Layerwise Activation Similarity to Training Data for Assessing Non-Conforming Events

Eryk Banatt
Johns Hopkins University
Laurel, MD 20723
ebanatt1@jh.edu

Abstract

Uncertainty estimation methods for classification generally fall into one of two categories: they rely on changes in the output distribution ("Label space"), or they rely on the representation constructed by the model, usually by pulling out the penultimate layer prior to softmax ("Feature space"). The vast majority of uncertainty estimation methods lie in one of these two categories, but both of these methods fall victim to a well-known but largely-undocumented failure case: stochastic overlapping feature values between in-distribution and out-of-distribution data. In this work we present Layerwise Activation Similarity to Training Data for Assessing Non-Conforming Events (LASTDANCE), a new form of anomaly detection which is robust to these points by leveraging an input's full trajectory through the layers. We show that this method sets a new state-of-the-art in anomaly detection under spurious feature overlap, despite not requiring any supervision from out-of-distribution data.

1 Introduction

Machine Learning has revolutionized computer vision, leading to impressive results spanning a wide variety of tasks including classification, object detection, and semantic segmentation [14]. However, unlike hand-designed feature engineering approaches, machine learning models rely upon training data in order to infer upon unseen data, under the assumption that the training data was independent and identically distributed [16]. If the test data does not resemble the training data, performance suffers as a result of a phenomenon known as **Domain shift** [25]. Since this data often comes without labeled ground truth in deployed machine learning models, this effect is often invisible, and is therefore highly important to detect when encountered.

Identifying a testing point which is out-of-distribution (OOD) in the wild after deployment is an open problem in machine learning, and one which is of critical importance to operators leveraging machine learning models. Understanding when a model ought to be truly confident in a point (due to its resemblance to the training data) and when the model is simply guessing (because machine learning models are typically overconfident out-of-distribution [7]) can mean the difference between life and death in high-stakes scenarios.

Common methods for uncertainty estimation which are more sophisticated than model confidence include Monte Carlo Dropout [4], ALICE [22], Trust Score [9] and others. Despite some of these models leveraging the constructed representation [22], and others leveraging the variance of the outputs [4] [12], both of these share a common failure case: spuriously overlapping OOD features.

This failure case has previously been informally documented [3], demonstrating that out-of-distribution data sometimes finding it's way into the "center" of the representation. With enough out-of-distribution examples, this would occasionally cause it to overlap with the genuinely con-

Preprint. Under review.

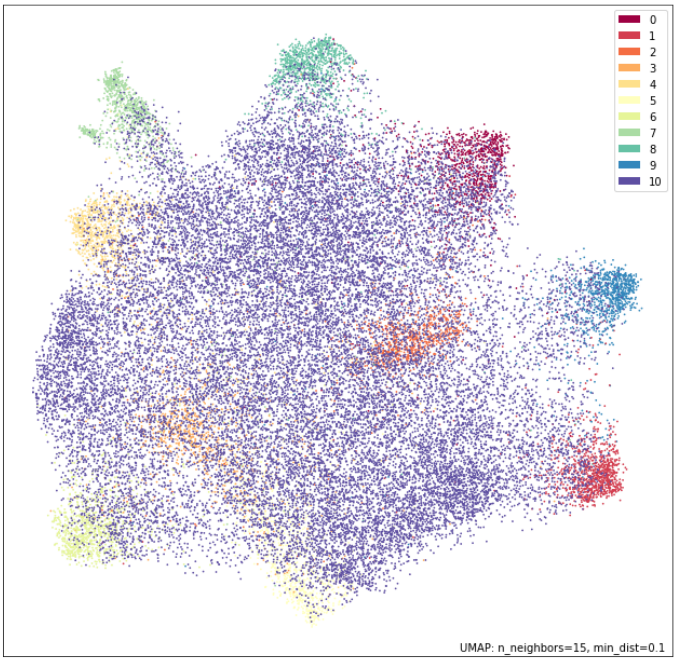

Figure 1: CIFAR10 (in-distribution) vs SVHN (out-of-distribution, Purple) penultimate layer features projected down into two dimensions using UMAP [13]. Figure was generated by training a ResNet18 model, then evaluating with model.train() active in order to keep the batch normalization layers on.

structed representation for in-domain data. Note that this causes uncertainty estimation failure in both classes of methods: if an OOD point is indistinguishable from an in-domain point, it will also have an overconfident label output, which means it will be overconfident across all methods.

Also note that this problem is highly architecture dependent. For example, methods such as monte carlo (MC) dropout [4] are often implemented by enabling all layers which are frequently turned off at test time, one of which is the Batch Normalization layer [8]. Batch Normalization is a useful layer, it improves performance [8] and aides in training models to generalize [23], but it also increases the likelihood that out-of-distribution features are randomly overlapping with the representation constructed of in-distribution data. It would be both useful and convenient to be able to leverage these techniques while also being able to stay aware of out-of-distribution points, but current methods force algorithm developers to choose one or the other: if you use MC dropout, you are significantly increasing the chances out-of-distribution points will appear in-distribution.

Our work makes three primary contributions. First, it formally presents stochastic feature overlap as a failure case of uncertainty estimation methods, both in feature-based and label-based uncertainty estimation methods. Second, it presents Layerwise Activation Similarty to Training Data for Assessing Non-Conforming Events (LASTDANCE) as a method for detecting out-of-distribution points, even if those points overlap in-distribution features at a layer of interest. Finally, we perform experiments to verify the performance of LASTDANCE over current anomaly detection methods, and set a new state-of-the-art benchmark for anomaly detection in this setting.

2 Related Work

2.1 Uncertainty Estimation

The field of uncertainty estimation is broad and deeply researched, and remains an open topic of interest to the research community to the present day. Broadly, uncertainty estimation methods fall into one of two categories: label-space-based estimators and feature-space-based estimators.

Label-space-based estimators are estimators which construct an output distribution, and uses this distribution to compute variance (and therefore a measure of uncertainty). The most prominent among these is Monte Carlo Dropout [4]. In MC dropout, training-time dropout layers are enabled, and the forward pass is calculated several times, creating a distribution over output space which can be measured to generate a notion of uncertainty. As mentioned, this often underperforms with out-of-distribution data, since batch normalization layers will "pull" those features back in-distribution erroneously.

Feature-space-based estimators are estimators which leverage the constructed internal representation built by the model in order to determine if a point ought to be certain or uncertain. There are a number of methods which fall into this category, including Trust Score [9], ALICE [22] and Deep k-Nearest Neighbors [17]. Similar in spirit to ours is [20], which takes a more directly bayesian approach by modeling layerwise the expected distribution of parameters. While more direct, our approach leveraging fitted gaussians upon reduced features is less computationally expensive and therefore much more scalable.

Additional approaches to uncertainty estimation, such as test-time augmentation methods, training strategies, weight sharing methods, etc. fall outside the scope of this work due to requiring modification to the model architetures themselves. For more information on these methods, we refer the reader to Gawlikowski et al 2022 [5].

2.2 Anomaly Detection

Adjacent to uncertainty estimation is the field of Anomaly detection, which seeks to identify whether or not a point is similar to the training distribution. ODIN [12] notes that classifiers calibrated via temperature scaling [6] will produce separated in- and out-of-distribution outputs under small perturbations.

Mahalanobis distance is the de facto standard for anomaly detection in deep neural networks [11], and other methods [22] leverage mahalanobis distance as a proxy for distributional uncertainty in order to compute a more holistic estimate of uncertainty.

2.3 Layerwise Feature Selection

Early exit ensembles [24] [21] don't preserve the best performance of the penultimate layer, so performance goes down. OODL [1] is generally robust to stochastic feature overlap, but requires supervision by training on out-of-distribution data, which is generally unavailable in practice and also comes attached with no guarantees that you select the optimal OOD layer for a different distribution which is different from both your training data and your "training OOD" data.

3 Methods

We first lay out our desiderata for an uncertainty estimate which is robust to stochastic overlap of out-of-distribution inputs on any particular layer. First, we want to be able to predict with the best layer with respect to in-distribution inputs, i.e. the penultimate layer. Second, we want to be able to determine if a point is out of distribution, even if that point randomly happens to fall in-distribution on a particular layer. Finally, we want to use this regardless of what the out-of-distribution domain actually is: that is, given in-domain set A and out-of-domain sets B and C, we want to be reasonably confident that both domain B and domain C reliably land out-of-distribution, compared to a method which can only work for a known a priori OOD set.

Methods like early exit ensembles seem to function well for the second two criteria, in exchange for damaging the performance on in-distribution datapoints by diluting the prediction from the end of the network. Methods like OODL seem to work for the first two criteria, but require a priori OOD data, and also do not dynamically select an optimal OOD layer for each individual point. We propose Layerwise Activation Similarity to Training Data for Assessing Non-Conforming Events (LASTDANCE) as a method which satisfies all three criteria.

In order to not require OOD data, we leverage the intuition that points belonging to the same class will "ride through" the layers of a neural network together, whereas points that randomly overlap in-distribution data will do so in an erratic manner. That is to say, it is unlikely that the trajectory

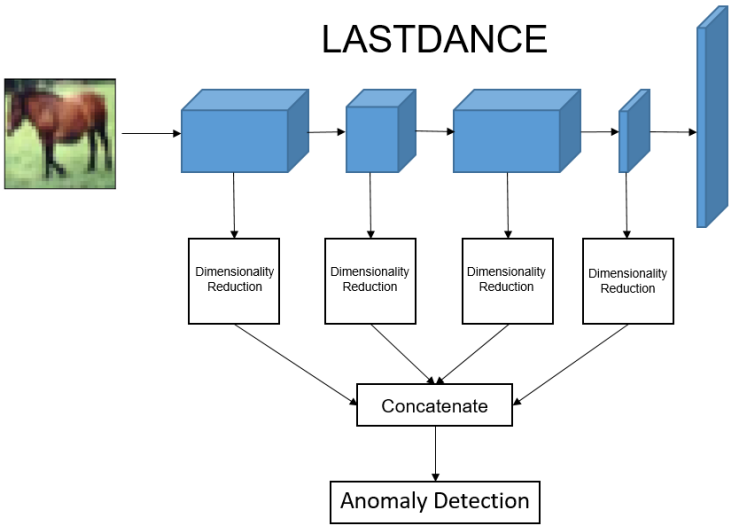

Figure 2: The fundamental structure of LASTDANCE: leveraging a feature vector consisting of concatenating several reduced feature vectors from every layer, instead of the penultimate representation.

through the layers resembles that of any in-distribution point, even if that point lands in-distribution on every layer (due to bouncing around between classes).

In order to make this trajectory computationally tractable, we fit a dimensionality reduction module upon each layer of a neural network, such that we can get a reduced-size representation of the features at every layer. Given n datapoints, L layers, and k dimensions from each layer, we can concatenate the reduced feature vectors from each layer together to get a feature vector of size (n, L, k), or flattened to (n, Lk), which represents the point's trajectory through each of the layers. From there, we can behave as normal; we use mahalanobis distance upon the features in order to determine whether or not that point is out-of-distribution.

4 Experiments

We perform a series of experiments to show that anomalies are much easier to detect when leveraging the trajectory of the features through the layers. We run experiments with and without the spurious overlap condition, in order to examine performance with and without this case present.

We use the following datasets:

- **CIFAR10** [10] CIFAR10 is a benchmark computer vision dataset containing 10 classes, spanning natural and human-made objects. CIFAR10 is provided under the MIT License.
- **SVHN** [15] The Street View House Numbers (SVHN) dataset is a dataset consisting of photographs of various numbers painted upon house number signs. SVHN was released under the CC0 1.0 public domain license.

For all experiments, we train a model on the training set of the "in-distribution" set, and then infer upon both the test set of the "in-distribution" set and the test set of the "out-of-distribution" set. To artificially ensure stochastic feature overlap, we use a ResNet18 model pretrained on Imagenet [2] with the batch norm layers enabled at test time (i.e. using $model.train()$ for the model's PyTorch [18] implementation). All experiments take place on a single NVIDIA 1070TI, and code to reproduce these experiments can be found in the supplemental material. A more complete description of the implementation details and hyperparameters are also available in the supplemental material [1].

[1] Code for these experiments can be found on Github: https://github.com/ambisinister/LASTDANCE

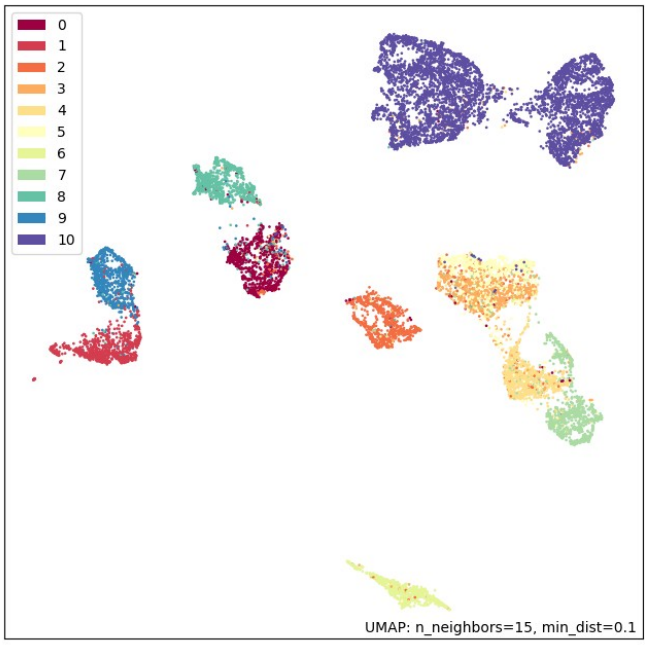

Figure 3: CIFAR10 (in-distribution) vs SVHN (out-of-distribution, Purple) trajectories through each layer of ResNet18, reduced to 2 dimensions using UMAP. Despite using the same features as 1, the out-of-distribution points cleanly cluster far apart from the in-distribution points.

Method	ID Set	OOD Set	Mean Dist ID	Mean Dist OOD
Penultimate [11]	CIFAR10	SVHN	0.11	0.46
Trajectory (ours)	CIFAR10	SVHN	**0.09**	**0.63**
Penultimate[11]	SVHN	CIFAR10	0.12	**0.45**
Trajectory (ours)	SVHN	CIFAR10	**0.09**	0.40

Figure 4: Mean mahalanobis Distance for in-domain (ID) data vs out-of-domain (OOD) data, scaled between 0 and 1

The first experiment is a simple measure of different mahalanobis distance shifts for various pairs of in-domain and out-of-domain data. We scale the distances between 0 and 1 using a scikit-learn MinMaxScaler [19], and show that under many different shifts with stochastic overlap, the trajectories much more significantly separate compared to using just the features from the penultimate layer. Results for this experiment can be found in table 4.

The second experiment we show an ablation study, upon features generated by a model which does *not* have spurious overlapping features between in- and out-of-distribution data. We show that in this scenario, LASTDANCE achieves similar-or-better performance to current methods, and does not *only* work when such an overlap exists. Results for this experiment can be found in table 5

Method	ID Set	OOD Set	Mean Dist ID	Mean Dist OOD
Penultimate [11]	CIFAR10	SVHN	0.14	0.42
Trajectory (ours)	CIFAR10	SVHN	**0.08**	**0.68**
Penultimate[11]	SVHN	CIFAR10	0.09	**0.43**
Trajectory (ours)	SVHN	CIFAR10	**0.05**	0.31

Figure 5: Mean mahalanobis Distance for in-domain (ID) data vs out-of-domain (OOD) data, scaled between 0 and 1, with no spurious feature overlap.

5 Discussion

In this work, we demonstrated a new method for anomaly detection which is robust to spurious overlap of out-of-distribution features on any particular layer. Compared to other feature-based anomaly detection methods, it works on dataset shifts which are not known a priori, and also does not force the network to use a weaker representation for classification of in-distribution points. Dimensionality reduction allows this method to remain computationally tractable despite leveraging multiple distinct representations of a given input point, and uncertainty quantification of a trained neural network is lightweight and fast after the gaussians and dimensionality reduction models have been fit on training data features.

Aside from the obvious practical benefit of an anomaly detection method which is robust to spurious overlap, there are a number of more subtle implications from this work. For one, with LASTDANCE it becomes more tractable to perform uncertainty estimation upon models which specifically leverage batch normalization layers, for example Monte Carlo Dropout [4]. In a sense, LASTDANCE is not that fundamentally different from any other mahalanobis-based anomaly detection, but the simple decision to construct a feature vector from *all* of the layers rather than focus on a single layer in particular increases the span of available models from which reasonable uncertainty estimation would be possible.

Future work will revolve around methods of calculating trajectories without destroying as much information in each layer. For example, one method which warrants further study is scaling every feature vector to unit length, calculating class conditional unit vectors, and then calculating cosine distance between the scaled unit feature vector of each layer for a given point. Since the volume of a ball in high dimensions is concentrated near its shell, scaling each vector to unit length could potentially destroy less information than projecting down to two dimensions, and cosine distance between two vectors would neatly yield a scalar value for the degree of anomaly between a vector and each of the other classes. Likewise, numerous small improvements are apparent: using a gaussian mixture instead of a single gaussian, using distance to every class-conditional gaussian rather than the single minimum, calibrating the distance beyond simply scaling it between 0 and 1, etc. All of these could yield substantial improvements to the anomaly detection results.

Acknowledgments and Disclosure of Funding

This paper was completed with no external sources of funding.

References

[1] ABDELZAD, Vahdat ; CZARNECKI, Krzysztof ; SALAY, Rick ; DENOUNDEN, Taylor ; VERNEKAR, Sachin ; PHAN, Buu: Detecting out-of-distribution inputs in deep neural networks using an early-layer output. In: *arXiv preprint arXiv:1910.10307* (2019)

[2] DENG, Jia ; DONG, Wei ; SOCHER, Richard ; LI, Li-Jia ; LI, Kai ; FEI-FEI, Li: Imagenet: A large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition* Ieee (Veranst.), 2009, S. 248–255

[3] DIETTERICH, Thomas G.: https://twitter.com/tdietterich/status/1424203516034228226. (2021)

[4] GAL, Yarin ; GHAHRAMANI, Zoubin: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: *international conference on machine learning* PMLR (Veranst.), 2016, S. 1050–1059

[5] GAWLIKOWSKI, Jakob ; ROVILE NJIEUTCHEU TASSI, Cedrique u. a.: A Survey of Uncertainty in Deep Neural Networks. (2022)

[6] GUO, Chuan ; PLEISS, Geoff ; SUN, Yu ; WEINBERGER, Kilian Q.: On calibration of modern neural networks. In: *International conference on machine learning* PMLR (Veranst.), 2017, S. 1321–1330

[7] HEIN, Matthias ; ANDRIUSHCHENKO, Maksym ; BITTERWOLF, Julian: Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, S. 41–50

[8] IOFFE, Sergey ; SZEGEDY, Christian: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *International conference on machine learning* PMLR (Veranst.), 2015, S. 448–456

[9] JIANG, Heinrich ; KIM, Been ; GUAN, Melody ; GUPTA, Maya: To trust or not to trust a classifier. In: *Advances in neural information processing systems* 31 (2018)

[10] KRIZHEVSKY, Alex ; HINTON, Geoffrey u. a.: Learning multiple layers of features from tiny images. (2009)

[11] LEE, Kimin ; LEE, Kibok ; LEE, Honglak ; SHIN, Jinwoo: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In: *Advances in neural information processing systems* 31 (2018)

[12] LIANG, Shiyu ; LI, Yixuan ; SRIKANT, Rayadurgam: Enhancing the reliability of out-of-distribution image detection in neural networks. In: *arXiv preprint arXiv:1706.02690* (2017)

[13] MCINNES, Leland ; HEALY, John ; MELVILLE, James: Umap: Uniform manifold approximation and projection for dimension reduction. In: *arXiv preprint arXiv:1802.03426* (2018)

[14] MICHALSKI, Ryszard S. ; CARBONELL, Jaime G. ; MITCHELL, Tom M.: *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013

[15] NETZER, Yuval ; WANG, Tao ; COATES, Adam ; BISSACCO, Alessandro ; WU, Bo ; NG, Andrew Y.: Reading digits in natural images with unsupervised feature learning. (2011)

[16] NOURETDINOV, Ilia ; VOVK, Volodya ; VYUGIN, Michael ; GAMMERMAN, Alex: Pattern recognition and density estimation under the general iid assumption. In: *International Conference on Computational Learning Theory* Springer (Veranst.), 2001, S. 337–353

[17] PAPERNOT, Nicolas ; MCDANIEL, Patrick: Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. In: *arXiv preprint arXiv:1803.04765* (2018)

[18] PASZKE, Adam ; GROSS, Sam ; MASSA, Francisco ; LERER, Adam ; BRADBURY, James ; CHANAN, Gregory ; KILLEEN, Trevor ; LIN, Zeming ; GIMELSHEIN, Natalia ; ANTIGA, Luca u. a.: Pytorch: An imperative style, high-performance deep learning library. In: *Advances in neural information processing systems* 32 (2019)

[19] PEDREGOSA, Fabian ; VAROQUAUX, Gaël ; GRAMFORT, Alexandre ; MICHEL, Vincent ; THIRION, Bertrand ; GRISEL, Olivier ; BLONDEL, Mathieu ; PRETTENHOFER, Peter ; WEISS, Ron ; DUBOURG, Vincent u. a.: Scikit-learn: Machine learning in Python. In: *the Journal of machine Learning research* 12 (2011), S. 2825–2830

[20] POSCH, Konstantin ; STEINBRENER, Jan ; PILZ, Jurgen: Variational Inference to Measure Model Uncerainty in Deep Neural Networks. (2019)

[21] QENDRO, Lorena ; CAMPBELL, Alexander ; LIO, Pietro ; MASCOLO, Cecilia: Early exit ensembles for uncertainty quantification. In: *Machine Learning for Health* PMLR (Veranst.), 2021, S. 181–195

[22] RAJENDRAN, Vickram ; LEVINE, William: Accurate layerwise interpretable competence estimation. In: *Advances in Neural Information Processing Systems* 32 (2019)

[23] SEGU, Mattia ; TONIONI, Alessio ; TOMBARI, Federico: Batch normalization embeddings for deep domain generalization. In: *arXiv preprint arXiv:2011.12672* (2020)

[24] SUN, Tianxiang ; ZHOU, Yunhua ; LIU, Xiangyang ; ZHANG, Xinyu ; JIANG, Hao ; CAO, Zhao ; HUANG, Xuanjing ; QIU, Xipeng: Early exiting with ensemble internal classifiers. In: *arXiv preprint arXiv:2105.13792* (2021)

[25] ZHOU, Kaiyang ; LIU, Ziwei ; QIAO, Yu ; XIANG, Tao ; LOY, Chen C.: Domain generalization in vision: A survey. In: *arXiv preprint arXiv:2103.02503* (2021)

A Further Results

B Further Details

For CIFAR10 -> SVHN, the models were trained on ResNet18 pretrained with Imagenet. We trained for 40 epochs, a batch size of 32, with a transform that resized the images to ((63, 63)). We used cross entropy loss, and as an optimizer we used SGD with a learning rate of 0.001 and momentum of 0.9. The models, trained this way, achieve approximately 91% accuracy.

For SVHN -> CIFAR10, we use the same setup as the above, but we train for 15 epochs instead of 40. Under this scenario, we achieve approximately 94% accuracy. For memory constraints, we take only the first 10 thousand points from the test set, instead of the full test set, since the feature vectors from the full test sets from SVHN and CIFAR10 do not fit in memory.

All UMAP modules project down to $k = 2$, purely for memory constraint reasons. We expect that using a higher k would leave to superior results, and that using the last $l < L$ layers instead of all L layers could allow for a trade-off between number of layers and k size for each layer in the trajectory.