Supplementary Material for Out of Distribution Detection via Neural Network Anchoring

1. APPENDIX

Details of Benchmark Datasets : We use commonly used benchmarks to evaluate AMP, these include the following datasets – iSUN (Xu et al., 2015), LSUN (R), LSUN (C) (Yu et al., 2015), Places365 (Zhou et al., 2017), Texture (Cimpoi et al., 2014), and SVHN (Netzer et al., 2011)

Consistency training details The transformation $\mathcal{T}$ was applied to the anchors using a pre-specified schedule, every 5^{th} batch for CIFAR-10/100 and every 10^{th} batch for ImageNet, while the clean anchors were used directly in the other batches. However, from our experiments, we found that the choice of this schedule is not sensitive and the detection performance was similar even with other schedules. During the inference step, we did not utilize any transformation $\mathcal{T}$, and fixed the number of anchors $K = 5$ while making predictions for a test image. We performed an ablation on the number of anchors (reported at the end of the section), and observed that even a small number of random anchors was sufficient to obtain good detection performance, thus making our approach efficient in practice.

During training we always use $K = 1$ anchor, which is typically chosen by randomly shuffling the current batch so that every input sample is assigned a random anchor from that batch. During training we use `RandomCrop, RandomHorizontalFlip` augmentations in Pytorch. For the test set and the OOD set, we normalize data to the same mean and standard deviation as the training set without any additional transformations.

Hyperparameter settings **CIFAR-10/100:** We use standard training protocol for both CIFAR-10/100 datasets using all our networks – WideResNet, ResNet-18, ResNet-34 (He et al., 2016). We use an SGD optimizer with an initial learning rate of 0.1, momentum of 0.9, and weight decay of $5e - 4$. This learning rate is scaled down by a $\gamma = 0.2$ using a schedule of [60, 120, 160] epochs out of the total 200 epochs for training. We use a batch size of 128 in all our training experiments for CIFAR datasets. **ImageNet:** We also follow standard training protocol for ResNet-50 on ImageNet as well. We use an SGD optimizer with a learning rate of 0.1, weight decay of $1e - 4$, momentum of 0.9. We decay the learning rate by 0.1 every 30 epochs, and train for a total of 120 epochs. We use a batch size of 128 to train the model.

Method	AUROC ↑
ResNet-18 (He et al., 2016)	91.77 ± 1.85
DUQ (Van Amersfoort et al., 2020)	92.70 ± 1.30
Deep Ens (Lakshminarayanan et al., 2017)	94.70
AMP	**97.41 ± 0.72**

Table 1: OOD Detection with uncertainties on CIFAR-SVHN with ResNet-18.

1.1. Modification to anchor a model

We demonstrate with more detailed pseudo-code, the simple modification to be able to train with anchoring.

1.2. Additional Results

We report detailed results for individual datasets on various benchmarks used in the paper here. Table 3 and Table 2 report 4 performance metrics for the SCOOD benchmark (Yang et al., 2021), where we use the re-sampled OOD set following the SCOOD protocol. We observe competitive performance on CIFAR-10 and state-of-the-art on CIFAR-100 with AMP. Next, Table 4, we report detailed performance numbers on the second OOD benchmark used in the paper. We note that our method consistently performs either the best or second best as compared to GM (Sastry and Oore, 2020), while being better on average across the various datasets. In particular, we see that on challenging datasets like near-OOD AMP is significantly better than all competing baselines. Finally, in Table 1 we show uncertainty based OOD on a CIFAR-10 vs SVHN benchmark, compared to other uncertainty based approaches. We see once again that AMP is significantly better than sophisticated methods including Deep Ensembles that requires multiple models to be trained.

References

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

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.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with gram matrices. In *International Conference on Machine Learning*, pages 8491–8501. PMLR, 2020.

Algorithm 1: PyTorch-style pseudo-code for anchoring.

```
def create_anchored_model(model):
    model.conv1 = nn.Conv2d(in_channels=6, 64)
    return model

Tx = transforms.Compose([
    transforms.RandomResizedCrop(size=224),
    transforms.RandomHorizontalFlip(),
    transforms.RandomApply([color_jitter,blurr], p=0.8),
    ])
## load model and change the first conv layer

model_basic = ResNet50(pre_trained=False,n_class=1000)
model = create_anchored_model(model_basic)

## load datasets, setup optimizer, define criterion etc.
for images, targets in train_loder:
    batch_order = np.arange(images.shape[0])
    np.random.shuffle(batch_order)
    anchors = images[batch_order,:,:,:]
    diff = images-anchors
    if i % 10 ==0:
        tx_anchors = Tx(anchors)
    else:
        tx_anchors = anchors

    batch = torch.cat([tx_anchors,diff],axis=1)
    output = model(batch)
    loss = criterion(output, target)
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, pages 9690–9700. PMLR, 2020.

Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

Jingkang Yang, Haoqi Wang, Litong Feng, Xiaopeng Yan, Huabin Zheng, Wayne Zhang, and Ziwei Liu. Semantically coherent out-of-distribution detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8301–8309, 2021.

Method	Dataset	FPR95 ↓	AUROC ↑	AUPR(In/Out) ↑
ODIN	Texture	42.52	84.06	86.01 / 80.73
	SVHN	52.27	83.26	63.76 / 92.60
	CIFAR-100	56.34	78.40	73.21 / 80.99
	Tiny-ImageNet	59.09	79.69	79.34 / 77.52
	LSUN	47.85	84.56	81.56 / 85.58
	Places365	53.94	82.01	54.92 / 93.30
	Mean	**52.00**	**82.00**	**73.13** / **85.12**
EBO	Texture	52.11	80.70	83.34 / 75.20
	SVHN	30.56	92.08	80.95 / 96.28
	CIFAR-100	56.98	79.65	75.09 / 81.23
	Tiny-ImageNet	57.81	81.65	81.80 / 78.75
	LSUN	50.56	85.04	82.80 / 85.29
	Places365	52.16	83.86	58.96 / 93.90
	Mean	**50.03**	**83.83**	**77.15** / **85.11**
MCD	Texture	83.92	81.59	90.20 / 63.27
	SVHN	60.27	89.78	85.33 / 94.25
	CIFAR-100	74.00	82.78	83.97 / 79.16
	Tiny-ImageNet	78.89	80.98	85.63 / 72.48
	LSUN	68.96	84.71	85.74 / 81.50
	Places365	72.08	83.51	69.44 / 92.52
	Mean	**73.02**	**83.89**	**83.39** / **80.53**
OE	Texture	51.17	89.56	93.79 / 81.88
	SVHN	20.88	96.43	93.62 / 98.32
	CIFAR-100	58.54	86.22	86.17 / 84.88
	Tiny-ImageNet	58.98	87.65	90.9 / 82.16
	LSUN	57.97	86.75	87.69 / 85.07
	Places365	55.64	87.00	73.11 / 94.67
	Mean	**50.53**	**88.93**	**87.55** / **87.83**
UDG	Texture	20.43	96.44	98.12 / 92.91
	SVHN	13.26	97.49	95.66 / 98.69
	CIFAR-100	47.20	90.98	91.74 / 89.36
	Tiny-ImageNet	50.18	91.91	94.43 / 86.99
	LSUN	42.05	93.21	94.53 / 91.03
	Places365	44.22	92.64	87.17 / 96.66
	Mean	**36.22**	**93.78**	**93.61** / **92.61**
AMP (ours)	Texture	52.43	88.74	91.91 / 80.48
	SVHN	12.53	97.60	95.58 / 98.83
	CIFAR-100	48.10	89.61	88.99 / 88.47
	Tiny-ImageNet	50.40	90.26	92.01 / 85.74
	LSUN	23.01	95.17	94.94 / 94.78
	Places365	34.45	93.25	83.95 / 97.19
	Mean	**36.82**	**92.40**	**91.23** / **90.91**

Table 2: **Detailed results on SCOOD benchmark (Yang et al., 2021) using CIFAR-10/ResNet-18.** AMP performs very close to methods that use outlier exposure, while outperforming all the baselines that do not. We use results for baselines as reported in (Yang et al., 2021)

Method	Dataset	FPR95 ↓	AUROC ↑	AUPR(In/Out) ↑
ODIN	Texture	79.47	77.92	86.69 / 62.97
	SVHN	90.33	75.59	65.25 / 84.49
	CIFAR-10	81.82	77.90	79.93 / 73.39
	Tiny-ImageNet	82.74	77.58	86.26 / 61.38
	LSUN	80.57	78.22	86.34 / 63.44
	Places365	76.42	80.66	66.77 / 89.66
	Mean	**81.89**	**77.98**	**78.54** / **72.56**
EBO	Texture	84.29	76.32	85.87 / 59.12
	SVHN	78.23	83.57	75.61 / 90.24
	CIFAR-10	81.25	78.95	80.01 / 74.44
	Tiny-ImageNet	83.32	78.34	87.08 / 62.13
	LSUN	84.51	77.66	86.42 / 61.40
	Places365	78.37	80.99	68.22 / 89.60
	Mean	**81.66**	**79.31**	**80.54** / **72.82**
MCD	Texture	83.97	73.46	83.11 / 56.79
	SVHN	85.82	76.61	65.50 / 85.52
	CIFAR-10	87.74	73.15	76.51 / 67.24
	Tiny-ImageNet	84.46	75.32	85.11 / 59.49
	LSUN	86.08	74.05	84.21 / 58.62
	Places365	82.74	76.30	61.15 / 87.19
	Mean	**85.14**	**74.82**	**75.93** / **69.14**
OE	Texture	86.56	73.89	84.48 / 54.84
	SVHN	68.87	84.23	75.11 / 91.41
	CIFAR-10	79.72	78.92	81.95 / 74.28
	Tiny-ImageNet	83.41	76.99	86.36 / 60.56
	LSUN	83.53	77.10	86.28 / 60.97
	Places365	78.24	79.62	67.13 / 88.89
	Mean	**80.06**	**78.46**	**80.22** / **71.83**
UDG	Texture	75.04	79.53	87.63 / 65.49
	SVHN	60.00	88.25	81.46 / 93.63
	CIFAR-10	83.35	76.18	78.92 / 71.15
	Tiny-ImageNet	81.73	77.18	86.00 / 61.67
	LSUN	78.70	76.79	84.74 / 63.05
	Places365	73.86	79.87	65.36 / 89.60
	Mean	**75.45**	**79.63**	**80.69** / **74.10**
AMP (ours)	Texture	68.39	83.76	90.69 / 72.16
	SVHN	34.12	94.21	90.11 / 97.24
	CIFAR-10	80.47	78.74	81.36 / 74.07
	Tiny-ImageNet	80.70	78.34	86.95 / 63.03
	LSUN	83.60	76.64	85.80 / 60.63
	Places365	74.77	81.67	69.97 / 90.09
	Mean	**70.34**	**82.22**	**84.14** / **76.20**

Table 3: **Detailed results on SCOOD benchmark (Yang et al., 2021) using CIFAR-100/ResNet-18.** AMP consistently outperforms all methods including those that use outlier exposure. We use results for baselines as reported in (Yang et al., 2021)

In-dist (model)	OOD	TNR at TPR 95% ↑	AUROC ↑	Detection Acc. ↑
		MSP / ODIN / Gram Matrices / Ours		
CIFAR-10 (ResNet-34)	iSUN	44.6 / 73.2 / **97.3** / 91.8	91.0 / 94.0 / **99.1** / 98.2	85.0 / 86.5 / **96.2** / 93.8
	LSUN (R)	49.8 / 82.1 / **98.2** / 92.4	91.0 / 94.1 / **99.2** / 98.7	85.3 / 86.7 / **96.7** / 94.9
	LSUN (C)	48.6 / 62.0 / 91.7 / **98.5**	91.9 / 91.2 / 98.3 / **99.5**	86.3 / 82.4 / 94.1 / **97.0**
	TinyImgNet (R)	41.0 / 67.9 / **95.9** / 88.8	91.0 / 94.0 / **98.9** / 97.0	85.1 / 86.5 / **95.6** / 92.1
	TinyImgNet (C)	46.4 / 68.7 / 77.6 / **94.5**	91.4 / 93.1 / 96.2 / **98.7**	85.4 / 85.2 / 90.8 / **94.9**
	SVHN	50.5 / 70.3 / **95.3** / 91.2	89.9 / 96.7 / **99.0** / 98.1	85.1 / 91.1 / **95.2** / 93.7
	CIFAR-100	33.3 / 42.0 / 40.2 / **56.5**	86.4 / 85.8 / 83.6 / **90.2**	80.4 / 78.6 / 76.4 / **83.5**
CIFAR-100 (ResNet-34)	iSUN	16.9 / 45.2 / **66.2** / 48.7	75.8 / 85.5 / **94.6** / 90.2	70.1 / 78.5 / **88.3** / 82.6
	LSUN (R)	18.8 / 23.2 / **61.4** / 54.2	75.8 / 85.6 / **94.4** / 91.7	69.9 / 78.3 / **88.6** / 84.3
	LSUN (C)	18.7 / 44.1 / 43.7 / **67.8**	75.5 / 82.7 / 89.7 / **94.0**	69.2 / 75.9 / 82.4 / **86.4**
	TinyImgNet (R)	20.4 / 36.1 / **66.8** / 45.9	77.2 / 87.6 / **94.7** / 89.2	70.8 / 80.1 / **88.6** / 81.3
	TinyImgNet (C)	24.3 / 44.3 / 41.4 / **61.5**	79.7 / 85.4 / 89.7 / **92.9**	72.5 / 78.3 / 82.8 / **85.4**
	SVHN	20.3 / **62.7** / 54.5 / 56.5	79.5 / **93.9** / 92.1 / 91.9	73.2 / **88.0** / 84.9 / 83.7
	CIFAR-10	**19.1** / 18.7 / 16.9 / 17.5	77.1 / 77.2 / 74.5 / **79.9**	71.0 / 71.2 / 68.9 / **73.8**

Table 4: Detailed results on the OOD detection benchmark with ResNet-34. Note, different from the main paper we report TNR here (instead of FPR95) which is 100-FPR95, as this was used in (Sastry and Oore, 2020). We observe that AMP performs comparably to Gram Matrices, while being better on average. Our method has significant advantages on more challenging datasets like near OOD.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.