Feature Space Singularity: Out-of-Distribution Samples Concentrate in Trained Neural Networks

Haiwen Huang, Zhihan Li, Lulu Wang, Sishuo Chen, Xinyu Zhou, Bin Dong

SafeAI, Feburary 2021

Contents

Background:
OOD detection for AI Safety

OOD detection for AI safety

Traffic signs in the training set

Unseen traffic signs

- OOD samples can cause unintended and harmful behaviors of current machine learning systems
- We want to detect OOD samples and make later actions accordingly.

Madhav Iyengar, Michael Opitz and Horst Bischof. Detecting Out-of-Distribution Traffic Signs. Proceedings of the ARW & OAGM Workshop 2019

Current methods for OOD detection are not good enough

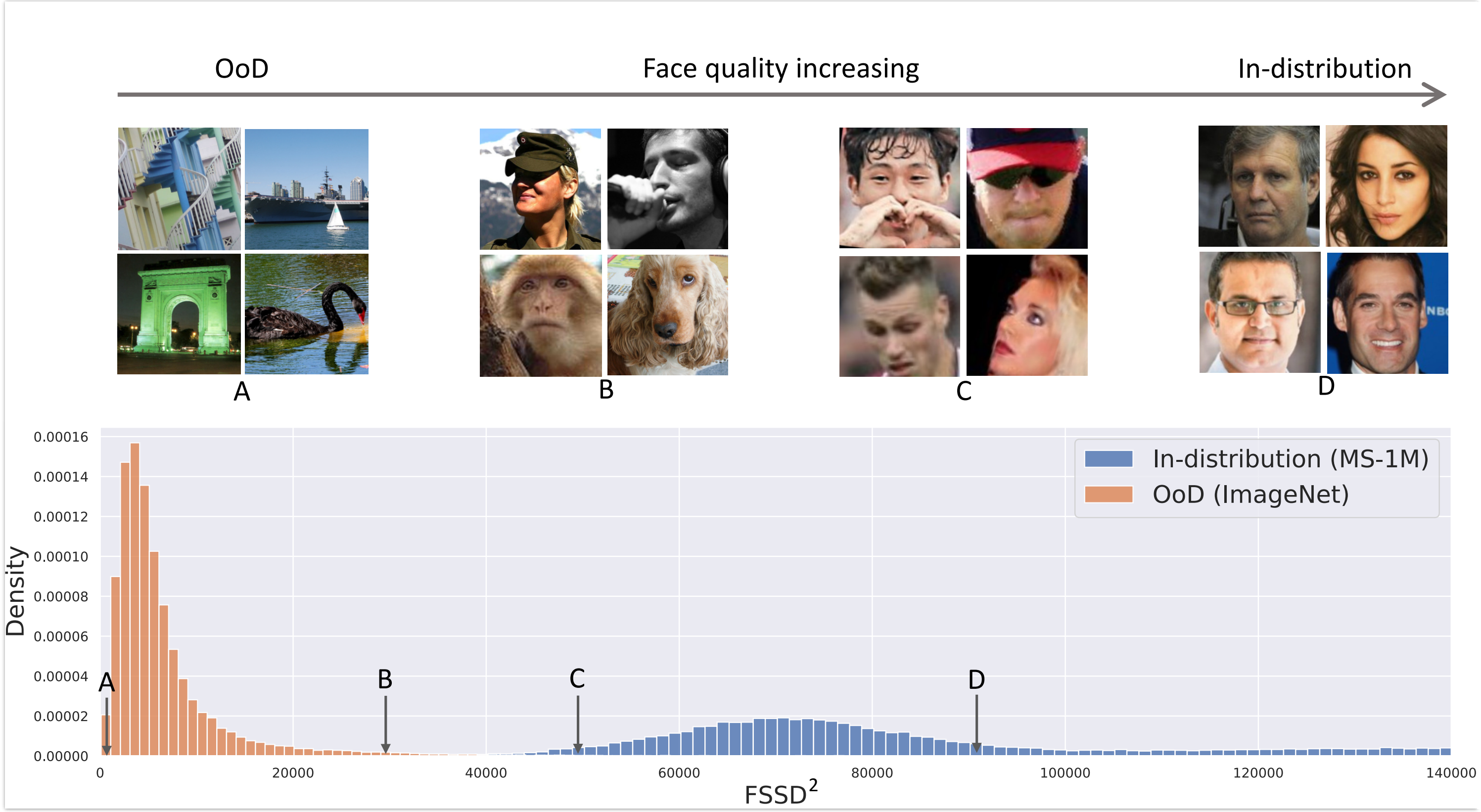
- Two major concerns: performance and computational cost
  - Single-model methods (e.g., ODIN, MC Dropout) don't perform well
  - Ensemble-based methods (Deep Ensemble) require training multiple randomly initialized NNs

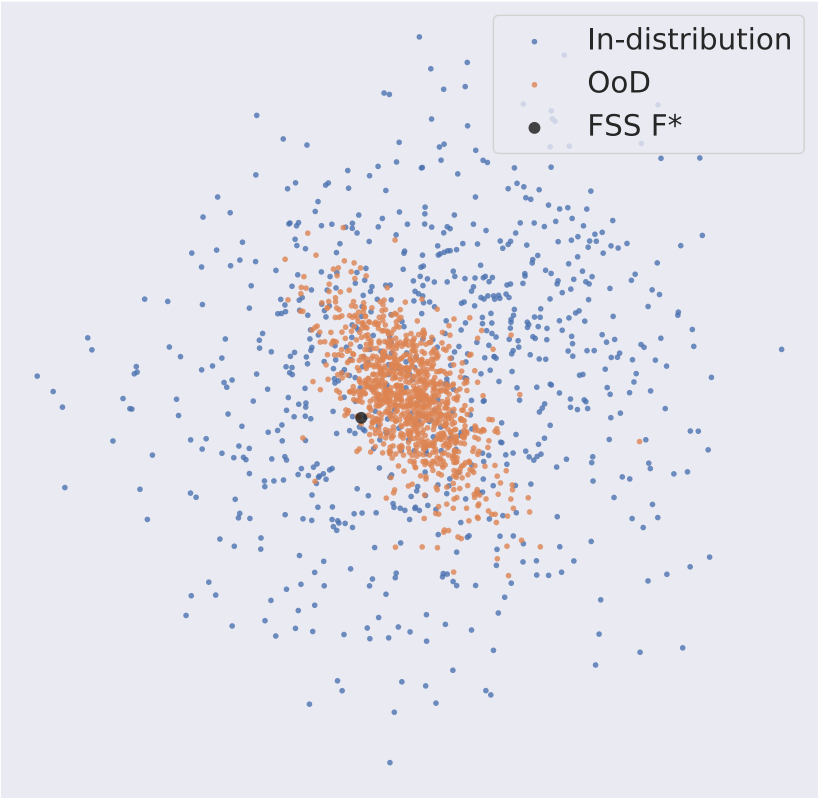
- Can we use a single model and still achieve high performance?

Y. Ovidia et al. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. NeurIPS 2019

Observation and Analysis:
OOD samples concentrate in the feature space

OOD samples concentrate in the feature space

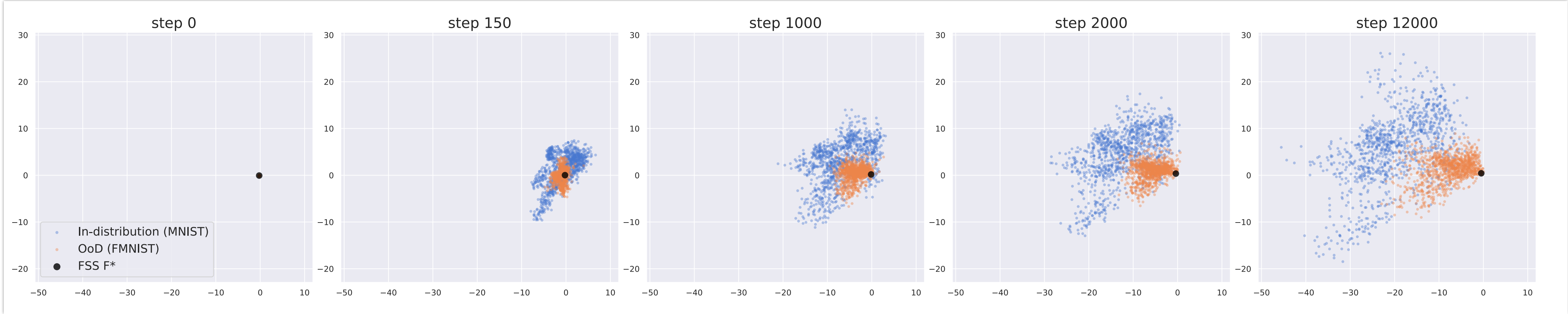


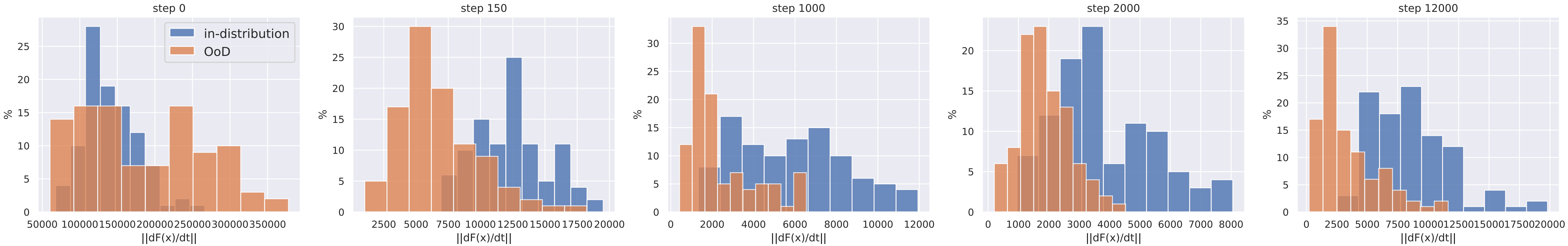


The phenomenon seems universal

- We have tested:
  - Architecture: MLP, LeNet, ResNet, DenseNet, bi-LSTM
  - Activation function: ReLU, Tanh
  - Loss: Cross Entropy loss, Triplet loss, L2-loss
  - Datasets: MNIST, FMNIST, CIFAR10, SVHN, ImageNet, CelebA, MS1M ……
  - Supervised learning, unsupervised learning (Instance Discrimination)

Explanation: Different moving speeds of features





The "moving speed" of features: $\frac{\mathrm{d}F_{\theta_t}(x)}{\mathrm{d}t} = \frac{\partial F_{\theta_t}(x)}{\partial \theta_t}\frac{\mathrm{d}\theta_t}{\mathrm{d}t}$

Note x can be any input; x_m are the training data $= -\sum_{m=1}^{M} \frac{\partial F_{\theta_t}(x)}{\partial \theta_t}\frac{\partial F_{\theta_t}(x_m)}{\partial \theta_t}^{\mathsf{T}} \partial_m \mathcal{L}_\phi .$

Algorithms and Experiments

Algorithm: Ensemble different layers

Algorithm 1: Computation of FSSD-Ensem

Input: Test samples $x = \{x_n^{\text{test}}\}_{n=1}^{N}$, noise samples $\{x_s^{\text{noise}}\}_{s=1}^{S}$, ensemble weights α_k, perturbation magnitude ϵ, feature extractors $\{F_{(k)}\}_{k=1}^{K}$

for *each feature extractor* $\{F_{(k)}\}_{k=1}^{K}$ **do**

1. Estimate FSS $F_{(k)}^{*} = \sum_{s=1}^{S} F_{(k)}\left(x_s^{\text{noise}}\right)/S$, where $x_s^{\text{noise}} \sim \mathcal{U}[0,1]$, $s = 1, \cdots, S$
2. Add perturbation to test sample: $\tilde{x} = x + \epsilon\, \text{sign}(\nabla_{\boldsymbol{x}} \left\| F_{(k)}(x) - F_{(k)}^{*} \right\|)$
3. Calculate $\text{FSSD}^{(k)}(x) = \left\| F_{(k)}(\tilde{x}) - F_{(k)}^{*} \right\|$

end

Return $\text{FSSD-Ensem}(x) = \sum_{k=1}^{K} \alpha_k\, \text{FSSD}^{(k)}(x)$

More on perturbation: Shiyu Liang, Yixuan Li, R. Srikant. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. ICLR 2018

Evaluation metrics

- AUROC: Area Under the Receiver Operating Characteristic curve.
- AUPRC: Area Under the Precision-Recall Curve.
- FPR80: False Positive Rate when the true positive rate is 80%.

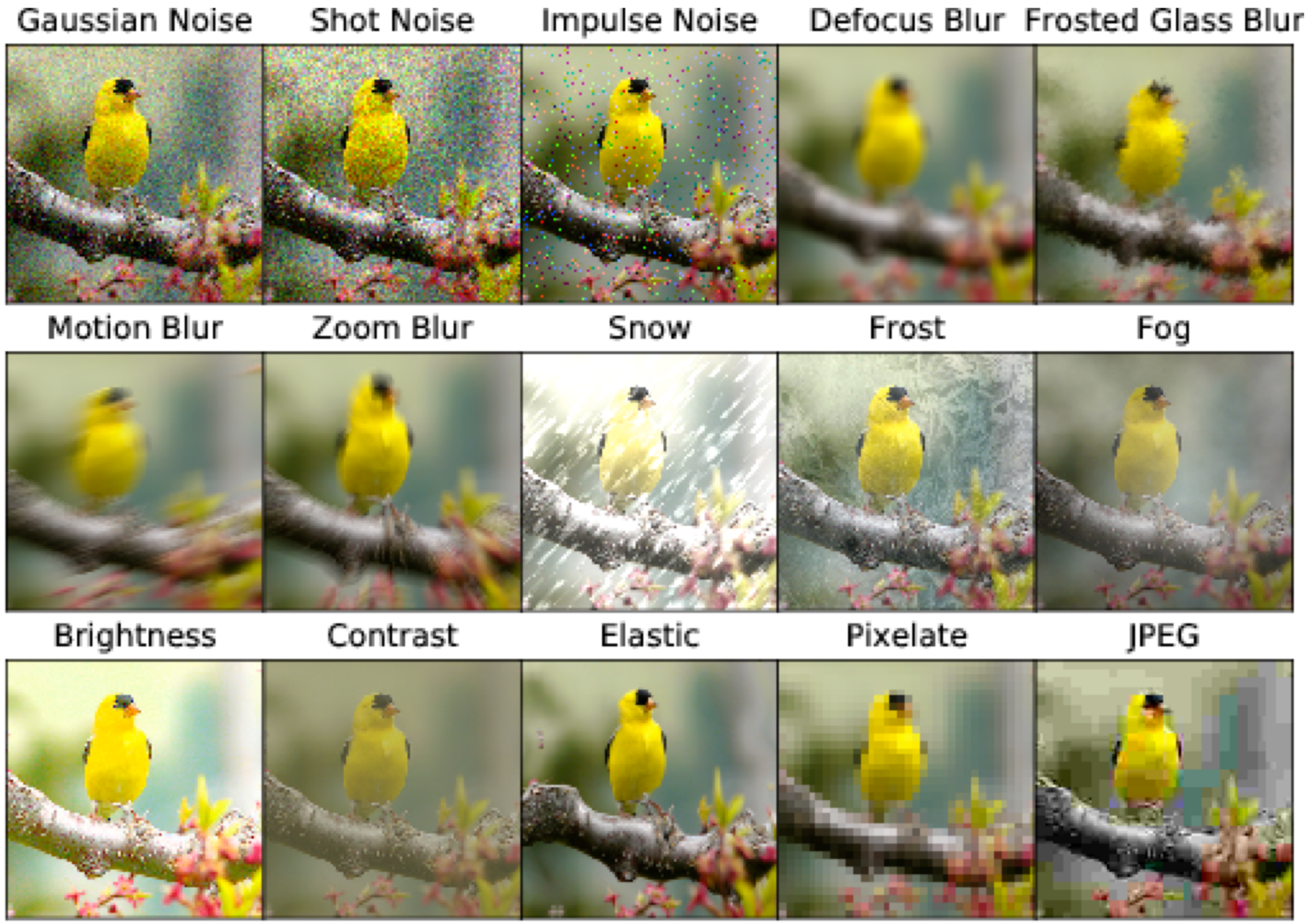
Experiments: Detecting OOD datasets

Table 2: Main results. All values are in %.

	Datasets (Architecture)	Metrics	*Base*	*ODIN*	*Maha*	*DE*	*MCD*	*OE*	*FSSD*
Small-scale benchmarks	FMNIST vs. MNIST (LeNet)	AUROC	77.3	96.9	**99.6**	83.9	81.7	**99.6**	**99.6**
		AUPRC	79.2	93.0	**99.7**	83.3	85.3	99.6	**99.7**
		FPR80	43.5	2.5	**0.0**	27.5	36.8	**0.0**	**0.0**
	CIFAR10 vs. SVHN (ResNet34)	AUROC	89.9	96.7	99.1	93.7	96.7	90.4	**99.5**
		AUPRC	85.4	92.5	98.1	90.6	93.9	89.8	**99.5**
		FPR80	10.1	4.7	**0.3**	3.7	2.4	12.5	0.4
	ImageNet dogs vs. non-dogs (ResNet34)	AUROC	88.5	90.8	83.3	89.0	67.2	92.5	**93.1**
		AUPRC	86.1	88.6	83.0	89.0	66.9	**92.6**	92.5
		FPR80	19.5	15.2	30.1	18.8	59.2	**7.9**	10.2
Large-scale benchmarks	CelebA non-blurry vs. blurry (ResNeXt50)	AUROC	71.7	73.3	73.9	74.5	69.8	71.5	**78.3**
		AUPRC	89.9	91.4	90.9	91.4	88.7	90.7	**92.8**
		FPR80	52.0	50.3	46.0	47.1	53.2	54.2	**39.2**
	MS-1M vs. IJB-C (ResNeXt50)	AUROC	60.0	61.3	82.5	63.0	65.5	52.6	**86.7**
		AUPRC	53.3	55.9	80.6	56.1	59.4	46.6	**86.1**
		FPR80	61.8	59.4	29.6	56.7	58.8	64.2	**22.1**
Sequence benchmark	Bacteria Genome (LSTM)	AUROC	69.6	70.6	70.4	70.0	69.3	NA	**74.8**
		AUPRC	69.9	71.9	69.3	56.0	70.2	NA	**75.8**
		FPR80	57.4	55.9	53.7	**30.0**	58.3	NA	47.4

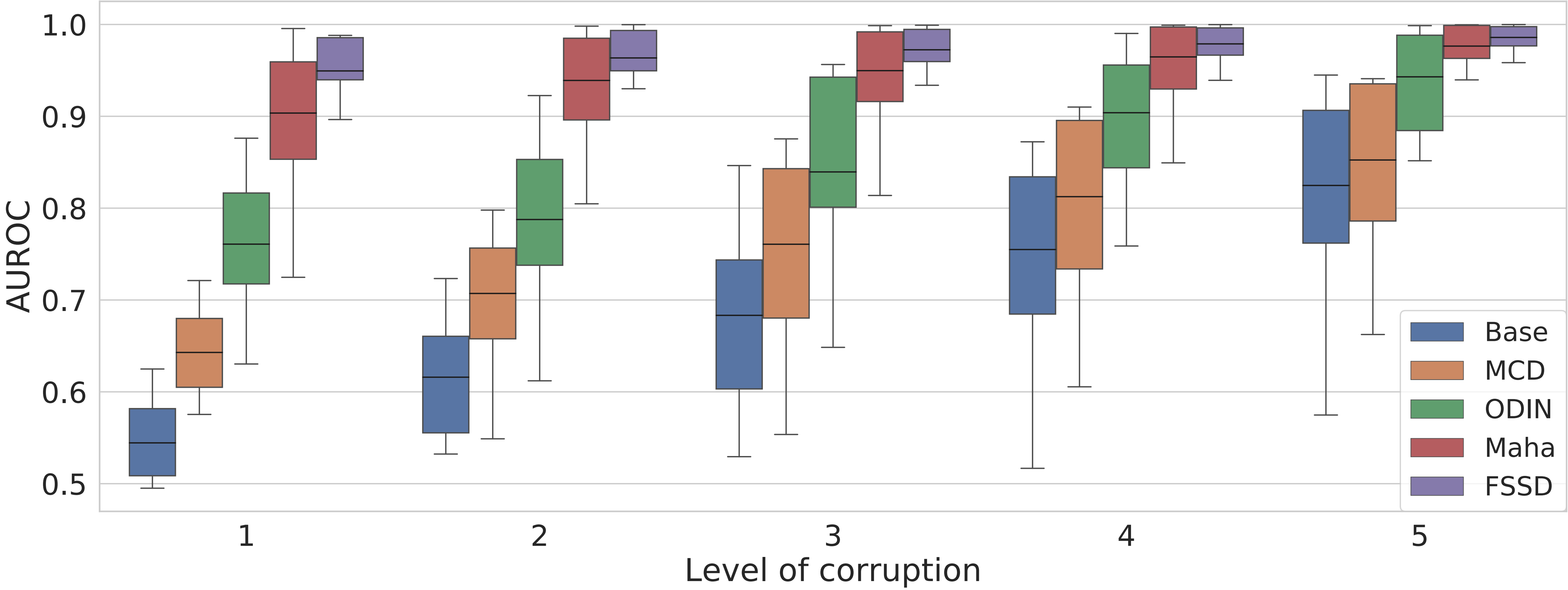
Different scales, different data types

Experiments: Detecting image corruption



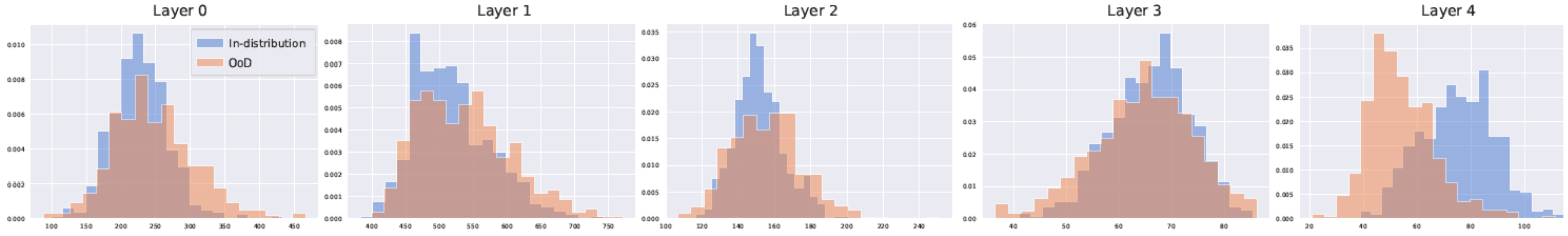
Corruption datasets come from ImageNet-C datasets, which contain 16 types of corruptions and each corruption has 5 different levels

Experiments: Detecting image corruption

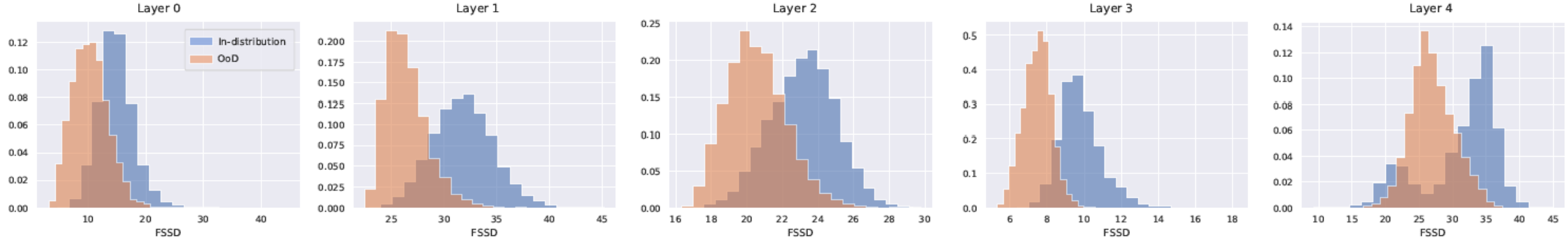


FSSD has higher mean and smaller variance across different corruptions.

Analysis of ensemble



(a) ImageNet (dogs) vs. ImageNet (non-dogs)



(b) CIFAR10 vs. SVHN

Early layers correspond to low-level statistics; deeper layers correspond to high-level semantics.

Outlook:
What's next?

Failure cases?

- We have just seen some successful cases of exploiting the feature space singularity to detect OOD samples. But there are also failure cases, e.g., CIFAR-10 vs CIFAR-100.
- Explaining and mitigating the failure cases can lead to clean separation of OOD and in-distribution data/features.
- Future directions:
  - Inductive bias of neural network (neural tangent kernel) on OOD inputs, e.g., input norm;
  - Disentangling spurious features shared by OOD data;
  - Constraining the learning process to for better OOD detection, e.g., adding bi-Lipschitz constraints, using OOD-related loss functions.

THANK YOU!

Code repository:

https://github.com/megvii-research/FSSD_OoD_Detection

Implemented Algorithms

In this repository, we implement the following algorithms.

Algorithm	Paper	Implementation
FSSD	Feature Space Singularity for Out-of-Distribution Detection	test_fss.py
Baseline	A BASELINE FOR DETECTING MISCLASSIFIED AND OUT-OF-DISTRIBUTION EXAMPLES IN NEURAL NETWORKS	test_baseline.py
ODIN	Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks	test_odin.py
Maha	A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks	test_maha.py
DE	Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles	test_de.py
OE	Deep Anomaly Detection with Outlier Exposure	test_baseline.py