Mapping DNN Embedding Manifolds for Network Generalization Prediction

Molly O'Brien[1]
Johns Hopkins University
3400 N Charles St, Baltimore, MD
mollynnobrien@gmail.com

Brett Wolfinger[1]
bwolfin1@jhu.edu

Julia Bukowski
Villanova University
800 Lancaster Ave, Villanova, PA
julia.bukowski@villanova.edu

Mathias Unberath[1]
unberath@jhu.edu

Aria Pezeshk*
Center for Devices and
Radiological Health, U.S. FDA
Silver Spring, MD
aria.pezeshk@gmail.com

Greg Hager[1] *
hager@jhu.edu

Abstract

Deep Neural Networks(DNN) often fail in surprising ways, and predicting how well a trained DNN will generalize in a new, external operating domain is essential for deploying DNNs in safety critical applications, e.g., perception for self-driving vehicles or medical image analysis. Recently, the task of Network Generalization Prediction (NGP) has been proposed to predict how a DNN will generalize in an external operating domain. Previous NGP approaches have leveraged multiple labeled test sets or labeled metadata. In this study, we propose an embedding map, the first NGP approach that predicts DNN performance based on how unlabeled images from an external operating domain map in the DNN embedding space. We evaluate our proposed Embedding Map and other recently proposed NGP approaches for pedestrian, melanoma, and animal classification tasks. We find that our embedding map has the best average NGP performance, and that our embedding map is effective at modeling complex, non-linear embedding space structures.

1. Introduction

It is well known that Deep Neural Networks (DNNs) are black box systems that achieve state of the art performance in essentially every perception task proposed in the last decade. DNNs are composed of tens to hundreds of layers with millions of learnable weights, and they excel at tasks such as image classification, object detection, and semantic segmentation. It is also well documented that DNN performance often degrades when DNNs are deployed in operating domains that are different from their training and testing domains [13]. For instance, in perception for self-driving vehicles, differences in camera characteristics, lighting and weather conditions, and foreground and background objects can impact DNN performance. In medical image analysis, the input data distribution can be impacted by choice of scanner vendors, pre- and post-processing algorithms, dose levels, image compression, and patient and disease distributions. Because of this performance degradation, even as DNN performance continues to improve and approach human performance in many benchmark datasets, it is challenging to deploy DNNs in commercial products that perform safety critical tasks in unconstrained environments. In order for DNNs to reach their full potential for commercial use, we need techniques that can predict how a DNN will perform in an external operating domain before it causes automated, harmful failures [28].

While DNNs are different from traditional software in that the learned weights cannot be read and interpreted, there is still structure in the mappings that DNNs learn. Feed-forward DNNs perform a high-dimensional, non-linear projection of input data into an embedding space, and the final prediction is a linear projection of the embedding. We are interested in identifying structure in the DNN embedding space as it relates to the DNN performance.

Our primary contribution is a new NGP approach that can accurately predict DNN performance in a novel operating domain based on how unlabeled images from the novel operating domain map in the DNN embedding space. We evaluate our NGP method on pedestrian, melanoma, and animal classification tasks and demonstrate accurate NGP across different DNN architectures, external datasets, and classification tasks. Additionally, we visualize the DNN embedding space for a pedestrian classification experiment and propose that DNN architecture impacts what NGP approach is most accurate in a given experiment.

2. Network Generalization Prediction

DNNs are typically trained and tested on different partitions of a labeled, internal dataset before the DNN is used in multiple external operating domains; see Figure 1. In unconstrained environments, e.g., perception for self-driving vehicles, medical image analysis, etc., the external operating domain may vary significantly from the internal test set. Typically, labeled examples from the external operating domain are not available, though some unlabeled data from the external operating domain may be available. Network Generalization Prediction (NGP) is the task of predicting how a DNN will generalize in an external operating domain, without requiring labeled test data from the given operating domain.

NGP has been the focus of growing attention over the last year and has been denoted Automatic Model Evaluation (AutoEval) [6], Accuracy Estimation [8], Network Generalization Prediction (NGP) [27], and Detection Performance Modeling [29] in different prior works. Unlike other ML tasks, the goal of NGP is to accurately predict what the DNN performance will be in a novel, external operating domain; it is not to improve the overall DNN performance.

2.1. Previous NGP Approaches

Prior NGP works have relied on unlabeled operating domain images [6], [8] or expensive context annotations to predict DNN generalization [27], [29]. Deng et al. propose to simulate different labeled test sets by applying sequential image transformations to the internal test images [6]. They modify the image background and foregrounds separately, which is possible in binary images, e.g., MNIST [16], or object detection images with object masks, e.g., MS COCO [20], but it is unclear how these image transformations could be performed on a broad set of natural images. They generate 1600 simulated test datasets, and then they use the Fréchet distance to measure the distribution shift between the unmodified internal test set and the modified test sets. They then fit a regressor, linear or DNN, to learn the generalization gap for a given magnitude distribution shift. This approach requires significant computational time and memory to test the DNN on additional test sets.

Guillory et al. propose a similar approach where they use ImageNet as the internal dataset and ImageNet-V2, ImageNet-VidRobust, ImageNet-Rendition, and ImageNet-Sketch as labeled test sets from different distributions [8]. Again Guillory et al. model the decrease in model accuracy as a function of the magnitude of the distribution shift. They investigate multiple distance metrics to measure distribution shift including the Fréchet distance and several metrics that they propose. The most effective distance metric is the Difference of Confidences (DoC). DoC measures the change in the average predicted softmax score between the internal test sets and the external operating set. The authors train a regressor to predict the change in accuracy based on the magnitude of the distribution shift observed across multiple labeled test sets.

O'Brien et al. proposed an interpretable context subspace (CS) that identifies context features, i.e., metadata, or image statistics like brightness, that are informative for NGP [27]. The previous work can accurately predict DNN performance for changes in context feature distribution, but does not capture changes that occur when moving from one dataset to another, e.g., changes in camera parameters or changes in the image structure. Ponn et al. trained a random forest on image attributes, e.g., pedestrian occlusion, bounding box size, presence of rain, etc., to predict whether a pedestrian would be detected [29].

In contrast to previous NGP methods, we propose the first NGP approach that can predict DNN performance directly from the way in which unlabeled images map in the DNN embedding space.

3. Related Works

NGP is related to the fields of Domain Generalization and Out-of-Distribution detection, that aim to improve the performance of DNNs in unconstrained environments and detect when an input is outside the known distribution, respectively.

3.1. Domain Generalization

DNNs trained using domain generalization algorithms aim to perform well in operating domains that differ from the training or testing domains. While many domain generalization algorithms have been proposed in the last decade [2], none has been shown to consistently out-perform standard Empirical Risk Minimization (ERM) [9].

Jiang et al. demonstrate that the margin distribution of internal layers in a DNN is correlated with the generalization gap between the internal training set and the internal test set [12]. Recent work has proposed that Generalized Reweighting optimization techniques do not out-perform ERM because current DNNs are over parameterized and the optimization techniques carry the same biases as ERM [41]. See [9] for a review of Domain Generalization techniques. It has been proposed that underspecification can cause DNN performance to degrade when deployed in operating domains different from the training domains [5]. To facilitate domain generalization research, the WILDS benchmark was released to provide datasets with “in-the-wild” distribution shifts between the training and test data [13].

An emerging topic in domain generalization is hidden stratification: the idea that average performance can obscure subpopulations of data where the DNN performs poorly.

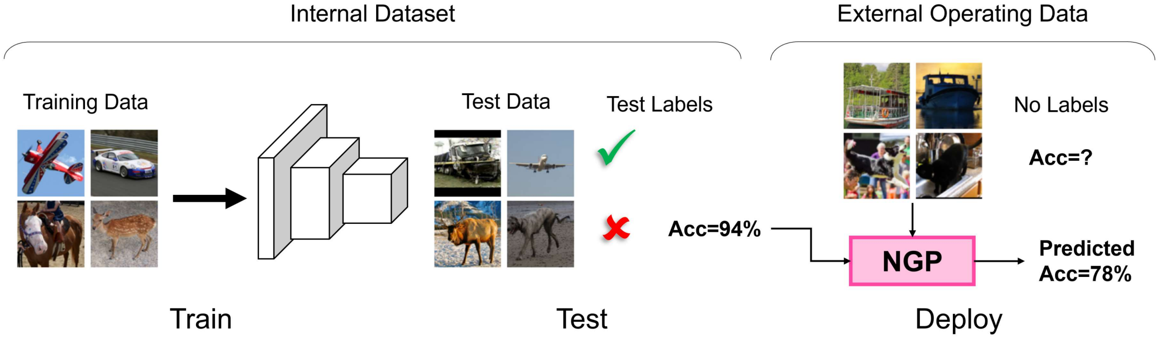


Figure 1. Overview of Network Generalization Prediction (NGP). A DNN is trained using internal training data. Next, the DNN is tested with the internal test data. The internal test results are used to predict the DNN performance for the external operating data.

This can lead to harm if the task is safety critical, e.g., medical image analysis [25]. Sohoni et al. propose the framework GEORGE that identifies subgroups of data by clustering examples in the DNN embedding space and training classifiers that demonstrate robust performance across subgroups [35].

3.2. Out-of-Distribution Detection

DNNs are trained in limited environments, and are not typically capable of correct predictions for input samples that are unlike the data with which they were trained. Automatically recognizing Out-of-Distribution (OOD) samples is a broad area of research that is relevant to safely deploying DNNs in unconstrained environments. Many prior works use the DNN embedding, i.e., the output from the penultimate DNN layer, or the softmax scores to detect OOD samples [10],[19], [24], [22], [33]. The baseline in [10] uses the softmax scores to predict whether an image is misclassified in addition to OOD detection. Previous work has investigated input sample Euclidean or Mahalanobis distance from training data in the embedding space to identify OOD and adversarial examples [17], [7] , [23]. Recent work proposed the Multi-level Out-of-distribution Detection (MOOD) framework for computationally efficient OOD [21].

4. Methods

We consider a trained, feed-forward DNN, f, where $f(x)$ denotes the DNN prediction. The layers of f, excluding the final layer, are a feature extractor, denoted ϕ, that projects the input image x into a D dimensional DNN embedding space (embedding space), $\phi(x) \in \mathbb{R}^D$. The DNN f is tested with images from an internal test set, i.e., a test set drawn from the same distribution as the training data. The images in the internal test set are denoted $X = \{x_i\}_{i=1}^N$ and are labeled $\mathbf{y} = \{y_i\}_{i=1}^N$. Images from an external operating set $\hat{X} = \{\hat{x}_i\}_{i=1}^M$ are analogous data from a new distribution. However, for the external operating set we assume that labels $\hat{\mathbf{y}}$ are unknown.

We are interested in finding structure in the embedding space that provides information about the DNN performance; specifically we aim to link the embedding space to the DNN outcome. Depending on the task, the outcome of interest could be determined by the loss, e.g., success or failure, or by the loss and the label, e.g., True Positive (TP), False Positive (FP), False Negative (FN), True Negative (TN) for binary classification. We use the notation $o(f(x), y)$ to denote the outcome.

4.1. Decision Tree in Embedding Space

The internal test set embeddings, $\phi(X)$, lie on some manifold in the high-dimensional embedding space; see Figure 2 Test (1). Decision trees are able to identify a sparse set of the most informative features given high-dimensional feature data [32]. We fit a decision tree on the D-dimensional test set embeddings, $\phi(X)$, so that the decision tree can predict the observed test outcomes, $o(f(X), \mathbf{y})$. We set a maximum depth of the decision tree to prevent the decision tree from overfitting.

The decision tree recursively selects the dimension of the embedding feature that maximizes the information gain about the DNN outcome. After the decision tree is fit, each node in the tree corresponds to a value and a dimension in the embedding space, e.g., value 0.78 at dimension 45. The node in the DT corresponds to a hyper-plane in the embedding space, i.e., images that map above a hyper-plane drawn at 0.78 for dimension 45 follow the left branch, images that map below the hyper-plane follow the right branch. See Figure 2 Test (2) for an illustration of the hyper-planes shown as dashed lines.

Each leaf node in the DT corresponds to a contiguous region in the embedding space, defined using the hyper-planes in the DT along the path to the leaf node. The leaf nodes are represented by the color boxes in Figure 2 Test (3). The images that map to a leaf node in the tree correspond to images that map to the same region in the embedding space. Each leaf node in the DT is identified using a sparse subset of the embedding dimensions that give the most information (in a greedy sense) about the DNN outcome. We refer to the fitted decision tree as our embedding map, because it maps regions in the embedding space to observed DNN outcomes.

4.2. Approximating Internal Test Set Manifold

The embedding map found in Section 4.1 contains L leaf nodes that define contiguous regions in the embedding space, where leaf l is identified using a sparse subset $\mathbf{d}^l << D$ of the embedding space dimensions. Note that the number of dimensions in $\mathbf{d}^l$ is less than or equal to the maximum depth of the decision tree. Using the embedding map, we can partition the internal test samples $X = \{x_i\}_{i=1}^N$ by the leaf to which each sample maps, i.e.,

$$X = \cup_{l=1}^{L} X^l, \qquad X^i \cap X^j = \emptyset \quad \forall i \neq j \tag{1}$$

where X^l is the set of N^l test samples that map to leaf l.

$$X^l = \{x_i^l\}_{i=1}^{N^l} \tag{2}$$

The internal test set X has associated labels $\mathbf{y}$. Let $\mathbf{y} = \cup_{l=1}^{L} \mathbf{y}^l$ be the test set labels partitioned by the leaf to which each sample maps and $\mathbf{y}^l = \{y_i^l\}_{i=1}^{N^l}$ be the labels for the test samples that map to leaf l. Let $\mathbb{I}(\mathbf{a}, \mathbf{b})$ be an indicator function that is equal to 1 if $\mathbf{a} = \mathbf{b}$ and 0 otherwise. Assuming each test sample is equally likely, the probability of outcome a in leaf l can be computed as:

$$p(a|l) = \frac{1}{N^l} \sum_{i=1}^{N^l} \mathbb{I}(o(f(x_i^l), y_i^l), a) \tag{3}$$

Note, the boxes in Figure 2 Test (3) are colored to match the most likely test outcome in the leaf region to illustrate linking a region of embedding space to the DNN outcomes observed in testing.

4.3. Network Generalization Prediction

We leverage the embedding map on unlabeled, external operating data; see Figure 2 Operating (1). The external operating samples can be mapped into the DNN embedding space as $\phi(\hat{X})$; see Figure 2 Operating (2). The external operating samples can then be partitioned to the L leaf nodes:

$$\hat{X} = \cup_{l=1}^{L} \hat{X}^l, \qquad \hat{X}^i \cap \hat{X}^j = \emptyset \quad \forall i \neq j \tag{4}$$

where $\hat{X}^l$ is the set of M^l external operating samples that map to leaf l.

$$\hat{X}^l = \{\hat{x}_i^l\}_{i=1}^{M^l} \tag{5}$$

The probability that a sample in the external operating set maps to leaf l can be approximated by the fraction of the operating samples that map to leaf l:

$$p(l) = \frac{M^l}{M} \tag{6}$$

The probability of encountering outcome a in the operating domain is:

$$p(a) = \sum_{l \in L} p(a|l)p(l) \tag{7}$$

$p(a)$ can be computed for each outcome a observed in testing (assuming we have discrete outcomes and outcome possibilities) to perform NGP.

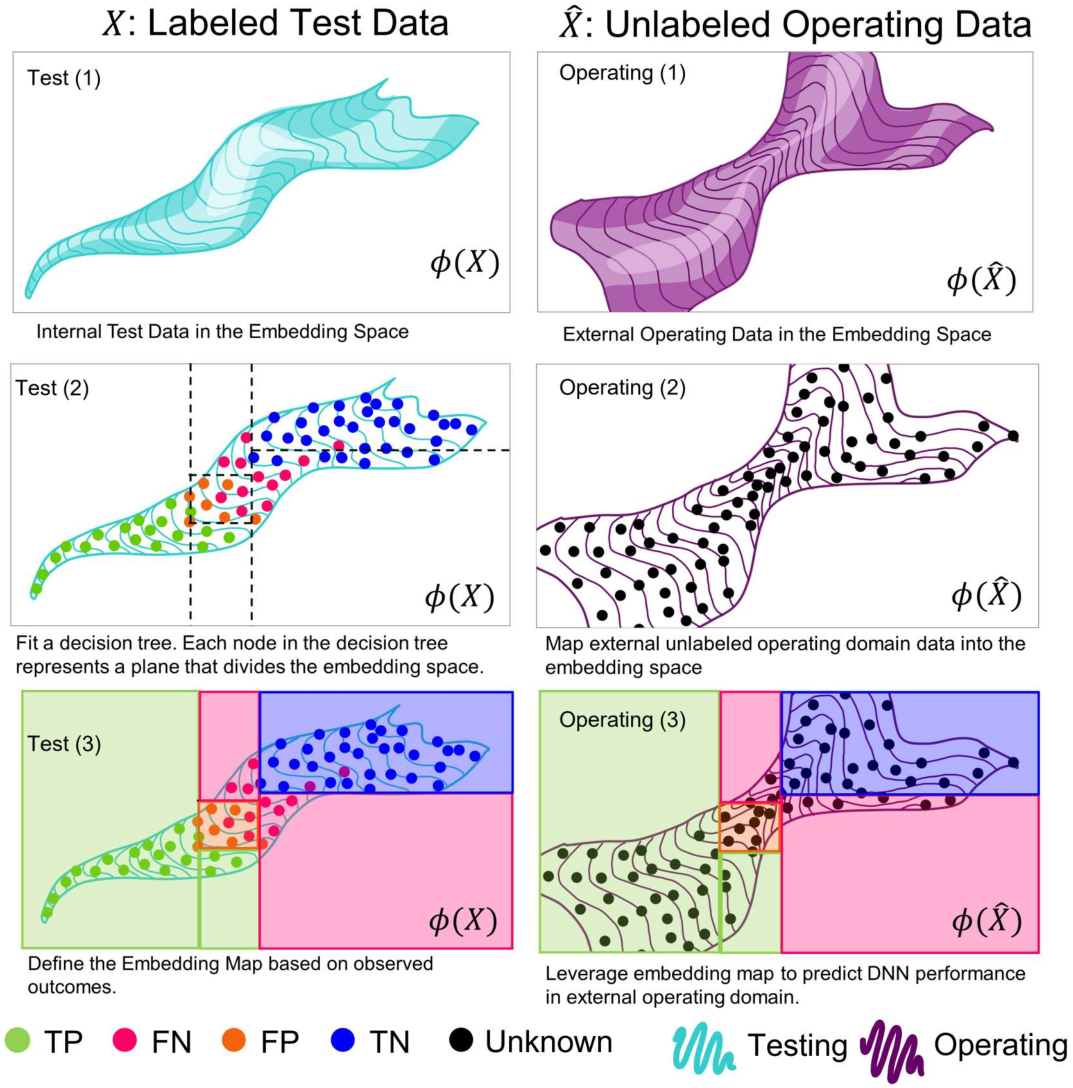
5. Experiments

5.1. Classification Tasks

In our experiments we demonstrate that our embedding map can be used to accurately predict DNN performance for unlabeled, external operating datasets for three binary image classification tasks: pedestrian classification, melanoma classification, and animal classification. Figure 3 shows examples of the images for each of these tasks. For pedestrian classification, we classify image patches as including a pedestrian (the positive class) or not (the negative class), where the negative image patches are randomly cropped patches from driving scene images without pedestrians. The Berkeley Deep Drive 100k (BDD) [40] dataset is the internal dataset and Cityscapes [4] and the Joint Attention in Autonomous Driving (JAAD) [30] are the external datasets. For melanoma classification, we classify an image of a skin lesion as melanoma (the positive class), or benign (the negative class). We use the Human Against Machine 10000 (HAM) dataset [36] for our internal dataset and the SIIM-ISIC Melanoma Classification (ISIC) dataset [31] for our external dataset. For animal classification, we classify an image as an animal (the positive class) or an object (the negative class), with STL10 [3] as the internal dataset and the Common Objects Day and Night (CODaN) [18] and CIFAR-10 [15] as external datasets. The internal and external datasets we investigate provide shifts in image statistics like brightness and saturation, shifts in scene structure, changes in image resolution, and unseen conditions like night images.

5.2. Experimental Setup

For each classification task, we fine-tune three classifiers with different DNN architectures: VGG [34], AlexNet [14], and DenseNet [11]; the pre-trained models are available in the PyTorch library. Each round of training considers 100 batches of images with a batch size of 8, where the images are sampled with a uniform probability for each class. The VGG and AlexNet models are trained with 10 rounds of



Internal Test Data in the Embedding Space

External Operating Data in the Embedding Space

Fit a decision tree. Each node in the decision tree represents a plane that divides the embedding space.

Map external unlabeled operating domain data into the embedding space

Define the Embedding Map based on observed outcomes.

Leverage embedding map to predict DNN performance in external operating domain.

Figure 2. An illustration of the decision tree for mapping DNN embeddings. Test data lie on a manifold in the embedding space. We identify structure in the embedding space as it relates to the DNN outcome. For binary classification the possible outcomes are true positive (TP), false negative (FN), false positive (FP) and true negative (TN). The structure identified using labeled test data can be leveraged to predict the DNN's performance on unlabeled operating data, where the outcome is unknown. Best viewed in color.

training, a learning rate of $1e-6$ and a weight decay of $1e-3$. The DenseNet models are trained with 4 rounds of training, a learning rate of $1e-4$, and a weight decay of $1e-3$. VGG and AlexNet have an embedding space of $4,096$ dimensions. DenseNet has an embedding space of $W \times H \times 1664$ where W and H depend on the initial image size. Like the full DenseNet architecture, we use a Global Average Pooling (GAP) layer to convert from the 3D embedding to a 1664 dimensional vector for each image.

5.3. Network Generalization Prediction

For each architecture in each task, we fit a decision tree with a maximum depth of 10 to distinguish four outcomes: TP, FP, FN, and TN. We refer to the fitted decision tree as our embedding map. Next, we project the external dataset images to the embedding space, $\phi(\hat{X})$, and map the external embeddings to leaves in the embedding map. We compute the probability of each outcome in the external dataset according to Equation 7. The predicted DNN accuracy is the sum of the probability of correct outcomes (TP + TN for binary classification).

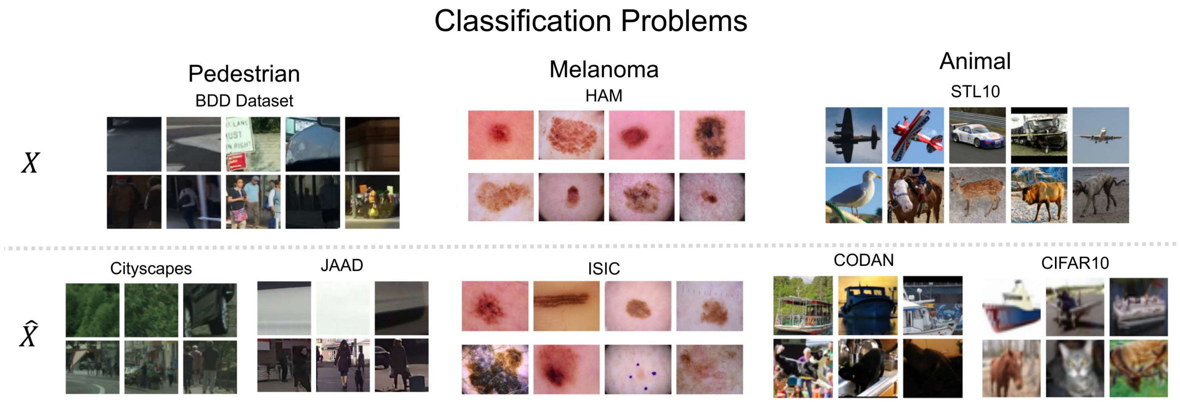


Figure 3. Classification tasks. X indicates the internal dataset that is used to train the DNN classifier and fit the embedding decision tree. $\hat{X}$ indicates the unlabeled, external operating dataset. For each dataset, the top row shows a random sampling of negative examples, and the bottom row shows a random sampling of positive examples.

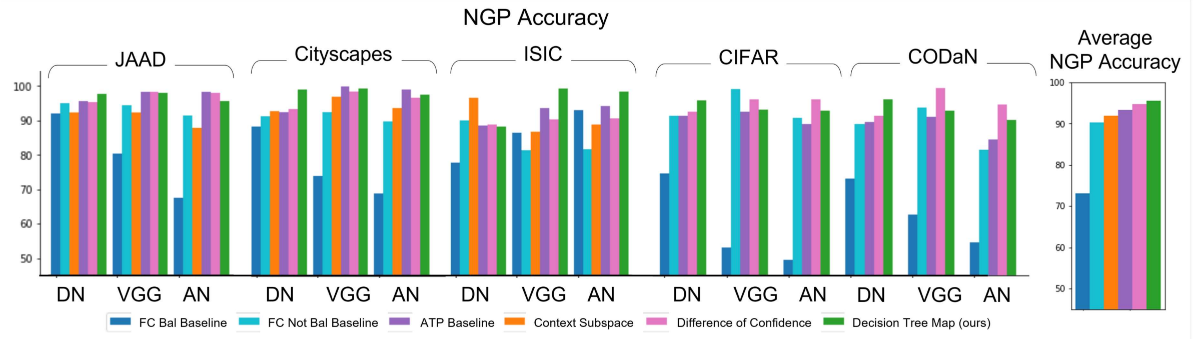


Figure 4. Left: NGP Accuracy for all external operating datasets and DNN architectures, (higher is better). DenseNet shown as DN. AlexNet shown as AN. Right: Average NGP Accuracy by method. Note, the CS baseline is not available for the animal classification tasks. The CS average NGP accuracy is calculated across pedestrian and melanoma classification only.

We compare our NGP results to the DoC NGP approach proposed in [8] and the Context Subspace NGP approach proposed in [27], denoted CS. Guillory et al. propose to measure the DoC and change in prediction accuracy using multiple internal test datasets for a given task and then fit a regressor for NGP. As we use only one internal test dataset for each prediction task, we predict that the change in DNN accuracy between the internal test set and the external operating set will be equivalent to the average difference in softmax score between the external operating dataset and the internal test dataset.

The CS NGP requires distributions of each class and the distribution of context features to make predictions. For pedestrian classification, we use image brightness, scene type, weather, and time of day as available context features, as in the experiments of [27]. For melanoma classification, we use average image hue, saturation, value, patient age, sex, and lesion location as possible context features, as in [26]. Labeled metadata are not available for the animal classification task, so we do not include a comparison to CS NGP for the animal external datasets.

In addition, we compare to an Average Test Performance (ATP) baseline and a baseline Fully Connected (FC) NGP approach. The ATP baseline predicts that the accuracy of the external operating domain will be the same as the internal test set. In FC baseline, an additional FC layer is trained that classifies the embedding as one of the four outcomes. For each DNN model, we trained two FC baselines using the internal test embeddings and outcomes 1) with balanced sampling of examples for each outcome (FC Bal), and 2) without balancing the training sampling by outcome (FC Not Bal).

5.4. NGP Results

The work in [8] presents the NGP results using the Mean Absolute Error (MAE), i.e., the absolute error between the observed model accuracy and the predicted model accuracy. To make the results easier to visualize across multi-

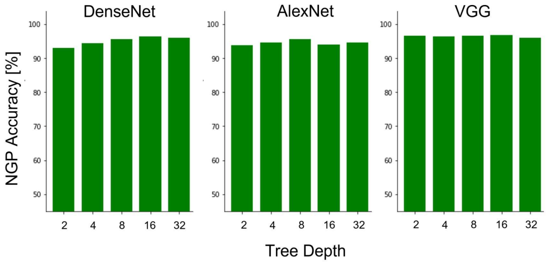


Figure 5. Ablation study by tree depth and DNN architecture.

ple datasets and architectures, we present our results using the NGP accuracy, i.e., $1-$ MAE. Figure 4 shows the NGP results for all tasks, external operating datasets, and DNN architectures. See Appendix A for F1 scores for the NGP task. The average NGP across all experiments is shown on the right in Figure 4. Overall, our DT Map has the best NGP performance, but it is closely followed by DoC and ATP. The NGP results are heterogeneous, and suggest that different NGP methods are more accurate depending on the DNN architecture and external operating data. To investigate these results further, we inspect the embedding space of the JAAD dataset for each architecture, see Section 5.6.

5.5. Tree Depth Ablation Study

In the previous results, we show our NGP prediction performance for a fixed tree depth of 10. We investigate how the tree depth impacts the NGP results across the different DNN architectures with maximum tree depths of 2, 4, 8, 16, and 32; see Figure 5. Decision trees can terminate before the maximum depth is reached if further splits do not improve the performance; when the maximum depth was set to 32, some decision trees terminated before a depth of 32. We see that the NGP performance is consistent across different tree depths for VGG and AlexNet. For the DenseNet architecture, there is some improvement in NGP when the tree depth is ≥ 16.

5.6. Interpreting the NGP Results

To understand these NGP results, we visualize the internal and external datasets in the embedding space using t-SNE [37]. For each external operating dataset and each DNN architecture, we first perform PCA [1] to reduce the corresponding internal test embeddings and external operating embedding to 50 dimensions. Then we use t-SNE to reduce the embedding dimensionality to 2 and plot the image embeddings, see Figure 6, for the embedding plots for the pedestrian internal test set, BDD100K, and external operating set, JAAD. We see a similar embedding structure in the VGG and AlexNet embedding spaces: a cluster of points with an approximately linear DNN decision boundary (drawn in a white dashed line in Figure 6) with TP to one side of the decision boundary, TN to the other side of the decision boundary, and FN and FP close to the decision boundary. In contrast, we see a more complicated embedding space structure for DenseNet. The embedded images form an "S" shape for both BDD100K and JAAD. We still see a separation of the TP and TN images in the embedding space, and mostly FN images in between them; however, it is not possible to draw a line in the embedding space for the DNN decision boundary.

Our embedding map defines regions in the embedding space associated with a given outcome. It is able to map complicated, non-linear manifolds, and therefore is more accurate in predicting the performance for the JAAD, DenseNet DNN. In fact, our embedding map has the best NGP performance in all DenseNet experiments except for the ISIC dataset. In contrast, DoC is leveraging the softmax scores; in situations with an approximately linear decision boundary, e.g., VGG and AlexNet in the JAAD experiments, the softmax score can effectively measure how close a prediction is to the decision boundary and DoC is an accurate NGP method.

6. Discussion

Our NGP experiments show that our proposed DT embedding map accurately performs NGP across a wide range of classification tasks. In particular, we find that our embedding map leverages the embedding space structure well in non-linear embedding manifolds. This is complimentary to the previously proposed DoC method which performs well with linear decision boundaries. In [8] the authors conclude it is "sobering" that the simple DoC approach outperforms other, more complicated baselines. We find that DoC does outperform our embedding map in some experiments, but the embedding space visualization in Figure 6 provides some insight as to why. If the distance from a linear decision boundary is a good indication for whether a DNN prediction is correct, as in VGG and AlexNet in Figure 6, then DoC will be an accurate NGP approach. In contrast, if the structure in the embedding space is more complicated, as in DenseNet in Figure 6, then our embedding map will be a more accurate NGP approach.

When we examine the results in Figure 4 we see that our embedding map tends to do well for the DenseNet architecture across different external datasets and tasks. We believe this is promising because it indicates there are patterns to the DNN embedding space structure for a given DNN architecture across multiple datasets. In addition, the embedding space structure is similar for AlexNet and VGG in the Figure 6 visualization. This is logical, because both AlexNet and VGG have a series of convolutional layers followed by fully connected layers, and both have the same embedding space dimensionality. This suggests the possibility that

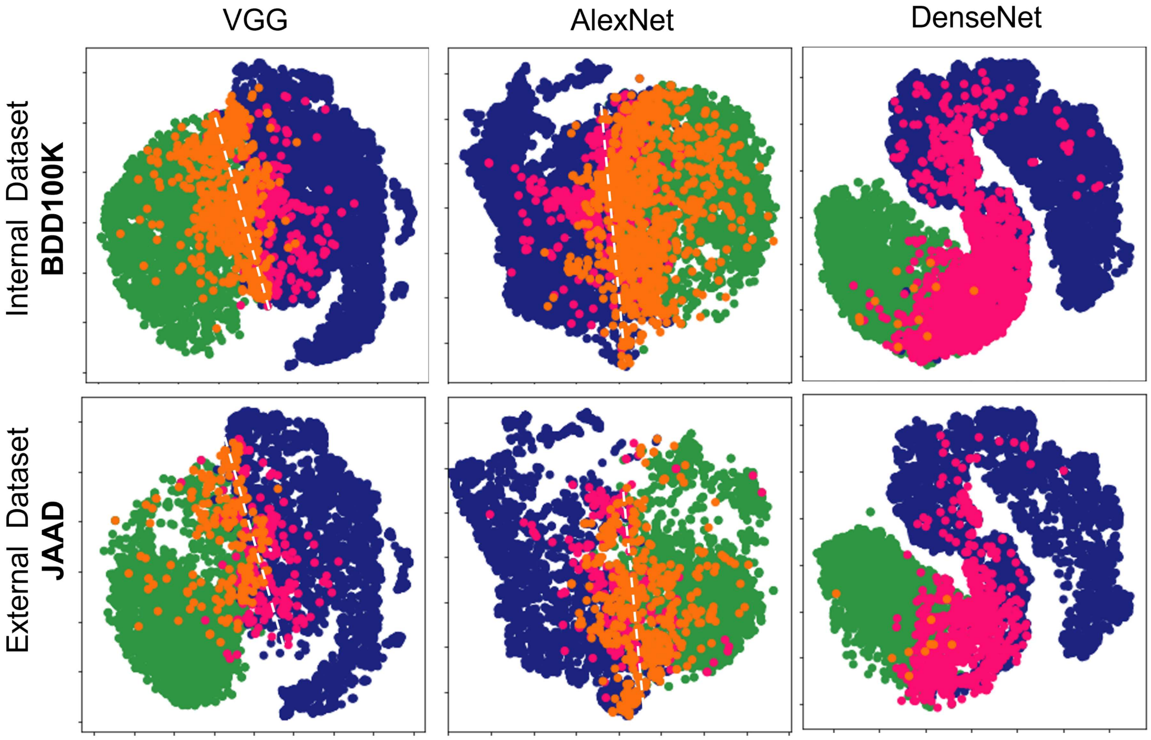


Figure 6. Embedding space visualization for the pedestrian internal test set (BDD100K, top), and external operating set (JAAD, bottom) for each DNN architecture. Each point indicates one image from the respective dataset in the embedding space. True Positive (TP) images shown in green, False Positive (FP) shown in orange, False Negative (FN) shown in pink, True Negative (TN) shown in indigo. Best viewed in color.

groups of architectures with similar layer structures and embedding space dimensionalities have similar DNN embedding space structures. An interesting avenue for future work is to investigate whether patterns in the embedding space structure are indeed seen across different tasks and datasets.

Our proposed approach is not restricted to binary classification problems, and is applicable for other feed-forward supervised learning problems, such as, multi-class classification and object detection. For example, to apply our embedding map to a multi-class classification problem, the outcomes can be defined as ‘class 1 correct’, ‘class 1 incorrect’, etc. for each class and the decision tree can be fit using the procedure we describe.

6.1. Societal Impact

Training DNNs that are fair to all subpopulations is essential to safely deploy DNNs in operation. There is evidence that both pedestrian detection [39] and melanoma classification [38] can have lower performance for some subpopulations, particularly people with darker skin tones. We believe that our approach could be used to recognize if a DNN’s performance will be poor for people from underrepresented subpopulations before harmful failures occur.

7. Conclusions

We propose an NGP method that maps the structure of the DNN embedding space and achieves the best average NGP performance across different tree depths, classification tasks, DNN architectures, and external operating domains. Our embedding map efficiently maps structure in the embedding space, and through embedding space visualizations, we see that in particular our proposed approach outperforms previous NGP approaches in complex, non-linear embedding space structures. More broadly, we believe that the growing body of work around NGP is a promising direction for further research and can be a step towards dependable and practical DNNs for safety critical tasks in unconstrained environments.

7.1. Acknowledgements

M.O. was supported through a Critical Path grant from the U.S. FDA, and by an appointment to the Research Participation Program at the Center for Devices and Radiological Health administered by the Oak Ridge Institute for Science and Education. The contents of this work are solely the responsibility of the authors.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[3] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020.

[6] Weijian Deng and Liang Zheng. Are labels always necessary for classifier accuracy evaluation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15069–15078, 2021.

[7] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[8] Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1134–1144, 2021.

[9] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2020.

[10] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *CoRR*, abs/1610.02136, 2016.

[11] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

[12] Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions. In *International Conference on Learning Representations*, 2018.

[13] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[14] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[16] Y LECUN. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

[17] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

[18] Attila Lengyel, Sourav Garg, Michael Milford, and Jan C. van Gemert. Zero-shot domain adaptation with a physics prior. 2021.

[19] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[21] Ziqian Lin, Sreya Dutta Roy, and Yixuan Li. Mood: Multi-level out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15313–15323, June 2021.

[22] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21464–21475. Curran Associates, Inc., 2020.

[23] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018.

[24] Sina Mohseni, Mandar Pitale, JBS Yadawa, and Zhangyang Wang. Self-supervised learning for generalizable out-of-distribution detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5216–5223, Apr. 2020.

[25] Luke Oakden-Rayner, Jared Dunnmon, Gustavo Carneiro, and Christopher Re. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, CHIL '20, page 151–159, New York, NY, USA, 2020. Association for Computing Machinery.

[26] Molly O'Brien, Julia Bukowski, Greg Hager, Aria Pezeshk, and Mathias Unberath. Evaluating neural network robustness for melanoma classification using mutual information. In *Medical Imaging 2022: Image Processing*, volume 12032, pages 173–177. SPIE, 2022.

[27] Molly O'Brien, Mike Medoff, Julia Bukowski, and Gregory D Hager. Network generalization prediction for safety critical tasks in novel operating domains. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 614–622, 2022.

[28] Molly O'Brien, William Goble, Greg Hager, and Julia Bukowski. Dependable neural networks for safety critical tasks. In *International Workshop on Engineering Dependable and Secure Machine Learning Systems*, pages 126–140. Springer, 2020.

[29] Thomas Ponn, Thomas Kröger, and Frank Diermeyer. Identification and explanation of challenging conditions for camera-based object detection of automated vehicles. *Sensors (Basel, Switzerland)*, 20(13), 2020.

[30] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 206–213, 2017.

[31] Veronica Rotemberg, Nicholas Kurtansky, Brigid Betz-Stablein, Liam Caffery, Emmanouil Chousakos, Noel Codella, Marc Combalia, Stephen Dusza, Pascale Guitera, David Gutman, et al. A patient-centric dataset of images and metadata for identifying melanomas using clinical context. *Scientific data*, 8(1):1–8, 2021.

[32] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991.

[33] Vikash Sehwag, Arjun Nitin Bhagoji, Liwei Song, Chawin Sitawarin, Daniel Cullina, Mung Chiang, and Prateek Mittal. Analyzing the robustness of open-world machine learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, AISec'19, page 105–116, New York, NY, USA, 2019. Association for Computing Machinery.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19339–19352. Curran Associates, Inc., 2020.

[36] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018.

[37] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[38] David Wen, Saad M Khan, Antonio Ji Xu, Hussein Ibrahim, Luke Smith, Jose Caballero, Luis Zepeda, Carlos de Blas Perez, Alastair K Denniston, Xiaoxuan Liu, et al. Characteristics of publicly available skin cancer image datasets: a systematic review. *The Lancet Digital Health*, 2021.

[39] Benjamin Wilson, Judy Hoffman, and Jamie Morgenstern. Predictive inequity in object detection. *arXiv preprint arXiv:1902.11097*, 2019.

[40] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.

[41] Runtian Zhai, Chen Dan, Zico Kolter, and Pradeep Ravikumar. Understanding why generalized reweighting does not improve over erm. *arXiv preprint arXiv:2201.12293*, 2022.