AdvProb: Adversarial Probes for Out-of-Distribution Detection

Anonymous Author(s)
Affiliation
Address
email

Abstract

Neural networks frequently yield overconfident predictions when encountering out-of-distribution (OOD) samples, undermining their reliability in critical real-world tasks. In this paper, we introduce **AdvProb:** Adversarial Probes, a novel diagnostic framework for robust OOD detection. Our approach applies multiple targeted adversarial perturbations to each input, systematically probing the local stability of model predictions. By analyzing how model confidence shifts under these perturbations, we construct a comprehensive behavioral fingerprint for each input and train an XGBoost ensemble to robustly discriminate between in-distribution (ID) and OOD data. AdvProb substantially improves the OOD detection performance of standard classifiers and can be seamlessly integrated into existing methods like ODIN and Mahalanobis, yielding consistent performance gains across architectures and datasets. Our results highlight adversarial probes as a flexible and highly effective tool for enhancing OOD detection robustness.

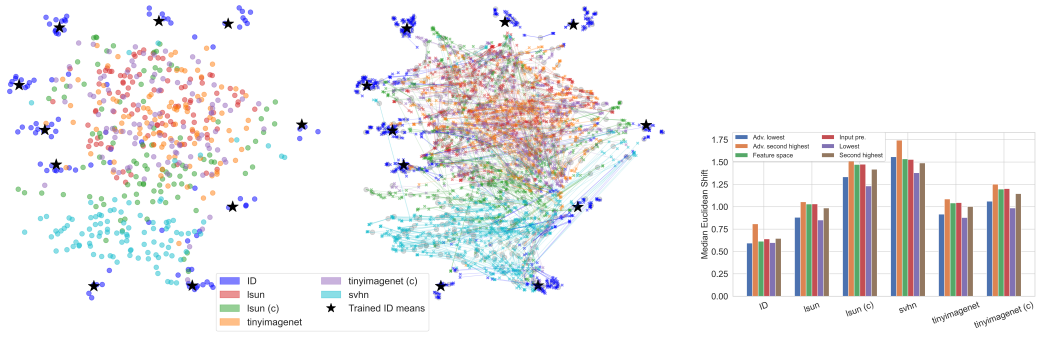

Figure 1: Experimental result on the ResNet model with 18 layers. Left: UMAP projection on the final feature layer of the ResNet model with (unperturbed) ID and OOD images. Middle: UMAP projection illustrating the displacements shift after the adversarial perturbations. Right: Mean Euclidean distance shift for each adversarial probe for ID and OOD images.

1 Introduction

Computer vision models often produce highly confident predictions on images that differ from the training distribution, known as out-of-distribution (OOD) samples. Such overconfidence threatens the reliability and safety of AI systems deployed in critical real-world settings. For example, in

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

autonomous driving, misclassifying novel roadside objects may lead to unsafe decisions, or medical image classifiers might incorrectly diagnose rare conditions, undermining their clinical reliability.

Previous OOD detection work have proposed adjusting or replacing the softmax confidence scores to improve the model's tendency on predicting high confidence scores on OOD samples. For instance, ODIN applies a high-temperature softmax to amplify the confidence gap between in- and out-of-distribution samples [1]. In contrast, Mahalanobis-based methods fit class-conditional Gaussian in intermediate feature space and use the negative Mahalanobis distance to the nearest class centroid as an OOD score [2], while energy-based approaches threshold the negative log-sum-exp of the logits to reduce overconfidence on anomalies [3].

Other OOD strategies instead try to modify the input image directly to improve ID-OOD separability. For example, ODIN [1] introduced an input Preprocessing method that adds a small perturbation on the original image to increase the model's confidence score. This strategy is inspired by adversarial attacks [4] but aims to increase confidence not reduce it. Generalized ODIN [5] extends this by learning both the temperature and perturbation magnitude on held-out in-distribution validation data and by training a lightweight calibration head on penultimate features—thereby removing any need for OOD tuning. Subsequent work has applied a broader range of test-time image manipulations, AdaScale [6] dynamically rescales inputs to exploit scale-dependent confidence shifts—ID samples remain stable across scales while OOD confidences fluctuate—while CoVer [7] subjects inputs to common corruptions (e.g. blur, noise) and averages the resulting predictions to accentuate the confidence drop of OOD samples.

A third line of work trains small auxiliary detectors on signals extracted from a fixed backbone. For example, Monte Carlo Dropout [8] approximates Bayesian uncertainty by averaging predictions over stochastic dropout masks, and Deep Ensembles [9] aggregate outputs from multiple independently trained models. Other approaches pull single-pass signals: OpenMax [10] fits an EVT-based meta-classifier on penultimate activations, GradNorm [11] thresholds the norm of input gradients, and ReAct [12] clips extreme activations to suppress OOD spikes.

In this work, we propose *adversarial probes*, a multi-directional probing framework that systematically diagnoses model prediction stability via targeted adversarial perturbations. Specifically, we generate multiple adversarial perturbations per input—each optimized to increase or decrease confidence in specific classes or to align feature embeddings with class prototypes. Our central hypothesis is that ID samples, residing within dense and stable regions of feature space, exhibit relatively stable confidence responses to these small perturbations. In contrast, OOD samples tend to occupy sparse or boundary regions, resulting in larger fluctuations in confidence scores under adversarial perturbations.

We collect per-probe confidence scores and binary flip indicators into a feature vector and train an XGBoost classifier to distinguish ID from OOD. This approach captures complex, non-linear interactions among multiple probe responses, yielding a robust OOD detector that generalizes across datasets (CIFAR-10/100, SVHN, TinyImageNet, LSUN) and architectures (ResNet, DenseNet). Our Contributions are summarized as follows:

- We introduce AdvProb, a multi-objective adversarial probing diagnostic framework.
- We combine the output confidence scores of the adversarial probes with an XGBoost to ensemble and learn complex interactions across perturbation responses, significantly improving OOD detection performance.
- We empirically demonstrate the effectiveness and robustness of our approach on standard OOD benchmarks and give intuition why it works.

2 Method

Notation. Let $\mathcal{X}$ denote the input space, where each $x \in \mathcal{X}$ is the input. Let $f : \mathcal{X} \to \mathbb{R}^K$ be a pre-trained classifier mapping each input image to a vector of K logits. The predicted class probabilities are given by

$$S(x) = \mathrm{softmax}\big(f(x)\big) \in \mathbb{R}^K,$$

where $S_i(x)$ denotes the predicted probability for class i.

We define the following terms for any input image x:

- $\hat{y} := \arg\max_i S_i(x)$, the index of the most confident prediction,
- $y_2 := \arg\max_{i \neq \hat{y}} S_i(x)$, the index of the second most confident prediction,
- $y_{\text{least}} := \arg\min_i S_i(x)$, the index of the least confident prediction,
- $h(x)$, the feature embedding of x extracted from an intermediate layer of f,
- μ_c, the empirical mean embedding of class c computed over the training set,
- $\cos\_\text{sim}(a, b) = \frac{a \cdot b}{\|a\| \, \|b\|}$, the cosine similarity between vectors a and b.

Six Perturbation Objectives. AdvProbes generates perturbed inputs $\{\tilde{x}_i\}_{i=1}^{6}$, by using the Fast Gradient Sign Method (FGSM) [4]) by ascending or descending the gradient of various objective functions, each meant to alter the model's behavior in a different way:

$$\tilde{x}_1 \leftarrow x + \epsilon \cdot \text{sign}\big(\nabla_x S_{\hat{y}}(x)\big) \qquad \triangleright \text{Increase primary class confidence}$$

$$\tilde{x}_2 \leftarrow x - \epsilon \cdot \text{sign}\big(\nabla_x S_{y_2}(x)\big) \qquad \triangleright \text{Decrease second-highest class confidence}$$

$$\tilde{x}_3 \leftarrow x - \epsilon \cdot \text{sign}\big(\nabla_x S_{y_{\text{least}}}(x)\big) \qquad \triangleright \text{Decrease lowest class confidence}$$

$$\tilde{x}_4 \leftarrow x + \epsilon \cdot \text{sign}\big(\nabla_x S_{y_2}(x)\big) \qquad \triangleright \text{Increase second-highest class confidence}$$

$$\tilde{x}_5 \leftarrow x + \epsilon \cdot \text{sign}\big(\nabla_x S_{y_{\text{least}}}(x)\big) \qquad \triangleright \text{Increase lowest class confidence}$$

$$\tilde{x}_6 \leftarrow x + \epsilon \cdot \text{sign}\big(\nabla_x \cos\_\text{sim}(h(x), \mu_{\hat{y}})\big) \qquad \triangleright \text{Increase similarity to class mean}$$

Note that $\tilde{x}_1$ is the input pre-processing strategy proposed by [1]. After generating the six perturbed inputs $\tilde{x}_1, \ldots, \tilde{x}_6$, we record the softmax confidence assigned to the model's original prediction $\hat{y}$, i.e., $S_{\hat{y}}(\tilde{x}_i)$. In addition to confidence scores, we include two binary indicators that track whether the original prediction $\hat{y}$ is preserved under the adversarial perturbations $\tilde{x}_4$ and $\tilde{x}_5$, defined as $\mathbb{I}(\hat{y} = \arg\max_i S_i(\tilde{x}_4))$ and $\mathbb{I}(\hat{y} = \arg\max_i S_i(\tilde{x}_5))$ respectively. Lastly, we also keep track of the original confidence score of the model $S_{\hat{y}}(x)$ which would be useful for the model to learn.

Together, these nine values form the vector $\mathbf{z}(x)$:

$$\begin{aligned}\mathbf{z}(x) = [&S_{\hat{y}}(x), S_{\hat{y}}(cos\_sim(h(x), \mu_{\hat{y}}),\\ &S_{\hat{y}}(\tilde{x}_1), S_{\hat{y}}(\tilde{x}_2), S_{\hat{y}}(\tilde{x}_3), S_{\hat{y}}(\tilde{x}_4), S_{\hat{y}}(\tilde{x}_5), S_{\hat{y}}(cos\_sim(h(x_6), \mu_{\hat{y}})),\\ &\mathbb{I}(\hat{y} = \arg\max_i S_i(\tilde{x}_4)), \mathbb{I}(\hat{y} = \arg\max_i S_i(\tilde{x}_5))]\end{aligned}$$

We then supply $\mathbf{z}(x)$ as input for an XGBoost classifier to obtain the final ID/OOD scores for each image.

3 Experimental Setup

Datasets. We use CIFAR-10 and CIFAR-100 [13] as in-distribution (ID) datasets. For out-of-distribution (OOD) evaluation, we use SVHN [14], ImageNet [15], and LSUN [16]. Since ImageNet and LSUN images have higher resolution than the CIFAR-trained models expect, we evaluate both their cropped (c) and resized (r) variants to match the input dimensions. Lastly, to train the XGBoost classifier we used iSUN [17] as the OOD dataset, and evaluate it on the remaining OOD datasets to test generalization to unseen OOD distribution.

Models. We evaluate on four models: a DenseNet-100-BC and a ResNet-18, each trained separately on CIFAR-10 and CIFAR-100. The DenseNet models follow the implementation from the official ODIN repository [1], and the ResNet-18 models are taken from the OpenOOD benchmark repository [18]. We use the publicly released pretrained weights for all models.

Evaluation. We use three standard OOD detection metrics: FPR@95%TPR (false positive rate when 95% of ID samples are correctly accepted), AUROC (area under the receiver operating characteristic curve), and Detection Accuracy (maximum average of ID and OOD classification accuracy across all thresholds). Note that the threshold τ is selected independently for each metric.

Setting	AUROC ↑		FPR@95 ↓		Acc ↑	
	Baseline	AdvProb	Baseline	AdvProb	Baseline	AdvProb
CIFAR-10 / DenseNet-100	93.3	**96.3**	21.1	**15.6**	87.4	**90.3**
CIFAR-10 / ResNet-18	92.4	**97.0**	22.3	**13.9**	87.4	**91.3**
CIFAR-100 / DenseNet-100	76.2	**86.2**	62.3	**46.7**	69.7	**78.7**
CIFAR-100 / ResNet-18	81.9	**93.0**	50.4	**27.3**	74.8	**86.1**

Table 1: AdvProb versus unperturbed baseline [19]. The results are the average of the 5 OOD datasets across. The values are percentages and the best values are highlighted in bold.

Perturbation Strength (ϵ). For all adversarial probes, we follow the Generalized ODIN strategy [5] and select the perturbation strength ϵ using only in-distribution validation data. In Section 5 we show that AdvProb is robust to a wide range of ϵ values, with strong performance across a wide range of values—unlike standard input pre-processing methods which require careful per-dataset tuning.

4 Results

AdvProb improves on Baseline. We begin by evaluating AdvProb as a standalone OOD detector. Table 1 summarizes the results averaging performance across five OOD benchmarks. AdvProb consistently outperforms the (unperturbed) baseline on both ResNet-18 and DenseNet-100 architectures trained on CIFAR-10 and CIFAR-100, improving AUROC by up to +11.1 points (from 81.9 to 93.0) and reducing FPR@95 by up to –23.1 points (from 50.4 to 27.3). These empirical gains highlight that even small, targeted adversarial perturbations can reveal meaningful differences in model behavior. By systematically probing a model in multiple adversarial directions, AdvProb captures the local prediction stability and robustness, which proves highly informative for OOD detection. For per-dataset results and comparisons with input pre-processing, see Appendix A.

Plugin enhancements via AdvProb. Because AdvProb operates purely at the input level, it can be dropped into any confidence-based OOD detector in place of its standard pre-processing step. Table 2 shows the OOD detection performance on 5 OOD benchmarks when AdvProb is used with Odin [1] and Mahalanobis strategies [2]. ODIN+AdvProb consistently outperforms vanilla ODIN, with AUROC mean gains between +0.4 pts and +13.8 pts and FPR@95 reductions of –3.8 pts to –61.8 pts, pushing all settings into the high-90s. When applied to Mahalanobis strategy, AdvProb yields even larger lifts: AUROC increases by +5.7 pts to +21.6 pts and FPR@95 drops by –25.1 pts to –60.8 pts. For example, on CIFAR-10 with DenseNet-100 model, Mahalanobis+AdvProb achieves an average of 99.0 % AUROC (vs. 68.9 %) and 4.0 % FPR@95 (vs. 64.8 %) showing nearly perfect detection[1]

5 Ablation and Objective Analysis

Effects of Probes on Confidence and Stability. We analyze how each adversarial probe $\tilde{x}_i$ alters the confidence and prediction stability for in-distribution (ID) and out-of-distribution (OOD) inputs of the model. Figure 2 shows that the adversarial probes $\tilde{x}_1$ (increase top-class confidence), $\tilde{x}_2$ (decrease second-highest class confidence), and $\tilde{x}_3$ (decrease lowest class confidence) produce noticeably larger confidence shifts on ID samples than on OOD samples, signaling tighter alignment with ID decision boundaries. In contrast, the adversarial probes $\tilde{x}_4$ (increase second-highest class confidence) and $\tilde{x}_5$ (increase lowest class confidence) cause OOD predictions to flip classes far more frequently specially at smaller perturbation strengths, while ID predictions remain largely stable as shown by the flip-rate curves in Figure 3. Finally, $\tilde{x}_6$ induces only mild confidence changes yet still keeps ID samples more tightly clustered around their prototypes in feature space. Notice that smaller ϵ values (e.g. 0.015) tent to have a better separation between ID and OOD samples. By combining these different set adversarial probes we are essentially stress testing the model's prediction

[1]For both ODIN and Mahalanobis, we select the perturbation magnitude ϵ per model via in-distribution validation—following the Generalized ODIN protocol [5]—instead of tuning on held-out OOD samples, mirroring our AdvProb setup described in Sec. 3.

ID Dataset	OOD Dataset	Baseline/ ODIN / ODIN+AdvProb			Baseline/ Mahalanobis / Mahalanobis+AdvProb		
		AUROC ↑	FPR@95 ↓	Acc ↑	AUROC ↑	FPR@95 ↓	Acc ↑
CIFAR-10 (DenseNet)	LSUN(r)	95.5 / 99.2 / **99.4**	14.6 / 4.0 / **2.72**	90.3 / 95.6 / **96.4**	95.5 / 94.7 / **99.9**	14.6 / 26.1 / **0.2**	90.3 / 87.9 / **98.5**
	LSUN(c)	93.2 / 94.2 / **94.6**	20.5 / 26.8 / **19.5**	87.3 / 87.2 / **88.3**	93.2 / 56.2 / **97.5**	20.5 / 99.9 / **10.2**	87.3 / 61.0 / **92.7**
	TinyImageNet(r)	94.1 / 98.5 / **98.6**	19.3 / 7.2 / **6.6**	88.3 / 94.0 / **94.3**	94.1 / 93.8 / **99.8**	19.3 / 34.1 / **1.0**	88.3 / 87.0 / **97.8**
	TinyImageNet(c)	93.9 / 97.6 / **97.7**	20.0 / **11.4** / 11.6	88.0 / **92.3** / 91.9	93.8 / 87.3 / **99.4**	20.0 / 63.7 / **2.8**	88.0 / 80.7 / **96.4**
	SVHN	90.0 / 89.8 / **91.1**	31.0 / 37.5 / **27.4**	83.3 / 81.5 / **84.4**	90.0 / 12.6 / **98.2**	31.1 / 100.0 / **6.0**	83.3 / 50.0 / **94.7**
	Average	93.3 / 95.9 / **96.3**	21.1 /17.4 / **13.6**	87.4 / 90.1 / **91.1**	93.3 / 68.9 / **99.0**	21.1 / 64.8 / **4.0**	87.4 / 73.3 / **96.0**
CIFAR-10 (ResNet)	LSUN(r)	93.9 / 94.6 / **98.5**	18.8 / 35.9 / **7.0**	88.6 / 89.3 / **94.1**	93.9 / 99.2 / **99.9**	18.8 / 3.7 / **0.4**	88.6 / 95.8 / **98.4**
	LSUN(c)	92.9 / 88.5 / **96.9**	21.3 / 70.8 / **15.7**	87.8 / 83.6 / **90.9**	92.9 / 89.7 / **96.5**	21.3 / 41.4 / **14.9**	87.8 / 82.7 / **91.3**
	TinyImageNet(r)	91.9 / 90.7 / **97.2**	24.9 / 58.3 / **13.4**	86.9 / 82.5 / **91.4**	91.9 / 98.9 / **99.7**	24.9 / 5.5 / **1.0**	86.9 / 94.8 / **97.8**
	TinyImageNet(c)	92.6 / 91.1 / **97.0**	22.0 / 55.7 / **13.4**	87.6 / 85.5 / **91.3**	92.6 / 97.2 / **99.3**	22.0 / 14.8 / **2.7**	87.6 / 91.2 / **96.3**
	SVHN	90.6 / 85.8 / **95.6**	24.6 / 67.5 / **17.9**	86.0 / 79.3 / **89.3**	90.6 / **98.2** / 97.9	24.6 / **7.2** / 8.0	86.0 / **94.0** / 93.5
	Average	92.4 / 83.2 / **97.0**	22.3 / 75.3 / **13.5**	87.4 / 77.0 / **91.4**	92.4 / 96.6 / **98.7**	22.3 / 14.5 / **5.4**	87.4 / 91.7 / **95.5**
CIFAR-100 (DenseNet)	LSUN(r)	69.1 / 84.7 / **94.0**	71.9 / 51.2 / **24.7**	64.0 / 76.8 / **87.1**	69.1 / 78.0 / **99.5**	71.9 / 50.5 / **1.5**	64.0 / 73.7 / **97.3**
	LSUN(c)	80.1 / **91.1** / 84.1	59.2 / **35.6** / 52.4	72.8 / **83.2** / 76.7	80.1 / 74.3 / **90.4**	59.2 / 77.4 / **34.2**	72.8 / 68.8 / **83.4**
	TinyImageNet(r)	71.2 / 85.5 / **94.3**	70.5 / 52.2 / **22.7**	65.5 / 77.5 / **87.5**	71.2 / 79.8 / **99.2**	70.5 / 53.1 / **3.9**	65.5 / 73.7 / **95.7**
	TinyImageNet(c)	77.0 / 89.7 / **92.6**	65.2 / 42.6 / **30.6**	69.9 / 81.7 / **85.1**	77.0 / 71.0 / **97.1**	65.2 / 71.3 / **15.3**	69.9 / 66.4 / **91.0**
	SVHN	83.4 / **88.2** / 84.6	44.8 / **37.8** / 51.2	76.2 / **81.0** / 76.7	83.4 / 86.9 / **96.9**	44.8 / 58.0 / **11.3**	76.2 / 79.6 / **91.9**
	Average	76.2 / 87.8 / **89.9**	62.3 / 43.9 / **36.3**	69.7 / 80.0 / **82.6**	76.2 / 78.0 / **96.6**	62.3 / 62.1 / **13.2**	69.7 / 72.4 / **92.5**
CIFAR-100 (ResNet)	LSUN(r)	81.9 / 90.5 / **96.1**	50.8 / 34.7 / **15.5**	74.6 / 82.9 / **90.7**	81.9 / 95.2 / **99.2**	50.8 / 22.6 / **4.0**	74.6 / 88.1 / **95.7**
	LSUN(c)	81.4 / **89.5** / 89.4	50.9 / **38.7** / 41.7	74.2 / **81.8** / 81.3	81.4 / 83.5 / **91.8**	50.9 / 64.3 / **32.3**	74.2 / 75.4 / **84.9**
	TinyImageNet(r)	81.8 / 90.5 / **95.9**	50.6 / 35.0 / **16.3**	74.7 / 83.0 / **90.1**	81.8 / 94.5 / **98.8**	50.6 / 27.4 / **6.0**	74.7 / 87.7 / **94.7**
	TinyImageNet(c)	84.2 / 90.5 / **94.3**	46.6 / 35.1 / **24.0**	76.7 / 82.8 / **87.6**	84.2 / 90.1 / **97.7**	46.6 / 43.3 / **11.5**	76.7 / 81.8 / **92.5**
	SVHN	80.3 / 91.7 / **92.5**	52.9 / 31.3 / **27.6**	73.9 / 84.1 / **85.7**	80.3 / 89.2 / **95.1**	52.9 / 37.9 / **16.4**	73.9 / 81.7 / **89.7**
	Average	81.9 / 90.53 / **93.6**	50.4 / 35.0 / **25.0**	74.8 / 82.9 / **87.1**	81.9 / 90.5 / **96.5**	50.4 / 39.1 / **14.0**	74.8 / 82.9 / **91.5**

Table 2: Comparison of OOD detection performance across ODIN (left) and Mahalanobis (right) with and without AdvProb. The values are percentages and the best values are highlighted in bold.

stability and the XGBoost head learns a non-linear combination of the resulting per-probe scores to maximise ID–OOD separability.

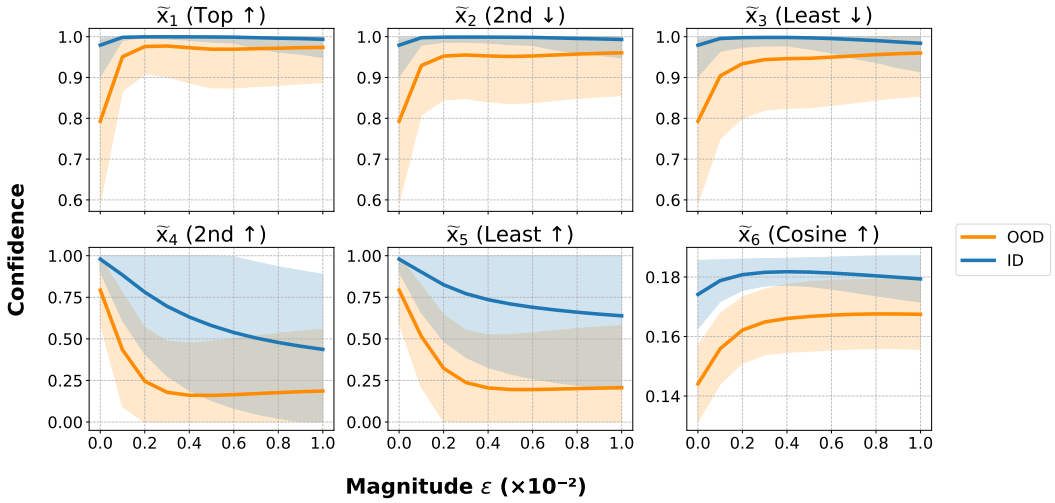

Figure 2: Confidence bounds across perturbation magnitude ϵ for each of the six AdvProb objectives - see section 2 for definitions. Lines show the mean confidence gain for in-distribution data (ID, blue) and the mean OOD response (bold orange), with the shaded band indicating ± 1 std.

Better OOD Performance with More Adversarial Probes. As shown in Figure 4, adding more adversarial probes consistently improves OOD detection performance. As we include probes one by one (curves labeled "+1 probe" through "+6 probes"), both AUROC increases and FPR@95 decreases, with the full six-probe configuration yielding the best results. We also sweep the perturbation magnitude ϵ from 0 to 0.01 and observe that AdvProb maintains strong performance across this entire range—unlike standard input preprocessing, which peaks narrowly around $\epsilon \approx 0.0015$ and degrades sharply outside it, consistent with findings in [1]. These results indicate that (1) each probe contributes complementary signals useful for OOD discrimination, and (2) AdvProb is robust to a wide range of ϵ values, offering greater flexibility than single-step methods. Additional results

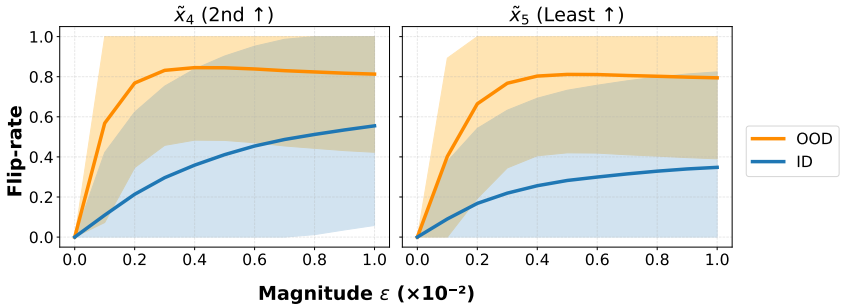

Figure 3: Flip-rate vs. perturbation magnitude ϵ ($\times 10^{-2}$). Flip-rate measures the proportion of samples whose predicted class changes after applying the perturbation. Lines indicate flip-rate for in-distribution data (ID, blue) and mean OOD response (bold orange), with the shaded area representing ± 1 std across OOD datasets.

in Appendix G further analyze the effect of each individual probe and their combinations across different model architectures.

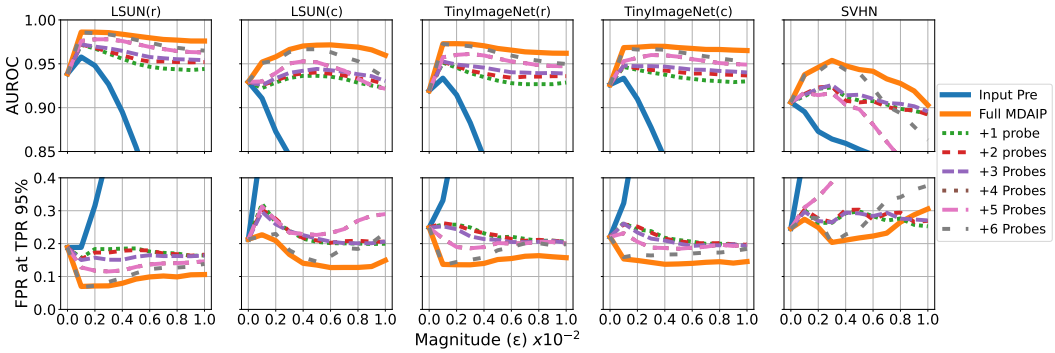

Figure 4: Incremental performance gain on AUROC and FPR@95 as more adversarial probes are included in AdvProb.

Feature Space Displacement under Perturbation. We further examine how adversarial probes affect the model's internal feature representations. As shown in the right panel of Figure 1, the mean Euclidean distance between the original and perturbed feature embeddings is consistently smaller for ID samples than for OOD samples across all probes. This indicates that ID inputs reside in more stable regions of feature space, while OOD inputs are more sensitive to small, targeted perturbations. These findings support our central assumption that ID samples are more locally stable under small, targeted perturbations—a property that AdvProb exploits through confidence-based probing to distinguish ID from OOD inputs.

6 Future Work

Future Work. While our proposed set of adversarial probes captures distinct aspects of model robustness, it is likely that additional perturbation objectives could reveal complementary signals relevant for OOD detection. Exploring a broader space of perturbation directions-particularly those informed by gradient alignment, feature attribution, or class-wise prototypes-may further enrich the behavioral signature. Moreover, extending beyond single-step FGSM-style attacks to multi-step or optimization-based perturbations could enhance sensitivity to more subtle robustness gaps. Finally, applying this probing framework to other modalities, such as text or audio, presents a promising direction for generalizing adversarial diagnostics across domains

7 Conclusion

We introduced *Adversarial Probes* (AdvProb), a new method for out-of-distribution (OOD) detection in computer vision models. AdvProb applies multiple adversarial perturbations to assess model stability across a diverse range of objectives. It can be integrated with existing OOD detection methods, such as ODIN and Mahalanobis distance, to further enhance performance. While the need to compute multiple perturbations limits real-time scalability, AdvProb remains a practical and general framework for improving detection in high-stakes or batch-processing scenarios. Future work will explore stronger perturbation strategies, adaptive probe selection, and extensions to other data modalities.

References

[1] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. 6 2017.

[2] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper$_f iles/paper/2018/file/abdeb6f575ac5c6676b747bca8d09cc2-Paper.pdf$.

[3] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 33, pages 21464–21475, 2020.

[4] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. 3 2015. URL http://arxiv.org/abs/1412.6572.

[5] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. 3 2020. URL http://arxiv.org/abs/2002.11297.

[6] Sudarshan Regmi. Adascale: Adaptive scaling for ood detection. 3 2025. URL http://arxiv.org/abs/2503.08023.

[7] Boxuan Zhang, Jianing Zhu, Zengmao Wang, Tongliang Liu, Bo Du, and Bo Han. What if the input is expanded in ood detection? 10 2024. URL http://arxiv.org/abs/2410.18472.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. 10 2016. URL http://arxiv.org/abs/1506.02142.

[9] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. 11 2017. URL http://arxiv.org/abs/1612.01474.

[10] Abhijit Bendale and Terrance Boult. Towards open set deep networks. 11 2015. URL http://arxiv.org/abs/1511.06233.

[11] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. 6 2018. URL http://arxiv.org/abs/1711.02257.

[12] Jindong Sun, Yilun Jin, and Yixuan Li. React: Out-of-distribution detection with rectified activations. In *Advances in Neural Information Processing Systems*, 2021.

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. URL https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf.

[14] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011$_h ousenumbers.pdf$.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[16] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015. URL `http://arxiv.org/abs/1506.03365`.

[17] iSUN dataset. `https://paperswithcode.com/dataset/isun`. Accessed: 2024-07-14.

[18] Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, Wenxuan Peng, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Dan Hendrycks, Yixuan Li, and Ziwei Liu. Openood: Benchmarking generalized out-of-distribution detection. 10 2022. URL `http://arxiv.org/abs/2210.07242`.

[19] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

A AdvProb Vs Input Preprocessing

ID Dataset	ID dataset	AUROC ↑	FPR@95 ↓	Detection acc. ↑
		baseline / Input pre. / AdvProb		
CIFAR-10 (DenseNet)	lsun(r)	95.5 / 98.7 / **99.3**	14.6 / 7.0 / **3.2**	90.3 / 94.0 / **96.0**
	lsun(c)	93.2 / 94.2 / **94.3**	**20.5** / 23.6 / 23.6	**87.3** / 87.1 / 86.7
	tinyimagenet(r)	94.1 / 97.6 / **98.3**	19.3 / 11.4 / **8.4**	88.3 / 91.9 / **93.3**
	tinyimagenet(c)	93.8 / 96.6 / **97.3**	20.0 / 14.8 / **14.3**	88.0 / 90.5 / **91.1**
	svhn	90.0 / **92.3** / **92.3**	31.1 / 28.7 / **28.4**	83.3 / **84.3** / **84.3**
	average	93.3 / 95.9 / **96.3**	21.1 / 17.1 / **15.6**	87.4 / 89.6 / **90.2**
CIFAR-10 (ResNet)	lsun(r)	93.9 / 91.21/ **98.5**	18.8 / 54.2 / **7.3**	88.6 / 85.1 / **93.9**
	lsun(c)	92.9 / 83.5 / **96.9**	21.3 / 82.3 / **14.9**	87.8 / 78.7 / **90.9**
	tinyimagenet(r)	91.9 / 86.4 / **97.2**	24.9 / 71.0 / **13.6**	86.9 / 80.4 / **91.4**
	tinyimagenet(c)	92.6 / 86.0 / **97.0**	22.0 / 72.3 / **13.4**	87.6 / 80.1 / **91.3**
	svhn	90.6 / 86.2 / **95.2**	24.6 / 69.8 / **20.1**	86.0 / 79.4 / **89.0**
	average	92.4 / 86.7 / **96.9**	22.3 / 69.9 / **13.9**	87.4 / 80.7 / **91.3**
CIFAR-100 (DenseNet)	lsun(r)	69.1 / 76.2 / **92.4**	71.9 / 65.4 / **29.0**	64.0 / 68.8 / **85.2**
	lsun(c)	80.1 / **84.7** / 72.9	59.2 / **52.8** / 79.4	72.8 / **75.6** / 66.1
	tinyimagenet(r)	71.2 / 78.8 / **92.8**	70.5 / 63.0 / **30.3**	65.5 / 70.8 / **85.3**
	tinyimagenet(c)	77.0 / 81.6 / **90.1**	65.2 / 59.7 / **41.1**	69.9 / 73.5 / **81.9**
	svhn	83.4 / **89.3** / 82.9	44.8 / **41.0** / 53.6	76.2 / **79.9** / 75.1
	average	76.2 / 82.1 / **86.2**	62.3 / 56.4 / **46.7**	69.7 / 73.7 / **78.7**
CIFAR-100 (ResNet)	lsun(r)	81.9 / 88.1 / **95.5**	50.8 / 42.2 / **18.2**	74.6 / 80.5 / **89.3**
	lsun(c)	81.4 / **88.7** / 88.1	50.9 / **40.9** / 45.1	74.2 / 80.4 / **80.5**
	tinyimagenet(r)	81.8 / 88.8 / **95.4**	50.6 / 40.9 / **18.5**	74.7 / 81.1 / **89.0**
	tinyimagenet(c)	84.2 / 88.8 / **94.0**	46.6 / 40.0 / **24.9**	76.7 / 81.1 / **87.0**
	svhn	80.3 / 91.0 / **91.8**	52.9 / 34.9 / **29.6**	73.9 / 82.6 / **84.8**
	average	81.9 / 89.0 / **93.0**	50.4 / 39.8 / **27.3**	74.8 / 81.1 / **86.1**

Table 3: Comparison of OOD detection results across baselines.

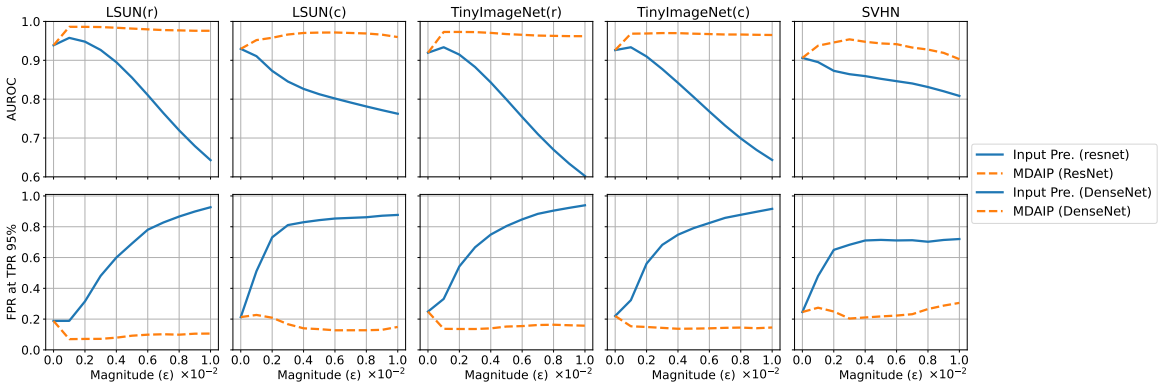

Figure 5: **Robustness of OOD detection to perturbation magnitude**. We compare AUROC (top) and FPR@95 (bottom) across a range of perturbation magnitudes (ϵ) for standard Input Preprocessing (blue) and our proposed AdvProb (orange dashed), using a ResNet model trained on CIFAR-10.

Table **??** shows the improvements of AdvProb compared to the traditional input preprocessing technique with a DenseNet model trained on the CIFAR-100 dataset and evaluated using the SVHN

as OOD dataset [2]. While standard preprocessing alone raises AUROC from 69.1% to 76.2% (and reduces FPR@95 from 71.9% to 65.4%), integrating AdvProb further boosts AUROC to 92.4% and cuts FPR@95 to 29.0%, demonstrating a substantial improvement over both the unprocessed baseline and the preprocessing-only variant. Table **??** extends the comparison between our method and Input Preprocessing method to all in- and out-of-destribution datasets. In 18 out of 20 cases, AdvProb strictly outperforms the Input Preprocessing strategy. In particular, on CIFAR-100 with a DenseNet backbone evaluated against LSUN(r) as OOD, AdvProb raises AUROC from $76.2\% \rightarrow 92.4\%$ and and cuts the false positive rate at 95% true positive rate (FPR@95) from $65.4\% \rightarrow 29.0\%$. On a small number of pairs (e.g. LSUN(c) on CIFAR-100 with DenseNet), AdvProb matches or slightly trails the Input Preprocessing strategy (explain why not perfect - isun as ood samples?). These results confirm that multi-directional adversarial preprocessing reliably enhances OOD separation across both datasets and network architectures.

[2] Since AdvProb inherently incorporates the standard input-preprocessing step, it is marked in both the "Input Preprocessing" and "AdvProb" columns in Table **??**.

B AdvProb blugin performance

ID Dataset	OOD Dataset	Baseline/ ODIN / ODIN+AdvProb			Baseline/ Mahalanobis / Mahalanobis+AdvProb		
		AUROC ↑	FPR@95 ↓	Acc ↑	AUROC ↑	FPR@95 ↓	Acc ↑
CIFAR-10 (DenseNet)	LSUN(r)	95.5 / 99.2 / **99.4**	14.6 / 4.0 / **2.72**	90.3 / 95.6 / **96.4**	95.5 / 94.7 / **99.9**	14.6 / 26.1 / **0.2**	90.3 / 87.9 / **98.5**
	LSUN(c)	93.2 / 94.2 / **94.6**	20.5 / 26.8 / **19.5**	87.3 / 87.2 / **88.3**	93.2 / 56.2 / **97.5**	20.5 / 99.9 / **10.2**	87.3 / 61.0 / **92.7**
	TinyImageNet(r)	94.1 / 98.5 / **98.6**	19.3 / 7.2 / **6.6**	88.3 / 94.0 / **94.3**	94.1 / 93.8 / **99.8**	19.3 / 34.1 / **1.0**	88.3 / 87.0 / **97.8**
	TinyImageNet(c)	93.9 / 97.6 / **97.7**	20.0 / **11.4** / 11.6	88.0 / **92.3** / 91.9	93.8 / 87.3 / **99.4**	20.0 / 63.7 / **2.8**	88.0 / 80.7 / **96.4**
	SVHN	90.0 / 89.8 / **91.1**	31.0 / 37.5 / **27.4**	83.3 / 81.5 / **84.4**	90.0 / 12.6 / **98.2**	31.1 / 100.0 / **6.0**	83.3 / 50.0 / **94.7**
	Average	93.3 / 95.9 / **96.3**	21.1 /17.4 / **13.6**	87.4 / 90.1 / **91.1**	93.3 / 68.9 / **99.0**	21.1 / 64.8 / **4.0**	87.4 / 73.3 / **96.0**
CIFAR-10 (ResNet)	LSUN(r)	93.9 / 94.6 / **98.5**	18.8 / 35.9 / **7.0**	88.6 / 89.3 / **94.1**	93.9 / 99.2 / **99.9**	18.8 / 3.7 / **0.4**	88.6 / 95.8 / **98.4**
	LSUN(c)	92.9 / 88.5 / **96.9**	21.3 / 70.8 / **15.7**	87.8 / 83.6 / **90.9**	92.9 / 89.7 / **96.5**	21.3 / 41.4 / **14.9**	87.8 / 82.7 / **91.3**
	TinyImageNet(r)	91.9 / 90.7 / **97.2**	24.9 / 58.3 / **13.4**	86.9 / 82.5 / **91.4**	91.9 / 98.9 / **99.7**	24.9 / 5.5 / **1.0**	86.9 / 94.8 / **97.8**
	TinyImageNet(c)	92.6 / 91.1 / **97.0**	22.0 / 55.7 / **13.4**	87.6 / 85.5 / **91.3**	92.6 / 97.2 / **99.3**	22.0 / 14.8 / **2.7**	87.6 / 91.2 / **96.3**
	SVHN	90.6 / 85.8 / **95.6**	24.6 / 67.5 / **17.9**	86.0 / 79.3 / **89.3**	90.6 / **98.2** / 97.9	24.6 / **7.2** / 8.0	86.0 / **94.0** / 93.5
	Average	92.4 / 83.2 / **97.0**	22.3 / 75.3 / **13.5**	87.4 / 77.0 / **91.4**	92.4 / 96.6 / **98.7**	22.3 / 14.5 / **5.4**	87.4 / 91.7 / **95.5**
CIFAR-100 (DenseNet)	LSUN(r)	69.1 / 84.7 / **94.0**	71.9 / 51.2 / **24.7**	64.0 / 76.8 / **87.1**	69.1 / 78.0 / **99.5**	71.9 / 50.5 / **1.5**	64.0 / 73.7 / **97.3**
	LSUN(c)	80.1 / **91.1** / 84.1	59.2 / **35.6** / 52.4	72.8 / **83.2** / 76.7	80.1 / 74.3 / **90.4**	59.2 / 77.4 / **34.2**	72.8 / 68.8 / **83.4**
	TinyImageNet(r)	71.2 / 85.5 / **94.3**	70.5 / 52.2 / **22.7**	65.5 / 77.5 / **87.5**	71.2 / 79.8 / **99.2**	70.5 / 53.1 / **3.9**	65.5 / 73.7 / **95.7**
	TinyImageNet(c)	77.0 / 89.7 / **92.6**	65.2 / 42.6 / **30.6**	69.9 / 81.7 / **85.1**	77.0 / 71.0 / **97.1**	65.2 / 71.3 / **15.3**	69.9 / 66.4 / **91.0**
	SVHN	83.4 / **88.2** / 84.6	44.8 / **37.8** / 51.2	76.2 / **81.0** / 76.7	83.4 / 86.9 / **96.9**	44.8 / 58.0 / **11.3**	76.2 / 79.6 / **91.9**
	Average	76.2 / 87.8 / **89.9**	62.3 / 43.9 / **36.3**	69.7 / 80.0 / **82.6**	76.2 / 78.0 / **96.6**	62.3 / 62.1 / **13.2**	69.7 / 72.4 / **92.5**
CIFAR-100 (ResNet)	LSUN(r)	81.9 / 90.5 / **96.1**	50.8 / 34.7 / **15.5**	74.6 / 82.9 / **90.7**	81.9 / 95.2 / **99.2**	50.8 / 22.6 / **4.0**	74.6 / 88.1 / **95.7**
	LSUN(c)	81.4 / **89.5** / 89.4	50.9 / **38.7** / 41.7	74.2 / **81.8** / 81.3	81.4 / 83.5 / **91.8**	50.9 / 64.3 / **32.3**	74.2 / 75.4 / **84.9**
	TinyImageNet(r)	81.8 / 90.5 / **95.9**	50.6 / 35.0 / **16.3**	74.7 / 83.0 / **90.1**	81.8 / 94.5 / **98.8**	50.6 / 27.4 / **6.0**	74.7 / 87.7 / **94.7**
	TinyImageNet(c)	84.2 / 90.5 / **94.3**	46.6 / 35.1 / **24.0**	76.7 / 82.8 / **87.6**	84.2 / 90.1 / **97.7**	46.6 / 43.3 / **11.5**	76.7 / 81.8 / **92.5**
	SVHN	80.3 / 91.7 / **92.5**	52.9 / 31.3 / **27.6**	73.9 / 84.1 / **85.7**	80.3 / 89.2 / **95.1**	52.9 / 37.9 / **16.4**	73.9 / 81.7 / **89.7**
	Average	81.9 / 90.53 / **93.6**	50.4 / 35.0 / **25.0**	74.8 / 82.9 / **87.1**	81.9 / 90.5 / **96.5**	50.4 / 39.1 / **14.0**	74.8 / 82.9 / **91.5**

Table 4: Comparison of OOD detection performance across ODIN (left) and Mahalanobis (right) with and without AdvProb. The values are percentages and the best values are highlighted in bold.

C Mahalanobis performance when trained and validated with the same OOD dataset

ID Dataset	OOD dataset	AUROC ↑	FPR@95 ↓	Detection acc. ↑
		baseline / Mahalanobis [**?**]/ Mahalanobis + AdvProb (ours)		
CIFAR-10 (Resnet)	lsun(r)	93.9 / 99.2 / **99.9**	18.8 / 4.2 / **0.3**	88.6 / 95.6 / **98.7**
	lsun(c)	92.9 / 95.0 / **98.4**	21.3 / 21.2 / **9.1**	87.8 / 88.4 / **93.5**
	tinyimagenet(r)	91.9 / 98.9 / **99.7**	24.9 / 5.7 / **0.7**	86.9 / 95.0 / **98.3**
	tinyimagenet(c)	92.6 / 97.7 / **99.6**	22.0 / 10.9 / **1.1**	87.6 / 92.2 / **97.1**
	svhn	90.6 / 98.9 / **99.8**	24.6 / 4.5 / **1.1**	86.0 / 95.7 / **98.3**
CIFAR-10 (DenseNet)	lsun(r)	95.5 / 93.8 / **100.0**	14.6 / 32.2 / **0.2**	90.3 / 88.1 / **99.2**
	lsun(c)	93.2 / 87.6 / **98.8**	20.5 / 52.5 / **5.5**	87.3 / 78.8 / **95.0**
	tinyimagenet(r)	94.1 / 94.0 / **99.8**	19.3 / 33.7 / **0.6**	88.3 / 87.7 / **97.8**
	tinyimagenet(c)	93.8 / 87.8 / **99.5**	20.0 / 53.6 / **1.7**	88.0 / 80.4 / **97.2**
	svhn	90.0 / 98.3 / **99.9**	31.1 / 8.2 / **0.6**	83.3 / 93.4 / **98.4**
CIFAR-100 (Resnet)	lsun(r)	81.9 / 96.1 / **99.5**	50.8 / 20.4 / **3.1**	74.6 / 89.8 / **96.3**
	lsun(c)	81.4 / 90.1 / **96.2**	50.9 / 41.5 / **21.0**	74.2 / 81.9 / **90.0**
	tinyimagenet(r)	81.8 / 95.4 / **99.2**	50.6 / 24.5 / **3.9**	74.7 / 88.3 / **95.9**
	tinyimagenet(c)	84.2 / 91.6 / **98.4**	46.6 / 33.4 / **7.4**	76.7 / 83.7 / **94.2**
	svhn	80.3 / 97.5 / **99.5**	52.9 / 9.0 / **2.5**	73.9 / 93.2 / **96.8**
CIFAR-100 (DenseNet)	lsun(r)	69.1 / 78.0 / **99.6**	71.9 / 50.5 / **1.3**	64.0 / 73.7 / **97.6**
	lsun(c)	80.1 / 74.3 / **95.5**	59.2 / 77.4 / **17.3**	72.8 / 68.8 / **89.4**
	tinyimagenet(r)	71.2 / 79.8 / **99.3**	70.5 / 53.1 / **3.2**	65.5 / 73.7 / **96.3**
	tinyimagenet(c)	77.0 / 71.0 / **98.2**	65.2 / 71.3 / **8.5**	69.9 / 66.4 / **93.6**
	svhn	83.4 / 86.9 / **99.4**	44.8 / 58.0 / **2.4**	76.2 / 79.6 / **96.5**

Table 5: Comparison of OOD detection performance across baselines when models are trained and evaluated on the same OOD dataset. Results include: (i) baseline (no preprocessing), (ii) Mahalanobis detector with standard input preprocessing, and (iii) Mahalanobis detector integrated with our proposed AdvProb method (using XGBoost).

D Detailed Analysis of Individual Perturbation Objectives

E Relative Confidence gain across perturbation maginutes for all OOD objects - full visualisation

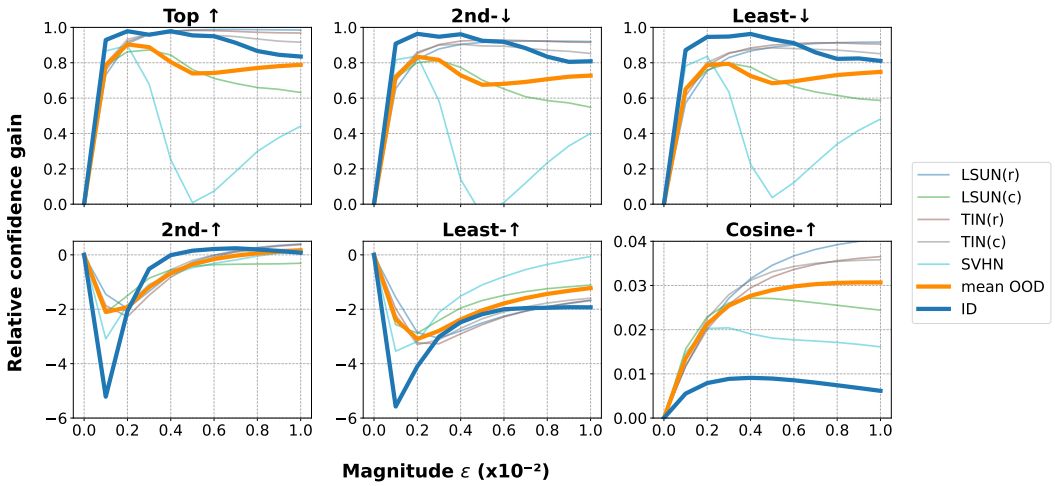

Figure 6: Relative confidence gain across perturbation magnitude ϵ for each of the six AdvProb objectives. This is the same plot as Figure 2 but adding each individual OOD dataset (thin colored lines). This results are form the DenseNet model train on CIFAR-10.

F Pseudo algorithm

Algorithm 1 Multi-Directional Adversarial Input Preprocessing (AdvProb)

Neural network model $f(\cdot)$, temperature scaling T, perturbation magnitude ϵ, input sample x, precomputed class mean embeddings μ_y, predicted softmax probabilities $S(x)$, feature extractor $h(\cdot)$ Compute initial predictions:

$$\hat{y} \leftarrow \arg\max_i S_i(x)$$

$$y_2 \leftarrow \text{Second highest predicted class from } S(x)$$

$$y_{\text{least}} \leftarrow \text{Lowest predicted class from } S(x)$$

Generate adversarial perturbations:

$$\tilde{x}_1 \leftarrow x + \epsilon \cdot \text{sign}\left(\nabla_x S_{\hat{y}}(x)\right) \quad \triangleright \text{ Increase primary class confidence}$$

$$\tilde{x}_2 \leftarrow x - \epsilon \cdot \text{sign}\left(\nabla_x S_{y_2}(x)\right) \quad \triangleright \text{ Decrease second-highest class confidence}$$

$$\tilde{x}_3 \leftarrow x - \epsilon \cdot \text{sign}\left(\nabla_x S_{y_{\text{least}}}(x)\right) \quad \triangleright \text{ Decrease lowest class confidence}$$

$$\tilde{x}_4 \leftarrow x + \epsilon \cdot \text{sign}\left(\nabla_x S_{y_2}(x)\right) \quad \triangleright \text{ Increase second-highest class confidence}$$

$$\tilde{x}_5 \leftarrow x + \epsilon \cdot \text{sign}\left(\nabla_x S_{y_{\text{least}}}(x)\right) \quad \triangleright \text{ Increase lowest class confidence}$$

$$\tilde{x}_6 \leftarrow x + \epsilon \cdot \text{sign}\left(\nabla_x \text{cos\_sim}(h(x), \mu_{\hat{y}})\right) \quad \triangleright \text{ Increase similarity to class mean}$$

Compute feature vector $\mathbf{z}(x)$ for XGBoost:

$$\mathbf{z}(x) = [S_{\hat{y}}(\tilde{x}_1), S_{\hat{y}}(\tilde{x}_2), S_{\hat{y}}(\tilde{x}_3), S_{\hat{y}}(\tilde{x}_4), S_{\hat{y}}(\tilde{x}_5), S_{\hat{y}}(\tilde{x}_6),$$
$$\mathbb{I}(\hat{y} = \arg\max_i S_i(\tilde{x}_4)), \mathbb{I}(\hat{y} = \arg\max_i S_i(\tilde{x}_5)),$$

Train an XGBoost classifier on the feature vector $\mathbf{z}(x)$ for ID/OOD discrimination.

G Perturbation Performance

Perturbation	AUROC ↑	FPR_{95} ↓	Accuracy ↑	AUROC ↑	FPR_{95} ↓	Accuracy ↑
$z(x) = \tilde{x}_0 + \cdots + \tilde{x}_7$	**98.45**	**7.28**	**93.88**	**99.30**	**3.20**	**96.03**
$\tilde{x}_0$	93.86	18.82	88.56	95.49	14.62	90.28
$\tilde{x}_1$	91.21	54.18	85.11	**98.65**	**7.04**	**94.04**
$\tilde{x}_2$	91.25	55.78	86.04	98.43	7.40	93.96
$\tilde{x}_3$	91.18	53.66	85.42	97.42	9.10	93.03
$\tilde{x}_4$	87.24	32.42	82.29	91.44	25.76	85.38
$\tilde{x}_5$	92.92	24.14	85.94	94.86	21.16	87.94
$\tilde{x}_6$	**96.43**	21.54	**90.85**	98.13	8.80	93.21
$\tilde{x}_7$	96.35	**13.96**	90.53	96.81	13.28	91.14
$\tilde{x}_0 + \tilde{x}_1$	95.87	18.64	88.65	**99.08**	**4.16**	**95.48**
$\tilde{x}_0 + \tilde{x}_2$	95.92	18.22	89.15	98.72	5.28	95.00
$\tilde{x}_0 + \tilde{x}_3$	95.96	18.40	89.42	98.00	7.12	94.13
$\tilde{x}_0 + \tilde{x}_4$	97.08	13.52	91.63	99.02	5.00	95.17
$\tilde{x}_0 + \tilde{x}_5$	96.46	15.14	90.80	98.15	7.64	93.75
$\tilde{x}_0 + \tilde{x}_6$	**97.34**	13.22	91.41	98.33	7.72	93.71
$\tilde{x}_0 + \tilde{x}_7$	97.32	**11.32**	**92.16**	97.05	12.46	91.74
$\tilde{x}_0 + \tilde{x}_1 + \tilde{x}_2$	96.15	17.74	89.47	99.08	4.24	95.48
$\tilde{x}_0 + \tilde{x}_1 + \tilde{x}_3$	96.57	15.72	90.51	99.08	4.26	95.45
$\tilde{x}_0 + \tilde{x}_1 + \tilde{x}_4$	97.43	12.44	92.07	**99.24**	3.56	95.76
$\tilde{x}_0 + \tilde{x}_1 + \tilde{x}_5$	97.02	13.62	91.10	99.13	4.36	95.35
$\tilde{x}_0 + \tilde{x}_1 + \tilde{x}_6$	97.53	12.68	91.60	99.13	3.94	95.64
$\tilde{x}_0 + \tilde{x}_1 + \tilde{x}_7$	**97.87**	**10.00**	**92.70**	99.16	**3.44**	**95.83**
$\tilde{x}_0 + \tilde{x}_6 + \tilde{x}_1$	97.53	12.68	91.60	**99.13**	**3.94**	**95.64**
$\tilde{x}_0 + \tilde{x}_6 + \tilde{x}_2$	97.55	12.56	91.65	98.90	4.62	95.26
$\tilde{x}_0 + \tilde{x}_6 + \tilde{x}_3$	97.56	12.36	91.75	98.77	5.42	94.92
$\tilde{x}_0 + \tilde{x}_6 + \tilde{x}_4$	**98.00**	**9.80**	**92.95**	99.12	3.96	95.58
$\tilde{x}_0 + \tilde{x}_6 + \tilde{x}_5$	97.85	11.06	92.39	98.78	5.72	94.71
$\tilde{x}_0 + \tilde{x}_6 + \tilde{x}_7$	**98.00**	10.42	92.50	98.50	6.86	94.11
$z_{-0}(x) = z(x) \notin \tilde{x}_0$	98.43	7.58	93.84	99.28	3.26	96.09
$z_{-1}(x) = z(x) \notin \tilde{x}_1$	98.45	7.32	93.84	99.19	3.94	95.72
$z_{-2}(x) = z(x) \notin \tilde{x}_2$	98.41	7.44	93.89	99.28	3.26	96.09
$z_{-3}(x) = z(x) \notin \tilde{x}_3$	98.44	7.44	93.86	99.30	3.22	96.05
$z_{-4}(x) = z(x) \notin \tilde{x}_4$	98.30	8.32	93.50	99.19	4.00	95.65
$z_{-5}(x) = z(x) \notin \tilde{x}_5$	98.45	7.56	93.87	99.29	3.02	96.01
$z_{-6}(x) = z(x) \notin \tilde{x}_6$	98.42	7.60	93.81	99.30	3.22	96.05
$z_{-7}(x) = z(x) \notin \tilde{x}_7$	98.23	8.92	93.23	99.27	3.50	95.78

Table 6: Performance of each $\tilde{x}_i$ trained on the CIFAR-10 and validated on the LSUN dataset. **Left:** Resnet model. **Right:** DenseNet model.

H Dropout epsilon Robustness

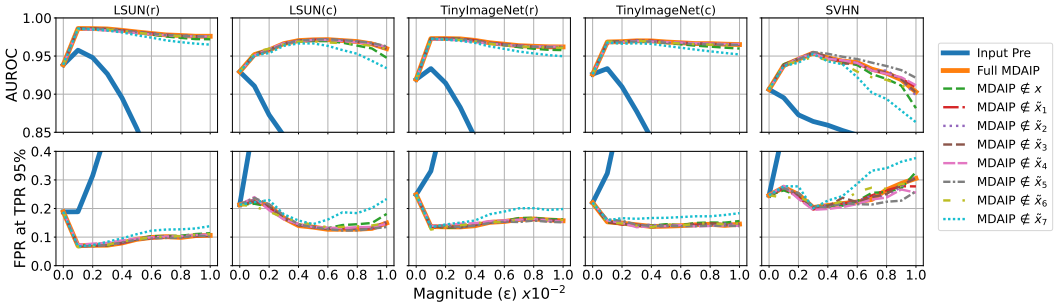

Figure 7: Resnet model epsilon robustness experiment for each dropout perturbation $\tilde{x}_i$

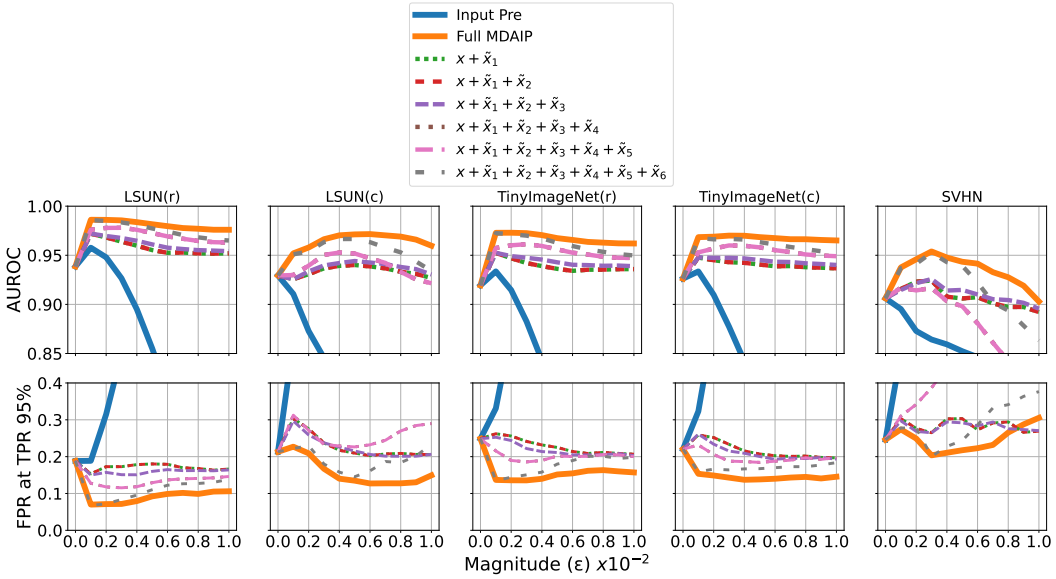

Figure 8: Resnet model epsilon robustness experiment, incremental perturbation robustness gain.

I Dropout epsilon Robustness (most Crit)

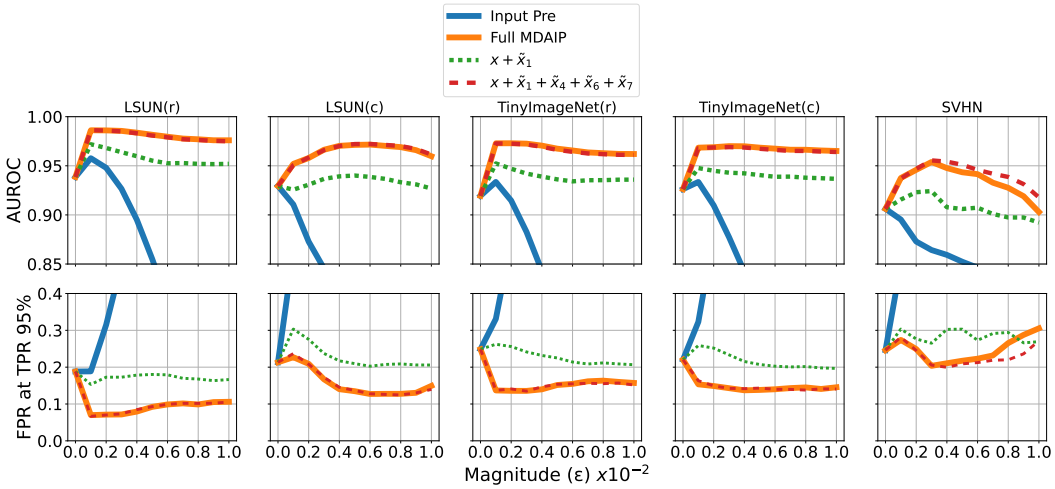

Figure 9: Discovering that most of the information are stored on 5 perturbation (instead of 8) for the **Resnet** model

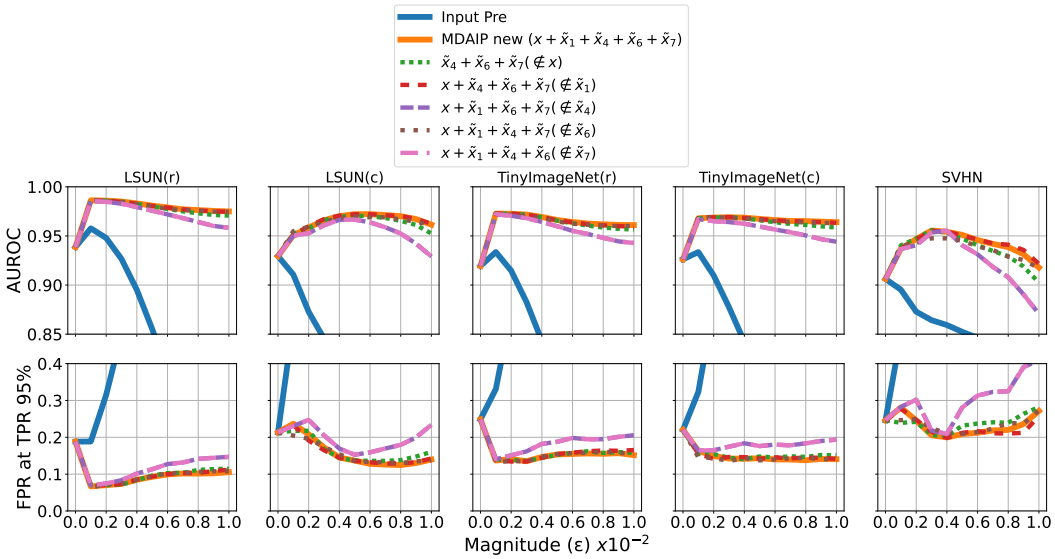

Figure 10: Dropout of the most critical perturbations (for the **Resnet** model).

J UMAP means projection shifts

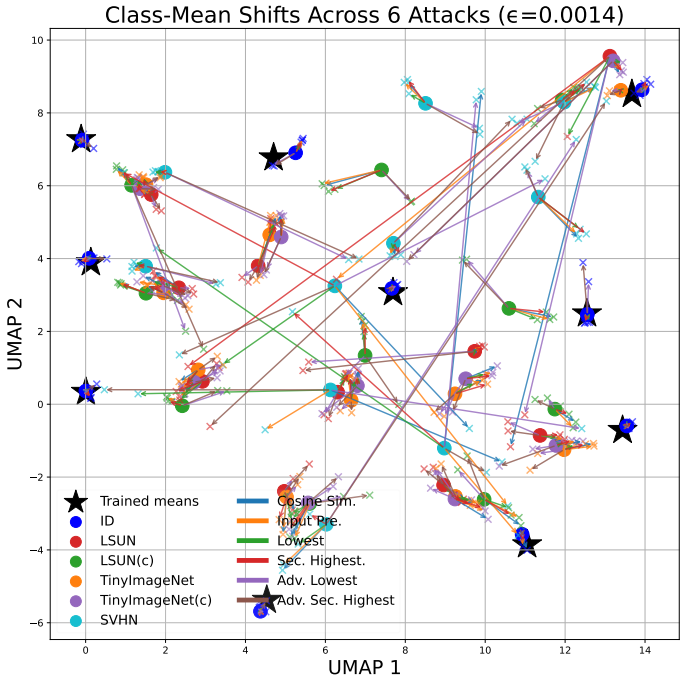

Figure 11: Caption

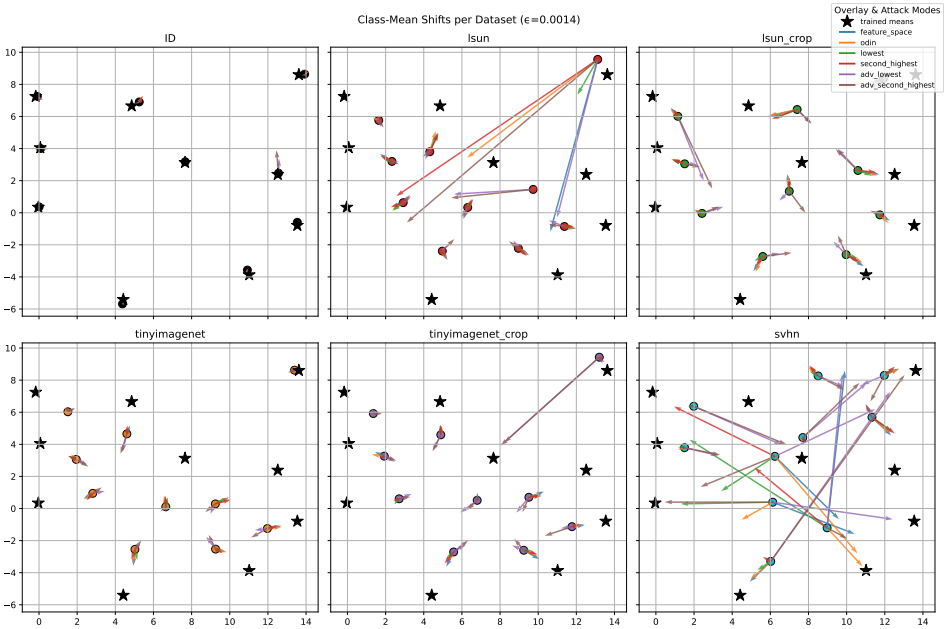

Figure 12: Caption

K Computational Complexity and Empirical Timing

Theoretical Complexity. The proposed AdvProb strategy applies P one-step adversarial perturbations per input, each requiring one forward and one backward pass. Let C_{fwd} be the cost of a single forward pass and $C_{\mathrm{bwd}} \approx 2\,C_{\mathrm{fwd}}$ the cost of its backward. Then the per-sample cost is

$$T_{\mathrm{AdvProb}} = P\left(C_{\mathrm{fwd}} + C_{\mathrm{bwd}}\right) + C_{\mathrm{fwd}} \;\approx\; P\,(3\,C_{\mathrm{fwd}}) + C_{\mathrm{fwd}} = (1+3P)\,C_{\mathrm{fwd}}.$$

Over a dataset of N samples (and model FLOPs proportional to M), this scales as

$$T_{\mathrm{AdvProb}}(N) = O\big(N\,P\,M\big).$$

Empirical Timing (Canonical Dataset). We measure elapse time to compute a confidence score for each perturbation. To compute this time take used the ResNet18 model and used the CIFAR-10 dataset average 100 batch sizes. The experiments were run on one NVIDIA GeForce RTX 4060 (Driver 566.50, CUDA 12.7) with PyTorch 2.5.1+cu124 and cuDNN 90100:

Method	Batch size	Time (ms per batch)	Time (ms per image)
Baseline (no AdvProb)	64	8.42 ± 0.01	0.13 ± 0.00
Average single perturbation	64	42.48 ± 0.08	0.66 ± 0.00
AdvProb ($P = 8$ directions)	64	263.49 ± 0.51	4.12 ± 0.01

Table 7: Batch-level and per-image timing for AdvProb on the CIFAR-10 test set (100 batches, mean $\pm$ std), in ms.