Using Local Complexity to Evaluate Out-of-Distribution Generalization

Grace O'Brien
Pacific Northwest National Laboratory
University of Washington
`grace.obrien@pnnl.gov`

Andrew Aguilar
Pacific Northwest National Laboratory
University of Washington

Robert Jasper
Pacific Northwest National Laboratory

Henry Kvinge
Pacific Northwest National Laboratory
University of Washington

Sarah Scullen
Pacific Northwest National Laboratory

Helen Jenne
Pacific Northwest National Laboratory

Abstract

Despite their growing ubiquity, the inner workings of deep neural networks are still largely a black box. Even in the case of classification tasks, common methods used to assess model performance do not give insight into whether the model will generalize to unseen data. In this extended abstract, we investigate local complexity (LC) Humayun et al. [2024], a geometric measure of the input space, as a predictor of model performance on out-of-distribution (OOD) data. We find that LC alone is not sufficient to predict model generalization, but that it does capture meaningful information about the correctness of individual predictions, suggesting it may be useful as part of a larger set of tools to understand OOD generalization.

1 Introduction

Out-of-distribution (OOD) generalization–reliable performance when the deployment distribution differs from the training distribution–remains a central challenge in machine learning. Research addressing this problem ranges from developing algorithms to improve OOD generalization to identifying model properties that facilitate robust generalization, such as the stability of estimations under small data perturbations Gupta and Rothenhäusler [2023].

Recently, Humayun et al. [2024] introduced *local complexity (LC)*, a data-dependent geometric measure that approximates the local density of linear regions around inputs in networks with piecewise-linear activation functions. They proposed local complexity as a progress measure Barak et al. [2022] and linked decreases in LC in the final phase of training to grokking (delayed generalization) and increased adversarial robustness. Building on this work and motivated by the intuition that larger linear regions around the training data promote generalization, we investigate whether LC can serve as an effective predictor of model generalization capabilities.

The key question we ask is whether LC can predict OOD classification performance at the model or the per-example level. At the model level, we examine whether qualitative training-time LC trajectories predict performance on OOD data. We find that these dynamics alone are insufficient for reliable predictions about model generalization, primarily because LC dynamics are highly dependent on model architecture. However, at the per-example level, we find that LC is significantly lower for correctly classified OOD inputs compared to misclassified ones, suggesting that it captures a meaningful component of OOD generalization. This indicates that while LC may not work as a standalone predictor at the model level, it may complement other uncertainty measures, a direction we plan to explore in future work.

2 Local complexity

Local complexity, introduced by Humayun et al. [2024], measures the density of spline partition regions that tile a deep neural network's (DNN) input space. DNNs map an input vector x to an output vector y through a composition of affine and nonlinear functions. In particular, a network with activation function **a** and K layers can be written as

$$y = b_K + W_K \mathbf{a}\Big[b_{K-1} + W_{K-1}\mathbf{a}\big[\dots b_2 + W_2\mathbf{a}\left[b_1 + W_1\mathbf{a}\left[b_0 + W_0 x\right]\right]\dots\big]\Big]. \tag{1}$$

where b_k is the vector of biases for hidden layer $k+1$ and W_k is the weights matrix applied to the kth layer. In this work, the activation function **a** is always ReLU. Balestriero and Baraniuk [2018] showed that for any piecewise-linear activation function, (1) is a continuous piecewise-affine spline operator. That is, there is a partition Ω of the input space such that the network acts affinely on any region $\omega \in \Omega$. These are the spline partition regions whose density local complexity tracks.

Measuring local complexity For a convex region $\mathcal{V}$ in the input space of our network, local complexity can be computed in terms of the hyperplanes stemming from each neuron. For the kth layer of a network with weight matrix W_k, bias vector b_k, and output dimension d_k, the spline partition Ω_k of the input space to layer k can be written as the hyperplane arrangement where each hyperplane is associated to a neuron in layer k, that is, $\partial\Omega_k = \bigcup_{i=1}^{d_k} \mathcal{H}_k^{(i)}$, where $\mathcal{H}_k^{(i)} = \Big\{x \in \mathbb{R}^{d_{k-1}} \colon \langle w_k^{(i)}, x\rangle + b_k^{(i)} = 0\Big\}$, $w_k^{(i)}$ is the ith row of W_k, and $b_k^{(i)}$ is the ith entry of b_k.

To approximate the LC induced by the kth layer on $\mathcal{V}$, we simply count the number of regions in $\bigcup_i^{d_k} \Phi \cap \mathcal{H}_k^{(i)}$, where Φ is the embedded representation of $\mathcal{V}$ after being passed through layers 1 through $k-1$ of the network. To simplify computation, we consider the number of hyperplanes passing through Φ as a proxy for the LC of $\mathcal{V}$ at layer k. Following Humayun et al. [2024] and utilizing their code base[1], to measure how local complexity changes throughout training, we randomly sample data points and construct randomly-oriented, P-dimensional ℓ_1-norm balls with radius r centered at each data point. We then count the number of hyperplanes passing through these neighborhoods to approximate the local complexity in that area for a given layer. In this work, we take $P = 2$ and $r = 0.5$. These choices are discussed further in Appendix B.2.

Training dynamics of local complexity In Section 5, we will compare the dynamics of local complexity across models. We use the word "dynamics" to refer to the general qualitative behavior, as well as phases described in Humayun et al. [2024]. In that work, the authors describe the *two descent phases* of LC. After initialization, they note the first descent. This phase does not always occur; it is dependent on the network parameterization and initialization. Then, in the ascent phase, region density accumulates around training and testing points until training interpolation is reached. Finally, in the second descent phase, also called the *region migration* phase, the nonlinearities shift towards the decision boundary leading to increased LC near the boundary and decreased LC away from the training data. They document these dynamics for a variety of model architectures and datasets. Additionally, the authors study how architecture and regularization influence LC dynamics in both Humayun et al. [2024] and Humayun et al. [2023b].

3 Related work

OOD generalization A recent survey Yu et al. [2024] categorizes OOD evaluation into three groups based what test data it requires. OOD performance testing evaluates models when labeled test data is available, OOD performance prediction evaluates models when unlabeled test data is available, and OOD intrinsic property characterization aims to discover properties of models that inform OOD generalization when no test data is available. Examples of intrinsic properties include characteristics

[1]`https://github.com/AhmedImtiazPrio/grok-adversarial`; released with an MIT License

like stability of estimates under small perturbations Gupta and Rothenhäusler [2023] and flatness Foret et al. [2020]. When unlabeled test data is available, Garg et al. [2022] proposes the method Average Thresholded Confidence to predict accuracy on OOD data using model confidence.

Local complexity Patel and Montúfar [2024] uses a slightly different definition of local complexity and develops theory that explains some of the results in Humayun et al. [2024]. Namely, they show that their formulation of local complexity is an upper bound on the total variation of the network over the input space. They also connect local complexity to local rank, the average dimension of the feature manifold at intermediate layers. Though they do not discuss local complexity directly, Hanin and Rolnick [2019] studies spline partitions and investigates alternative ways to quantify the changing partition regions.

4 Experimental set up

To test model performance on out-of-distribution data, we train 25 different models on CIFAR-10[2] Krizhevsky et al. [2009]. These models include architectures from the ResNet, DenseNet, and VGG families, among others. We trained models with and without data augmentation (random crop, random horizontal flip, and random erasing) to compare its effect on LC. A full list of models can be found in Appendix B.1, and details on model training in Appendix B.2. To simulate OOD data, we evaluated each of our models on a subset of images from CIFAR-10-Warehouse[3] (CIFAR-10-W) Sun et al. [2024], a collection of 180 datasets motivated by the observation that many collections of OOD testsets have a small number of domains or rely on synthetic corruptions. Specifically, we used the subset of CIFAR-10-W sourced from the internet image search engine, 360. This dataset has over 60,000 images separated into 12 different colors across the same classes as CIFAR-10. The images were chosen through keyword searches of the form "*color class*" (i.e., "red airplane").

5 Results and Discussion

Different LC dynamics, similar OOD performance As in Humayun et al. [2023b], our work corroborates that training-time LC trajectories depend on model architecture. Figure 1 shows three examples of LC trajectories, measured at the final layer before the classification layer. DenseNet-161 (left, top) is closest to having two descent phases[4]: after rising upon initialization, the mean number of regions intersected decreases slightly, and then flattens, before decreasing for the remainder of training. In contrast, for both ResNet-34 and Inception-V3 the sharp spike after initialization is followed by a much faster decrease. (Note that the maximum number of possible hyperplane intersections is directly related to the number of neurons, making it difficult to compare LC values directly between models with different architectures). Despite the qualitative differences in local complexity dynamics, DenseNet-161 and ResNet-34 perform very similarly on CIFAR-10-W, with 55.18% and 54.34% accuracy, respectively. While Inception-V3 and ResNet-34 have very similar LC dynamics, Inception-V3 outperforms ResNet-34, getting 58.41% accuracy. As shown in Figure 1, Inception-V3, DenseNet-161, and ResNet-34 are 87.74%, 91.44%, 88.83% accurate on the CIFAR-10 test set, respectively. Overall, the trends we saw in LC dynamics confirmed findings in Humayun et al. [2023b] that LC is highly dependent on model architecture. (Figure 6 in Appendix E show the similarity in LC dynamics between models from the same architecture families.) Consequently, LC dynamics alone are not sufficient to predict a model's ability to generalize to OOD data.

Comparing LC of correctly and incorrectly classified examples Though LC alone is not enough to predict model performance on OOD data overall, we found that it does reflect some information about the correctness of individual predictions. We evaluate DenseNet-161 and ResNet-34 on 128 random samples from each of the 12 color groups of CIFAR-10-W's 360 dataset, separate the data based on whether it was correctly or incorrectly classified by the model, and measure the difference in LC at the end of training (we omitted Inception-V3 because of its similarity in LC dynamics to ResNet-34). We find, on average, that the number of hyperplanes intersecting a neighborhood is higher among the incorrectly classified points than those correctly classified (Figure 3). In fact, computing

[2]`https://www.cs.toronto.edu/~kriz/cifar.html`; released with an MIT License

[3]Licensed under CC BY-NC 4.0 1

[4]All three models use batch normalization, which could be the reason we see only a single descent phase.

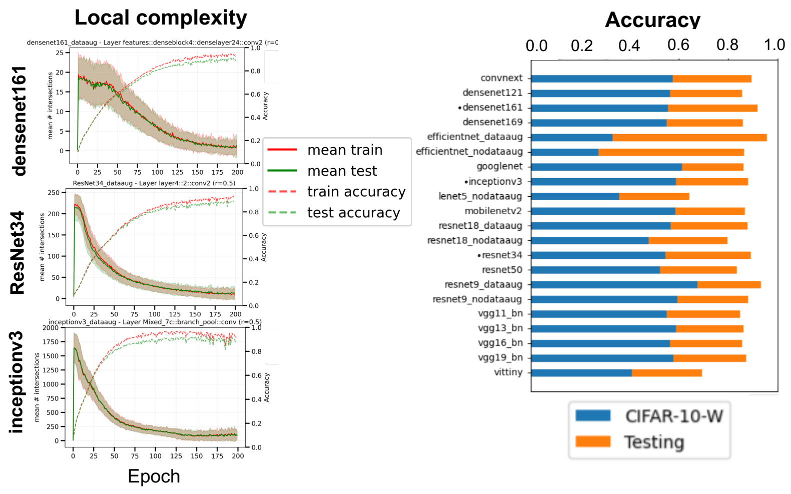

Figure 1: Left: LC on training and testing data (CIFAR-10) throughout training for three models. Differences in model architecture lead to different qualitative behavior in LC. Right: Accuracy on CIFAR-10-W is shown in blue, with accuracy on the CIFAR-10 test set overlaid in orange for comparison.

t-tests on LC measurements by color and model reveals that the majority of these differences are statistically significant ($p \leq 0.05$). We found 19 of 24 t-tests were statistically significant. Further details can be found in Appendix C. A possible future direction is to train a classifier that considers LC, among other information, as input and predicts if an individual OOD example will be correctly classified.

LC and model confidence Since average thresholded confidence is used to predict accuracy on unlabeled test data Garg et al. [2022], as an initial step towards understanding what features could be used in conjunction with LC, we study the relationship between LC and confidence. Our intuition is guided by the idea that the spline partitions will migrate towards the decision boundary throughout training Humayun et al. [2024]. So, we expect that an example with low local complexity will be firmly located within a particular label's region, implying that the model is confident in its prediction. Conversely, an example near the decision boundary will have high LC and low confidence. The plots of model confidence vs. LC on our in-distribution testing data shown in Figure 2, top, support this. We see that for both DenseNet-161 and ResNet-34 there are a cluster of correctly classified points in the upper left, while incorrectly classified points more often fall in the lower right.

Figure 2, bottom, shows model confidence vs. LC but for CIFAR-10-W data. In this case, we still see a distinct cluster of correctly classified points in the upper left, but also see more incorrectly classified examples in this area. This suggests that these examples fall firmly within the region of the input space for a particular label (far from the decision boundary), but that it is the incorrect label, reflecting one way in which data can be OOD. Additionally, we see many examples in the OOD data that fall in the lower left. These examples are not easily explained by our current understanding and warrant further investigation. One could use a tool like SplineCam Humayun et al. [2023a] to better understand where in the input space these OOD examples lie.

6 Future directions

The statistically significant difference in mean LC between correctly and incorrectly classified OOD samples indicates this measure captures meaningful aspects of OOD generalization and suggests several avenues for future study. Future work could use the full distribution of LC values rather than just the means, including class-specific patterns, to further understand model generalization and robustness. Notably, since this approach computes LC at the end of training, it can be extended to pretrained models, broadening its practical applications.

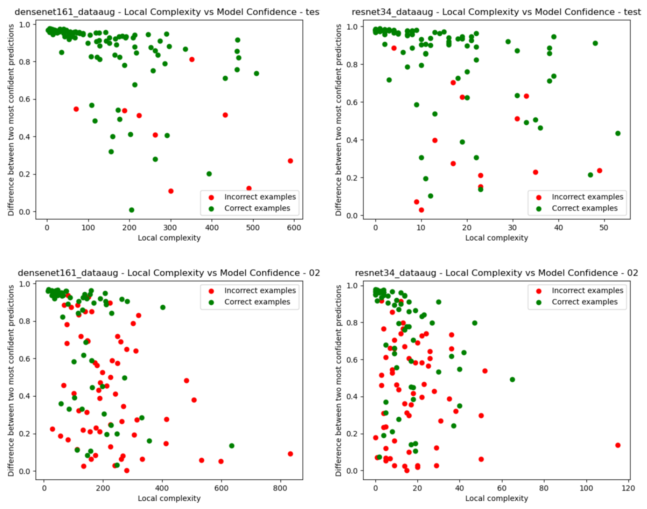

Figure 2: Top: Model confidence vs. local complexity for examples from the in-distribution CIFAR-10 holdout test set. To evaluate model confidence, we take the difference of the two highest softmax logits. Bottom: Model confidence vs. local complexity for examples from CIFAR-10-W in color '02' ("orange") on three models.

It would also be interesting to use LC to identify confusing training examples or important data features. Examples of questions include how removing high-LC training examples affects generalization performance, and whether analyzing which hyperplanes intersect neighborhoods most often could reveal key features. While the present work uses CIFAR-10 and CIFAR-10-W, in future work we plan to expand to other datasets and evaluate our hypotheses on a larger collections of models.

References

Randall Balestriero and Richard Baraniuk. A spline theory of deep networks. In *International Conference on Machine Learning*. PMLR, 2018.

Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: SGD learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Saurabh Garg, Sivaraman Balakrishnan, Zachary Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. In *International Conference on Learning Representations*, 2022.

Suyash Gupta and Dominik Rothenhäusler. The s-value: evaluating stability with respect to distributional shifts. *Advances in Neural Information Processing Systems*, 36:72058–72070, 2023.

Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. In *International Conference on Machine Learning*. PMLR, 2019.

Ahmed Imtiaz Humayun, Randall Balestriero, Guha Balakrishnan, and Richard Baraniuk. SplineCam: Exact visualization and characterization of deep network geometry and decision boundaries. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3789–3798, 2023a.

Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. Training dynamics of deep network linear regions. *arXiv preprint arXiv:2310.12977*, 2023b.

Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. Deep networks always grok and here is why. *arXiv preprint arXiv:2402.15555*, 2024.

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

Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Norman Mu and Justin Gilmer. MNIST-C: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906:0.2337*, 2019.

Niket Patel and Guido Montúfar. On the local complexity of linear regions in deep ReLU networks. *arXiv preprint arXiv:2412.18283*, 2024.

Xiaoxiao Sun, Leng Xingjian, Zijian Wang, Yang Yang, Zi Huang, and Liang Zheng. CIFAR-10-Warehouse: Broad and more realistic testbeds in model generalization analysis. *arXiv preprint arXiv:2310.04414*, 2024.

Han Yu, Jiashuo Liu, Xingxuan Zhang, Jiayun Wu, and Peng Cui. A survey on evaluation of out-of-distribution generalization. *ArXiv*, abs/2403.01874, 2024.

A Related work

B Additional details on experimental setup

B.1 Complete list of trained models

We trained 25 models on CIFAR-10. Unless stated otherwise, all models were trained with data augmentation. Below is the complete list:

- ConvNeXt
- DenseNet-121
- DenseNet-161
- DenseNet-169
- EfficientNet
- EfficientNet without data aug.
- GoogLeNet
- Inception-V3
- LeNet-5 without data aug.
- MobileNet-V2
- ResNet-18
- ResNet-18 without data aug.
- ResNet-34
- ResNet-50
- ResNet-9
- ResNet-9 without data aug.
- VGG-11
- VGG-11 with batch norm.
- VGG-13
- VGG-13 with batch norm.
- VGG-16
- VGG-16 with batch norm.
- VGG-19
- VGG-19 with batch norm.
- ViTTiny

Many of these models were listed here[5] as suggestions for use on CIFAR-10. Of these models, we chose to exclude VGG-11, VGG-13, VGG-16, and VGG-19 without batch normalization from analysis because they never learned better than random chance.

B.2 Experimental choices

Local complexity hyperparameters We made choices in our experimental design based on observations of preliminary MNIST experiments. For example, we set the radius $r = 0.5$ for the ℓ_1-neighborhoods (Section 2). We experimented with various sizes of radii and settled on 0.5 as it seemed to capture the most change in LC–with larger r, the number of intersections was always quite high and smaller caused the neighborhoods to be too small to consistently intersect any of the hyperplanes.

We chose the dimension of the neighborhoods to be $P = 2$. Our experiments with MNIST showed that increasing the dimension increased the scale of the number of intersections per neighborhood, but did not tend to change the overall dynamics. Thus, we chose the minimum dimension for computational efficiency.

Dataset selection We chose to use CIFAR-10-W over a dataset with synthetic corruptions to better simulate real-world encounters of unseen data. We chose to experiment on only a subset of CIFAR-10-W. There are many different datasets from various internet search engines within CIFAR-10-W. In addition, for some search engines, they create cartoon datasets by searching "*color class* cartoon" to further push the data out of distribution. The dataset also includes images generated using diffusion models. Within CIFAR-10-W, we focused on the search engine 360 for simplicity and the existence of an analogous cartoon version, though we ultimately did not analyze that data.

Model confidence In Section 5, we evaluate model confidence using the difference of the two highest softmax logits. We chose this computation instead of simply taking the highest value as it suggests the model prefers a single label over all others which we interpret as a data point lying far from the decision boundary.

[5] `https://zenodo.org/badge/latestdoi/195914773`

Training details When training each of our models, we used stochastic gradient descent as our optimizer and trained for 200 epochs. We set weight decay to be 0.01, learning rate to be 0.1, momentum to be 0.9, and used a batch size of 128. We used a scheduler to set the learning rate to follow a linear warmup schedule followed by a cosine annealing schedule. We used the default train/test split of CIFAR-10 with 50,000 training points and 10,000 test points. All models were trained on a single NVIDIA A100.

C Full t-test results

We chose to run independent t-tests as we wish to compare the means of a statistic between two different populations. We sample 128 points randomly from CIFAR-10-W and assume that the LC of each point is an independent observation. We find that the correctly and incorrectly classified examples have similar variances. The table below shows the p-values for each of the t-tests described in Section 5. The bold values are statistically significant ($p \leq 0.05$).

Color	DenseNet-161	ResNet-34
Red '01'	**0.01688**	**0.00176**
Orange '02'	**0.00021**	0.07751
Yellow '03'	**3.05161 e-07**	**1.30424 e-05**
Green '04'	**0.00059**	**5.04387 e-06**
Light Blue '05'	**0.00812**	0.27444
Blue '06'	**4.41172 e-09**	**0.00011**
Purple '07'	**0.00019**	0.25146
Pink '08'	**1.0775 e-06**	**8.61807 e-05**
Brown '09'	**0.00041**	**9.86942 e-05**
Gray '10'	**2.17557 e-06**	**7.99038 e-08**
White '11'	**0.04719**	**6.84716 e-05**
Black '12'	0.69873	0.08998

In addition to studying differences between means, we also examined how LC changes throughout training, both overall (Figure 3) and for individual examples (Figure 4). Although there is notable overlap between the correctly and incorrectly classified examples, the statistically significant difference between the two distributions suggests that we may be able to discern which examples will be correctly classified using local complexity.

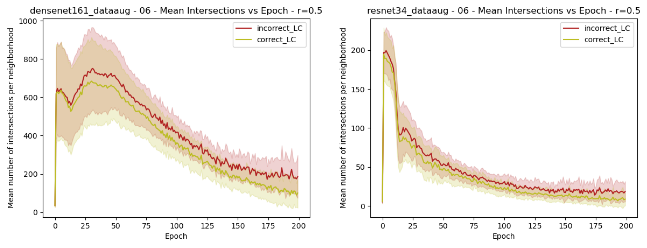

Figure 3: Mean intersections per neighborhood for examples from CIFAR-10-W in color '06' ("blue") from three models. Examples are separated based on if they were correctly (green) or incorrectly (red) classified by the model. The mean among the incorrectly classified examples is higher than among the correctly classified examples. This is true across all colors. The statistical significance is discussed in Section 5.

D Preliminary MNIST experiments

Preliminary experiments on MNIST[6] LeCun et al. [2010] informed our approach to studying CIFAR-10. We trained 10 different models on MNIST for 10 epochs each and computed local complexity throughout training on 100 training and 100 testing examples. We then evaluated each of the models

[6]Licensed under MIT License

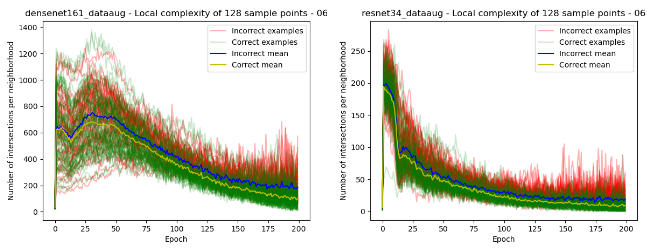

Figure 4: Number of intersections per neighborhood for 128 examples taken from CIFAR-10-W in color ‘06’ (“blue”). Examples that are correctly classified are shown in green and those incorrectly classified in red. This allows us to see how local complexity changes throughout training for each example individually.

on MNIST-C[7] Mu and Gilmer [2019], a corrupted, synthetic version of MNIST. We compared accuracy and LC across models and across the 15 corruptions in MNIST-C. We found that even when a model was able to achieve high accuracy on corrupted data, the LC dynamics for “brightness” and “fog” were qualitatively different than for other corruptions. These two corruptions are the only two that edit the contrast of the original images leading us to wonder if contrast is a particularly important feature in the model’s decision-making process. Further study could reveal if local complexity can identify how influential certain data features are in model predictions.

E Additional figures

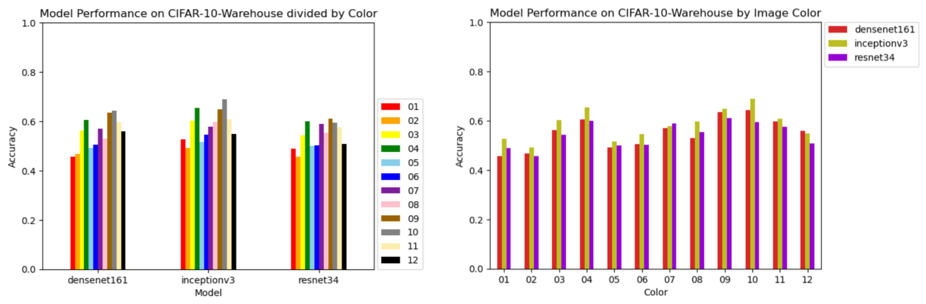

Figure 5: Two bar plots showing model performance separated by each of the color groups within CIFAR-10-W. Left: comparing the performance between the 12 colors by each of the 3 models. Right: comparing performance between models on each of the 12 colors.

[7]Licensed under Apache License Version 2.0

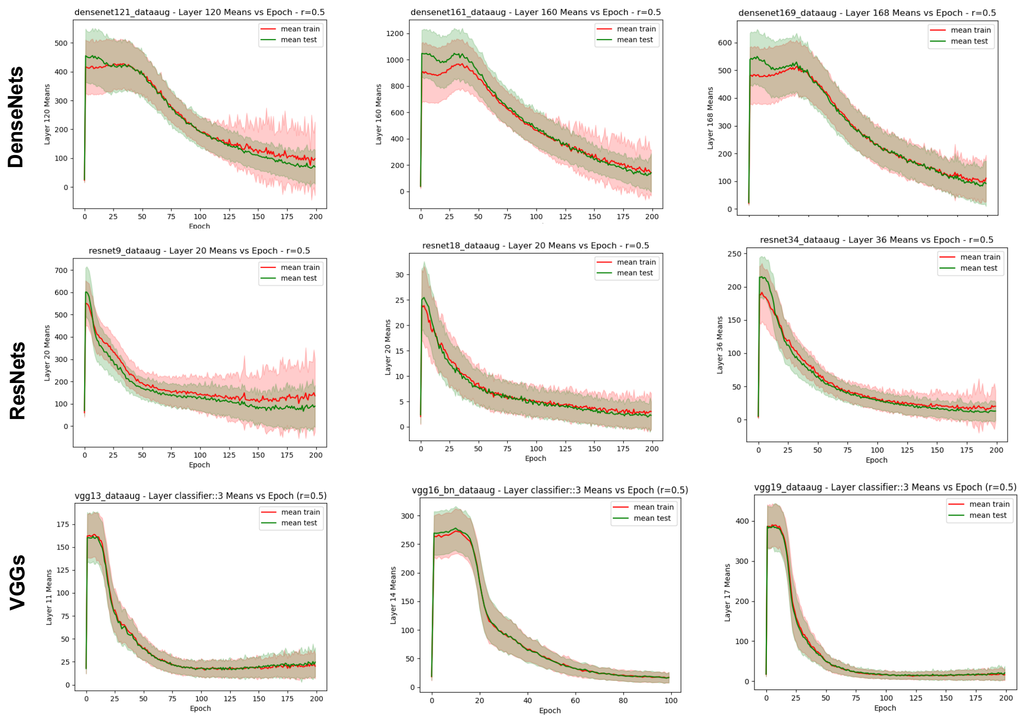

Figure 6: LC at the final layer before the classification layer for three DenseNet architectures (DenseNet-121, DenseNet-161, DenseNet-169), three ResNet architectures (Resnet-9, ResNet-18, ResNet-34) and three VGG architectures (VGG13, VGG16, and VGG19). LC dynamics look very similar between models with similar architecture.