A Reproducibility

Authors of the paper recognize the importance and value of reproducible research. We summarize our efforts below to facilitate reproducible results:

1. **Dataset.** We use publicly available datasets, which are described in detail in *Section 4.2* and *Section 4.1*.
2. **Assumption and proof.** The complete proof of our theoretical contribution is provided in *Appendix E*, which supports our theoretical claims made in *Section 6*.
3. **Baselines.** The description and hyperparameters of baseline methods are specified in *Appendix B*.
4. **Model.** Our main results on ImageNet are based on ResNet50 [17] provided by Pytorch. Due to the post hoc nature of our method, this allows the research community to reproduce our numbers provided with the same model and evaluation datasets.
5. **Implementation.** The simplicity of the DICE eases the reproducibility, as it only requires a few lines of code modification in the PyTorch model specification. Specifically, one can replace the weight matrix in the penultimate layer of deep networks using the following code:

```
        threshold = numpy.percentile(V, p)
        M = V > threshold
        W_new = W * M
```

6. **Open Source.** The codebase and the dataset is available in https://github.com/deeplearning-wisc/dice.git.
7. **Hardware**: We conduct all the experiments on NVIDIA GeForce RTX 2080Ti GPUs.

B Details of Baselines

For the reader's convenience, we summarize in detail a few common techniques for defining OOD scores that measure the degree of ID-ness on a given input. By convention, a higher (lower) score is indicative of being in-distribution (out-of-distribution).
MSP [20] This method proposes to use the maximum softmax score as the OOD score.
ODIN [34] This method improves OOD detection with temperature scaling and input perturbation. In all experiments, we set the temperature scaling parameter $T = 1000$. For ImageNet, we found the input perturbation does not further improve the OOD detection performance and hence we set $\epsilon = 0$. Following the setting in [34], we set ϵ to be 0.004 for CIFAR-10 and CIFAR-100.
Mahalanobis [32] This method uses multivariate Gaussian distributions to model class-conditional distributions of softmax neural classifiers and uses Mahalanobis distance-based scores for OOD detection. We use 500 examples randomly

selected from ID datasets and an auxiliary tuning dataset to train the logistic regression model and tune the perturbation magnitude ϵ. The tuning dataset consists of adversarial examples generated by FGSM [14] with a perturbation size of 0.05. The selected ϵ's are 0.001, 0.0, and 0.0 for ImageNet-1k, CIFAR-10, and CIFAR-100, respectively.

Generalized ODIN [22] This method proposes a specialized network to learn temperature scaling and a novel strategy to choose perturbation magnitude, in order to replace manually-set hyperparameters. Our training configurations strictly follow the original paper, where we train the DeConf-C network for 200 epochs without applying the weight decay in the final layer of the Deconf classifier (notated as $h_i(x)$ in [22]). The other settings such as learning rate, momentum and training batch size are the same as ours. Note that G-ODIN has a slight advantage due to a longer training time than ours (100 epochs). We choose the best perturbation magnitude ϵ by maximizing the confidence scores on 1,000 examples randomly selected from ID datasets. The selected ϵ value is 0.02 for all (ImageNet-1k, CIFAR-10, and CIFAR-100).

Energy [36] This method proposes using energy score for OOD detection. The energy function maps the logit outputs to a scalar $E(\mathbf{x}; f) \in \mathbb{R}$, which is relatively lower for ID data. Note that [36] used the *negative energy score* for OOD detection, in order to align with the convention that $S(\mathbf{x})$ is higher (lower) for ID (OOD) data. Energy score does not require hyperparameter tuning.

ReAct [51] This method also uses energy score for OOD detection. It further truncates the internal activations of neural networks, which provides more distinctive feature patterns for OOD distributions. The truncation threshold is set with the validation strategy in [51].

C Validation Strategy

We use a validation set of `Gaussian noise` images, which are generated by sampling from $\mathcal{N}(0, 1)$ for each pixel location. The optimal p is selected from $\{0.1, 0.3, 0.5, 0.7, 0.9, 0.99\}$, which is 0.9 for CIFAR-10/100 and 0.7 for ImageNet. We also show in Figure 3 using Gaussian can already find the near-optimal one averaged over all OOD test datasets considered.

D More results on the effect of Sparsity Parameter p

We characterize the effect of sparsity parameter p on other ID datasets. In Table 6, we summarize the OOD detection performance and classification performance for DenseNet trained on CIFAR-10 and ImageNet, where we vary $p = \{0.1, 0.3, 0.5, 0.7, 0.9, 0.99\}$. A similar trend is observed on CIFAR-100 as discussed in the main paper.

Table 6. Effect of varying sparsity parameter p. Results are averaged on the test datasets described in Section 4.

Sparsity	**CIFAR-10**			**ImageNet**		
	FPR95 ↓	**AUROC** ↑	**Acc.** ↑	**FPR95** ↓	**AUROC** ↑	**Acc.** ↑
$p = 0.99$	57.57	84.29	60.81	75.79	66.07	63.28
$p = 0.9$	21.76	94.91	94.38	40.10	89.09	73.36
$p = 0.7$	21.76	94.91	94.35	34.75	90.77	73.82
$p = 0.5$	21.76	94.91	94.35	34.58	90.80	73.80
$p = 0.3$	21.75	94.91	94.35	34.70	90.69	73.57
$p = 0.1$	21.92	94.90	94.33	40.25	89.44	73.38
$p = 0$	26.55	94.57	94.50	58.41	86.17	75.20

E Variance Reduction with Correlated Variables

Extension of Lemma 2. We can show variance reduction in a more general case with correlated variables. The variance of output f_c without sparsification is:

$$\mathrm{Var}[f_c] = \sum_{i=1}^{m} \sigma_i^2 + 2 \sum_{1 \le i < j \le m} \mathrm{Cov}(v_i, v_j),$$

where $\mathrm{Cov}(\cdot, \cdot)$ is the covariance. The expression states that the variance is the sum of the diagonal of the covariance matrix plus two times the sum of its upper triangular elements.

Similarly, the variance of output *with* directed sparsification (by taking the top units) is:

$$\mathrm{Var}[f_c^{\mathrm{DICE}}] = \sum_{i=t+1}^{m} \sigma_i^2 + 2 \sum_{t < i < j \le m} \mathrm{Cov}(v_i, v_j).$$

Therefore, the variance reduction is given by:

$$\sum_{i=1}^{t} \sigma_i^2 + 2 \sum_{1 \le i < j \le m} \mathrm{Cov}(v_i, v_j) - 2 \sum_{t < i < j \le m} \mathrm{Cov}(v_i, v_j),$$

We show in Fig. 5 that the covariance matrix of unit contribution v primarily consists of elements of 0, which indicates the independence of variables by large. The covariance matrix is estimated on the CIFAR-10 model with DenseNet-101, which is consistent with our main results in Table 2.

Moreover, the summation of non-zero entries in the full matrix (i.e., the second term) is greater than that of the submatrix with top units (i.e., the third term), resulting in a larger variance reduction than in Lemma 1. In the case of OOD data (SVHN), we empirically measure the variance reduction, where $\sum_{i=1}^{t} \sigma_i^2 + 2 \sum_{1 \le i < j \le m} \mathrm{Cov}(v_i, v_j)$ equals to **6.8** and $2 \sum_{t < i < j \le m} \mathrm{Cov}(v_i, v_j)$ equals to **2.2**. Therefore, DICE leads to a significant variance reduction effect.

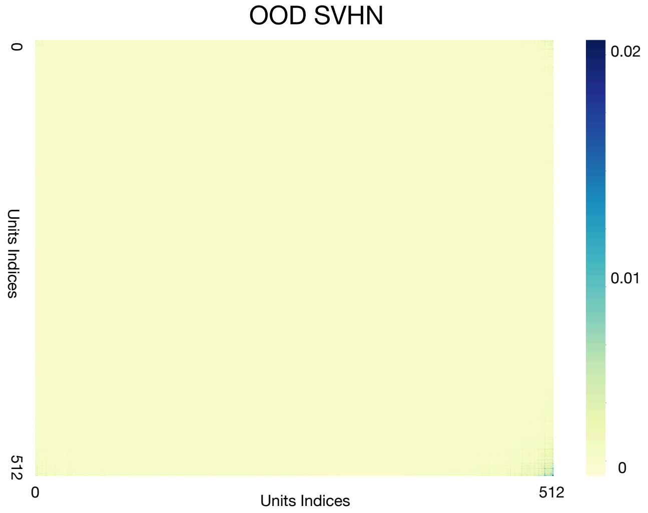

Fig. 5. Covariance matrix of unit contribution estimated on the OOD dataset SVHN. Model is trained on ID dataset CIFAR-10. The unit indices are sorted from low to high, based on the expectation value of ID's unit contribution (airplane class, same as in Figure 1). The matrix primarily consists of elements with 0 value.

F Effect of DICE on MSP

Our theory shows the variance reduction effect directly in the logit output space, which is more compatible with the energy score. As a further investigation in Table 7, we find empirically that using DICE for MSP can improve the performance for MSP though it does not yield better performance than our main results.

Table 7. Effect of applying DICE with MSP on DenseNet101 pretrained on CIFAR-10. The number is reported in FPR95.

Method	SVHN	LSUN-c	LSUN-r	iSUN	Texture	places365	Average
MSP [20]	48.25	33.80	42.37	41.42	63.99	62.57	48.73
DICE+MSP	**45.94**	**24.36**	**35.68**	**34.60**	**62.06**	**59.40**	**43.67**

G Detailed OOD Detection Performance for CIFAR

We report the detailed performance for all six test OOD datasets for models trained on CIFAR10 and CIFAR-100 respectively in Table 8 and Table 9.

Table 8. Detailed results on six common OOD benchmark datasets: `Textures` [6], `SVHN` [45], `Places365` [68], `LSUN-Crop` [66], `LSUN-Resize` [66], and `iSUN` [64]. For each ID dataset, we use the same DenseNet pretrained on **CIFAR-10**. ↑ indicates larger values are better and ↓ indicates smaller values are better.

Method Type	Method	SVHN		LSUN-c		LSUN-r		iSUN		Textures		Places365		Average	
		FPR95 ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑
Non-Sparse	MSP	47.24	93.48	33.57	95.54	42.10	94.51	42.31	94.52	64.15	88.15	63.02	88.57	48.73	92.46
	ODIN	25.29	94.57	4.70	98.86	3.09	99.02	3.98	98.90	57.50	82.38	52.85	88.55	24.57	93.71
	GODIN	6.68	98.32	17.58	95.09	36.56	92.09	36.44	91.75	35.18	89.24	73.06	77.18	34.25	90.61
	Mahalanobis	6.42	98.31	56.55	86.96	9.14	97.09	9.78	97.25	21.51	92.15	85.14	63.15	31.42	89.15
	Energy	40.61	93.99	3.81	99.15	9.28	98.12	10.07	98.07	56.12	86.43	39.40	91.64	26.55	94.57
	ReAct	41.64	93.87	5.96	98.84	11.46	97.87	12.72	97.72	43.58	92.47	43.31	91.03	26.45	94.67
Sparse	Unit-Droput	89.16	60.96	72.97	81.33	87.03	68.78	87.29	68.07	88.53	60.10	94.82	59.18	86.63	66.40
	Weight-Droput	81.34	80.03	21.06	96.15	54.70	90.33	58.88	89.80	83.34	73.31	73.42	81.10	62.12	85.12
	Unit-Pruning	40.56	93.99	3.81	99.15	9.28	98.12	10.07	98.07	56.1	86.43	39.47	91.64	26.55	94.57
	Weight-Pruning	28.61	95.40	3.01	99.30	8.58	98.19	9.08	98.16	49.45	88.20	46.78	89.77	24.25	94.84
	DICE (ours)	$25.99^{\pm 5.10}$	$95.90^{\pm 1.08}$	$0.26^{\pm 0.11}$	$99.92^{\pm 0.02}$	$3.91^{\pm 0.56}$	$99.20^{\pm 0.15}$	$4.36^{\pm 0.71}$	$99.14^{\pm 0.15}$	$41.90^{\pm 4.41}$	$88.18^{\pm 1.80}$	$48.59^{\pm 1.53}$	$89.13^{\pm 0.31}$	$20.83^{\pm 1.58}$	$95.24^{\pm 0.24}$

Table 9. Detailed results on six common OOD benchmark datasets: `Textures` [6], `SVHN` [45], `Places365` [68], `LSUN-Crop` [66], `LSUN-Resize` [66], and `iSUN` [64]. For each ID dataset, we use the same DenseNet pretrained on **CIFAR-100**. ↑ indicates larger values are better and ↓ indicates smaller values are better.

Method Type	**Method**	**SVHN**		**LSUN-c**		**LSUN-r**		**iSUN**		**Textures**		**Places365**		**Average**	
		FPR95 ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑
Non-Sparse	MSP	81.70	75.40	60.49	85.60	85.24	69.18	85.99	70.17	84.79	71.48	82.55	74.31	80.13	74.36
	ODIN	41.35	92.65	10.54	97.93	65.22	84.22	67.05	83.84	82.34	71.48	82.32	76.84	58.14	84.49
	GODIN	36.74	93.51	43.15	89.55	40.31	92.61	37.41	93.05	64.26	76.72	95.33	65.97	52.87	85.24
	Mahalanobis	22.44	95.67	68.90	86.30	23.07	94.20	31.38	93.21	62.39	79.39	92.66	61.39	55.37	82.73
	Energy	87.46	81.85	14.72	97.43	70.65	80.14	74.54	78.95	84.15	71.03	79.20	77.72	68.45	81.19
	ReAct	83.81	81.41	25.55	94.92	60.08	87.88	65.27	86.55	77.78	78.95	82.65	74.04	62.27	84.47
Sparse	Unit-Droput	91.43	54.71	56.24	85.25	91.06	57.79	90.88	57.90	89.59	54.57	94.15	56.15	85.56	61.06
	Weight-Droput	92.97	64.39	18.96	95.62	88.67	65.48	87.12	67.82	88.45	64.38	88.69	71.87	77.48	71.59
	Unit-Pruning	87.52	81.83	14.73	97.43	70.62	80.18	74.46	79.00	84.20	71.02	79.32	77.70	68.48	81.19
	Weight-Pruning	77.99	84.14	5.17	99.05	59.42	87.13	61.80	86.09	72.68	73.85	82.53	75.06	59.93	84.22
	DICE (ours)	$54.65^{\pm4.94}$	$88.84^{\pm0.39}$	$0.93^{\pm0.07}$	$99.74^{\pm0.01}$	$49.40^{\pm1.99}$	$91.04^{\pm1.49}$	$48.72^{\pm1.55}$	$90.08^{\pm1.36}$	$65.04^{\pm0.66}$	$76.42^{\pm0.35}$	$79.58^{\pm2.34}$	$77.26^{\pm1.08}$	$49.72^{\pm1.69}$	$87.23^{\pm0.73}$