Supplementary material for: Distance Matters For Improving Performance Estimation Under Covariate Shift

Mélanie Roschewitz
Imperial College London
mb121@ic.ac.uk

Ben Glocker
Imperial College London
b.glocker@imperial.ac.uk

Supp Note 1: Additional information on datasets used for this study

Dataset family	ID datasets / splits	OOD datasets / splits	N classes	Type of shift
ImageNet	Train / Val	ImageNet-Sketch	1,000	Photographs to Sketches
PACS	Photo train / val	Painting test Cartoon test Sketches test	7	Art type
MNIST	MNIST	SVHN	10	Binary images to house numbers
CIFAR10	Train / Val	CIFAR10-C	10	Synthetic corruptions
Entity30	Train / Val	OOD test	30	Subpopulation shift (random)
Entity13	Train / Val	OOD test	30	Subpopulation shift (random)
Living17	Train / Val	OOD test	17	Subpopulation shift (random)
NonLiving26	Train / Val	OOD test	26	Subpopulation shift (random)
WILDS Camelyon	Train / id-Val	ood-test, ood-val	2	Site / staining protocol
WILDS iCam	Train / id-Val	ood-test, ood-val	182	Location of camera
WILDS FmoW	Train / id-Val	ood-test, ood-val	62	Location and time
WILDS RxRx1	Train / id-Val	ood-test, ood-val	1,189	Experimental session
PathMNIST	Train / Val	Test	9	Site / staining protocol

Table 1. **Additional information for the dataset used.** "Dataset family" denotes the name used in the tables in the main paper to refer to this task. For all datasets we used the official splits as denoted in the columns.

Supp Note 2: Additional information on model training

All our training and evaluation code as well as data augmentation and training configurations are available in our codebase https://github.com/melanibe/distance_matters_performance_estimation.

ImageNet models For ImageNet, we used readily available trained models from the `timm` [15] package. We evaluate all available models from the following 14 family of model architectures: ConvNext [4], ConvMixer [11], DarkNet [7], CSPNet [13], EfficientNet [10], Inception ResNet [9], ResNext [16], ResNeSt [17], TResNet [8], DenseNet [3], ResNet [2], ResNetv2 [9], ECA-Net [14], Res-SE-Net [12]. This amounted to testing a total of 259 trained models.

Trained models For each model / training configuration we repeated training for 3 different seeds. Details of training configurations are listed in the table below. In total, we train 18 models from random initialisation and 12 models from ImageNet weights, amounting to 30 models for each dataset, except for the BREEDS datasets for which we only used the models trained from scratch (as they are subsets of ImageNet). The "standard" training procedure uses Adam optimiser, automatic learning rate adaptation after 10 epochs without improvement, early stopping when the accuracy did not improve anymore for 15 epochs. Unless specified otherwise, we used data augmentation during training (incl. random rotation, color jittering, flips, cropping), all data augmentations configurations can be found in the codebase.

Model architecture	Type of init.	Training procedure	Data augmentation
ResNet18 [2]	Random	Standard	Yes
ResNet18 [2]	Random	Standard + Weight decay $1e-3$	Yes
ResNet18 [2]	ImageNet weights	Standard	Yes
ResNet50 [2]	Random	Standard	Yes
ResNet50 [2]	Random	Standard	No
ResNet50 [2]	ImageNet weights	Standard	Yes
ResNet50 [2]	ImageNet weights	Standard + Weight decay $1e-3$	Yes
DenseNet [3]	Random	Standard	Yes
EfficientNet-S [10]	Random	Standard	Yes
EfficientNet-S [10]	ImageNet weights	Standard	Yes

Table 2. Details of training configurations. Standard training procedure is described above. ImageNet weights are obtained from `torchvision` [6] package.

Implementation details for baselines For COT, we used the suggestion of the authors to counter the cubic runtime of the method: we used batches of 2,500 images to estimate accuracy and averaged the accuracy over the batches, using up to 25,000 randomly sampled samples. To compute the Wasserstein distance we used the POT package [1] as suggested by the authors of COT [5]

[More on next page.]

Supp Note 3: Scatter plots Predicted versus True accuracy

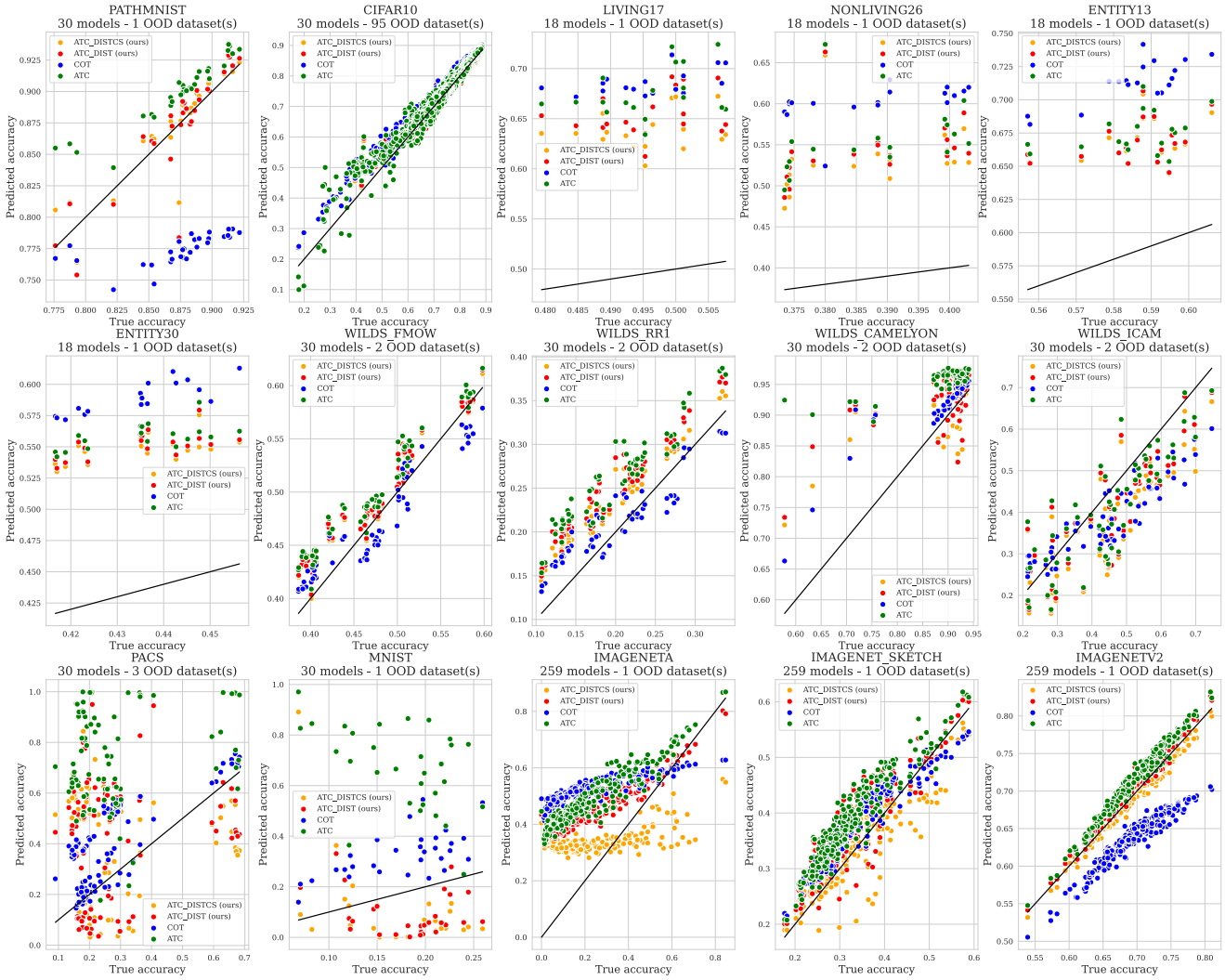

Figure 1. **Predicted versus true accuracy for all models and datasets.**

Supp Note 4: Additional results for ablation study on the choice of distance estimation method.

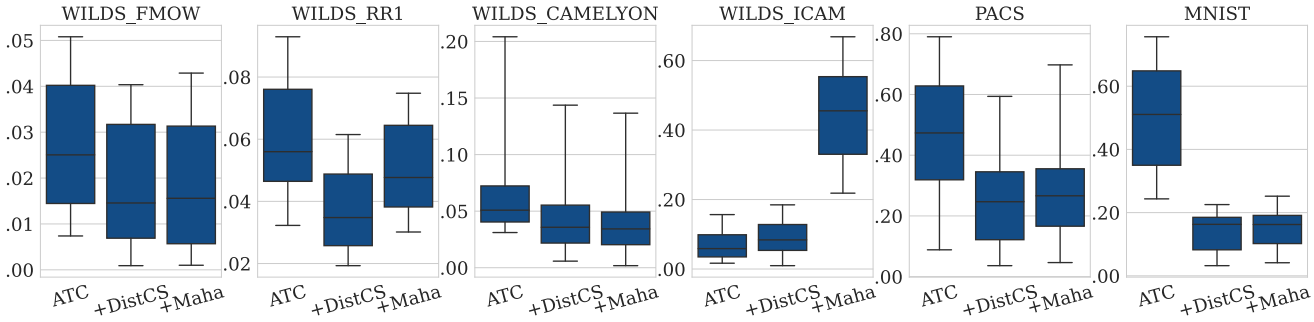

Figure 2. **Ablation study for the choice of distance estimation method: K-NN (DistCS) versus Mahalanobis distance (Maha).** Each boxplot shows the distribution of the Mean Absolute Error for accuracy estimation. Whiskers denote the [5%;95%]-percentiles of the distribution, outliers omitted for readability. Using distance improves the results for all but one dataset, no matter if K-NN or Mahalanobis distance. However, K-NN distance is better than Mahalanobis overall.

Supp Note 5: K-NN ablation study

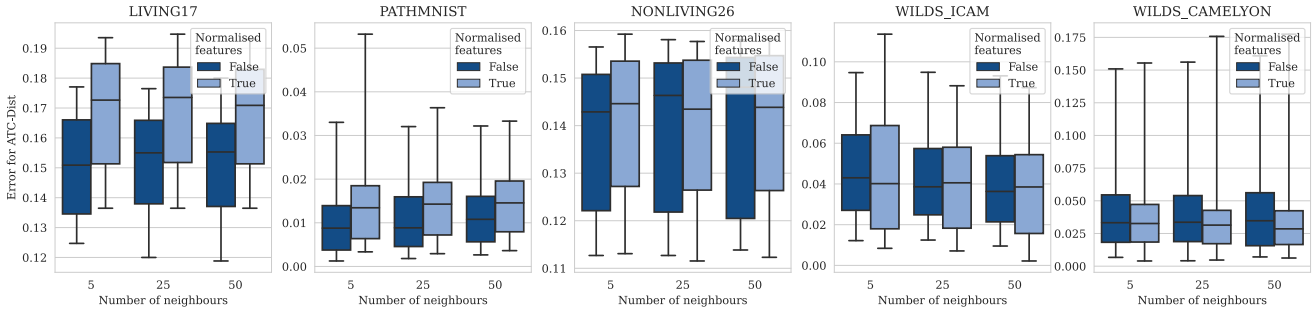

Figure 3. **Ablation study for parameters of nearest neighbours for ATCDist.** The method is not sensitive to the choice of number of neighbours and to normalisation of the features (i.e. dividing the features by their norm) does not significantly impact the performance. We compare the performance of the ATC-Dist estimator for 5 tasks in different settings. Each boxplot represents the distribution of absolute errors for accuracy estimation over all trained models. Whiskers denote the $[5;95]^{th}$ percentiles of the distribution. Outliers are omitted for readability.

Supp Note 6: Ablation study on the distance threshold choice

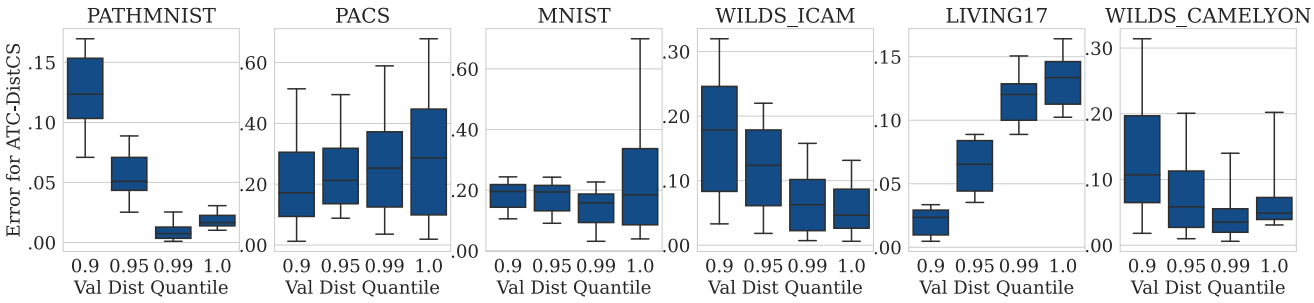

Figure 4. **Ablation study: MSE of ATC-Dist in function of distance threshold** (i.e. observed quantile on validation set). Our choice of threshold offers good generalisation across all tasks.

References

[1] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. 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. 1, 2

[3] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1, 2

[4] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 1

[5] Yuzhe Lu, Zhenlin Wang, Runtian Zhai, Soheil Kolouri, Joseph Campbell, and Katia P Sycara. Predicting out-of-distribution error with confidence optimal transport. In *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*, 2023. 2

[6] TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. https://github.com/pytorch/vision, 2016. 2

[7] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1

[8] Tal Ridnik, Hussam Lawen, Asaf Noy, Emanuel Ben Baruch, Gilad Sharir, and Itamar Friedman. Tresnet: High performance gpu-dedicated architecture. In *proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1400–1409, 2021. 1

[9] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 1

[10] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 1, 2

[11] Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022. 1

[12] V Varshaneya, S Balasubramanian, and Darshan Gera. Res-se-net: Boosting performance of resnets by enhancing bridge connections. *Machine Learning Algorithms and Applications*, pages 61–75, 2021. 1

[13] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 390–391, 2020. 1

[14] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11534–11542, 2020. 1

[15] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. 1

[16] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 1

[17] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2736–2746, 2022. 1