Explaining Model Parameters Using the Product Space [*]

Ethan Payne[1][0000−0002−6620−8336], David Patrick[1,2][0000−0003−2556−8818], and Amanda S. Fernandez[1][0000−0003−2397−0838]

[1] The University of Texas at San Antonio, TX, USA
[2] Texas State University, San Marcos, TX, USA

Abstract. With the increasing interest in explainable attribution for deep neural networks, it is important to consider not only the importance of individual inputs, but also the model parameters themselves. Existing methods, such as Neuron Integrated Gradients [18] and Conductance [6], attempt model attribution by applying attribution methods, such as Integrated Gradients, to the inputs of each model parameter. While these methods seem to map attributions to individual parameters, these are actually aggregated feature attributions which completely ignore the parameter space and also suffer from the same underlying limitations of Integrated Gradients. In this work, we compute parameter attributions by leveraging the recent family of measures proposed by Generalized Integrated Attributions, by instead computing integrals over the product space of inputs and parameters. This usage of the product space allows us to now explain individual neurons from varying perspectives and interpret them with the same intuition as inputs. To the best of our knowledge, ours is the first method which actually utilizes the gradient landscape of the parameter space to explain each individual weight and bias. We confirm the utility of our parameter attributions by computing exploratory statistics for a wide variety of image classification datasets and by performing pruning analyses on a standard architecture, which demonstrate that our attribution measures are able to identify both important and unimportant neurons in a convolutional neural network.

Keywords: Attribution · Saliency · Influence · Integrated Gradients · Expected Gradients · Explainability · Causal Inference · Pruning · Unlearning

1 Introduction

As deep learning architectures grow in size and complexity, the push for explainability of model predictions and performance continues. While existing attribution methods are able to generate importance values for model inputs and

[*] Supported by the National Science Foundation under Grant No. 2134237. Code: https://github.com/UTSA-VAIL/Explaining-Model-Parameters-Using-the-Product-Space

extracted features, it is also critical to consider the model parameters themselves in the context of the ambient parameter space.

We first briefly summarize the several types of attributions for clarity of terminology in the following subsections. Figure 1 provides an illustration of these attribution types.

Input attributions assign importance values, or some other value of interest, to each dimension of the input. For the particular case of image recognition this means assigning importance values to each pixel of the input, or ideally to each pixel's color channels individually.

Intermediate Feature Attributions assign importance values to the feature maps generated by layers and modules within a neural network, in a method similar to input attribution. Since these feature maps are simply transformations of the input data, we can conveniently treat them using the same attribution methodology as we used for inputs.

Parameter Attributions facilitate computing importance values for individual model parameters, i.e. the weights and biases in a convolutional neural network. Since the model itself is an entirely different class of object than either the inputs or extracted features, we must develop a new methodology for extending the theory of attributions to the parameter space.

1.1 Our Contribution

In this work, we achieve a more complete and faithful method of parameter attribution, leveraging the reformulated attribution framework of Generalized Integrated Attributions [21]. Similar to Neuron Integrated Gradients [18] and Conductance [6], we assign an integrated measure to a parameter within a model. However, unlike these previous methods, we do not aggregate path-integrated feature attributions, but rather use the generalized volume-integral formulation proposed in [21], and account for the parameter space by integrating over the product space of inputs and parameter values. By integrating over a set in the parameter space, we are able to interpret the resulting parameter attributions using the same theory as for input and feature attributions. Additionally, this formulation allows us to assign unique attribution values to each weight and bias in a convolutional neural network, which was not possible using previous methods.

To ensure that the computed measures reflect the dataset of interest rather than some arbitrary or counterfactual baseline value, we follow the approaches of Expected Gradients [7] and Generalized Integrated Attributions [21] and take the expectation in the input space over a set from the training dataset. Using this new formulation of *Parameter Explanations using the Product Space (PEPS)*, we are able to extend each of the measures proposed in [21] to model parameters in addition to inputs. Our experiments confirm that our measures are able to successfully identify important and unimportant neurons, and we summarize our findings regarding the distribution of these measures for several datasets, model training statuses, and hyperparameter combinations using exploratory plots. Furthermore, our measures are able to un-learn a specific class from the trained model without destroying the model's performance on the remaining classes, and we identify several trends within the distributions of our attributions which might be used for future training diagnostics and improved robustness.

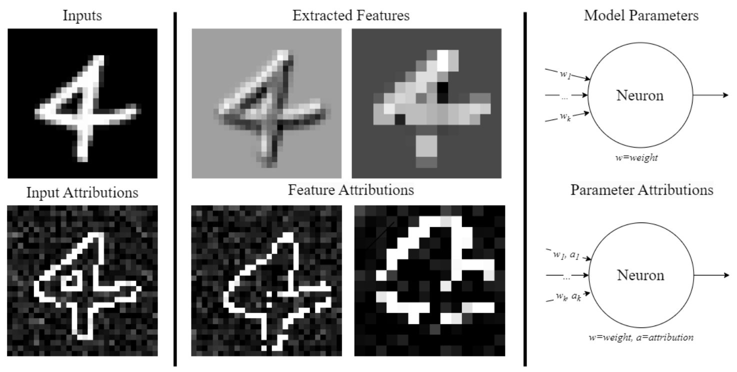

Fig. 1: Mock-up visualization of attributions corresponding to different types of explainable objects. Even once the extracted features cease to resemble the original input, we can still compute attributions using the same methodology as used for input attributions. However, model parameters require a different approach in order to reflect the behavior of the model's own gradient landscape.

2 Related Work

2.1 Input Attributions

Many methods for attributing model predictions to specific features of the input utilize information contained in the model gradients [20,19,24,22,3,8]. Sundararajan et al. devised an attribution method called Integrated Gradients [22] which computes the integral of feature gradients over a linear path from an input to a reference. The value of this integral can then be tracked with respect to each pixel of the input in order to obtain a pixel-wise attribution map. In a similar approach, DeepLift [17] propagates contribution scores according to differences from a reference, which, akin to integrated gradients, requires some justification for correct or appropriate reference values. By incorporating the theory associated with classical Shapley values, Lundberg et al.[14] contextualize several attribution methods such as DeepLift as additive explanations, and uses the theory of SHAP values to unify these different approaches to attribution under a single framework. Ancona et al. [2] show that many gradient-based attribution methods are closely related and can be described with a unified formulation, and propose the metric Sensitivity-n to evaluate these methods.

In contrast to gradient-based methods, other approaches to quantifying feature importance such as the search-based Parallel Local Search (PLS) [10] are able to outperform other state-of-the-art attribution methods. Still, while these search-based methods can be highly useful, they lack much of the a-priori and intuitive explainability offered by gradient-based methods.

A recent reformulation known as Expected Gradients [7] was proposed by Erion et al. to directly improve upon the original method of Integrated Gradients, by computing attributions as an expected value of gradients over the input dataset, thus alleviating the issue of counterfactual baselines. This prompted yet another recent reformulation in the form of Generalized Integrated Gradients [21].

2.2 Intermediate Feature Attributions

It can often be beneficial to consider the intermediate features extracted by machine learning models when collecting attribution information, either for the purpose of improving input attributions as in [3], or in order to assign attributions to model parameters. Using an integration-based method similar to Integrated Gradients, Leino et al. [13] compute an influence-directed attribution measure which can be applied to internal neurons within a model. In two very similar approaches, Shrikumar et al. [18] and Dhamdhere et al. [6] propose Neuron Integrated Gradients and Conductance respectively by applying the principles of Integrated Gradients to the inputs of a given parameter. These feature attributions are then pooled and assigned to the parameter. Although these methods do assign attribution values to individual model parameters, these values are actually aggregated feature attributions rather than representations of the model's own gradient landscape. Additionally, since both of these methods integrate feature gradients over a path in the input space, these aggregated feature attributions are relevant only to this path, entirely neglecting the parameter space itself.

2.3 Parameter Attributions

While methods like [24] intentionally avoid using gradient information when determining neuron importance, our experiments demonstrate that there is a wealth of useful information available within model gradients. Other work on neuron-level analysis has been conducted in the Natural Language Processing (NLP) space [16], but it remains to extend these methods to other domains such as vision. Our method is also similar to the Shapley-value approach of [9], which uses permutations of model parameters in a multi-armed bandit algorithm to determine neuron importance values, as well as the neuron ablation approach of [1] which uses majority voting. While these two methods can be effective for specific use cases, such algorithmic approaches lack the flexibility, intuition, and generalizability of our method for accomodating diverse subjective user needs.

3 Approach

While previous methods consider only integrals within the input space with respect to a single point in parameter space (the model weight state), we instead consider the product space of inputs and parameter values (Figure 2). By integrating over a set in the product space, we can explain a parameter not only with respect to a given set of inputs, but also with respect to other possible parameter values. We verify the effectiveness of this method in section 4.

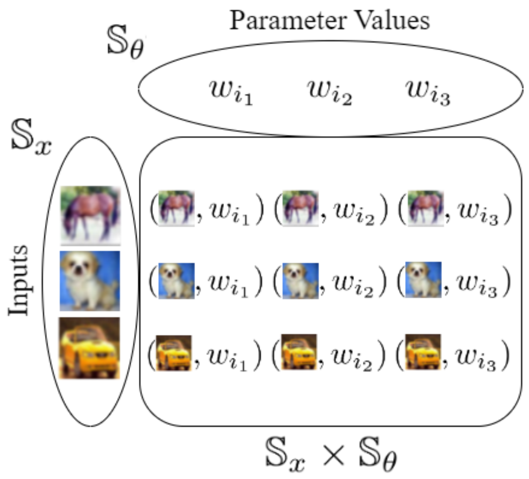
Parameter Values
$\mathbb{S}_\theta$
w_{i_1} w_{i_2} w_{i_3}
$\mathbb{S}_x$
Inputs
$, w_{i_1})$ $, w_{i_2})$ $, w_{i_3})$
$\mathbb{S}_x \times \mathbb{S}_\theta$

Fig. 2: For each neuron, we integrate over the product space of inputs and parameter values in order to collect importance information about that parameter's relation to the set of inputs. Pictured above, we visualize the product space corresponding to three samples for weight w_i and three input samples from the dataset.

3.1 Parameter Explanations using the Product Space (PEPS)

We first recall the formulation of Generalized Integrated Gradients in Equation 1 from [21] for an input x and a model F, over a set $\mathbb{S}_x$ in the input space, as well as the even more general Equation 2 for an attribution function $\mathcal{A}$ and a distribution $p_{\mathbb{S}_x}$:

$$\begin{aligned}\text{GeneralizedIntegratedGrads}(\mathbb{S}_x) &::= \frac{1}{|\mathbb{S}_x|}\int_{\mathbb{S}_x} \nabla F(x)dx \\ &= \mathbb{E}_{\mathbb{S}_x}\left[\nabla F\right]\end{aligned} \tag{1}$$

$$\begin{aligned}&\text{GeneralizedIntegratedAttribution}(\mathcal{A}, F, \mathbb{S}_x, p_{\mathbb{S}_x}) \\ &\qquad ::= \int_{\mathbb{S}_x} \mathcal{A}(F, x)p_{\mathbb{S}_x}(x)dx\end{aligned} \tag{2}$$

We apply this formulation to model attributions rather than input attributions by treating a parameter or parameter group (such as a convolutional filter) θ as we treated the input x for input attribution. This simultaneous effect of inputs and parameters yields a tuple representing a single point (x, θ) in the input $\times$ parameter product space. We can now compute the gradients of the parameter θ with respect to the input x, so we then integrate over a set of interest $\mathbb{S}_\theta$ and distribution $p_{\mathbb{S}_\theta}$ in the parameter space as well as over a set of interest $\mathbb{S}_x$ and distribution $p_{\mathbb{S}_x}$ in the input space to obtain Equations 3 and 4:

$$\begin{aligned}
&\text{GeneralizedModelIntegratedGradients}(\mathbb{S}_x, \mathbb{S}_\theta) \\
&\qquad ::= \frac{1}{|\mathbb{S}_x|}\int_{\mathbb{S}_x} \frac{1}{|\mathbb{S}_\theta|}\int_{\mathbb{S}_\theta} \nabla F_\theta\left(x\right) d\theta dx \\
&\qquad = \frac{1}{|\mathbb{S}_x||\mathbb{S}_\theta|}\int_{\mathbb{S}_x}\int_{\mathbb{S}_\theta} \nabla F_\theta\left(x\right) d\theta dx \\
&\qquad = \mathbb{E}_{\mathbb{S}_x}\left[\mathbb{E}_{\mathbb{S}_\theta}\left[\nabla F_\theta\left(x\right)\right]\right] \\
&\qquad = \mathbb{E}_{\mathbb{S}_x \times \mathbb{S}_\theta}\left[\nabla F_\theta\left(x\right)\right]
\end{aligned} \tag{3}$$

$$\begin{aligned}
&\text{GeneralizedIntegratedModelAttribution}(\mathcal{A}, \mathbb{S}_x, p_{\mathbb{S}_x}, \mathbb{S}_\theta, p_{\mathbb{S}_\theta}) \\
&\qquad ::= \int_{\mathbb{S}_x}\left[\int_{\mathbb{S}_\theta} \mathcal{A}(\theta, x) p_{\mathbb{S}_\theta}(\theta) d\theta\right] p_{\mathbb{S}_x}(x) dx \\
&\qquad = \int_{\phi \in \mathbb{S}_x \times \mathbb{S}_\theta} \mathcal{A}(\phi) p_{\mathbb{S}_x \times \mathbb{S}_\theta}(\phi) d\phi
\end{aligned} \tag{4}$$

Using this method, we can compute model attributions corresponding to each of the input attribution statistics proposed in [21]: Expected/Integrated Gradients, Gradient Variance, Stability, and Consistency. We compute these attributions for each parameter (weights, biases, etc.), and just as individual pixel attributions can be aggregated to obtain an attribution for an entire image, we can also aggregate parameter attributions as desired to obtain coarser attributions for parameter groups, modules, layers, or the entire model (Equations 5 6).

$$\text{Attribution(Parameter Group)} = \underset{\theta \in \text{Parameter Group}}{\mathbb{E}}\left[\text{Attributions}(\theta)\right] \tag{5}$$

$$\text{Attribution(Model)} = \underset{\text{Parameter Group} \in \text{Model}}{\mathbb{E}}\left[\text{Attribution(Parameter Group)}\right] \tag{6}$$

Note that this aggregation step is sensitive, as positive and negative measure values can potentially cancel out and obfuscate the true effect at the filter and model-level. For more accurate and thorough attributions, alternative aggregation methods can be applied to combine hierarchically nested attributions, or individual filters and even the entire model might be treated as a single entity for computation of each integrated measure, though this approach may require additional overhead or special considerations with respect to sampling the latent parameter space.

4 Evaluation

We perform two types of quantitative analysis to demonstrate the utility of our method. We first perform pruning experiments to verify that our novel attribution measures are able to identify important and unimportant neurons in a trained model. We then collect additional distributional information for a wider variety of datasets to confirm that our attributions can distinguish between a trained model and an untrained model. For each integrated attribution measure, we select our set $\mathbb{S}_\theta$ to be the ball centered at a parameter of locality radius hyperparameter ε proposed in [21]. We choose our ε values from a large range in our experiments in an attempt to sample both local and nonlocal gradient behavior. We also fix $\mathbb{S}_x$ to be the training set, and follow the same Monte Carlo integration method of [21] in which we sample points from $\mathbb{S}_x$ and $\mathbb{S}_\theta$.

4.1 Pruning

We demonstrate the effectiveness of our method in determining neuron importance by performing a series of pruning experiments (Figures 3, 4, and 6) as was the approach of [9,1], in which we set a proportion of individual model parameters to zero. If model performance degrades faster or slower when pruning according to ranked attribution values compared to pruning randomly, then we will have successfully identified the important or unimportant neurons respectively. In each experiment, we prune the same proportion of weight and bias parameters from each layer, except for the final output layer which we leave intact.

Pruning Experiments We perform experiments for the CIFAR-10 dataset [11], shown in Figure 3, using a ResNet-18 architecture trained for 20 epochs using a standard categorical cross-entropy with learning rate of 0.01 with no momentum or weight decay, and a batch size of 32. We also perform similar experiments for the ImageNet dataset [5], shown in Figure 4, using a pretrained ResNet-34 architecture. For all pruning experiments, we show the mean F1-Score on the respective test set over 10 replicates and the associated 95% confidence interval. We investigate the effect of our locality radius ε for 32 input sample points and 32 parameter sample points. We also include an investigation of the effect of the number of sample points in the input space and parameter space for in our supplemental information, and we briefly show the benefit of larger sample sizes in Figure 5. We can notice from Figures 3 and 4 that our Integrated Gradients and Gradient Variance measures are able to successfully identify important neurons for all three of the tested locality radii, but that our Stability measure identifies important neurons for a large radius and unimportant neurons for a small radius. We note the our Consistency measure currently only appears useful for large radii, so this may indicate that we need to explore a wider range of radii and sample points in future studies. We also assume that first multiplying our attributions by -1 will result in the opposite behavior as observed in Figures 3 and 4, but we must confirm this in future studies.

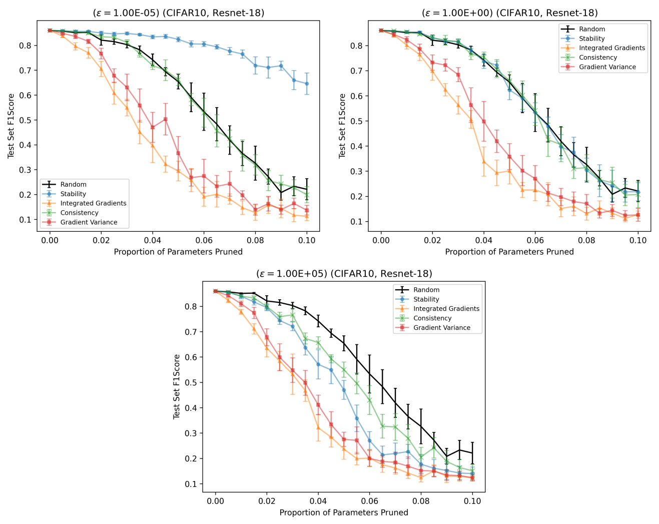

Fig. 3: Pruning of a ResNet-18 model trained on CIFAR-10. We can observe that different choices of locality radius ε reveal different types of attribution information for certain measures. For each of the four measures, we observe statistically significant differences from random pruning, verifying that we have collected information relevant to the model's performance.

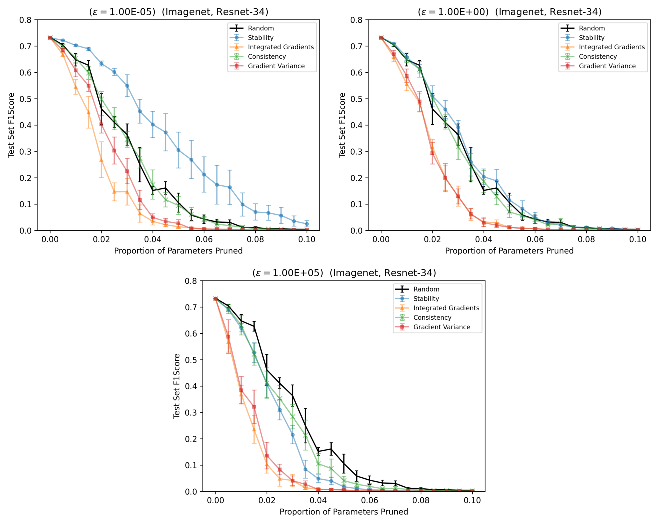

Fig. 4: Pruning of a ResNet-34 model trained on Imagenet. Note the same overall trends for each measure as observed for the CIFAR-10 dataset in Figure 3, indicating that the utility of our measures successfully generalizes to larger datasets.

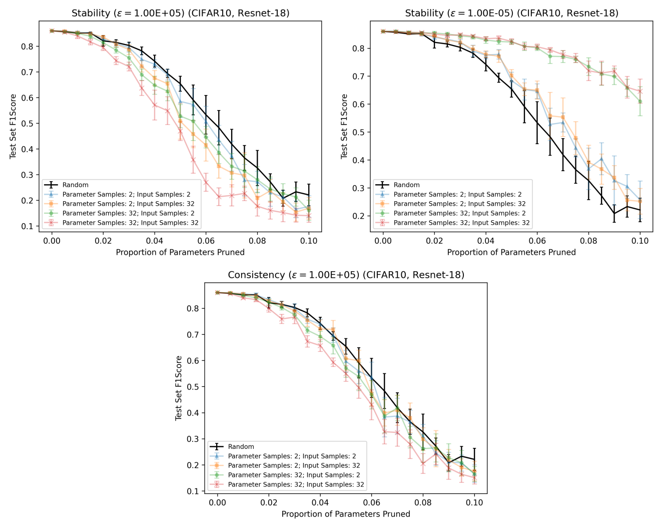

Fig. 5: Pruning of a ResNet-18 model trained on CIFAR-10 for varying input and parameter sample sizes. In general we observe that increased sample size results in more accurate attributions reflecting a neurons relative importance or unimportance.

Pruning to Target a Single Class We also demonstrate the ability of our method to select for neurons important to a specific class. By fixing $\mathbb{S}_x$ as a specific subset of the input space (i.e. a specific class), we can determine a neuron's importance with respect to that class. We otherwise perform these experiments using the same methodology as Figures 3 and 4. Shown in Figure 6 are the two experiments which demonstrate the most selectivity for a single class while preserving performance on the remaining classes, the remaining experiments are available in supplemental information. Our results show resounding success in destroying the model's test set F1-Score for the targeted class, which may prove to be useful for applications related to unlearning [4].

4.2 Distributional Analysis

We investigate the distribution of our attribution measures for trained and randomly initialized models using several small-scale image classification datasets: CIFAR-10 [11], MNIST [12], FashionMNIST [23], and SVHN [15], using a ResNet-18 architecture trained using the same methodology as for the pruning experiments, except for that on the smaller datasets (MNIST, FashionMNIST, and SVHN) we trained only for 10 epochs. We also include data for partial training on the CIFAR-10 dataset (10 out of 20 epochs) to further illustrate how the distributions converge as the model trains. For these distribution analyses we use 10 sample points from the input space and 20 sample points from the parameter

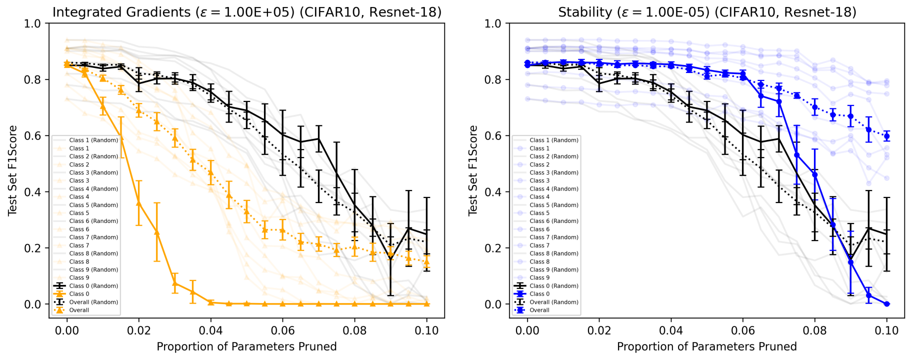

Fig. 6: Single-class targeted pruning of a ResNet-18 model trained on CIFAR-10. Note that we are able to reduce the F1-Score for the targeted class to zero while maintaining $\sim .60$ F1-Score on the overall dataset. This indicates that our measures could be used to very effectively unlearn specific classes as needed for privacy or security applications.

space, and we investigate the distributions of measures corresponding to $\varepsilon = 1.0$ and $\varepsilon = 10^4$.

We demonstrate below that the distributions of each of our four integrated model attribution measures are highly dependent on model training status. See Figures 7, 8, 9, and 10 for histograms and discussion of Integrated Gradients, Gradient Variance, Stability, and Consistency. We can observe distributions resembling several parametric families in the generated histograms, namely the Student's t-distribution for Integrated Gradients, the Gamma distribution for Gradient Variance, and possibly Beta distributions for Stability and Consistency. While we do not in this study fit any parametric distributions to the data, these distinct distributions offer a quantitative and parametric source of model explanation and evaluation.

5 Conclusions

We have developed a novel methodology for explaining model parameters, and have verified its ability to identify important and unimportant neurons. By considering the product space of inputs and parameter values, we are now able to generalize the family of integrated attribution measures proposed in [21] from only input and intermediate feature attribution to also accommodate parameter attribution. We have identified several unique sources of parameter attribution information by using our new formulation to compute the four measures proposed in [21] for varying locality radius. Since the underlying family of attributions is very diverse, our initial success in identifying important neurons justifies a much more thorough investigation into the various relevant hyperparameters (input sample size, parameter sample size, locality radius, etc.) and merits the

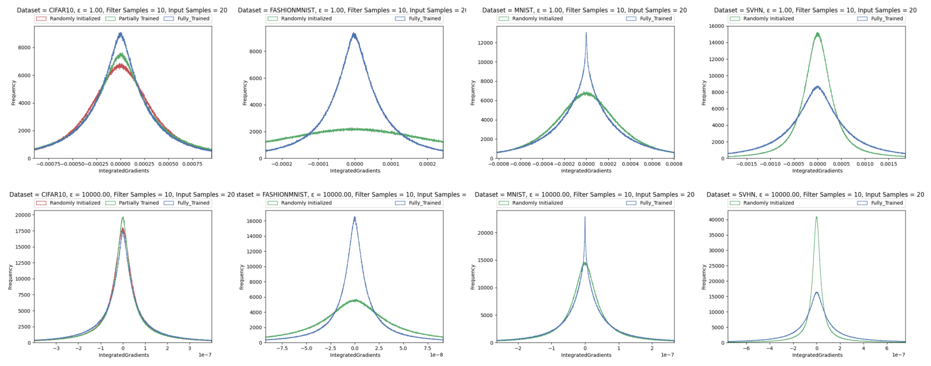

Fig. 7: Parameter-level histograms for the Integrated Gradients measure for models trained on CIFAR-10, FashionMNIST, MNIST, and SVHN datasets. Values outside the .9 quantile are excluded from the histogram range. The attributions appear to be roughly distributed according to a Student's t-distribution for both choices of locality radius ε, which is consistent with the fact that Integrated Gradients is computed as a sample mean of gradients in the Monte Carlo integral assuming gradients in the underlying latent space are normally distributed.

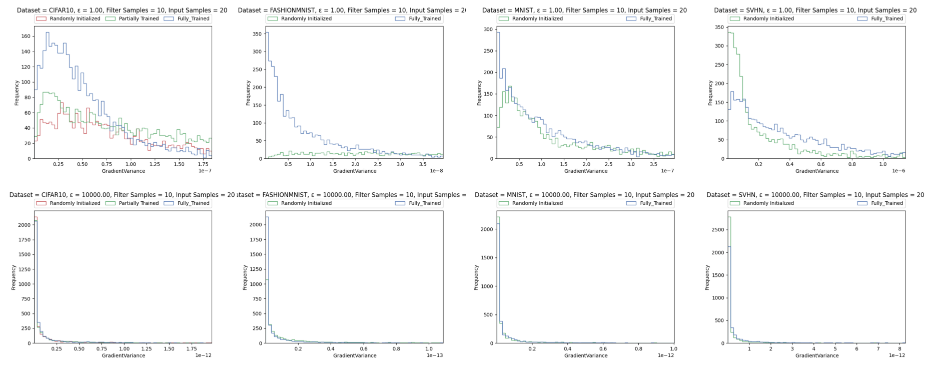

Fig. 8: Filter-level histograms for the Gradient Variance measure for models trained on CIFAR-10, FashionMNIST, MNIST, and SVHN datasets. For these histograms we present filter-level attributions rather than parameter-level attributions due to the large volume of extreme values at the parameter-level. Values outside the .9 quantile are excluded from the histogram range, but even though we are plotting the filter-level distribution, due to the large number of extreme values this range restriction still results in distributions which are difficult to qualitatively evaluate for locality radius $\varepsilon = 10^4$. The attributions appear to be distributed according to a gamma distribution for both choices of locality radius ε, which is consistent with the fact that filter attributions are sums of per-parameter Gradient Variances, which each follow a chi-square distribution assuming that gradients in the underlying latent space are normally distributed.

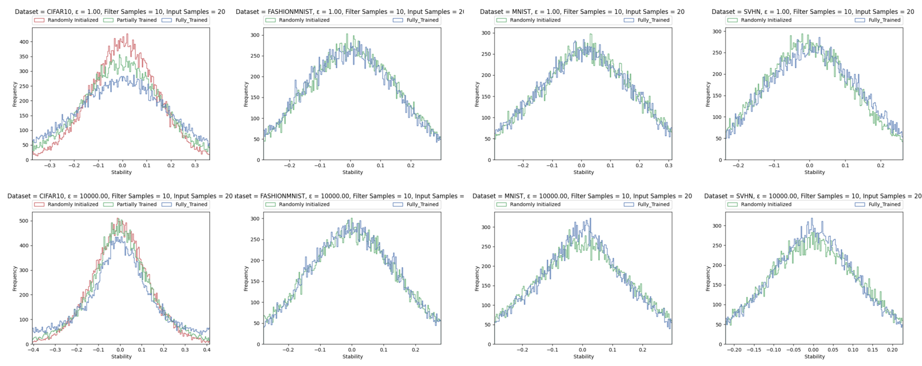

Fig. 9: Parameter-level histograms for the Stability measure for models trained on CIFAR-10, FashionMNIST, MNIST, and SVHN datasets. Values outside the .9 quantile are excluded from the histogram range. The attributions appear to be similarly distributed for both choices of locality radius ε, and while there does appear to be some dependence on model training for the CIFAR-10 ($\varepsilon = 1.0$) results, any additional trends are not immediately qualitatively clear. Since the Stability measure is defined on a bounded support of $[-1, 1]$ as the expectation of a cosine, we should strongly consider distributions such as the beta distribution for development of future statistical tests and inferences based on Stability.

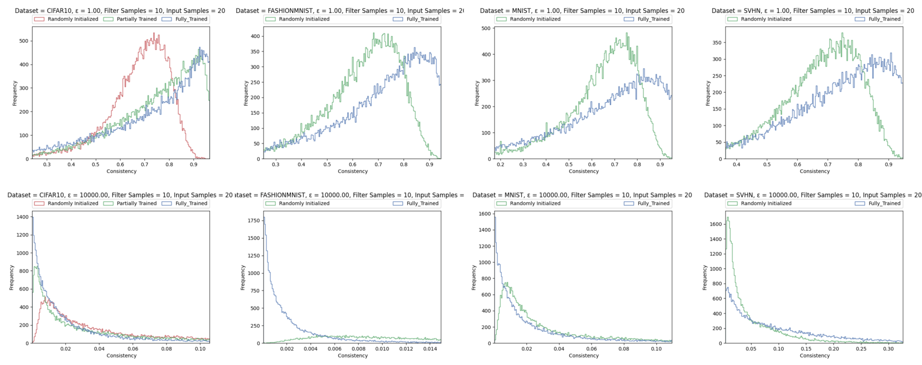

Fig. 10: Parameter-level histograms for the Consistency measure for models trained on CIFAR-10, FashionMNIST, MNIST, and SVHN datasets. Values outside the .9 quantile are excluded from the histogram range. We can note the two distinct paradigms for locality radius $\varepsilon = 1.0$ and $\varepsilon = 10^4$, and a clear distinction between the distributions for trained and untrained models in both cases. Like the Stability measure, since the Consistency measure is defined on a bounded support of $[-1, 1]$, we should be mindful of this when developing any statistical tests.

development of additional measures beyond the four which we have explored in this work. Furthermore, we have demonstrated our method's ability to identify neurons important specifically for single classes. If our parameter attributions are incorporated into more sophisticated unlearning and model reduction methods, we will likely observe even better utility. We have also identified several preliminary trends and patterns with respect to the distributions of Integrated Gradients, Gradient Variance, Stability, and Consistency for trained versus untrained models. Our study confirms that this family of attribution measures is a rich source of relevant model information which begs further study toward the end of both explaining and improving model behavior. Since our methodology is immediately applicable to any machine learning model for which parameter gradients can be computed, and can accommodate any new measures developed using the framework of Generalized Integrated Attributions, we should expect the utility of this method to only increase as additional novel and useful integrated measures continue to be proposed.

5.1 Future Work

While we include much additional pruning data in the supplemental information, we should still explore a wider range of locality radii and perform experiments in which we prune in ascending order of attribution value as opposed to descending value, or in which we sort each parameter using multiple attribution measures at once. Additionally, While we have collected data for the Resnet-18 and Resnet-34 architectures for several image datasets, we can also investigate a wider variety of model architectures and data tasks beyond image recognition.

In the future we may be able to visualize the semantic effect and role of the model's important parameters by inspecting the types of features extracted by these parameters. This, coupled with input and feature attribution, may give us a broader understanding of model attention.

The qualitative trends observed in the distributional study above justify a more rigorous statistical analysis such as Analysis of Variance (ANOVA) in the future to search for higher order and mixed effects. The distributions of each attribution measure can also be fit to parametric distributions such as Student's t, gamma, and beta distributions in order to directly quantify the effect of model training and other hyperparameters. In particular, since we directly derived that Integrated Gradients and Gradient Variance measures should follow Student's t and gamma distributions respectively, we can immediately begin developing statistical tests to explain and improve models based on these two measures. We can additionally continue to collect more data regarding how attributions depend on model training and accuracy. We can similarly continue to investigate how hyperparameters such as the locality radius ε, the training dataset, number of classes, and filter size affect the distribution of attributions. We might also inspect class-wise attribution distributions as a means of further quantifying and explaining how each model responds to a particular class.

Finally, though out-of-scope for this work, future studies should pursue training models using the method of attribution priors proposed by [7]. It is possible that the attribution measures studied in this work could be used to train models more robustly and accurately, so any such opportunities for explainable improvement should be investigated.

Acknowledgements This material is based upon work supported by the National Science Foundation under Grant No. 2134237. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Alqahtani, A., Xie, X., Essa, E., Jones, M.W.: Neuron-based network pruning based on majority voting. 2020 25th International Conference on Pattern Recognition (ICPR) pp. 3090–3097 (2021), https://api.semanticscholar.org/CorpusID:233877899
2. Ancona, M., Ceolini, E., Öztireli, C., Gross, M.: Towards better understanding of gradient-based attribution methods for deep neural networks. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=Sy21R9JAW
3. Barkan, O., Elisha, Asher, Y., Eshel, A., Koenigstein, N.: Visual explanations via iterated integrated attributions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2073–2084 (October 2023)
4. Chundawat, V.S., Tarun, A.K., Mandal, M., Kankanhalli, M.: Zero-shot machine unlearning. Trans. Info. For. Sec. **18**, 2345–2354 (jan 2023). https://doi.org/10.1109/TIFS.2023.3265506, https://doi.org/10.1109/TIFS.2023.3265506
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
6. Dhamdhere, K., Sundararajan, M., Yan, Q.: How important is a neuron. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=SylKoo0cKm
7. Erion, G., Janizek, J., Sturmfels, P., Lundberg, S., Lee, S.I.: Improving performance of deep learning models with axiomatic attribution priors and expected gradients. Nature Machine Intelligence **3**, 1–12 (07 2021). https://doi.org/10.1038/s42256-021-00343-w
8. Fel, T., Picard, A., Béthune, L., Boissin, T., Vigouroux, D., Colin, J., Cadène, R., Serre, T.: Craft: Concept recursive activation factorization for explainability. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2711–2721 (June 2023)
9. Ghorbani, A., Zou, J.Y.: Neuron shapley: Discovering the responsible neurons. ArXiv **abs/2002.09815** (2020), https://api.semanticscholar.org/CorpusID:211258568
10. Hase, P., Xie, H., Bansal, M.: The out-of-distribution problem in explainability and search methods for feature importance explanations. Advances in Neural Information Processing Systems **34** (2021)
11. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009)
12. LeCun, Y., Cortes, C., Burges, C.: Mnist handwritten digit database. ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist **2** (2010)

13. Leino, K., Li, L., Sen, S., Datta, A., Fredrikson, M.: Influence-directed explanations for deep convolutional networks. 2018 IEEE International Test Conference (ITC) pp. 1–8 (2018)
14. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 4768–4777. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
15. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning (2011)
16. Sajjad, H., Durrani, N., Dalvi, F.: Neuron-level interpretation of deep nlp models: A survey. Transactions of the Association for Computational Linguistics **10**, 1285–1303 (2021), https://api.semanticscholar.org/CorpusID:237353268
17. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70. p. 3145–3153. ICML'17, JMLR.org (2017)
18. Shrikumar, A., Su, J., Kundaje, A.: Computationally efficient measures of internal neuron importance. CoRR **abs/1807.09946** (2018), http://arxiv.org/abs/1807.09946
19. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. In: Proceedings of the International Conference on Learning Representations (ICLR). ICLR (2014)
20. Srinivas, S., Fleuret, F.: Full-gradient representation for neural network visualization. Advances in neural information processing systems **32** (2019)
21. anonymous submission: Generalized integrated gradients. In: under review. p. supplemental materials (2023)
22. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International conference on machine learning. pp. 3319–3328. PMLR (2017)
23. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017)
24. Zhang, J., Bargal, S.A., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-down neural attention by excitation backprop. International Journal of Computer Vision **126**(10), 1084–1102 (2018)