Extremely Simple Activation Shaping for Out-of-Distribution Detection

Andrija Djurisic
ML Collective
andrija.m.djurisic@gmail.com

Nebojsa Bozanic
ML Collective
strance@gmail.com

Arjun Ashok
ML Collective
arjun.ashok.psg@gmail.com

Rosanne Liu
Google Research, Brain Team
ML Collective
rosanneliu@google.com

Abstract

The separation between training and deployment of machine learning models implies that not all scenarios encountered in deployment can be anticipated during training, and therefore solely relying on better training data or methods has its limits. Out-of-distribution (OOD) detection is an important area that stress-tests a model's ability to handle unseen situations in deployment: do models know when they don't know? Existing OOD detection methods either incur extra training steps, additional data or make nontrivial modifications of the trained network. In contrast, in this work, we propose an extremely simple, post-hoc, activation shaping method, **ASH**, where a large portion (e.g. 90%) of a sample's activation at a later layer is removed, and the rest (e.g. 10%) simplified or lightly adjusted. The shaping is applied at inference time, on-the-fly, and does not require a global threshold calculated from training or test data . Experiments show that such a simple treatment enhances in-distribution and out-of-distribution sample distinction so as to allow state-of-the-art OOD detection on CIFAR-10, CIFAR-100 and ImageNet, and does not noticeably deteriorate the in-distribution accuracy.

1 Introduction

Machine learning works by iteration. We develop better and better training techniques (validated in a closed-loop validation setting) and once a model is trained, we observe problems, shortcomings, pitfalls and misalignment in deployment, which drive us to go back to modify or refine the training process. However, as we are entering an era of large models, recent progress is driven heavily by the advancement of scaling, seen on all fronts including the size of models, data, physical hardware as well as team of researchers and engineers. As a result, it is getting more difficult to perform multiple iterations in the usual train-deployment interaction loop; for that reason *post hoc* methods that improve model capability *without* the need of further training are greatly preferred. Methods like zero-shot learning [19], plug-and-play controlling [2], as well as feature post processing make use of general and flexible large models combined with post hoc operations to create more adaptive models in various applications.

The out-of-distribution (OOD) generalization failure is one of such pitfalls often observed in deployment. The central question around OOD detection is "Do models know that they don't know?" Ideally, NNs after sufficient training should produce low confidence or high uncertainty measures for data outside of the training distribution. However, that's not always the case [18]. Differentiating out-of-distribution (OOD) from in-distribution (ID) samples proves to be a much harder task than

Preprint. Under review.

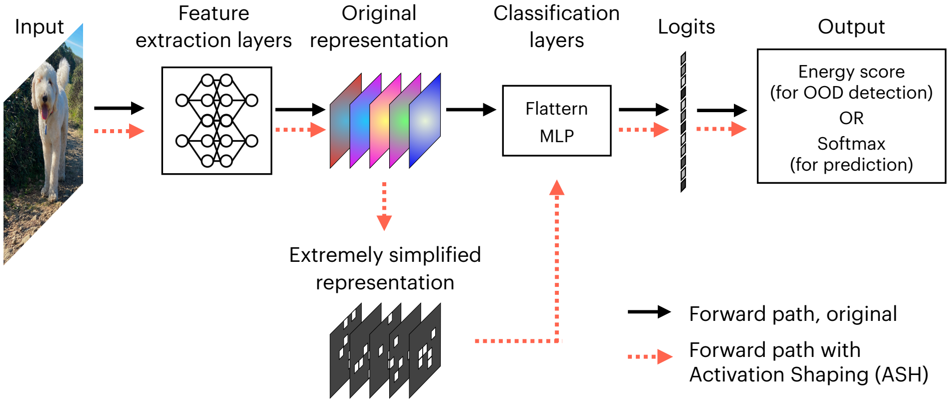
Input
Feature extraction layers
Original representation
Classification layers
Flattern
MLP
Logits
Output
Energy score (for OOD detection)
OR
Softmax (for prediction)
Extremely simplified representation
Forward path, original
Forward path with Activation Shaping (ASH)

Figure 1: Overview of the Activation Shaping (ASH) method applied to the forward path of an input sample. Black arrows indicate the regular forward path. Red dashed arrows indicate our proposed ASH path, adding one additional step to remove a large portion of the feature representation and simply or lightly adjusted the remaining, before routing back to the rest of the network. Note: we default to using the energy score calculated from logits for OOD detection, but the softmax score can also be used for OOD, and we have tested that in our ablation study.

expected. Many attribute the failure of OOD detection to neural networks being poorly calibrated. Many work have been attempting to improve calibration [5, 16]. With all these efforts OOD detection has progressed vastly, however there's still room to establish a Pareto frontier that offers the best OOD detection and ID accuracy tradeoff.

A recent work, ReAct [21], observed that the unit activation patterns of a particular (penultimate) layer show significant difference between ID data and OOD data, and hence proposed to rectify the activations at an upper limit—in other words, clipping the layer output at a certain value drastically improves the separation of ID and OOD data. A separate work, DICE [23], employs weight sparsification on a certain layer, and when combined with ReAct, achieves state-of-the-art on OOD detection on a number of benchmarks. Similarly, in this paper, we tackle OOD detection by making slight modifications to a certain layer of a pretrained network, assuming no knowledge of training or test data distributions. We show that an unexpectedly effective, and new state-of-the-art OOD detection can be achieved by a post hoc, one-shot *simplification* operation applied to input representations.

The extremely simple **A**ctivation **SH**aping (**ASH**) method takes an input's feature representation (usually from a late layer) and perform a two-stage operation: 1) remove a large portion (e.g. 80%) of the activations based on a simple top-K criterion, and 2) adjust the remaining (e.g. 20%) activation values by scaling them up, or simply assigning them a constant value. The resulting, simplified representation is then populated throughout the rest of the network, generating scores for classification and OOD detection as usual. Figure 1 illustrates this process.

The hypothesis is that overparameterized neural networks produce excessive representations for inputs, which are likely largely redundant for the task at hand. In particular, to distinguish between ID and OOD samples, the full representations learned from a trained network are not as effective as their *simplified, cleaned* counterparts.

ASH is similar to ReAct [21] in its post-training, one-shot manner taken in the activation space in the middle of a network, and uses the energy score for OOD detection. And similar to DICE [23], it performs a sparsification operation. However, we offer a number of advantages compared to ReAct: no global thresholds calculated from training data, and therefore completely *post hoc*; more flexible in terms of layer placement; better OOD detection performances across the board; better accuracy preservation on ID data, and hence establishing a much better Pareto frontier. As to DICE, we make no modification of the trained network whatsoever, and only operates in the activation space. Additionally, our method is plug-and-play, and can be combined with other existing methods, including ReAct (results shown in Table 4).

In the rest of the paper we develop and evaluate ASH via the following contributions:

1. We propose an extremely simple, easy to implement, post hoc and one-shot activation reshaping method, ASH. We extensively evaluate ASH on a suite of OOD detection tasks, where the ID datasets include CIFAR-10, CIFAR-100 and ImageNet-1k, and OOD datasets span 10 vision tasks (Section 2; Figure 1).
2. ASH immediately improves OOD detection performances across the board on all datasets, establishing a new state of the art, meanwhile best retains the ID classification accuracy, proving the optimal ID-OOD trade-off among all competitive methods in the literature establishing the new Pareto frontier (Section 3; Figure 2).
3. We perform detailed ablation analyses on different design choices of the method, and discuss what the unexpected effectiveness of such a simple operation implies about the potential excessiveness of neural network representations (Section 4).

2 Method and Experimental Setup

A trained network converts raw input data (e.g. RGB pixel values) into useful representations (e.g. a stack of spatial activations). Modern networks are highly over-parameterized. We argue that representations produced are excessive for the task at hand, and therefore could be largely simplified without much deterioration on the original performance (e.g. classification accuracy), and with a surprising gain on other tasks (e.g. OOD detection).

The proposed activation shaping (ASH) method simplifies an input's feature representation with the following recipe:

1. Remove a large portion of the activation by setting any activation value lower than the threshold t to be 0. The threshold t is picked to reflect the p-th percentile of the entire representation.
2. For the remaining $(1 - p)\%$ of activation values, apply one of the three treatments:
    - **ASH-P** (Algorithm 1): Do nothing. **P**runing is all we need. This is used as a baseline to show the gains of the following two treatments.
    - **ASH-B** (Algorithm 2): Assign all of them to be the same positive constant so the entire representation is **B**inary.
    - **ASH-S** (Algorithm 3): **S**cale their values up by a ration calculated the mass of activation values before and after pruning.

After the ASH treatment on an input sample's feature representation at an intermediate layer, it continues down the forward path throughout the rest of the network, as seen in Figure 1. To classify the sample, we calculate the Softmax probabilities of the logits as usual. In the case of OOD detection, a score function is used to convert the model output (logits) into a scalar, after which a thresholding mechanism is applied to distinguish between ID and OOD samples. Commonly used OOD scoring functions include the softmax confidence [], ODIN [], and the energy score, introduced in [] and adopted by ReAct [21] in the large-scale OOD detection setting.

Algorithm 1 ASH-P - Activation shaping with pruning

```
Input: $x, p$
Calculate the $p$-th percentile of the input sample $x \rightarrow$ threshold $t$
Set every value in $x$ less then threshold $t$ to $0.0$
Return $x$
```

In our experiments, we default to adopting the energy score following the setting of ReAct. In the ablation study we demonstrated the effectiveness of ASH when combined with other score functions, as well as with ReAct.

We perform extensive experiments on a suite of OOD detection tasks, on both moderate scale CIFAR benchmarks [11], adopted by most of the OOD detection methods, namely, MSP [7], Mahalanobis [12], and Energy [15], which we make comprehensive comparisons to, as well as the large-scale ImageNet benchmark [10], following ReAct [21].

Algorithm 2 ASH-B - Activation shaping by binarizing

```
Input: $x$, $p$
Calculate the sum of the input sample $x \to s$
Calculate the $p$-th percentile of the input sample $\to$ threshold $t$
Set every value in $x$ less then threshold $t$ to 0.0
Calculate the number of unpruned activations in $x \to n$
Set every non-zero value in $x$ to $s/n$
Return $x$
```

Algorithm 3 ASH-S - Activation shaping with scaling

```
Input: $x$, $p$
Calculate the sum of the input sample $x \to s1$
Calculate the $p$-th percentile of the input sample $\to$ threshold $t$
Set every value in $x$ less then threshold $t$ to 0.0
Calculate the sum of unpruned activations in $x \to s2$
Multiply every non-zero value in $x$ with $e^{\frac{s1}{s2}}$
Return $x$
```

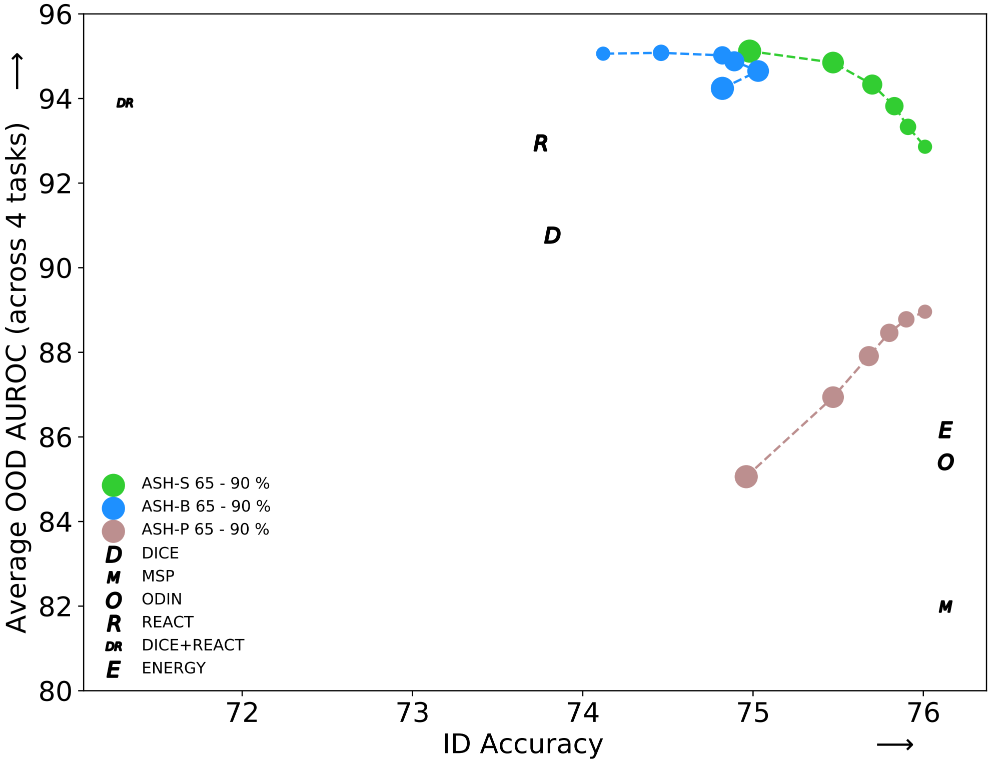

Figure 2: **Accuracy-detection trade-off on ImageNet.** Average OOD detection rate (AUROC; averaged across 4 OOD datasets) vs ID classification accuracy (Top-1 accuracy in percentage on ImageNet-1k validation set) of all OOD detection methods and their variants used in this paper. "Energy" indicates the upper bound of ID accuracy since it makes no modification of the network or the features, but just calculates energy scores from logits. ReAct improves on OOD detection significantly, but comes with an accuracy drop. ASH methods combined with ODIN or Softmax do not work as well. Our proposed ASH versions (dots) offers the best trade-off and form a Pareto front.

CIFAR experimental setup. For CIFAR-10 and CIFAR-100 experiments, we used the OOD datasets presented in DICE[23]: SVHN [17], LSUN C [28], LSUN R [28], iSUN [26], Places365

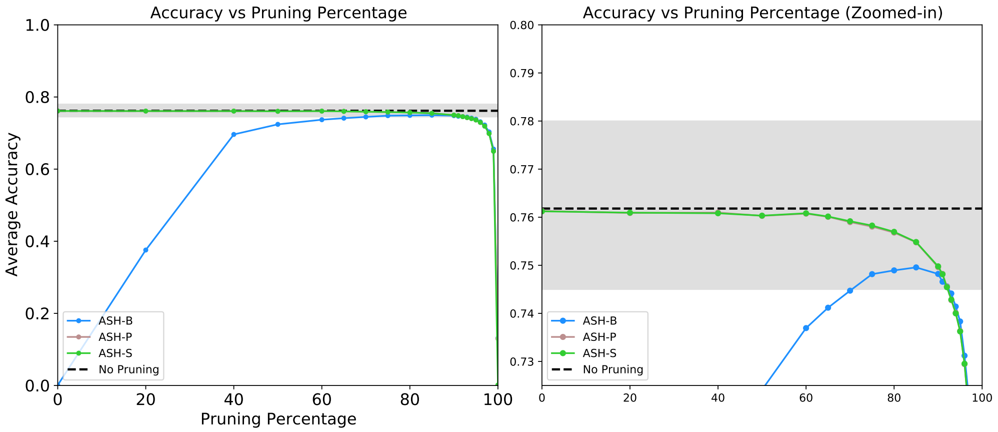

Figure 3: Accuracy degradation across pruning percentage. Shown is the penultimate layer, after Average Pooling and before the last Linear layer that generates logits.

[29] and Textures [1]. And the ID dataset is the respective CIFAR. The architecture is DenseNet-101 architecture [9]. We apply ASH after the penultimate layer, where the feature size is $342 \times 1 \times 1$.

ImageNet experimental setup. For the large-scale OOD detection, we follow the exact ImageNet setup used in ReAct, where the ID dataset is ImageNet-1k, and OOD datasets include iNaturalist [24], SUN [25], Places [29], Textures [1]. We used ResNet50 [6] and MobileNetV2 [20] network architectures. Both networks are pretrained with ID data and never modified post-training; weights frozen at the OOD detection phase. We can apply ASH at various places throughout the network, and the performance would differ. We found the most effective placements are towards the late layers of the network, for example, any convolutional layers in the last residual block of the ResNet, as well as the penultimate layer (same as ReAct). The main results shown in this paper are from applying ASH after the penultimate layer for ResNet, where the feature size is $2048 \times 1 \times 1$, and right before the classifier block in the case of MobileNet, where the feature size is $1280 \times 1 \times 1$. Results for other ASH placements are included in the Supplementary Information.

3 Activation Shaping for OOD detection

We evaluate our method using standard metrics for OOD detection task introduced in [7]: (i) AUROC: the Area Under the Receiver Operating Characteristic curve metric which is a threshold-independent performance evaluation metric, (ii) FPR95: False Positive Rate is the probability that a negative (OOD) example is misclassified as positive (ID) when the true positive rate (TPR) is as high as 95% [14] and (iii) AUPR: Area Under the Precision-Recall curve. In addition to OOD metrics, we also evaluate each method on their ID accuracy performance, which is the classification accuracy on in-distribution data, e.g. Top-1 accuracy on the ImageNet validation set.

3.1 On OOD detection for ImageNet

ASH is highly effective at OOD detection. Our ImageNet results in Table 1 indicate that when compared with competitive OOD detection methods in the literature, including the previous state-of-the-art ReAct, our method establishes the new state-of-the-art (SOTA), across almost all OOD datasets and evaluation metrics. All three versions of the algorithm (differing in only the treatments to the un-pruned activation values) perform similarly, outperforming existing methods by a large margin. In the table, ASH-P is with a percentile threshold $p = 60\%$, that is, 60% of lower value activations for every image are removed. ASH-B is using $p = 65\%$, and ASH-S $p = 90\%$. The fact that we can remove 90% of activations in the penultimate layer and produce SOTA OOD detection performances on many datasets is a surprising observation.

Figure 2 displays performances across different percentile thresholds. Variants within the same method with only different pruning thresholds are connected by a dashed line. We compare the three versions of our proposed method (ASH-P, ASH-B, ASH-S), each with various pruning thresholds, against benchmarks like ReAct (previous SOTA) and Energy (best ID accuracy). We also include ablations (e.g. ASH with ODIN) for further comparison. We can see that all variants of ASH that include a two-stage operation (ASH-B and ASH-S) perform well on OOD detection. The pruning-only ASH-P falls behind ASH-S, suggesting that simply by scaling up un-pruned activations, we obtain a great performance gain at OOD detection.

3.2 On accuracy preservation for ImageNet

Most OOD detection methods come with the cost of deteriorating classification accuracies on the original ID dataset. Figure 2 shows that ReAct [21], the previous SOTA, comes with a drop of ID accuracy from 76.13% to 74.82% (Top-1 on ImageNet validaton set). Our method, on the other hand, tend to preserve the ID accuracy. Towards the low end of pruning percentage (65%) both ASH-S and ASH-P come with only a slight drop of ID accuracy (76.01%). The more we prune, the larger drop of ID accuracy is seen in ASH-S and ASH-P. However, the opposite trend is observed in ASH-B: the best accuracy is preserved when 80% or 90% of activations are pruned.

Taking a closer look at the accuracy preservation with regards to pruning rate, shown in Figure 3, we realize that ASH-B is indeed following the opposite trend in the middle region of pruning (50%-90%). The reason is that the rather extreme binarizing operation in ASH-B (setting all remaining activations to a constant) has a bigger impact when the pruning rate is lower (more values are being modified). To the extreme of 0% pruning, ASH-B simply sets all activation values to their average, which prompts a drastic accuracy drop to 0 (Figure 3, left end). Therefore, it is not so baffling that ASH-B exhibits a reversed trend compared to ASH-S and ASH-P.

We also notice that the effects of ASH-P and ASH-S on ID accuracy match almost exactly, validating that scaling up activations does not change the network output. Connecting back to Figure 2, though, we learned that the simple scaling generates a large profit at OOD detection.

		OOD Datasets									
Model	**Methods**	**iNaturalist**		**SUN**		**Places**		**Textures**		**Average**	
		FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
ResNet	MSP [7]	54.99	87.74	70.83	80.86	73.99	79.76	68.00	79.61	66.95	81.99
	ODIN [14]	47.66	89.66	60.15	84.59	67.89	81.78	50.23	85.62	56.48	85.41
	Mahalanobis [12]	97.00	52.65	98.50	42.41	98.40	41.79	55.80	85.01	87.43	55.47
	Energy [15]	55.72	89.95	59.26	85.89	64.92	82.86	53.72	85.99	58.41	86.17
	ReAct [21]	20.38	96.22	24.20	94.20	33.85	91.58	47.30	89.80	31.43	92.95
	Dice [23]	25.63	94.49	35.15	90.83	46.49	87.48	31.72	90.30	34.75	90.77
	Dice + React [23]	18.64	96.24	25.45	93.94	36.86	90.67	28.07	92.74	27.25	93.40
	ASH-P (Ours)	44.57	92.51	52.88	88.35	61.79	85.58	42.06	89.70	50.32	89.04
	ASH-B (Ours)	14.21	97.32	**22.08**	**95.10**	**33.45**	**92.31**	21.17	95.50	**22.73**	95.06
	ASH-S (Ours)	**11.49**	**97.87**	27.98	94.02	39.78	90.98	**11.93**	**97.60**	22.80	**95.12**
MobileNet	MSP [7]	64.29	85.32	77.02	77.10	79.23	76.27	73.51	77.30	73.51	79.00
	ODIN [14]	55.39	87.62	54.07	85.88	57.36	84.71	49.96	85.03	54.20	85.81
	Mahalanobis [12]	62.11	81.00	47.82	86.33	52.09	83.63	92.38	33.06	63.60	71.01
	Energy [15]	59.50	88.91	62.65	84.50	69.37	81.19	58.05	85.03	62.39	84.91
	ReAct [21]	42.40	91.53	47.69	88.16	**51.56**	86.64	38.42	91.53	45.02	89.47
	Dice [23]	43.09	90.83	38.69	90.46	53.11	85.81	32.80	91.30	41.92	89.60
	Dice + React [23]	41.61	89.91	38.81	90.45	54.05	84.23	19.72	95.91	38.55	90.12
	ASH-P (Ours)	54.92	90.46	58.61	86.72	66.59	83.47	48.48	88.72	57.15	87.34
	ASH-B (Ours)	**31.46**	**94.28**	**38.45**	**91.61**	51.80	**87.56**	20.92	95.07	**35.66**	**92.13**
	ASH-S (Ours)	39.10	91.94	43.62	90.02	58.84	84.73	**13.12**	**97.10**	38.67	90.95

Table 1: **Main ImageNet results**. We follow the exact same metrics and format as ReAct [21]. Both ResNet and MobileNet are trained with ID data (ImageNet-1k) only. ↑ indicates larger values are better and ↓ indicates smaller values are better. All values are percentages. Rows except for those indicated **Ours** are taken directly from the Table 1 in ReAct. ASH consistently perform better than benchmarks, across all the OOD datasets. The rightmost two rows are averaged metrics across all 4 OOD datasets. All results except those noted as "Ours" are directly taken from ReAct[21].

3.3 On CIFAR results

CIFAR-10: SOTA CIFAR-100: SOTA

Method	CIFAR-10		CIFAR-100	
	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
MSP	48.73	92.46	80.13	74.36
ODIN	24.57	93.71	58.14	84.49
GODIN	34.25	90.61	52.87	85.24
Mahalanobis	31.42	89.15	55.37	82.73
Energy	26.55	94.57	68.45	81.19
ReAct	26.45	94.95	62.27	84.47
DICE	$20.83^{\pm 1.58}$	$95.24^{\pm 0.24}$	$49.72^{\pm 1.69}$	$87.23^{\pm 0.73}$
ASH-P (Ours)	23.45	95.22	64.53	82.71
ASH-B (Ours)	20.23	96.02	48.73	88.04
ASH-S (Ours)	**15.05**	**96.61**	**41.40**	**90.02**

Table 2: Cifar results. ↑ indicates larger values are better and ↓ indicates smaller values are better. All values are percentages. Results are averaged accross 6 OOD datasets.

4 Discussion

ASH as post hoc regularization. ASH can be thought of as a simple "feature cleaning" step, or a post hoc regularization of features. As we all now acknowledge that neural networks are overparameterized, consequently, we can think of the representation learned by such networks are likely "over represented." While the power of overparameterization shines mostly through making the training easier—adding more dimensions to the objective landscape while the intrinsic dimension of the task at hand remains constant [13], we argue that from the lens of representation learning, overparameterized networks "overdo" feature representation, i.e. the representation produced of an input contain too much redundancy. We conjecture that a simple feature cleaning step post training can help ground the resulting learned representation better. Future work to validate, or invalidate this conjecture would include testing if simplified or regularized representation works well in other problem domains, from generalization, to transfer learning and continual learning.

Connection to a modified ReLU Another lens to interpret ASH with is that it is simply a modified ReLu function, adapted on-the-fly per input. Since we operate on feature activations pre-ReLU, as shown in Figure 4, the most basic version, ASH-P, combined with the subsequent ReLU fuction, becomes simply an adjusted ReLU. Since the cut-off is determined per-input and on-the-fly, it is essentially an data-dependent activation function. The success of ASH highlights the need to look into more flexible, adaptive, data-dependent activation functions at inference.

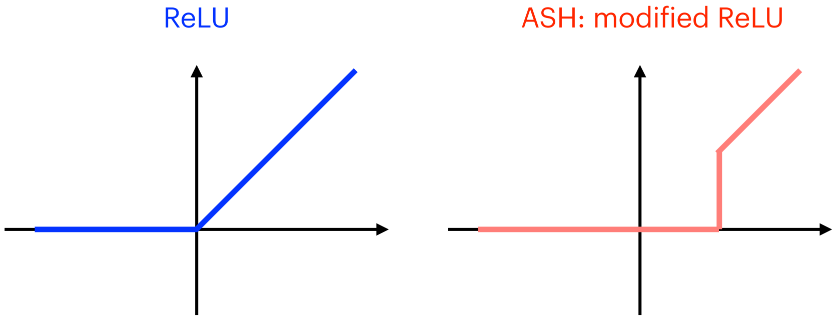

Figure 4: Comparison of a regular ReLU activation funciton (left) with a modified ReLU (right), which is equivalent to the ASH-P operation.

Where to ASH? Since ASH deals with an input's activation tensor, it can be applied at different locations of a network. In practice we found it most impactful at the later layers of a network, since that's where feature formation becomes stable. In Figure 5 we show the effect of performing ASH-P on different layers of a network—in this case a ResNet-50 on ImageNet. We can see that the accuracy deterioration over pruning rate becomes more severe as we move to earlier parts of the network.

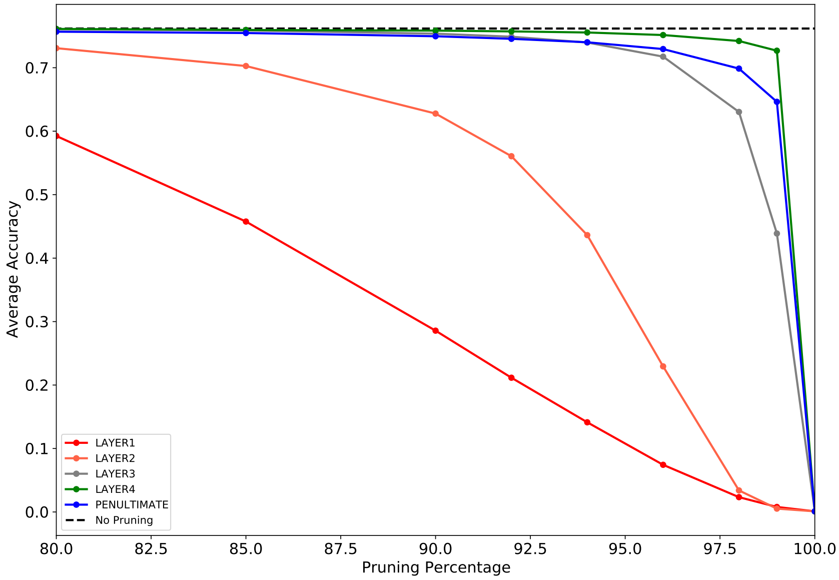

Figure 5: Accuracy (Top-1 ImageNet validation) degradation across pruning percentage, when ASH is applied to different layers throughout the network: the last convolutional layer of the third block, the last block of a ResNet-50, and the penultimate layer. Earlier layers see a more severe accuracy degradation.

Magnitude vs value. Lots of existing pruning methods rely on the *magnitude* of a number (weights or activations). However in this work we use the direct values. That is, a large negative will be pruned if it is within the p-percentile of value distribution. The reason is that we operate on activations either *before ReLU*, in which case all negative values will be removed subsequently, or on the penultimate layer where all values are already non-negative.

ASH-RAND: randomizing remaining values. With ASH-B and ASH-S we experimented different treatments on un-pruned activation values. What if we simply set them to random values? Table 4 shows that it still works!

		CIFAR-10			CIFAR-100			ImageNet	
Method	**FPR95** ↓	**AUROC** ↑	**AUPR** ↑	**FPR95** ↓	**AUROC** ↑	**AUPR** ↑	**FPR95** ↓	**AUROC** ↑	**AUPR** ↑
Softmax score	48.69	92.52	80.75	80.06	74.45	76.99	64.76	82.82	95.94
Softmax score + ASH	48.86	92.61	77.03	76.04	75.00	78.61	37.86	90.90	97.97
Energy score	26.59	94.63	95.61	68.29	81.23	83.64	57.47	87.05	97.15
Energy score + ASH	15.05	96.61	96.88	41.40	90.02	91.23	22.80	95.12	98.90

Table 3: Compatibility table. ↑ indicates larger values are better and ↓ indicates smaller values are better. All values are percentages. CIFAR10 and CIFAR100 results are averaged across 6 different OOD tasks and ImageNet results are averaged across 4 different OOD tasks.

5 Related Work

OOD Detection OOD detection came under the spotlight once it was shown that not-seen-during-training samples can have extreme overconfidence [18], specially in domains where failure to classify

	ImageNet benchmark			
Method	**FPR95** ↓	**AUROC** ↑	**AUPR** ↑	**ID ACC** ↑
ASH-RAND@65	45.37	90.80	98.09	72.16
ASH-RAND@70	46.93	90.67	98.05	72.87
ASH-RAND@75	46.93	90.67	98.05	73.19
ASH-RAND@80	51.24	89.94	97.89	73.57
ASH-RAND@90	59.35	87.88	97.44	73.51
ASH-B@65	22.73	95.06	98.94	74.12
ASH-S@90	22.80	95.12	98.90	74.98

Table 4: We experiment another variant of ASH: setting un-pruned activations to random numbers between 0 and 10. All methods are based on a model trained on ID data only (ImageNet-1k), without using any auxiliary outlier data. ↑ indicates larger values are better and ↓ indicates smaller values are better. All values are percentages. The result is comparable to, although not better than, ASH-B and ASH-S.

	Local threshold			Global threshold		
Method	**FPR95** ↓	**AUROC** ↑	**AUPR** ↑	**FPR95** ↓	**AUROC** ↑	**AUPR** ↑
ASH-B@99	45.43	89.09	97.55	41.70	89.75	97.55
ASH-B@98	39.59	91.22	98.08	41.64	90.57	97.87
ASH-B@97	36.54	92.17	98.30	42.73	90.57	97.88
ASH-B@96	34.26	92.79	98.44	43.55	90.43	97.85
ASH-B@95	32.64	93.20	98.52	44.38	90.20	97.79
ASH-B@94	31.09	93.51	98.59	45.39	89.99	97.75
ASH-B@93	30.03	93.76	98.64	45.77	89.86	97.72
ASH-B@92	29.01	93.95	98.68	46.29	89.71	97.68
ASH-B@91	28.43	94.10	98.71	46.97	89.55	97.64
ASH-B@90	27.58	94.24	98.74	48.25	89.39	97.61
ASH-B@85	25.26	94.65	98.83	51.04	88.77	97.48
ASH-B@80	24.04	94.88	98.88	53.63	88.24	97.37
ASH-B@75	22.95	95.02	98.91	56.66	87.63	97.25
ASH-B@70	22.39	95.08	98.93	60.19	86.93	97.12

Table 5: Local vs global threshold. ↑ indicates larger values are better and ↓ indicates smaller values are better. All values are percentages. All are reproduced by us.

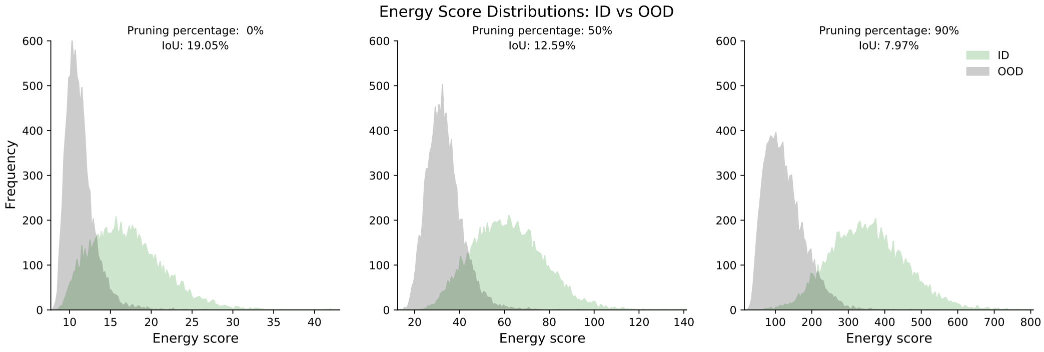

Figure 6: distributions, to do: make text larger

correctly would cause the safety risk. Many other studies worked on differentiating in-distribution (ID) and out-of-distribution (OOD) by assessing OOD uncertainty, which is later being used to recalibrate poorly calibrated NNs [27]. The most competitive ones find different ways to improve the differentiation. Such methods are: ODIN [14] which has a two-step approach. First by introducing a temperature scaling, and second by introducing small input perturbations that achieves better detection by making the larger separation between softmax scores of ID and OOD datasets; Mahalanobis distance-based score [12] which introduced the confidence score using the Mahalanobis distance between the test sample and the nearest class-conditional Gaussian distribution of training samples; Energy-based score [15] which introduced an energy-bounded learning objective at training time, lower energies to the ID data, and higher energies to the OOD data are assigned by the NN , thus inducing an energy gap; and Rectification-based [21] which introduced truncation of activations at the 90 percentile at the penultimate layer.

Activation pruning or reshaping A parallel can be drawn between ASH and activation pruning. Stochastic activation pruning (SAP) [3] has been proposed as a powerful technique against adversarial attacks. SAP prunes a random subset of low-magnitude activations during each forward pass and scales up the others. This stochasticity reduces the impact an adversary has, and thereby the model preserves greater accuracy under adversarial attacks. The authors of DICE [22] propose a method to directly learn a binary mask over the weights, that preserves those weights that contribute most to the prediction function. Unlike SAP, ASH prunes activations that fall below a significant percentile, to simplify the representations. It does not aim to promote sparsity or stochasticity, but rather aims to preserve only a given range of activations to simplify detection of OOD samples. Unlike DICE, we do not aim to identifying unimportant weights by using saliency information with respect to logits. Our model is simpler, does not involve any learnable parameters, and works at a per-input level and prunes activations only based on their currently observed range.

Targeted dropout Our method can also be interpreted through the lens of a targeted dropout on activations, instead of weights. Targeted dropout [4] is a method introduced to compress overparameterized machine learning models by dropping a set of units and weights that were stochastically selected from the set of the least important parameters (i.e. with the lowest magnitude), thus predicting which unit or weight may get pruned later [8]. A neural network trained using targeted dropout is shown to be extremely robust to post-hoc pruning of units and weights that repeatedly occur in the dropped set.

6 Conclusion

In this paper, we present ASH, an extremely simple, post hoc, on-the-fly, and plug-and-play activation shaping method applied to inputs at the inference time. ASH works by pruning a large portion of an input sample's activation and lightly adjusting the remaining. When combined with energy scores, it's shown to outperform all contemporary methods for OOD detection, on both common benchmark and large-scale image classification benchmarks. The extensive experimental setup on 3 ID datasets, 10 OOD datasets, and performance evaluated on 4 metrics, demonstrates the effectiveness of ASH across the board: reaching SOTA on OOD detection while providing the best trade-off between OOD detection and ID classification accuracy.

The unexpected effectiveness of ASH suggests that our overparameterized networks likely overdo representation learning—generating features for data that are largely redundant for the optimization task at hand. It is both an advantage and a peril: on the one hand the representation is less likely to overfit to a single task and might retain more potential to generalize, but on the other hand it serves a poorer discriminator between data seen and unseen.

References

[1] Mircea Cimpoi et al. "Describing textures in the wild". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 3606–3613.

[2] Sumanth Dathathri et al. "Plug and Play Language Models: A Simple Approach to Controlled Text Generation". In: *International Conference on Learning Representations*. 2020. URL: `https://openreview.net/forum?id=H1edEyBKDS`.

[3] Guneet S Dhillon et al. "Stochastic activation pruning for robust adversarial defense". In: *arXiv preprint arXiv:1803.01442* (2018).

[4] Aidan N Gomez et al. "Learning sparse networks using targeted dropout". In: *arXiv preprint arXiv:1905.13678* (2019).

[5] Chuan Guo et al. "On calibration of modern neural networks". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1321–1330.

[6] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[7] Dan Hendrycks and Kevin Gimpel. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *Proceedings of International Conference on Learning Representations* (2017).

[8] Torsten Hoefler et al. "Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks". In: *Journal of Machine Learning Research* 22.241 (2021), pp. 1–124.

[9] Gao Huang et al. "Densely Connected Convolutional Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2261–2269. DOI: `10.1109/CVPR.2017.243`.

[10] Rui Huang and Yixuan Li. "Mos: Towards scaling out-of-distribution detection for large semantic space". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8710–8719.

[11] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[12] Kimin Lee et al. "A simple unified framework for detecting out-of-distribution samples and adversarial attacks". In: *Advances in neural information processing systems* 31 (2018).

[13] Chunyuan Li et al. "Measuring the Intrinsic Dimension of Objective Landscapes". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: `https://openreview.net/forum?id=ryup8-WCW`.

[14] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. "Enhancing the reliability of out-of-distribution image detection in neural networks". In: *arXiv preprint arXiv:1706.02690* (2017).

[15] Weitang Liu et al. "Energy-based Out-of-distribution Detection". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[16] Matthias Minderer et al. "Revisiting the calibration of modern neural networks". In: *Advances in Neural Information Processing Systems* 34 (2021).

[17] Yuval Netzer et al. "Reading digits in natural images with unsupervised feature learning". In: (2011).

[18] Anh Nguyen, Jason Yosinski, and Jeff Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images". In: 2015, pp. 427–436.

[19] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763.

[20] Mark Sandler et al. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[21] Yiyou Sun, Chuan Guo, and Yixuan Li. "ReAct: Out-of-distribution Detection With Rectified Activations". In: ed. by M Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 144–157. URL: `https://proceedings.neurips.cc/paper/2021/file/01894d6f048493d2cacde3c579c315a3-Paper.pdf`.

[22] Yiyou Sun and Sharon Li. "DICE: A Simple Sparsification Method for Out-of-distribution Detection". In: (2021).

[23] Yiyou Sun and Yixuan Li. “DICE: Leveraging Sparsification for Out-of-Distribution Detection”. In: *European Conference on Computer Vision*. 2022.

[24] Grant Van Horn et al. “The inaturalist species classification and detection dataset”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8769–8778.

[25] Jianxiong Xiao et al. “Sun database: Large-scale scene recognition from abbey to zoo”. In: *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE. 2010, pp. 3485–3492.

[26] Pingmei Xu et al. “Turkergaze: Crowdsourcing saliency with webcam based eye tracking”. In: *arXiv preprint arXiv:1504.06755* (2015).

[27] Jingkang Yang et al. “Generalized out-of-distribution detection: A survey”. In: *arXiv preprint arXiv:2110.11334* (2021).

[28] Fisher Yu et al. “Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop”. In: *arXiv preprint arXiv:1506.03365* (2015).

[29] Bolei Zhou et al. “Places: A 10 million image database for scene recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.6 (2017), pp. 1452–1464.