

DIVERSIFYING DEEP ENSEMBLES: A SALIENCY MAP APPROACH FOR ENHANCED OOD DETECTION, CALIBRATION, AND ACCURACY

Anonymous authors

Paper under double-blind review

ABSTRACT

Deep ensembles are capable of achieving state-of-the-art results in classification and out-of-distribution (OOD) detection. However, their effectiveness is limited due to the homogeneity of learned patterns within ensembles. To overcome this issue, our study introduces Saliency Diversified Deep Ensemble (SDDE), a novel approach that promotes diversity among ensemble members by leveraging saliency maps. Through incorporating saliency map diversification, our method outperforms conventional ensemble techniques and improves calibration in multiple classification and OOD detection tasks. In particular, the proposed method achieves state-of-the-art OOD detection quality, calibration, and accuracy on multiple benchmarks, including CIFAR10/100 and large-scale ImageNet datasets.

1 INTRODUCTION

In recent years, deep neural networks achieved state-of-the-art results in many computer vision tasks, including object detection Liu et al. (2020), classification Beyer et al. (2020), image retrieval Musgrave et al. (2020), and face recognition Deng et al. (2019). In image classification in particular, DNNs have demonstrated results more accurate than what humans are capable of on several popular benchmarks, such as ImageNet He et al. (2015). However, while these benchmarks often source both training and testing data from a similar distribution, real-world scenarios frequently feature test sets curated independently and under varying conditions Malinin et al. (2021; 2022). This disparity, known as domain shift, can greatly damage the performance of DNNs Gulrajani & Lopez-Paz (2020); Koh et al. (2020); Yao et al. (2022). As such, ensuring robust confidence estimation and out-of-distribution (OOD) detection is paramount for achieving risk-controlled recognition Zhang et al. (2023).

There has been substantial research focusing on confidence estimation and OOD detection in deep learning. Some works consider calibration refinements within softmax classifications Guo et al. (2017), while other authors examine the nuances of Bayesian training Goan & Fookes (2020). In these studies, ensemble methods that use DNNs stand out due to achieving superior outcomes in both confidence estimation and OOD detection Lakshminarayanan et al. (2017); Zhang et al. (2023). The results of these methods can be further improved by diversifying model predictions and adopting novel training paradigms Shui et al. (2018); Pang et al. (2019); Ramé & Cord (2021). However, the works on this subject primarily focus on diversifying the model output without diversifying the feature space.

In this work, we introduce Saliency Diversified Deep Ensemble (SDDE), a novel ensemble training method. This method encourages models to leverage distinct input features for making predictions, as shown in Figure 1. This is implemented through computing saliency maps, or regions of interest, during the training process, and applying a special loss function for diversification. By incorporating these enhancements, we achieve new state-of-the-art (SOTA) results on multiple OpenOOD Zhang et al. (2023) benchmarks in terms of test set accuracy, confidence estimation, and OOD detection quality. Moreover, following past works Hendrycks et al. (2018), we extend our approach by adding OOD data to model training, which made it possible to obtain new SOTA results among methods that utilize OOD data during training.

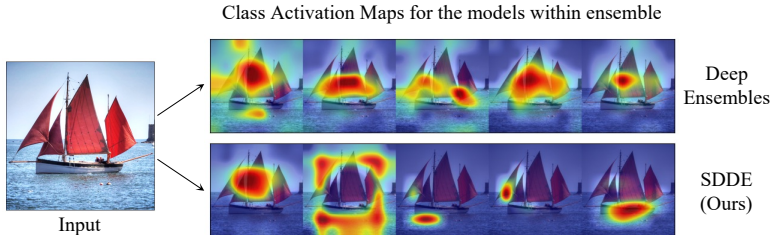


Figure 1: Saliency diversification: compared to Deep Ensembles, the models within the proposed SDDE ensemble use different features for prediction, leading to improved generalization and confidence estimation.

The main contributions of our work are the following:

1. We propose Saliency-Diversified Deep Ensembles (SDDE), a diversification technique that uses saliency maps in order to increase diversity among ensemble models.
2. We achieve new SOTA results in OOD detection, calibration, and classification on the OpenOOD benchmark. In particular, we improved on OOD detection, accuracy, and calibration results on CIFAR10/100 datasets. On ImageNet-1K, we enhance the accuracy and OOD detection scores.
3. We build upon our approach by adding OOD samples during ensemble training. The proposed approach improves ensemble performance and establishes a new SOTA on the CIFAR10 Near/Far and CIFAR100 Near OpenOOD benchmarks.

2 PRELIMINARIES

In this section, we will give a brief description of the model and saliency maps estimation approaches used in the proposed SDDE method. Suppose $x \in \mathbb{R}^{CHW}$ is an input image with height H , width W , and C channels. Let us consider a classification model $f(x) = h(g(x))$, which consists of Convolutional Neural Network (CNN) $g : \mathbb{R}^{CHW} \rightarrow \mathbb{R}^{C'H'W'}$ and classifier $h : \mathbb{R}^{C'H'W'} \rightarrow \mathbb{R}^L$ with L equal to the number of classes. CNN produces C' feature maps $M^c, c = \overline{1, C'}$ with spatial dimensions H' and W' . The output of the classifier is a vector of logits that can be mapped to class probabilities by a softmax layer.

The most common way to evaluate the importance of image pixels for output prediction is by computing saliency maps. In order to build a saliency map of the model, the simplest way is to compute output gradients w.r.t. input features Simonyan et al. (2013):

$$S_{Inp}(x) = \sum_{i=1}^L \nabla_x f_i(x). \quad (1)$$

However, previous works show that input gradients produce noisy outputs Simonyan et al. (2013). A more robust approach is to use Class Activation Maps (CAM) Zhou et al. (2016), or their generalization called GradCAM Selvaraju et al. (2017). Both methods utilize spatial directions of the structure of CNNs with bottleneck features. Unlike CAM, GradCAM can be applied at any CNN layer, which is useful for small input images. In GradCAM, the region of interest is estimated by analyzing activations of the feature maps. The weight of each feature map channel is computed as

$$\alpha_c = \frac{1}{H'W'} \sum_{i=1}^{H'} \sum_{j=1}^{W'} \frac{\delta f_y(x)}{\delta M_{i,j}^c}, \quad (2)$$

where $y \in \overline{1, L}$ is an image label. Having the weights of the feature maps, the saliency map is computed as

$$S_{GradCAM}(x, y) = ReLU \left(\sum_{c=1}^{C'} \alpha_c M^c \right), \quad (3)$$

where ReLU performs element-wise maximum between input and zero. The size of CAM is equal to the size of the feature maps. When visualized, CAMs are resized to match the size of the input image.

3 SALIENCY MAPS DIVERSITY AND ENSEMBLE AGREEMENT

Most previous works train ensemble models independently or by diversifying model outputs. Both of these approaches do not take into account the intrinsic algorithms implemented by the models. One way to distinguish classification algorithms is to consider the input features they use. Using saliency maps, which highlights input regions with the largest impact on the model prediction, is a popular technique for identifying these features. In this work, we suggest that ensemble diversity is related to the diversity of saliency maps. To validate this hypothesis, we analyzed the predictions and cosine similarity between saliency maps of the Deep Ensemble model Lakshminarayanan et al. (2017) trained on the MNIST, CIFAR10, CIFAR100, and ImageNet200 datasets. In particular, we compute agreement as a number of models which are consistent with the ensemble prediction. As shown in Figure 2, larger saliency maps similarity usually leads to a larger agreement between models. For more details, please, refer to Appendix F.

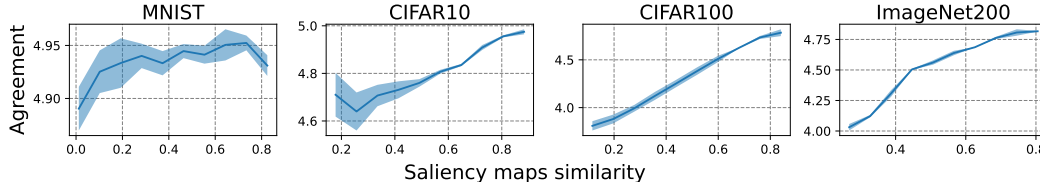


Figure 2: The dependency of ensemble predictions agreement on saliency maps cosine similarity. Saliency maps are computed using GradCAM. Mean and STD values w.r.t. multiple training seeds are reported.

Given this observation, we raise a question whether classification algorithms can be diversified by improving the diversity of saliency maps. We answer this question affirmatively by proposing a new SDDE method for training ensembles. In our experiments, we show the effectiveness of SDDE training in producing diverse high-quality models.

4 METHOD

4.1 SALIENCY MAPS DIVERSITY LOSS

In Deep Ensemble Lakshminarayanan et al. (2017), the models $f(x; \theta_k), k \in \overline{1, N}$ are trained independently. While there is a source of diversity in weight initialization, this method does not force different models to use different features. Therefore, we explore the following question: how can we train and implement different classification logic for models which rely on different input features? We answer this question by proposing a new loss function, which is motivated by previous research on saliency maps Selvaraju et al. (2017).

The idea behind SDDE is to make the saliency maps Simonyan et al. (2013) of the models as different as possible, as shown in Figure 1. Suppose we have computed a saliency map $S(x, y; \theta_k)$ for each model. The similarity of these models can be measured as a mean cosine similarity between their saliency maps. We thus propose the diversity loss function by computing a mean cosine similarity between the saliency maps of different models:

$$\mathcal{L}_{div}(x, y; \theta_1, \dots, \theta_N) = \frac{2}{N(N-1)} \sum_{k_1 > k_2} \langle S(x, y; \theta_{k_1}), S(x, y; \theta_{k_2}) \rangle. \quad (4)$$

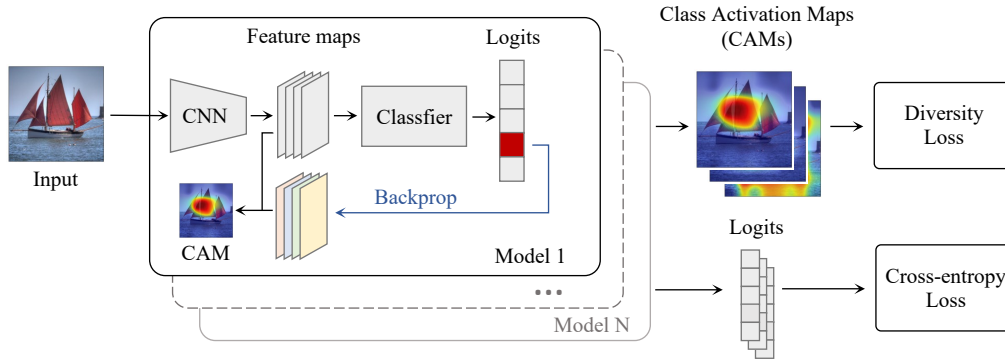


Figure 3: The training pipeline, where we compute saliency maps using the GradCAM method for each model and apply a diversification loss. The final loss also includes the cross-entropy term.

In SDDE, we use GradCAM Selvaraju et al. (2017), as it is more stable and requires less computation than input gradient method Simonyan et al. (2013). As GradCAM is differentiable, the diversity loss can be optimized via gradient descent along with cross-entropy loss.

The diversity loss is combined with cross-entropy during training in a single optimization objective. The final loss has the following form:

$$\mathcal{L}(x, y; \theta_1, \dots, \theta_N) = \lambda \mathcal{L}_{div}(x, y; \theta_1, \dots, \theta_N) + \frac{1}{N} \sum_k \mathcal{L}_{CE}(x, y; \theta_k), \quad (5)$$

where $\mathcal{L}_{CE}(x, y; \theta_k)$ is a standard cross-entropy loss and λ is a diversity loss weight. Example training algorithm is presented in Appendix H.

4.2 AGGREGATION METHODS FOR OOD DETECTION

The original Deep Ensembles approach computes the average over softmax probabilities during inference. The naive approach to OOD detection is to use the Maximum Softmax Probability (MSP) Hendrycks & Gimpel (2016). Suppose there is an ensemble $f(x; \theta_k)$, $k = \overline{1, N}$, where each model $f(x; \theta_k)$ maps input data to the vector of logits. Let us denote softmax outputs for each model as p_i^k and average probabilities as p_i :

$$p_i^k(x) = \frac{e^{f_i(x; \theta_k)}}{\sum_j e^{f_j(x; \theta_k)}} = \text{Softmax}_i(f(x; \theta_k)), \quad (6)$$

$$p_i(x) = \frac{1}{N} \sum_k p_i^k(x). \quad (7)$$

The rule of thumb in ensembles is to predict the OOD score based on the maximum output probability:

$$U_{MSP}(x) = \max_i p_i(x). \quad (8)$$

Some recent works follow this approach Abe et al. (2022). However, other works propose to use logits instead of probabilities for OOD detection Hendrycks et al. (2019). In this work, we briefly examine which aggregation is better for ensembles by proposing the Maximum Average Logit (MAL) score. The MAL score extends maximum logit approach for ensembles and is computed as:

$$U_{MAL}(x) = \max_i \frac{1}{N} \sum_k f_i(x; \theta_k). \quad (9)$$

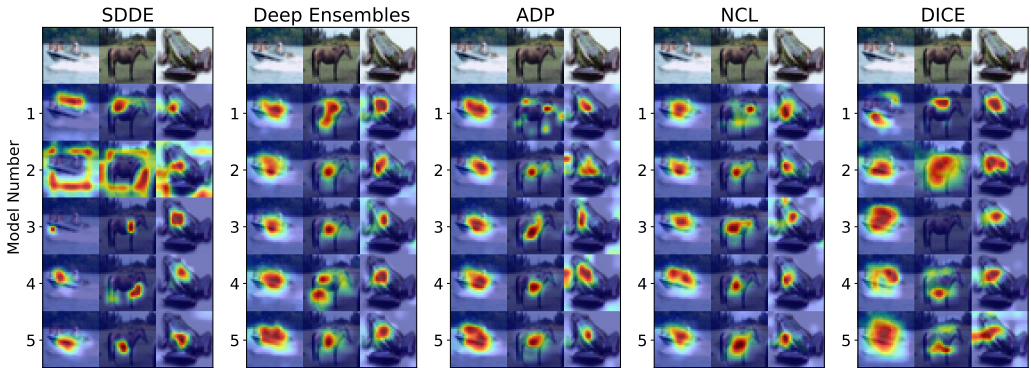


Figure 4: Class Activation Maps (CAMs) for SDDE and the baseline methods. SDDE increases CAMs diversity by focusing on different regions of the images.

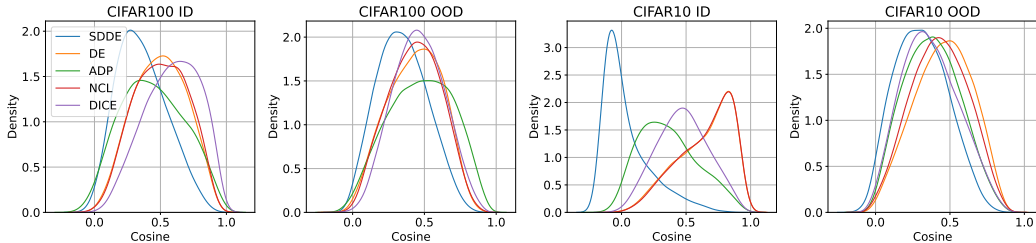


Figure 5: Pairwise distributions of cosine similarities between Class Activation Maps (CAMs) of ensemble models.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

In our experiments, we follow the experimental setup and training procedure from the OpenOOD benchmark (Zhang et al., 2023). We use ResNet18 (He et al., 2016) for the CIFAR10, CIFAR100 (Krizhevsky & Hinton, 2009), and ImageNet-200 datasets, LeNet (LeCun et al., 1998) for the MNIST dataset (LeCun et al., 1998), and ResNet50 for the ImageNet-1K dataset. All models are trained using the SGD optimizer with a momentum of 0.9. The initial learning rate is set to 0.1 for ResNet18 and LeNet, 0.001 for ResNet50, and then reduced to 10^{-6} with the Cosine Annealing scheduler (Loshchilov & Hutter, 2016). Training lasts 50 epochs on MNIST, 100 epochs on CIFAR10 and CIFAR100, 30 epochs on ImageNet-1K, and 90 epochs on ImageNet-200. In contrast to OpenOOD, we perform a multi-seed evaluation with 5 random seeds and report the mean and STD values for each experiment. In ImageNet experiments, we evaluate with 3 seeds. We compare the proposed SDDE method with other ensembling and diversification approaches, namely Deep Ensembles (DE) (Lakshminarayanan et al., 2017), Negative Correlation Learning (NCL) (Shui et al., 2018), Adaptive Diversity Promoting (ADP) (Pang et al., 2019), and DICE diversification loss (Ramé & Cord, 2021). In the default setup, we train ensembles of 5 models. In SDDE, we set the parameter λ from the Equation 5 to 0.1 for MNIST, and to 0.005 for CIFAR10/100, ImageNet-200, and ImageNet-1K. The comparison of saliency maps computation approaches can be found in Appendix B. The ablation study on the number of models is presented in Appendix D.

5.2 ENSEMBLE DIVERSITY

To demonstrate the effectiveness of the proposed diversification loss, we evaluate the saliency maps for all considered methods, as shown in Figure 4. The ground truth label is not available during inference, so we compute CAMs for the predicted class. It can be seen that SDDE models use different input regions for prediction. To highlight the difference between the methods, we analyze

the pairwise cosine similarities between CAMs of different models in an ensemble. The distributions of cosine similarities are presented in Figure 5. According to the data, the baseline methods reduce similarity compared to Deep Ensemble. However, among all methods, SDDE achieves the lowest cosine similarity. This effect persists even on OOD samples. In the following experiments, we study the benefits of the proposed diversification.

Prediction-based diversity metrics. Our saliency-based diversity measure primarily focuses on the variation of saliency maps, which may not fully capture the diversity in the prediction space. However, the benchmarked NCL, ADP, and DICE baselines mainly optimize the diversity in the prediction space. In order to fill this gap, we include additional diversity metrics, such as pairwise disagreement between networks, ratio-error, Q-statistics, correlation coefficient, and mutual information Aksela (2003). The values of metrics for SDDE and baselines are presented in Table 1. It can be seen that SDDE has a lesser impact on the quality of individual models in the ensemble compared to other methods, while achieving moderate diversity values.

Table 1: Diversification metrics. The best result in each column is **bolded**. The DICE method failed to converge on ImageNet.

Dataset	Method	Diversity					Error	
		Disagreement \uparrow	Correlation \downarrow	Q-value \downarrow	D/S Error Rate \uparrow	MI \downarrow	Mean	Ensemble
CIFAR 10	DE	4.37 \pm 0.26	58.85 \pm 1.43	97.38 \pm 0.31	0.22 \pm 0.02	2.06 \pm 0.01	4.89 \pm 0.13	3.93 \pm 0.10
	NCL	4.32 \pm 0.16	59.92 \pm 0.81	97.53 \pm 0.17	0.22 \pm 0.01	2.07 \pm 0.01	4.95 \pm 0.13	4.04 \pm 0.08
	ADP	4.89 \pm 0.06	57.52 \pm 0.39	97.03 \pm 0.09	0.32 \pm 0.01	2.04 \pm 0.00	5.14 \pm 0.04	3.93 \pm 0.12
	DICE	4.88 \pm 1.01	57.61 \pm 4.15	96.82 \pm 1.32	0.25 \pm 0.05	2.04 \pm 0.04	5.24 \pm 0.57	4.10 \pm 0.16
	SDDE (Our)	4.33 \pm 0.13	59.29 \pm 0.95	97.47 \pm 0.18	0.23 \pm 0.01	2.07 \pm 0.01	4.87 \pm 0.11	3.92 \pm 0.11
CIFAR 100	DE	20.81 \pm 2.09	62.90 \pm 2.25	92.70 \pm 1.42	0.86 \pm 0.10	3.63 \pm 0.09	23.12 \pm 1.28	19.05 \pm 0.54
	NCL	20.57 \pm 0.93	63.06 \pm 1.32	92.86 \pm 0.70	0.83 \pm 0.04	3.64 \pm 0.04	23.00 \pm 0.47	19.08 \pm 0.20
	ADP	25.53 \pm 0.28	63.02 \pm 0.78	92.44 \pm 0.40	1.78 \pm 0.08	3.43 \pm 0.01	25.13 \pm 0.12	18.82 \pm 0.06
	DICE	21.98 \pm 0.31	60.88 \pm 0.41	91.78 \pm 0.26	0.98 \pm 0.01	3.59 \pm 0.01	23.20 \pm 0.25	18.74 \pm 0.25
	SDDE (Our)	21.00 \pm 1.34	61.43 \pm 1.94	92.09 \pm 1.13	0.86 \pm 0.07	3.63 \pm 0.05	22.80 \pm 0.63	18.67 \pm 0.25
ImageNet	DE	4.37 \pm 0.02	93.65 \pm 0.03	99.84 \pm 0.00	0.09 \pm 0.00	5.99 \pm 0.00	25.41 \pm 0.00	24.84 \pm 0.06
	NCL	0.51 \pm 0.02	99.31 \pm 0.03	100.00 \pm 0.00	0.01 \pm 0.00	6.17 \pm 0.00	25.41 \pm 0.11	24.95 \pm 0.13
	ADP	30.17 \pm 0.47	80.93 \pm 0.67	98.05 \pm 0.10	2.43 \pm 0.04	4.50 \pm 0.03	34.56 \pm 0.20	24.90 \pm 0.01
	DICE							
	SDDE (Our)	4.36 \pm 0.07	93.80 \pm 0.09	99.85 \pm 0.00	0.09 \pm 0.00	5.99 \pm 0.00	25.39 \pm 0.00	24.80 \pm 0.01

5.3 ENSEMBLE ACCURACY AND CALIBRATION

Ensemble methods are most commonly used for improving classification accuracy and prediction calibration metrics. We compare these aspects of SDDE with other ensembling methods by measuring test set classification accuracy, Negative Log-Likelihood (NLL), Expected Calibration Error (ECE) (Naeni et al., 2015), and Brier score (Brier et al., 1950). All metrics are computed after temperature tuning on the validation set (Ashukha et al., 2020). The results are presented in Table 2. It can be seen that the SDDE approach outperforms other methods on CIFAR10 and CIFAR100 in terms of both accuracy and calibration. For the additional experiments with Wide ResNet architecture, which is not part of OpenOOD, please, refer to Appendix C.

Table 2: Accuracy and calibration metrics. The best results for each dataset and metric are **bolded**.

Dataset	Metric	DE	NCL	ADP	DICE	SDDE (Our)
MNIST	NLL ($\times 10$) \downarrow	0.38 \pm 0.02	0.36 \pm 0.02	0.35 \pm 0.02	0.37 \pm 0.02	0.37 \pm 0.03
	ECE ($\times 10^2$) \downarrow	0.21 \pm 0.07	0.19 \pm 0.07	0.31 \pm 0.05	0.29 \pm 0.07	0.28 \pm 0.08
	Brier score ($\times 10^2$) \downarrow	1.95 \pm 0.09	1.89 \pm 0.08	1.77 \pm 0.12	1.84 \pm 0.09	1.92 \pm 0.15
	Accuracy (%) \uparrow	98.72 \pm 0.07	98.76 \pm 0.03	98.85 \pm 0.11	98.81 \pm 0.09	98.73 \pm 0.14
CIFAR 10	NLL ($\times 10$) \downarrow	1.30 \pm 0.02	1.33 \pm 0.01	1.36 \pm 0.02	1.45 \pm 0.03	1.28 \pm 0.01
	ECE ($\times 10^2$) \downarrow	0.84 \pm 0.16	0.87 \pm 0.13	1.00 \pm 0.14	1.33 \pm 0.06	0.82 \pm 0.15
	Brier score ($\times 10^2$) \downarrow	5.96 \pm 0.08	6.03 \pm 0.08	5.99 \pm 0.07	6.33 \pm 0.23	5.88 \pm 0.07
	Accuracy (%) \uparrow	96.07 \pm 0.10	95.96 \pm 0.08	96.07 \pm 0.12	95.90 \pm 0.16	96.08 \pm 0.11
CIFAR 100	NLL ($\times 10$) \downarrow	7.21 \pm 0.14	7.22 \pm 0.05	7.54 \pm 0.03	7.46 \pm 0.054	6.93 \pm 0.05
	ECE ($\times 10^2$) \downarrow	3.78 \pm 0.17	3.86 \pm 0.23	3.73 \pm 0.28	4.46 \pm 0.36	3.48 \pm 0.36
	Brier score ($\times 10^2$) \downarrow	27.33 \pm 0.76	27.34 \pm 0.33	27.08 \pm 0.17	27.12 \pm 0.26	26.64 \pm 0.34
	Accuracy (%) \uparrow	80.95 \pm 0.54	80.92 \pm 0.20	81.18 \pm 0.06	81.26 \pm 0.25	81.33 \pm 0.28

5.4 OOD DETECTION

We evaluate SDDE’s OOD detection ability using the OpenOOD benchmark. Since SDDE does not use external data for training, we compare it to ensembles that only use in-distribution data. The results are presented in Table 3. It can be seen that SDDE achieves SOTA results in almost all cases, including the total scores on near and far tests. For the additional analysis on different logit aggregation methods, please, refer to Appendix E.

Table 3: OOD detection results. All methods are trained on the ID dataset and tested on multiple OOD sources. Mean and STD AUROC values are reported. The best results among different methods are **bolded**.

ID	Method	Near OOD		Far OOD				Total Near	Total Far
		Fashion MNIST	NotMNIST	CIFAR10	TIN	Texture	Places365		
MNIST	DE	95.34 ± 0.53	89.56 ± 0.84	99.06 ± 0.15	98.93 ± 0.18	96.32 ± 0.79	99.06 ± 0.14	92.45 ± 0.60	98.34 ± 0.31
	NCL	95.27 ± 0.42	89.19 ± 0.69	99.06 ± 0.11	98.97 ± 0.12	97.17 ± 0.45	99.03 ± 0.11	92.23 ± 0.53	98.56 ± 0.19
	ADP	96.49 ± 0.67	90.98 ± 0.65	99.48 ± 0.18	99.38 ± 0.21	97.52 ± 0.77	99.44 ± 0.15	93.74 ± 0.48	98.96 ± 0.32
	DICE	95.85 ± 0.22	89.88 ± 1.51	99.22 ± 0.19	99.10 ± 0.22	97.52 ± 1.05	99.20 ± 0.17	92.87 ± 0.67	98.76 ± 0.41
	SDDE (Our)	98.85 ± 0.38	94.27 ± 1.32	99.88 ± 0.03	99.84 ± 0.03	99.97 ± 0.01	99.82 ± 0.04	96.56 ± 0.67	99.88 ± 0.02
		CIFAR100	TIN	MNIST	SVHN	Texture	Places365		
CIFAR10	DE	90.31 ± 0.23	91.84 ± 0.16	95.35 ± 0.53	95.05 ± 0.47	92.52 ± 0.30	91.29 ± 0.59	91.07 ± 0.17	93.55 ± 0.20
	NCL	90.51 ± 0.12	91.93 ± 0.08	95.09 ± 0.29	94.81 ± 0.46	92.18 ± 0.34	92.00 ± 0.26	91.22 ± 0.09	93.52 ± 0.17
	ADP	89.96 ± 0.25	91.62 ± 0.13	95.41 ± 0.24	94.66 ± 0.40	92.55 ± 0.12	91.38 ± 0.53	90.79 ± 0.17	93.50 ± 0.16
	DICE	89.09 ± 0.79	90.89 ± 0.48	94.46 ± 1.10	94.89 ± 0.49	92.36 ± 0.45	89.84 ± 1.14	89.99 ± 0.63	92.89 ± 0.47
	SDDE (Our)	91.20 ± 0.13	92.92 ± 0.17	96.77 ± 0.45	95.74 ± 0.36	92.69 ± 0.62	93.19 ± 0.62	92.06 ± 0.13	94.60 ± 0.17
		CIFAR10	TIN	MNIST	SVHN	Texture	Places365		
CIFAR100	DE	80.83 ± 0.55	84.28 ± 0.52	80.24 ± 1.43	81.09 ± 1.27	80.16 ± 0.41	80.96 ± 0.26	82.55 ± 0.51	80.61 ± 0.51
	NCL	81.11 ± 0.20	84.48 ± 0.19	79.66 ± 0.66	80.91 ± 1.99	80.30 ± 0.52	81.24 ± 0.17	82.79 ± 0.18	80.53 ± 0.66
	ADP	81.12 ± 0.19	84.85 ± 0.22	79.54 ± 0.76	82.84 ± 2.16	81.80 ± 0.47	81.28 ± 0.11	82.98 ± 0.18	81.37 ± 0.54
	DICE	81.42 ± 0.18	84.94 ± 0.21	83.37 ± 1.11	82.40 ± 1.96	81.43 ± 0.39	81.41 ± 0.27	83.18 ± 0.14	82.15 ± 0.85
	SDDE (Our)	81.97 ± 0.10	85.34 ± 0.16	81.86 ± 1.80	83.40 ± 1.03	82.67 ± 0.35	81.63 ± 0.18	83.65 ± 0.09	82.39 ± 0.56

5.5 IMAGENET RESULTS

In addition to the above-mentioned datasets, we conduct experiments on the large-scale ImageNet-1K benchmark from OpenOOD. The results for accuracy, calibration, and OOD detection are presented in Table 4. It can be observed that on ImageNet-1K SDDE achieves the best accuracy and OOD detection score among all the methods. Furthermore, SDDE achieves calibration scores comparable to the best-performing method in each column. The DICE training failed to converge in this setup, and we excluded it from the comparison.

Table 4: ImageNet results. The best-performing method in each column is **bolded**.

Dataset	Method	Near OOD	Far OOD	NLL ($\times 10$)	ECE ($\times 10^2$)	Brier ($\times 10^2$)	Accuracy (%)
ImageNet	DE	75.02 ± 0.06	82.72 ± 0.26	9.79 ± 0.00	1.54 ± 0.06	34.40 ± 0.01	75.16 ± 0.06
	NCL	75.06 ± 0.05	82.85 ± 0.17	9.83 ± 0.00	1.51 ± 0.02	34.51 ± 0.14	75.05 ± 0.13
	ADP	75.26 ± 0.01	82.62 ± 0.08	9.95 ± 0.01	1.95 ± 0.11	34.54 ± 0.08	75.10 ± 0.01
	SDDE (Our)	75.26 ± 0.02	85.98 ± 0.61	9.79 ± 0.01	1.55 ± 0.05	34.43 ± 0.05	75.20 ± 0.01

5.6 DISTRIBUTION SHIFTS

In addition to OOD detection, accuracy, and calibration, we evaluate SDDE’s performance on datasets with distribution shifts (OOD generalization), such as the CIFAR10-C and CIFAR100-C datasets Hendrycks & Dietterich (2019). The accuracy and calibration metrics are reported in Table 5. These results demonstrate that SDDE outperforms the baselines in both accuracy and calibration metrics on CIFAR10-C and achieves accuracy on par with the best-performing method while improving the calibration metrics on CIFAR100-C.

5.7 LEVERAGING OOD DATA FOR TRAINING

In some applications, an unlabeled OOD sample can be provided during training to further improve OOD detection quality. According to previous studies Zhang et al. (2023), Outlier Exposure (OE) Hendrycks et al. (2018) is one of the most accurate methods on CIFAR10/100 and ImageNet-200

Table 5: Accuracy and calibration metrics on corrupted datasets.

Method	CIFAR10-C				CIFAR100-C			
	NLL ($\times 10$)	ECE ($\times 10^2$)	Brier ($\times 10^2$)	Accuracy	NLL ($\times 10$)	ECE ($\times 10^2$)	Brier ($\times 10^2$)	Accuracy
DE	1.06 \pm 0.01	4.67 \pm 0.12	44.68 \pm 0.58	67.74 \pm 0.38	12.91 \pm 0.10	4.74 \pm 0.26	43.28 \pm 0.59	68.55 \pm 0.55
NCL	1.06 \pm 0.01	4.83 \pm 0.37	44.98 \pm 0.49	67.43 \pm 0.40	12.83 \pm 0.12	4.69 \pm 0.10	43.07 \pm 0.36	68.71 \pm 0.30
ADP	1.06 \pm 0.02	4.67 \pm 0.12	44.64 \pm 0.58	67.81 \pm 0.38	13.14 \pm 0.11	4.97 \pm 0.10	42.61 \pm 0.32	69.20 \pm 0.22
DICE	1.10 \pm 0.02	7.41 \pm 0.53	45.86 \pm 0.70	67.14 \pm 0.60	13.57 \pm 0.09	5.23 \pm 0.26	44.40 \pm 0.50	67.91 \pm 0.48
SDDE (Ours)	1.04 \pm 0.01	4.84 \pm 0.38	44.32 \pm 0.31	67.96 \pm 0.28	12.58 \pm 0.08	4.00 \pm 0.09	42.43 \pm 0.32	69.08 \pm 0.27

Table 6: Evaluation results for methods trained with OOD data.

Dataset	Method	Near OOD	Far OOD	NLL ($\times 10$)	ECE ($\times 10^2$)	Brier ($\times 10^2$)	Accuracy (%)
CIFAR 10	OE Single	94.82 \pm 0.21	96.00 \pm 0.13	2.85 \pm 0.26	5.88 \pm 0.97	11.62 \pm 0.98	94.63 \pm 0.26
	OE Ensemble	96.23 \pm 0.08	97.45 \pm 0.15	2.65 \pm 0.05	4.86 \pm 0.25	10.89 \pm 0.19	95.84 \pm 0.19
	SDDE_{OOD} (Ours)	96.22 \pm 0.08	97.56 \pm 0.07	2.64 \pm 0.02	4.95 \pm 0.15	10.86 \pm 0.15	95.87 \pm 0.02
CIFAR 100	OE Single	88.30 \pm 0.10	81.41 \pm 1.49	9.77 \pm 0.28	8.41 \pm 0.47	34.11 \pm 0.73	76.84 \pm 0.42
	OE Ensemble	89.61 \pm 0.04	84.53 \pm 0.72	9.93 \pm 0.11	8.35 \pm 0.20	34.34 \pm 0.39	80.65 \pm 0.30
	SDDE_{OOD} (Ours)	89.70 \pm 0.22	85.47 \pm 1.76	10.13 \pm 0.28	8.70 \pm 0.31	34.86 \pm 0.60	80.30 \pm 0.25
ImageNet-200	OE Single	81.87 \pm 0.16	86.77 \pm 0.06	21.88 \pm 0.46	37.31 \pm 1.10	54.67 \pm 1.11	77.57 \pm 0.28
	OE Ensemble	84.41 \pm 0.15	89.45 \pm 0.26	21.71 \pm 0.07	36.68 \pm 0.14	54.37 \pm 0.12	83.36 \pm 0.07
	SDDE_{OOD} (Ours)	84.46 \pm 0.07	89.57 \pm 0.11	21.63 \pm 0.29	36.71 \pm 0.79	54.17 \pm 0.69	83.36 \pm 0.08

datasets. The idea behind OE is to make predictions on OOD data as close to the uniform distribution as possible. This is achieved by minimizing cross-entropy between the uniform distribution and the output of the model on OOD data:

$$\mathcal{L}_{OE}(x; \theta) = -\frac{1}{C} \sum_i \log \text{Softmax}_i(f(x; \theta)). \quad (10)$$

Given an unlabeled OOD sample \bar{x} , we combine OE loss with SDDE, leading to the following objective:

$$\mathcal{L}_{OOD}(x, y, \bar{x}; \theta_1, \dots, \theta_N) = \mathcal{L}(x, y; \theta_1, \dots, \theta_N) + \beta \frac{1}{N} \mathcal{L}_{OE}(\bar{x}; \theta_k), \quad (11)$$

We follow the original OE implementation and set β to 0.5. We call the final method SDDE_{OOD}.

The OpenOOD implementation of OE includes only a single model. In order to have a fair comparison, we train an ensemble of 5 OE models and average their predictions.

The results are presented in Table 6. It can be seen that SDDE_{OOD} achieves higher OOD detection accuracy compared to the current SOTA OE method in OpenOOD. Following SDDE experiments, we evaluate SDDE_{OOD} calibration and accuracy. SDDE_{OOD} demonstrates competitive accuracy and calibration scores on all benchmark datasets.

6 RELATED WORK

Confidence Estimation. DNNs are prone to overfitting, which limits their ability to generalize and predict confidence Guo et al. (2017). Multiple works attempted to address this issue. A simple approach to confidence estimation is to take the probability of a predicted class on the output of the softmax layer Hendrycks & Gimpel (2016). Several authors proposed improvements to this method for either better confidence estimation or higher OOD detection accuracy.

Some works studied activation statistics between layers to detect anomalous behavior Sun et al. (2021). Another approach is to use Bayesian training, which usually leads to improved confidence prediction at the cost of accuracy Goan & Fookes (2020). Other works attempted to use insights from classical machine learning, such as KNNs Sun et al. (2022) and ensembles Lakshminarayanan et al. (2017). It was shown that ensembles reduce overfitting and produce confidence estimates, which outperforms most existing methods Zhang et al. (2023). In this work, we further improve deep ensembles by applying a new training algorithm.

Ensembles Diversification. According to previous works, ensemble methods produce SOTA results in popular classification Lakshminarayanan et al. (2017); Ramé & Cord (2021) and OOD detection tasks Zhang et al. (2023). It was shown that the quality of the ensemble largely depends on the

diversity of underlying models Stickland & Murray (2020). Some works improve diversity by implementing special loss functions on predicted probabilities. The idea is to make different predicted distributions for different models. The NCL loss Shui et al. (2018) reduces correlations between output probabilities, but can negatively influence the prediction of the correct class. The ADP Pang et al. (2019) solves this problem by diversifying only the probabilities of alternative classes. On the other hand, the DICE Ramé & Cord (2021) loss reduces dependency between bottleneck features of multiple models. While previous works reduce the correlations of either model predictions or bottleneck features, we directly diversify the input features used by the models within an ensemble, which further improves OOD detection and calibration.

Out-of-Distribution Detection Risk-controlled recognition poses the problem of detecting out-of-distribution (OOD) data, i.e. data with distribution different from the training set, or data with unknown classes Zhang et al. (2023). Multiple OOD detection methods were proposed for deep models Sun et al. (2021); Hendrycks et al. (2019). Despite the progress made on single models, Deep Ensembles Lakshminarayanan et al. (2017) use a traditional Maximum Softmax Probability (MSP) Hendrycks & Gimpel (2016) approach to OOD detection. In this work, we introduce a novel ensembling method that offers enhanced OOD detection capabilities. Furthermore, we extend the Maximum Logit Score (MLS) Hendrycks et al. (2019) for ensemble application, underscoring its advantages over MSP.

7 LIMITATIONS

While the Saliency-Diversified Deep Ensembles (SDDE) method can offer promising advancements in OOD detection, calibration, and classification, there are some inherent limitations to consider. Firstly, while incorporating OOD samples during ensemble training can be beneficial, this approach could raise questions about the quality, diversity, and source of these samples, as well as whether such results would generalize when faced with unseen OOD categories. Additionally, while achieving SOTA results on the OpenOOD benchmark is notable, benchmarks constantly evolve, and the real-world applicability and robustness of the method across varied scenarios and datasets is yet to be extensively tested. Nevertheless, while it is important to take these limitations into account, they do not invalidate our conclusions. Instead, they qualify our results, serving as potential starting points for future research.

8 CONCLUSION

In this work, we proposed SDDE, a novel ensembling method for classification and OOD detection. SDDE forces the models within the ensemble to use different input features for prediction, which increases ensemble diversity. According to our experiments, SDDE performs better than several popular ensembles on the CIFAR10, CIFAR100, and ImageNet-1K datasets. At the same time, SDDE outperforms other methods in OOD detection on the OpenOOD benchmark. Improved confidence estimation and OOD detection make SDDE a valuable tool for risk-controlled recognition. We further generalized SDDE for training with OOD data and achieved SOTA results on the OpenOOD benchmark.

REFERENCES

- Taiga Abe, Estefany Kelly Buchanan, Geoff Pleiss, Richard Zemel, and John P Cunningham. Deep ensembles work, but are they necessary? *Advances in Neural Information Processing Systems*, 35:33646–33660, 2022.
- Matti Aksela. Comparison of classifier selection methods for improving committee performance. In *International Workshop on Multiple Classifier Systems*, pp. 84–93. Springer, 2003.
- Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*, 2020.
- Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.

- Glenn W Brier et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.
- Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2019.
- Ethan Goan and Clinton Fookes. Bayesian neural networks: An introduction and survey. *Case Studies in Applied Bayesian Data Science: CIRM Jean-Morlet Chair, Fall 2018*, pp. 45–87, 2020.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *ArXiv*, abs/2007.01434, 2020.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.
- 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*, pp. 770–778, 2016.
- Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.
- Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.
- Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.
- Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2019.
- Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanus Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. *arXiv*, 2020.
- Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.
- Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128:261–318, 2020.
- Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

- Andrey Malinin, Neil Band, German Chesnokov, Yarin Gal, Mark JF Gales, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, Vatsal Raina, et al. Shifts: A dataset of real distributional shift across multiple large-scale tasks. In *NeurIPS Dataset & Benchmark Track*, 2021.
- Andrey Malinin, Andreas Athanasopoulos, Muhamed Barakovic, Meritxell Bach Cuadra, Mark J. F. Gales, Cristina Granziera, Mara Graziani, Nikolay Kartashev, Konstantinos Kyriakopoulos, Po-Jui Lu, Nataliia Molchanova, Antonis Nikitakis, Vatsal Raina, Francesco La Rosa, Eli Sivena, Vasileios Tsarsitalidis, Efi Tsompopoulou, and Elena Volf. Shifts 2.0: Extending the dataset of real distributional shifts, 2022.
- Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pp. 681–699. Springer, 2020.
- Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.
- Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pp. 4970–4979. PMLR, 2019.
- Alexandre Ramé and Matthieu Cord. Dice: Diversity in deep ensembles via conditional redundancy adversarial estimation. In *ICLR 2021-9th International Conference on Learning Representations*, 2021.
- Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.
- Changjian Shui, Azadeh Sadat Mozafari, Jonathan Marek, Ihsen Hedhli, and Christian Gagné. Diversity regularization in deep ensembles. *arXiv preprint arXiv:1802.07881*, 2018.
- Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.
- Asa Cooper Stickland and Iain Murray. Diverse ensembles improve calibration. *arXiv preprint arXiv:2007.04206*, 2020.
- Yiyu Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 34:144–157, 2021.
- Yiyu Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, pp. 20827–20840. PMLR, 2022.
- Huaxiu Yao, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei Koh, and Chelsea Finn. Wild-time: A benchmark of in-the-wild distribution shift over time. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.
- Jingyang Zhang, Jingkang Yang, Pengyun Wang, Haoqi Wang, Yueqian Lin, Haoran Zhang, Yiyu Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, et al. Openood v1.5: Enhanced benchmark for out-of-distribution detection. *arXiv preprint arXiv:2306.09301*, 2023.
- Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

A ADVERSARIAL LOSS IN DEEP ENSEMBLES

The original Deep Ensemble approach Lakshminarayanan et al. (2017) applied an extra adversarial loss, which is not present in OpenOOD. Therefore, we evaluate the impact of this loss on accuracy, calibration and OOD detection scores. The adversarial loss is inspired by adversarial samples generation Goodfellow et al. (2014) and has the following form:

$$\mathcal{L}_{adv} = \frac{1}{N} \sum_k \mathcal{L}_{CE}(\hat{x}_k, y, \theta_k), \quad (12)$$

where \hat{x}_k is an adversarial version of the input image x with label y , computed as

$$\hat{x}_k = x + \epsilon \text{sign}(\nabla_x \mathcal{L}_{CE}(x, y, \theta_k)). \quad (13)$$

We follow the original implementation and set ϵ equal to 0.01 of the input data’s dynamic range.

We evaluated the adversarial loss for the CIFAR-10 and CIFAR-100 datasets. According to our results, presented in Table 7, adversarial loss slightly reduces the quality of the ensemble. We conclude that adversarial loss improves ensemble robustness to adversarial samples, but negatively affects accuracy and OOD detection.

Table 7: Diversity loss ablation study.

Dataset	Method	Near ODD \uparrow	Far OOD \uparrow	NLL ($\times 10$) \downarrow	ECE ($\times 10^2$) \downarrow	Brier score ($\times 10^2$) \downarrow	Accuracy (%) \uparrow
MNIST	DE	92.45 \pm 0.60	98.34 \pm 0.31	0.38 \pm 0.02	0.21 \pm 0.06	1.95 \pm 0.09	98.72 \pm 0.07
	DE with \mathcal{L}_{adv}	95.63 \pm 0.46	99.55 \pm 0.02	0.34 \pm 0.01	0.26 \pm 0.04	1.76 \pm 0.05	98.84 \pm 0.06
CIFAR10	DE	91.07 \pm 0.17	93.55 \pm 0.20	1.30 \pm 0.02	0.84 \pm 0.16	5.96 \pm 0.08	96.07 \pm 0.10
	DE with \mathcal{L}_{adv}	90.84 \pm 0.13	92.12 \pm 0.22	2.02 \pm 0.04	0.93 \pm 0.09	9.60 \pm 0.15	93.50 \pm 0.21
CIFAR100	DE	82.55 \pm 0.51	80.61 \pm 0.51	7.21 \pm 0.14	3.78 \pm 0.17	27.33 \pm 0.76	80.95 \pm 0.54
	DE with \mathcal{L}_{adv}	80.87 \pm 0.60	76.87 \pm 0.99	10.06 \pm 0.05	3.11 \pm 0.79	37.00 \pm 0.35	72.74 \pm 0.39

B INPUT GRADIENTS VS GRADCAM

In this work, we diversify saliency maps computed by GradCAM. We also evaluate saliency maps constructed by taking gradients w.r.t. the input image, i.e. *input gradients* Simonyan et al. (2013). The results from Table 8 demonstrate the superiority of GradCAM to input gradients in terms of OOD detection and classification accuracy.

Another difference between input gradients and GradCAM lies in computation speed. Diversity loss requires two backward passes: one for making a saliency map, and another for loss optimization. The complexity of both passes depends on the method used. Computing gradients w.r.t. the input images requires a backward pass through all layers, while GradCAM stops when the feature extraction layer is reached. This leads to a difference in training speed, making GradCAM about 2.4 times faster than input gradients. As GradCAM outperforms input gradients in both accuracy and training speed, we decided to use it as a default choice for ensemble diversification.

Table 8: Comparison of saliency map computation algorithms.

Dataset	Method	Near ODD \uparrow	Far OOD \uparrow	NLL ($\times 10$) \downarrow	ECE ($\times 10^2$) \downarrow	Brier score ($\times 10^2$) \downarrow	Accuracy (%) \uparrow
MNIST	Inp. grad.	92.04 \pm 0.57	98.59 \pm 0.30	0.37 \pm 0.02	0.28 \pm 0.40	1.92 \pm 0.09	98.79 \pm 0.05
	GradCAM	96.56 \pm 0.67	99.88 \pm 0.02	0.37 \pm 0.03	0.28 \pm 0.08	1.92 \pm 0.15	98.73 \pm 0.14
CIFAR10	Inp. grad.	91.18 \pm 0.13	93.56 \pm 0.24	1.35 \pm 0.05	0.76 \pm 0.12	6.23 \pm 0.34	95.89 \pm 0.21
	GradCAM	92.06 \pm 0.13	94.60 \pm 0.17	1.28 \pm 0.01	0.82 \pm 0.15	5.88 \pm 0.07	96.08 \pm 0.11
CIFAR100	Inp. grad.	82.80 \pm 0.18	80.06 \pm 0.58	7.14 \pm 0.03	3.76 \pm 0.26	27.06 \pm 0.27	81.06 \pm 0.14
	GradCAM	83.69 \pm 0.08	82.39 \pm 0.50	6.93 \pm 0.05	3.45 \pm 0.35	26.66 \pm 0.33	81.27 \pm 0.24

C WIDE RESNET-28-10 EXPERIMENTS

In addition to OpenOOD benchmarks, we conducted experiments with the Wide ResNet architecture. The results in Table 9 show that SDDE with WRN-28-10 outperforms other methods in terms of accuracy, calibration, and OOD detection scores.

Table 9: Wide ResNet 28-10 results.

Dataset	Method	Near OOD	Far OOD	NLL	ECE	Brier	Accuracy
CIFAR 10	DE	90.31 ± 0.18	93.70 ± 0.18	11.87 ± 0.10	0.94 ± 0.14	5.27 ± 0.11	96.59 ± 0.15
	NCL	90.26 ± 0.37	93.44 ± 0.20	12.20 ± 0.34	0.78 ± 0.16	5.39 ± 0.24	96.48 ± 0.19
	ADP	88.81 ± 0.49	92.86 ± 0.41	12.77 ± 0.29	0.75 ± 0.12	5.39 ± 0.15	96.57 ± 0.12
	DICE	87.37 ± 0.39	91.52 ± 0.20	14.44 ± 0.32	1.09 ± 0.10	6.03 ± 0.18	96.17 ± 0.16
	SDDE (Our)	90.40 ± 0.23	94.30 ± 0.22	11.80 ± 0.17	0.94 ± 0.15	5.16 ± 0.11	96.65 ± 0.06
CIFAR 100	DE	83.03 ± 0.20	81.27 ± 0.69	66.32 ± 0.47	3.55 ± 0.30	24.48 ± 0.35	82.99 ± 0.33
	NCL	83.02 ± 0.23	80.47 ± 0.20	66.46 ± 0.54	3.49 ± 0.30	24.45 ± 0.09	83.05 ± 0.11
	ADP	82.48 ± 0.26	79.78 ± 0.98	70.75 ± 0.55	3.50 ± 0.11	24.85 ± 0.33	83.02 ± 0.22
	DICE	82.17 ± 0.78	80.11 ± 1.17	72.36 ± 0.97	3.99 ± 0.12	25.46 ± 0.57	82.56 ± 0.56
	SDDE (Our)	83.54 ± 0.12	82.81 ± 0.42	65.66 ± 0.20	3.39 ± 0.09	24.17 ± 0.14	83.34 ± 0.15

D NUMBER OF MODELS

The key parameter of any ensemble is its number of models. A large ensemble usually provides a better confidence estimate, but its size negatively affects the computation speed. In order to determine the best number of models to use without incurring losses, we analyze the dependency of quality on ensemble size. The results for classification and OOD detection quality are presented in Figure 6. It can be seen that SDDE effectively uses multiple models, leading to better results for all ensemble sizes, starting from 3 in the CIFAR-10 and CIFAR-100 Near and Far setups.

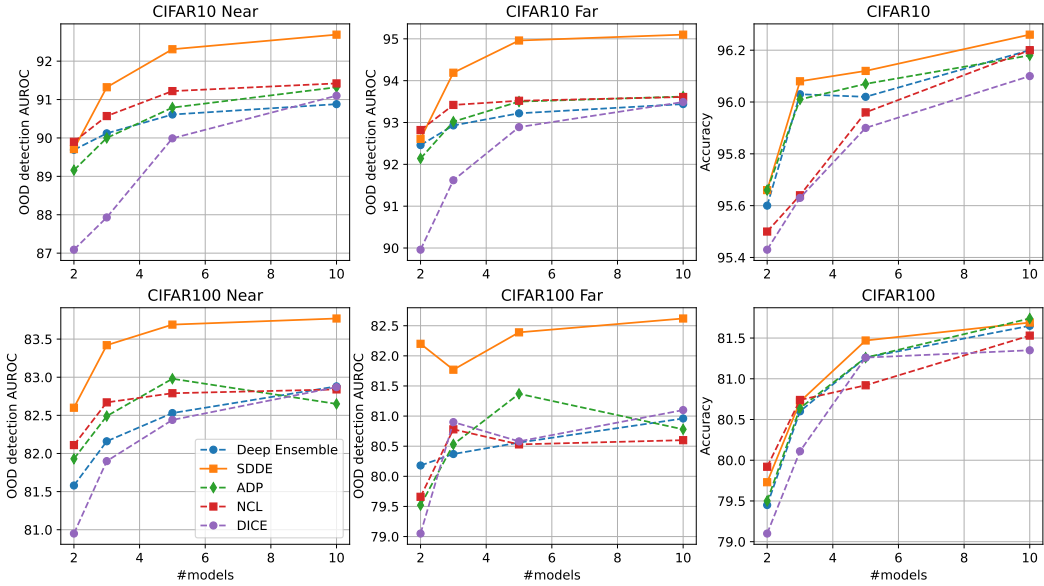


Figure 6: Classification accuracy and OOD detection quality depending on ensemble size.

E LOGIT AGGREGATION ABLATION

In Section 4, we introduced logit-based prediction aggregation for the proposed SDDE approach. Although a single-model maximal logit score can be used for OOD detection Hendrycks et al. (2019), it is still unclear whether logit averaging is applicable for ensemble OOD detection. To determine this, we analyze the outputs of individual models in the ensemble.

As demonstrated in Figure 7, the logits exhibit similar scales and distributions. While the figure presents only the CIFAR100 dataset and a single class, the figures for other datasets and classes exhibit similar structures. Based on this observation, we conclude that individual models contribute equally to the final OOD score, thereby improving the robustness of the OOD detection score estimation.

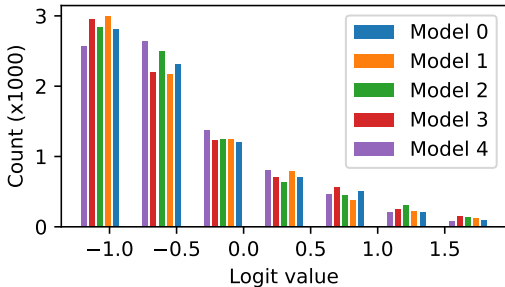


Figure 7: Logit distribution for individual models in the SDDE ensemble for the first class in CIFAR100.

As our baselines use probability averaging aggregation, we ablate logit aggregation for SDDE. In this experiment, we apply the proposed MAL aggregation to all baseline methods. The results are presented in Table 10. Our findings show that MAL improves the quality of the baseline models. At the same time, SDDE achieves higher scores in 6 out of 8 comparisons. The comparison of methods trained with OOD data is presented in Table 11. It can be seen that SDDE_{OOD} achieves the highest scores in 5 out of 6 comparisons.

Table 10: Near / Far OOD detection AUROC comparison for all methods with MAL aggregation.

Dataset	DE	NCL	ADP	DICE	SDDE (our)
MNIST	93.45 / 99.33	93.69 / 99.35	93.90 / 99.28	93.88 / 99.24	96.24 / 99.70
CIFAR10	91.66 / 94.36	91.67 / 94.24	91.08 / 94.02	90.44 / 93.64	91.90 / 94.55
CIFAR100	83.36 / 82.03	83.63 / 81.89	83.04 / 81.55	83.62 / 83.40	83.69 / 82.39
ImageNet	75.21 / 85.80	74.89 / 85.88	75.82 / 84.03	N/A	75.26 / 85.98

Table 11: Comparison of OOD score aggregation methods for the SDDE_{OOD} method.

Method	CIFAR10		CIFAR100		ImageNet200	
	Near	Far	Near	Far	Near	Far
OE-Ensemble _{MSP}	96.25 ± 0.14	97.34 ± 0.14	89.40 ± 0.02	88.34 ± 0.72	84.05 ± 0.19	88.77 ± 0.37
OE-Ensemble _{MAL}	96.27 ± 0.01	97.40 ± 0.19	89.59 ± 0.04	84.41 ± 0.97	84.41 ± 0.15	89.45 ± 0.26
SDDE_{MSP}	96.27 ± 0.07	95.57 ± 0.05	89.64 ± 0.17	85.38 ± 1.76	84.10 ± 0.10	88.95 ± 0.15
SDDE_{MAL}	96.22 ± 0.08	97.60 ± 0.08	89.70 ± 0.22	85.47 ± 1.76	84.46 ± 0.07	89.57 ± 0.11

F SALIENCY MAPS VS FEATURE DIVERSITY

In Section 3, we established a link between the diversity observed in saliency maps space and the diversity in the prediction space of models. However, an unresolved question remains: does the diversity in saliency maps contribute to diversity in the feature space? To address this, we analyzed

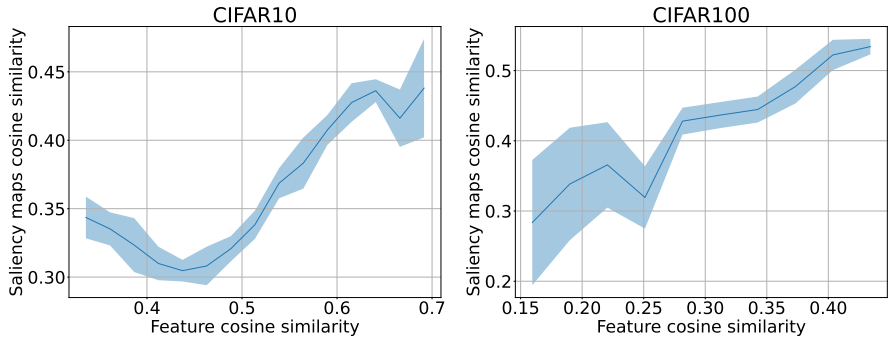


Figure 8: The dependency of ensemble saliency maps cosine similarity on feature cosine similarities. Saliency maps are computed using GradCAM. Mean and STD values w.r.t. multiple training seeds for the CIFAR10/100 datasets are reported.

the cosine similarities between the saliency maps and the models’ penultimate layers’ features. The findings are depicted in Figure 8. It’s evident from the data that higher similarities in the feature space correspond to increased similarities in saliency maps. Consequently, we infer that enhancing the diversity of saliency maps compels ensemble models to utilize distinct features for making predictions.

G COMPUTATIONAL COMPLEXITY

We have conducted a comparative analysis of the computational cost of SDDE training compared to other methods. According to our measurements on the NVIDIA V100 GPU, with training details described in Section 5.1 on CIFAR10, training a single SDDE model takes 2.9 hours using a single V100 GPU. Deep Ensemble takes 6.4 hours, and Deep Ensemble without adversarial loss takes 1.4 hours. The inference time is not affected by diversity loss and remains identical for all ensemble methods. Similarly, during training, SDDE, Deep Ensemble, and Deep Ensemble without adversarial loss consume 6.2 GB, 6.8 GB, and 3.9 GB of GPU memory, respectively.

H SDDE TRAINING ALGORITHM

SDDE training algorithm is presented in Listing 1. During cross-entropy (CE) loss computation, each model receives its own batch. The diversity loss is computed for a single batch, forwarded through each model. The training procedure with OOD sample is presented in Listing 2. The loss is extended with Outlier Exposure (OE) objective Hendrycks et al. (2018), computed on the OOD batch. Similar to OE training, both in-distribution and OOD batches are concatenated and processed via single forward pass. This way, batch normalization statistics are adapted to OOD data.

Algorithm 1 SDDE training**Input:** $\mathcal{D}, N, T, \lambda, \epsilon$ **Output:** Trained weights $\theta_k, k \in \overline{1, N}$

```

1: Initialize  $\theta_k, k \in \overline{1, N}$ 
2: for  $i \leftarrow 1$  to  $T$  do
3:    $l \leftarrow 0$ 
4:   for  $k \leftarrow 1$  to  $N$  do
5:     Sample batch  $(x_k^i, y_k^i)$  from  $\mathcal{D}$ 
6:      $l \leftarrow l + \frac{1}{N} \mathcal{L}_{CE}(x_k^i, y_k^i; \theta_k)$ 
7:   end for
8:   Sample batch  $(\hat{x}_k, \hat{y}_k)$  from  $\mathcal{D}$ 
9:    $l \leftarrow l + \lambda \mathcal{L}_{div}(\hat{x}_k, \hat{y}_k; \theta_1, \dots, \theta_N)$ 
10:  for  $k \leftarrow 1$  to  $N$  do
11:     $\theta_k \leftarrow \theta_k - \epsilon \nabla_{\theta_k} l$ 
12:  end for
13: end for

```

Algorithm 2 SDDE training with OOD data**Input:** $\mathcal{D}, \mathcal{D}_{OOD}, N, T, \lambda, \epsilon$ **Output:** Trained weights $\theta_k, k \in \overline{1, N}$

```

1: Initialize  $\theta_k, k \in \overline{1, N}$ 
2: for  $i \leftarrow 1$  to  $T$  do
3:    $l \leftarrow 0$ 
4:   for  $k \leftarrow 1$  to  $N$  do
5:     Sample batch  $(x_k^i, y_k^i)$  from  $\mathcal{D}$ 
6:     Sample unlabeled batch  $(\bar{x}_k^i)$  from  $\mathcal{D}_{OOD}$ 
7:      $l \leftarrow l + \frac{1}{N} \mathcal{L}_{CE}(x_k^i, y_k^i; \theta_k) + \frac{\beta}{N} \mathcal{L}_{OE}(\bar{x}_k^i; \theta_k)$ 
8:   end for
9:   Sample batch  $(\hat{x}_k, \hat{y}_k)$  from  $\mathcal{D}$ 
10:   $l \leftarrow l + \lambda \mathcal{L}_{div}(\hat{x}_k, \hat{y}_k; \theta_1, \dots, \theta_N)$ 
11:  for  $k \leftarrow 1$  to  $N$  do
12:     $\theta_k \leftarrow \theta_k - \epsilon \nabla_{\theta_k} l$ 
13:  end for
14: end for

```
