is a Normal distribution. **Exponential**: one-dimensional linear regression where the label distribution is an Exponential distribution. **MVN**: two-dimensional linear regression where the label distribution is a Multivariate Normal distribution.

Different extents of distribution skewness are studied as well. The results show that **1)** both GAI and BMC significantly outperforms Vanilla (*i.e.*, least square) and Reweighting, particularly when the skewness is high; **2)** the numerical implementation BMC shows comparable performance to the closed-form implementation GAI; **3)** using learned noise scale achieves a comparable performance to using the true noise scale.

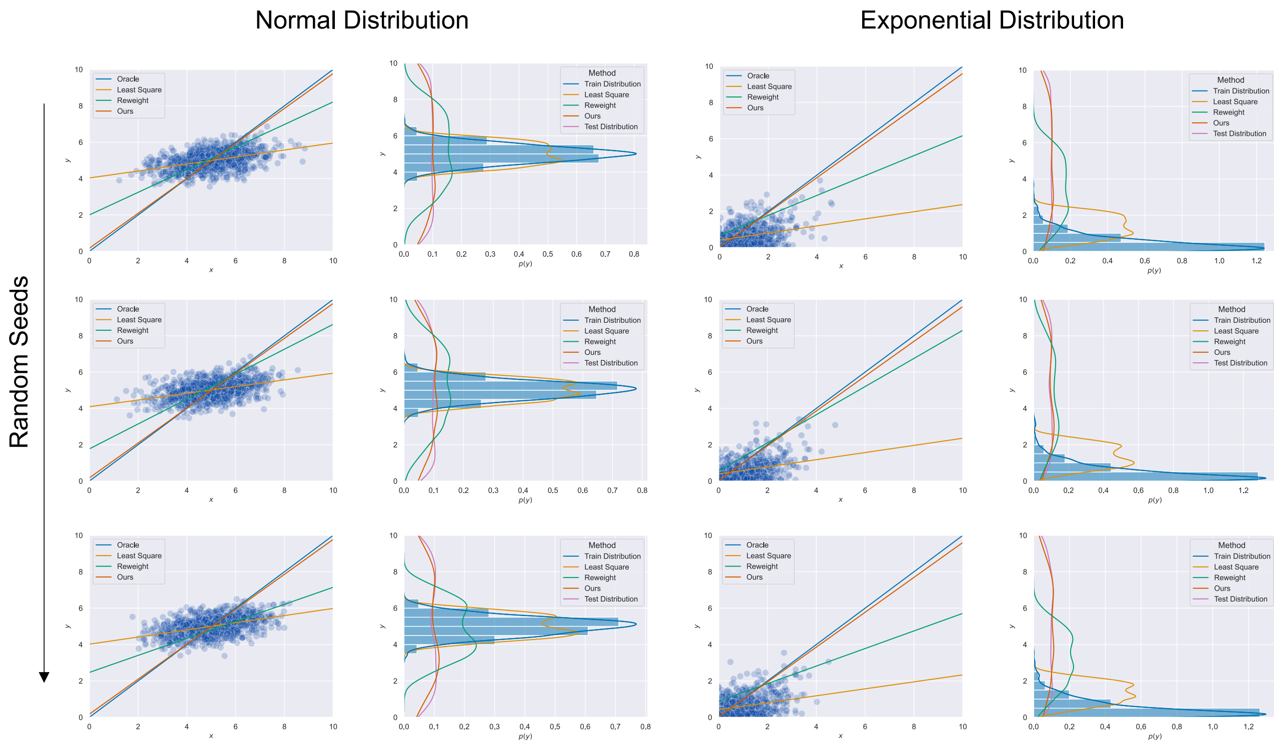

Figure 1. Synthetic benchmark on random seeds. Although the noise scale keeps the same, reweighting's performance varies drastically when different random seeds are used. In comparison, Balanced MSE is robust to different sampled noises.

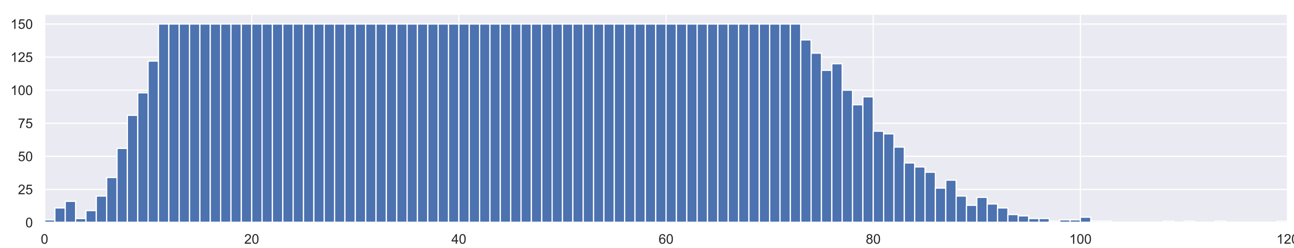

Figure 2. IMDB-WIKI-DIR test set visualization. We observe tail labels on both edges of the test distribution. Overall metrics will not sufficiently assess a model's performance on senior adults (age $>\sim 75$) and children (age $<\sim 15$).

5. IMDB-WIKI-DIR test set visualization

We visualize the label distribution of IMDB-WIKI-DIR's test set in Fig. 2.

6. Ablations

6.1. Effect of the noise scale

We study the effect of σ_{noise} on IMDB-WIKI-DIR, by fixing σ_{noise} at different values. We use the GAI option for study. We also compare fixed σ_{noise} (Fix.) with jointly optimized σ_{noise} (Joint.). Results are shown in Tab. 2. We observe that larger σ_{noise} trades the performance towards tail labels. We also observe that the jointly optimized σ_{noise} is effective in finding the optimal trade-off point.

6.2. Effect of number of components in GMM

We study the number of components K in GMM on IMDB-WIKI-DIR using the GAI variant. Results are shown in Tab. 3. We notice that the performance reaches optimal when K is larger or equal to 2. This may attribute to the fact that the training label distribution of IMDB-WIKE-DIR is relatively simple.

7. Additional Discussions and Analysis

Analysis on the Computational Cost. We compare Balanced MSE with other methods in terms of computational

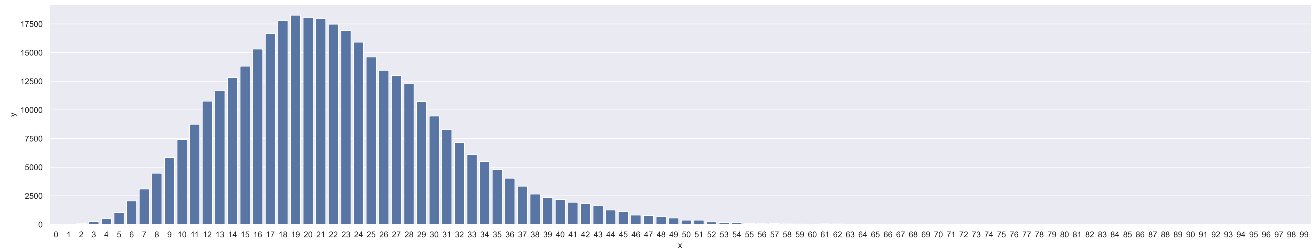

Figure 3. Visualization of the training label distribution of IHMR. The horizontal axis is 100 regions uniformly divided on the pose space according to their geodesic distance to the mean pose.

Table 1. Quantitative results for the synthetic benchmark. †: True noise scale used. For each type of distribution, we evaluate three extents of skewness: Low, Moderate, and High. Best results are bolded.

	Normal (MSE↓)			Exponential (MSE↓)			MVN (MSE↓)		
Method	High	Mod.	Low	High	Mod.	Low	High	Mod.	Low
Vanilla	5.521	3.275	1.936	18.61	13.14	6.038	5.522	3.809	2.570
Reweight	1.399	0.336	0.092	4.676	1.336	0.128	3.310	1.758	1.001
Ours (GAI)†	**0.031**	**0.001**	0.001	**0.001**	0.002	0.004	**0.122**	0.031	0.011
Ours (BMC)†	0.043	0.004	**0.000**	0.002	**0.000**	**0.000**	0.126	0.033	0.011
Ours (GAI)	0.089	0.008	0.005	0.130	0.082	0.023	0.184	**0.021**	**0.006**
Ours (BMC)	0.141	0.060	0.030	0.122	0.104	0.034	0.142	0.025	0.011

Table 2. Ablation on the choice of noise on IMDB-WIKI-DIR.

	bMAE↓				MAE↓			
Method	All	Many	Med.	Few	All	Many	Med.	Few
Fix. ($\sigma = 6$)	12.85	7.27	13.26	29.79	7.81	7.20	12.78	23.78
Fix. ($\sigma = 7$)	12.67	7.52	12.75	28.67	8.00	7.45	12.32	23.25
Fix. ($\sigma = 8$)	12.68	7.80	12.61	27.83	8.24	7.73	12.21	22.94
Joint.	12.66	7.65	12.68	28.14	8.12	7.58	12.27	23.05

Table 3. Ablation on the effect of the number of components K in the GMM.

	bMAE↓				MAE↓			
Method	All	Many	Med.	Few	All	Many	Med.	Few
K=1	12.72	7.70	12.94	28.08	8.18	7.63	12.47	23.17
K=2	12.66	7.65	12.68	28.14	8.12	7.58	12.27	23.05
K=4	12.67	7.62	12.68	28.26	8.09	7.55	12.26	23.03
K=128	12.66	7.61	12.87	28.11	8.09	7.53	12.44	23.18

cost in this section. We show the train-time computational cost on IMDB-WIKI-DIR in Tab. 4. Results are averaged on the first epoch. The overhead is negligible compared to overall cost. There is no additional computational cost during inference.

How is Balanced MSE connected to the Bayes-optimal prediction? We use $\boldsymbol{y}_{\text{pred}}$, the mean of the predicted Gaussian, to infer the final label. Since the mean and the mode are the same for a Gaussian distribution, it is by definition that $\boldsymbol{y}_{\text{pred}}$ estimated by Balanced MSE is the Bayes-optimal prediction for a balanced test set: $\boldsymbol{y}_{\text{pred}} = \text{argmax}_{\boldsymbol{y}} \mathcal{N}(\boldsymbol{y}; \boldsymbol{y}_{\text{pred}}, \sigma^2_{\text{noise}}\mathrm{I}) = \text{argmax}_{\boldsymbol{y}} p_{\text{bal}}(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\theta})$.

Why model the noisy prediction as an isotropic Gaussian? The isotropic Gaussian noise is assumed by ordinary least square [5]. More fine-grained noise correlations modeling can lead to better regression performance [5] but is out of the scope of Balanced MSE.

Will modeling the uncertainty explicitly help imbal-

Table 4. Computational cost comparison.

	Time (s/iter)	Memory	Remark
RRT	0.29	6502MB	-
LDS	0.29	6502MB	-
GAI	0.30	6502MB	K=2
GAI	0.30	6512MB	K=512
BMC	0.30	6504MB	B=256

anced regression? Balanced MSE estimates a constant noise and degrades to MSE when no imbalance exists, *i.e.*, the gain is from imbalance handling not from uncertainty modeling. However, sophisticated uncertainty modeling, *e.g.*, correlated noise [5] and input-dependent noise [9], could help regression in general.

Can we extend the analysis in Balanced MSE to L1 & Huber loss? Extending L1 & Huber loss to balanced versions will be important future works, which can be done via Theorem 1 by replacing Gaussian in this work to Laplacian and [12] respectively.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[3] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1043–1051. IEEE, 2019. 2

[4] Catalin Ionescu, Fuxin Li, and Cristian Sminchisescu. Latent structured models for human pose estimation. In *2011 International Conference on Computer Vision*, pages 2220–2227. IEEE, 2011. 2

[5] Prateek Jain and Ambuj Tewari. Alternating minimization for regression problems with vector-valued outputs. *Advances in Neural Information Processing Systems*, 28, 2015. 4, 5

[6] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, pages 1–11. British Machine Vision Association, 2010. 2

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. 2

[8] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2252–2261, 2019. 2

[9] Quoc V Le, Alex J Smola, and Stéphane Canu. Heteroscedastic gaussian process regression. In *Proceedings of the 22nd international conference on Machine learning*, pages 489–496, 2005. 5

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[11] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 international conference on 3D vision (3DV)*, pages 506–516. IEEE, 2017. 2

[12] Gregory P Meyer. An alternative probabilistic interpretation of the huber loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5261–5269, 2021. 5

[13] Timo von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 601–617, 2018. 2

[14] Yuzhe Yang, Kaiwen Zha, Ying-Cong Chen, Hao Wang, and Dina Katabi. Delving into deep imbalanced regression. In *International Conference on Machine Learning (ICML)*, 2021. 2