OPENPATCH: A 3D PATCHWORK FOR OUT-OF-DISTRIBUTION DETECTION SUPPLEMENTARY MATERIAL

Anonymous authors
Paper under double-blind review

1 TRAINING DETAILS

In our base experimental setup, we adopt distributed training using 4 NVIDIA v100 GPUs, where each GPU has a memory capacity of 16GB. In the following paragraphs, we denote as batch size the combined size across all GPUs. For the pre-training phase on Objaverse Deitke *et al.*(2023), we obtain point clouds by uniformly sampling 1024 points from the meshes. For all other datasets, we employ the same data pre-processing method outlined in 3DOS.

Pre-training on Objaverse. For EPN Chen *et al.*(2021) backbone, we employ the classification version proposed in Chen *et al.*(2021), with Max pooling applied for aggregating the rotation dimension. We train for 100 epochs with a batch size of 24 using the Adam Optimizer and a base learning rate of 1e-3. The learning rate is then reduced by a factor of 2 every 1000 training steps. For PointNet++ backbone, we employ the multi-scale grouping classification backbone. The network is trained for 250 epochs with a batch size of 128 using the Adam Optimizer and a base learning rate of 1e-3. The learning rate is reduced during training with a cosine annealing policy to a minimum of 1e-5.

Support set training. For the experiments involving support set training, we use the same setup described in Alliegro *et al.*(2022). EPN is trained for 250 epochs using Adam Optimizer with a base learning rate of 1e-3. We employ a cosine annealing scheduler that progressively decreases the initial learning rate to a value of 1e-5.

OpenShape Experiments. For both backbones we use off the shelf checkpoints trained on the proposed combination of 4 datasets that are publicly available from the github repos. When executing experiments that require a global embedding, like EVM Rudd *et al.*(2015) Mahalanobis Lee *et al.*(2018) and 1NN we use the embedding obtained after the projection layer. When extracting patches for OpenPatch we extract after the last transformer layer for PointBert Yu *et al.*(2022) and after the Conv5 layer for SPConv Choy *et al.*(2019). For SPConv we aggregate the extracted patches using a mean pooling operation with kernel and dilation 5 in order to reduce the number of patches extracted. We find that experimentally that this technique doesn't change significantly the results and has the effect the lowering of memory requirements

2 ADDITIONAL BENCHMARK TRACKS

The 3DOS benchmark defines two more track then the ones we presented, Synth to Synth and Real to Real, while they aren't of interest in our specific use case they offer additional tracks for evaluation to further assess our method. The two additional tracks are defined as follows:

Synthetic to Synthetic. The ShapeNetCore dataset Chang *et al.*(2015) classes are divided into three groups, with each one serving in turn as known class set while the other two define unknown classes. These setups are referred to as SN1, SN2, and SN3 and present increasing difficulty.

Real to Real. This track employs the same class sets as the Synthetic to Real track presented in the main paper, but both the support set and the test set contain real-world samples from ScanobjectNN Uy *et al.*(2019).

Pre-train Objaverse-LVIS	EPN Chen *et al.(2021)*								PointNet++ Qi *et al.(2017)*							
	SN1 (hard)		SN2 (med)		SN3 (easy)		Avg		SR3 (easy)		SR2 (med)		SR1 (hard)		Avg	
Method	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓
1NN	82.3	62.6	84.2	49.5	89.3	43.4	85.3	51.8	68.4	93.8	80.7	57.4	88.6	46.1	79.2	65.8
EVM Rudd *et al.* (2015)	82.7	62.1	84.4	39.8	91.3	29.7	86.1	43.9	78.2	81.1	73.0	62.6	85.9	46.3	79.0	63.3
Mahalanobis Lee *et al.(2018)*	78.5	65.8	88.3	36.2	93.1	44.9	86.6	49.0	66.9	94.2	83.5	56.9	90.7	45.4	80.4	65.5
OpenPatch	82.6	61.5	87.9	38.1	93.3	23.2	**87.9**	**40.9**	74.2	77.4	86.8	46.6	94.0	22.5	**85.0**	**48.8**
Pre-train Objaverse-LVIS	EPN Chen *et al.(2021)*								PointNet++ Qi *et al.(2017)*							
	SN1 (hard)		SN2 (med)		SN3 (easy)		Avg		SR3 (easy)		SR2 (med)		SR1 (hard)		Avg	
Method	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓
1NN	63.3	97.8	60.9	85.6	62.1	85.6	62.1	89.7	55.9	94.0	68.8	79.6	64.0	77.5	62.9	83.7
EVM Rudd *et al.* (2015)	66.3	87.8	73.9	81.1	78.1	72.0	72.8	80.3	68.3	74.7	72.8	76.1	66.0	86.0	69.0	78.9
Mahalanobis Lee *et al.(2018)*	55.1	89.6	46.2	98.3	55.9	98.2	52.4	95.4	61.2	85.6	63.6	90.9	52.6	93.5	59.1	90.0
OpenPatch	83.3	65.9	70.3	92.1	66.6	84.1	**73.4**	**80.7**	83.8	58.6	70.0	95.6	63.7	93.0	**72.5**	**82.4**

Table 1: Results on the Synthetic to Synthetic and Real to Real 3DOS Benchmark tracks Alliegro *et al.*(2022). For the Synthetic benchmark the column title indicates the chosen known class set (with the other two serving as unknown) for the Real to Real case the column title indicates the unknown class set (with the other two serving as known).

Results: The results from Tab. 1 confirm our hypothesis presented in the main paper, when closed set training is not possible OpenPatch performs best overall.

2.1 Distance metric

When applying distance based methods we extract the features from the penultimate layer if the network was trained using a cross entropy loss (SPCONV and PN2 experiments) or from the output of the network if it was trained using a contrastive loss (OpenShape and SimCLR experiments), this is in line with the evaluation proposed in the 3DOS benchmark Alliegro *et al.* (2022).

A natural extension to these methods is evaluating different types of distance metrics, mainly we test cosine similarity as it is often used in similar tasks. When testing backbones trained with cross entropy the change of metric incurred with very minimal/no improvements, while when using models trained with OpenShape and models trained on the closed set this strategy incurred in a loss of performance. Due to the unclear nature of these results we opt to report only the results with the l2 metric as it conforms to the benchmark protocols proposed in 3DOS Alliegro *et al.* (2022), and in general yields more stable results.

2.2 Unsupervised Pretraining

Another interesting type of pretraining we experiment on is unsupervised pretrainings using the SimCLR Chen *et al.* 2020 method. We train both on complete Objaverse and Objaverse-lvis. We use Adam Optimizer con lr scheduler 1e-4, cosine scheduler, train for 200 epochs with a warm up of 25 epochs, we use an effective batch size of 256. In Tab. 2 we report the results on the Synth to Real benchmark, we can confirm our hypothesis that a stronger pre-training can improve further the performance of our method and that OpenPatch is the overall best choice when using non-finetuned models. With these experiments we demonstrate that supervision isn't required for improving on the task, simply adding more data can create further improve results.

Comparison pre-training strategies - PointNet++ Chen *et. al* 2021							
		Synth to Real SR1		Synth to Real SR2		Avg	
Pre-Training	*Method*	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓
CE Obj-LVIS	1NN	60.1	90.7	57.5	87.8	58.8	89.3
	Mahalanobis Lee *et al.*(2018)	60.7	92.2	61.3	87.7	61.0	90.0
	EVM Rudd *et al.*(2015)	61.2	92.5	58.2	85.7	59.7	**89.1**
	OpenPatch	69.9	86.7	63.3	92.2	**66.6**	89.5
SimCLR Obj-LVIS	1NN	60.8	90.2	60.3	88.2	60.6	89.2
	Mahalanobis Lee *et al.*(2018)	61.9	91.3	63.7	86.6	62.8	**88.9**
	EVM Rudd *et al.*(2015)	55.7	94.4	60.7	87.2	58.2	90.8
	OpenPatch	68.1	87.0	60.7	92.9	**64.4**	90.0
SimCLR Obj-All	1NN	61.0	87.6	64.3	85.4	62.6	**86.5**
	Mahalanobis Lee *et al.*(2018)	57.1	95.8	65.5	83.3	61.3	89.6
	EVM Rudd *et al.*(2015)	60.0	94.8	65.6	84.7	62.8	89.8
	OpenPatch	67.9	90.8	65.8	85.6	**66.8**	88.2

Table 2: Results for fine-tuning free methods with different pre-training of the feature extractor.

2.3 Class Overlap

A cause of concern in OOD tasks is the possibility of data leakage of the training phase as seeing the test data at training time can cause inflated results. This isn't a big problem in our analysis due to the fact that we compare methods that are all based on the same exact pre-training and architectures meaning that even if there is some unfair advantage, at testing time this should be shared between all presented methods. By using the Synth to Real benchmark track presented in 3DOS we have a guarantee that there are no leakage of train or test samples between pre-training phase and testing

Pre-train Objaverse-LVIS no overlap	EPN Chen *et al.(2021)*					
	SR1		SR2		Avg	
Method	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓
1NN	62.6	88.3	61.6	88.8	62.1	88.6
EVM Rudd *et al.*(2015)	64.0	87.8	57.7	86.5	60.9	87.2
Mahalanobis Lee *et al.*(2018)	63.2	92.5	62.5	87.5	62.8	90.0
OpenPatch	70.8	80.0	63.4	89.0	**67.1**	**84.5**

Table 3: Results for Synth to Real benchmark when class overlap is removed from the pre-training dataset

phase, nonetheless we create an additional experiment for further assessing the performance of our method in the extreme case where no semantic leakage is present between test and pre-training phases. We recreate the pre-training on Objaverse-LVIS Deitke *et al.*(2023) as previously presented in the main paper but we eliminate from the training set all classes that are equal or semantically related to the classes of both the support set and the test set. In Tab 3 we report the results with the EPN backbone Chen *et al.*(2021), while overall performances decrease slightly OpenPatch remains the best choice for the task. The drop in performance could be attributed for the most part on the lower number of samples used for the pretraining phase.