On the Relationship Between Explanation and Prediction: A Causal View

Amir-Hossein Karimi
MPI for Intelligent Systems
& ETH Zurich

Krikamol Muandet
CISPA-Helmholtz Center for
Information Security

Simon Kornblith
Google Research

Bernhard Schölkopf
MPI for Intelligent Systems

Been Kim
Google Research

Abstract

Being able to provide explanations for a model's decision has become a central requirement for the development, deployment, and adoption of machine learning models. However, we are yet to understand what explanation methods can and cannot do. How do upstream factors such as data, model prediction, hyperparameters, and random initialization influence downstream explanations? While previous work raised concerns that explanations (E) may have little relationship with the prediction (Y), there is a lack of conclusive study to quantify this relationship. Our work borrows tools from causal inference to systematically assay this relationship. More specifically, we study the relationship between E and Y by measuring the treatment effect when intervening on their causal ancestors, i.e., on hyperparameters and inputs used to generate saliency-based Es or Ys. Our results suggest that the relationships between E and Y is far from ideal. In fact, the gap between 'ideal' case only increase in higher-performing models–models that are likely to be deployed. Our work is a promising first step towards providing a quantitative measure of the relationship between E and Y, which could also inform the future development of methods for E with a quantitative metric.

1 Introduction and Related Work

Being able to provide explanations for a machine learning (ML) model's decision has become central to the development, deployment, and adoption of ML models. Explanations are important not only to help practitioners better understand the model's underlying rationale to debug models (Adebayo et al., 2022; Rieger et al., 2020) and to influence the model's decision (Koh et al., 2020; Bau et al., 2020; Meng et al., 2022), but also to ensure that models comply with regulatory requirements (Parliament & of the European Union, 2016). However, Existing tools for interpretability have however elicited criticisms, often highlighting computational or qualitative user-study-based evidence that explanations generated from these tools may contain critical errors and must be used with care (Poursabzi-Sangdeh et al., 2018; Chu et al., 2020; Adebayo et al., 2018; Alqaraawi et al., 2020; Srinivas & Fleuret, 2021; Kindermans et al., 2019).

One focal point in many investigations is the relationship between explanations (E) and predictions (Y). In this work, we seek to formalize this relationship, inspired by the common cause principle of Reichenbach (1956) that states that if two variables are *statistically* dependent, there must be a common *cause* influencing both of them, and this common cause can be chosen such that it explains all the dependence. We develop a measure of dependence via the Potential Outcomes framework (Rubin, 2005). Viewed through a lens of causality, we evaluate the treatment effect

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

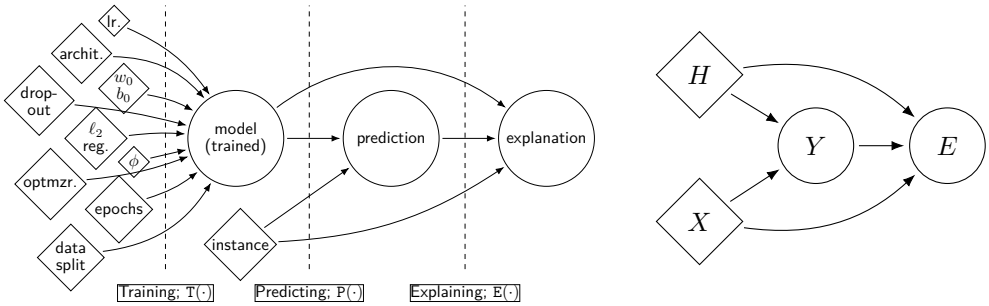

Figure 1: Explanation generating process involve three stages: training, predicting, and explaining (left). Intervening on factors (H, X) allow for studying their treatment effect (i.e., causal influence) on down-stream targets (i.e., Y, E) (right).

of hyperparameters of the model, H (i.e., H taking on value h', the counterfactual antecedent) on E and Y conditioned on a particular instance x. In other words, by measuring the treatment effect of each hyperparameter (e.g., choice of activation, initialization, training budget), we are measuring its influence on E and Y, and in particular, how the influence is *different or similar* in E and Y (Fig. 1; left). Furthermore, under a careful evaluation, we tease apart the direct influence of H on E vs. its indirect influence mediated through Y to better understand the flow of causation (Fig. 1; right).

Our study reveals a surprising relationship between E and Y (precisely, measured by how a causal ancestor of the two influences them). In particular, for top-performing models, the influence on E from Y *decreases* compared to relatively lower-performing models. For some methods, a causal ancestor of both Y and E directly influences E much more than Y, leaving Y's influence on E minimal, even though this ancestor, i.e, hyperparameter, should not inform the explanation of the model in any way. This finding was consistent across 30k pre-trained models with different hyperparameters across different datasets. Our work informs practitioners on what different explanation methods can and cannot be used for: if one's goal is to find E that is related to the prediction, Y, methods with little relationship between E and Y under our framework aren't the best choices. Our framework can also be used to drive the development of new methods by providing a quantitative metric.

2 Methodology

To understand the relationship between E and Y via H's impact on them, we perform an exploratory analysis on a class of ML models and then analyze their causal effects on the downstream E and Y.

2.1 Explanation Generating Process

At a high level, the *explanation generating process* (EGP) shown in Figure 1 describes a mechanical system that is engineered to train an ML model given an initial set of hyperparameters, h, which yields a prediction $\hat{y}_h(x)$ and an explanation $\hat{e}_h(x)$ given a test instance x. Formally, a supervised ML model is obtained through a *training procedure* $\mathtt{T} : \mathcal{H} \times \mathcal{D} \to \mathcal{F}$ given a set of training hyperparameters and a dataset $\mathcal{D} := (\mathcal{X}, \mathcal{Y})$. The training procedure typically contains initialization, optimization, and regularization. Once trained, the model can predict the target of a given test instance x via a *prediction procedure* $\mathtt{P} : \mathcal{F} \times \mathcal{X} \to \mathcal{Y}$. Finally, local explanations e are the result of an *explanation procedure* $\mathtt{E} : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to \mathcal{E}$ applied to a tuple of a trained model, test instance, and predicted target, $\hat{y}_h(x)$. Note the absence of noise variables; under a fixed random seed, the procedures above are deterministic.

Although these procedures may not be expressible in closed-form, e.g., one may not conclusively infer the trained weights of a neural network by only looking at the hyperparameters, each procedure is executable on a computer, e.g., the model weights can be obtained by training procedure under a training setting and given budget.

2.2 Potential Outcomes Framework

To study the causal effects of hyperparameters, we adopt the Potential Outcomes (PO) framework (Rubin, 2005). Given the temporal precedence of hyperparameters over the trained model parameters and in turn over the prediction and explanation, one may alternatively view the mechanical system in Figure 1a as the causal system shown in Figure 1b (with graphical and structural components). In this framing, the *causal influence* of up-stream factors (e.g., H, X) on down-stream targets (e.g., Y, E) can be measured as the *treatment effect* of a factor (e.g., treatment $H = h$ vs. control $H = h'$), on the down-stream target.

In what follows, we will refer to $Y_h^*(x)$ and $E_h^*(x)$ as *potential* prediction and explanation on an instance x when the model is trained with the hyperparameter h. For any pair $h, h' \in \mathcal{H}$, the individual treatment effect (ITE), which quantifies the treatment effect of assigning two different parameters, can be defined as

$$\text{ITE}_Y(x) = Y_h^*(x) - Y_{h'}^*(x). \tag{1}$$

Similarly defined, the treatment effect for explanation is denoted as ITE_E. In principle, it is possible to realize $Y_h^*(x)$ and $E_h^*(x)$ for all $h \in \mathcal{H}$ given unlimited computational resources. As a result, one can evaluate $\text{ITE}(x)$ in practice by contrasting the predictions of models trained on hyperparameters h and h'. However, when this process becomes computationally prohibitive, we might face the so-called *fundamental problem of causal inference*, i.e., for each $x \in \mathcal{X}$, we can only observe $Y_h^*(x)$ and $E_h^*(x)$ for a small number of hyperparameters h, but not the other $h' \neq h$. Furthermore, we may not be able to interpret the observed differences between Y and E that arise from two different H as a causal effect unless the assumption of *ceteris paribus*, i.e., all else being equal, is fulfilled. Retraining almost identical neural networks with all possible values of hyperparameters is however computationally prohibitive. Instead, we perform an observational study on a model zoo, a large collection of pre-trained models (Unterthiner et al., 2020; Jiang et al., 2019), to study the relationship between E and Y.

Since our research question seeks to investigate the impact of *multiple, potentially-non-binary* treatments (e.g., set of numerical and categorical H) on the target prediction/explanation (see Figure 1a), we amend the treatment definitions above as follows:

$$\underbrace{Y_{h=1}^*(x) - Y_{h=0}^*(x)}_{\substack{\text{effect of } h=1 \text{ w.r.t } h=0 \text{ on } x \in X \\ \text{(single binary treatment)}}} \tag{2}$$

$$\underbrace{\mathbb{E}_{m \neq n}\left[Y_{h=n}^*(x) - Y_{h=m}^*(x)\right]}_{\substack{\text{effect of } h=n \text{ w.r.t } h \neq n \text{ on } x \in X \\ \text{(single non-binary treatment)}}} \tag{3}$$

$$\underbrace{\mathbb{E}_{h_{\setminus i}}\left[\mathbb{E}_{m \neq n}\left[Y_{[h_i=n, h_{\setminus i}]}^*(x) - Y_{[h_i=m, h_{\setminus i}]}^*(x)\right]\right]}_{\substack{\text{effect of } h_i=n \text{ w.r.t } h_i \neq n \text{ on } x \in X \\ \text{(multiple non-binary treatments)}}} \tag{4}$$

which allows for answering queries of the form "*what is the treatment effect of optimizer choice ν_1 as opposed to ν_2 on the local prediction of x*?". Were the optimizer choice, ν, to be the only hyperparameter in the system, this query would be answered by (3). In the setting of Figure 1a, however, (4) is employed to also marginalize out the effect of other Hs. Although these expressions average over multiple set of Hs, they all refer to the prediction of the same individual (ITE); extensions to CATE and ATE, aggregated over $x \sim \mathcal{X}$, follow naturally.

3 Analysis and Results

This section provides details of our analysis and results of our observational study in both global setting (all models) and local setting (models in each performance buckets).

3.1 Details of Observational Study

Model zoo dataset and pre-processing explanations The dataset provided by Unterthiner et al. (2020) contains $30{,}000$ 3-layer CNNs ($4{,}970$ parameters; weights and biases) that were trained until convergence (or a maximum of 86 epochs) for multiple datasets. The hyperparameters are drawn "independently at random" from pre-specified ranges. Both the ranges and the training procedure are natural and resemble standard practice in machine learning, and the models are trained on commonly used CIFAR10, SVHN, MNIST, and FASHION MNIST datasets. The

Figure 2: Comparison of ITE values of $h_{\text{optimizer}}$ on Y (left) and E (right) for models across different performance buckets, showing the discrepancy in the effect of H on Y vs. that on E (top: CIFAR10; bottom: SVHN). Interestingly, there is a difference of ITE_E across accuracy buckets, and more importantly, none of the explainability methods resemble ITE_Y.

random seed (for mini-batch GD sampling and for weight initialization) and the architecture of the base models are fixed throughout. The diversity of hyperparameters allows for a representative study of treatment effects (details in Appendix A.3; code).

We study four commonly deployed saliency methods: *gradient* (Simonyan et al., 2013; Erhan et al., 2009; Baehrens et al., 2009), *SmoothGrad* (Smilkov et al., 2017), *Integrated Gradients* (IG) (Sundararajan et al., 2017), and *Grad-CAM* (Selvaraju et al., 2016). Note that many widely used methods are built based on these four methods Xu et al. (2020); Wang et al. (2021); Simonyan et al. (2013). The generated explanation maps are preprocessed as in Adebayo et al. (2018) (see Appendix A.3). Since some methods only produce positive attributions, we zero out any negative attributions for the methods that produce both positive and negative values; this is so that we can compare all methods on an equal footing. Finally, to measure the *goodness* of treatment effect values, we introduce and evaluate a reference explanation method, namely *Identity*, whereby E is set to be identical to Y. This is not a useful explanation for humans, but our goal is to create an ideal E that provides a point of comparison for our results.

3.2 Results

H influences Y (and E) differently across performance buckets: The relationship between E and Y when Y is from an untrained model v.s. a trained model should be qualitatively different. Teasing out how much Y influences E is one of the long-standing questions in interpretability; some have argued that E is visually indistinguishable when Y is from trained or untrained models Adebayo et al. (2018). How the relationship between E and Y changes as a function of the performance of the model is important for practitioners in deciding when E can or cannot be used. Thus, we conduct the remaining analysis by stratifying models into different accuracy buckets. In particular, we stratified the $30{,}000$ models into 8 buckets according to their accuracies to observe the treatment effect in each group (Figure 2). We use 0-20^{th}, 20-40^{th}, 40-60^{th}, 60-80^{th} and 80-90^{th}, 90-95^{th}, 95-99^{th} and 99-100^{th} percentiles as groups for all four datasets (finer granularity for top models that are more likely to be deployed; summarized in Table 2).

The control group: Calculating ITE for each performance bucket requires a decision on control groups, i.e., the point of comparison. There are two natural choices 1) select a control group within each accuracy bucket or 2) use the same control group across all buckets. Each choice means we are answering slightly different questions; (1) answers "the effect of $h_i = n$ w.r.t. $h_i \neq n$ on $x \in X$ such that training on $h_i \neq n$ gives a similarly performing model" while (2) answers "the effect of $h_i = n$ w.r.t $h_i \neq n$ on $x \in X$ such that training on $h_i \neq n$ gives a model with baseline performance". Although the latter enables comparison of performance buckets on similar footing, two factors are changing simultaneously: a) $h_i = n$ to $h_i \neq n$ and b) the change

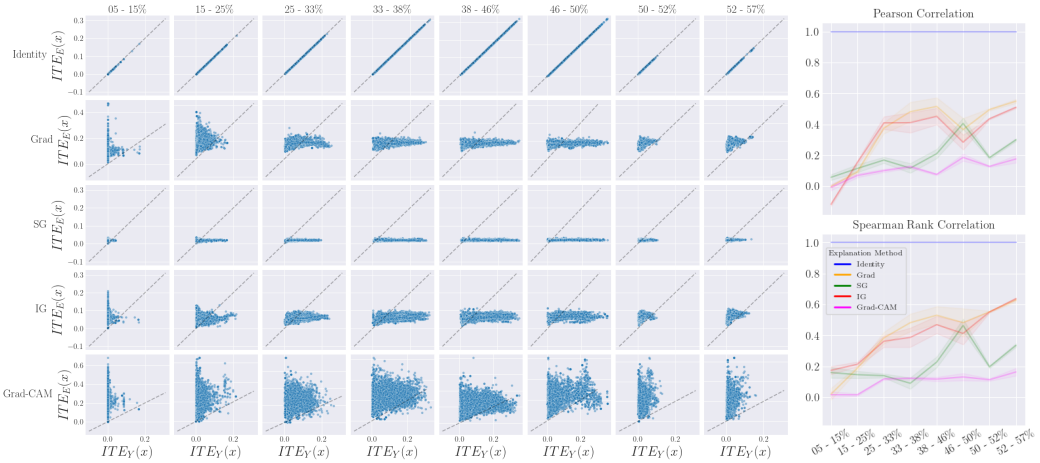

Figure 3: (left) Each column represents accuracy subsets, and each row denotes a distinct explanation method. For low-performing CIFAR10 models (first column), prediction changes are minimal as explanations vary, but high-performing models display the opposite trend. (right) Correlations from the left plots reveal reduced alignment in the top 1% models.

in performance bucket, making it difficult to tease apart hyperparameters' contributions to the ITE values. Therefore, we continue with within-accuracy-bucket control groups, and refrain from comparing absolute values of ITE (for Y or E) across buckets, but instead, look to *relative* ITE values of H on Y and E across buckets.

As seen in Figure 2, while both ITE_Ys (first column) and ITE_Es (the remainder of columns) vary across accuracy buckets, they appear not to follow the same pattern. This raises an important question: *how does the relationship between Y and E (measured by treatment effect of H on both) change as models' performance changes?*

Understanding the (odd) relationship between ITE_Y and ITE_E: We first investigate the extent of the relationship between ITE_Y and ITE_E by measuring their relative changes, before separating the direct influence of H on E from the indirect influence mediated through Y.

One way to compare ITE_Y and ITE_E is using scatterplots. Figure 3 (left) shows scatterplots for different performance buckets and explanation methods. Since the absolute value of each ITE is not directly comparable (due to different domains for Y and E, and different baseline control groups, as explained above), we summarize the scatter plot trends by measuring the Pearson and Spearman Rank correlations between the raw ITE values (Figure 3; right).

We observe that compared to the case of the Identity method,[1] whereby there is a perfect correlation between ITE_Y and ITE_E (the diagonal $x = y$ line), no other method seems to remotely follow a similar pattern. For most of the methods, the range of ITE_E values varies similarly regardless of low/mid/high accuracy models, while ITE_Y naturally shrinks in high accuracy models, which can be explained by the models becoming similar in their predictions. The correlation coefficient tells a similar, but more concise, story. While the correlation increases for Grad and IG in the higher accuracy bucket, both show only moderate correlation compared to the reference point (Identity). It is also unclear how the relationship between E and Y is similar in mid-accuracy (e.g., 33%) and top-accuracy models. The pattern described above is shared across all types of hyperparameters across four datasets (see Figure 15 and Figure 16).

To summarize, the $\text{corr}(\text{ITE}_Y, \text{ITE}_E)$ increases as the model accuracy increases, suggesting that E (for Grad and IG) becomes a better reflection of Y in higher-performing models,[2] which is desired. Despite this, the correlation values are substantially lower than a maximally informative explanation (i.e., the Identity method) suggests that *explanations may still be explaining something other than the prediction*.

[1] We remind that while the Identity explanation is not useful for humans in any way, it helps us to understand what a "good" explanation (where Y is a major factor in deciding E) may look like through the lens of the proposed ITE analysis.

[2] At least in the manner in which *changes in* E reflect *changes in* Y as a result of *changes in* H

Figure 4: Pearson correlation between ITE_Y and ITE_E in total vs. direct effect (first column). The second column highlights the difference between total and direct, with higher values signifying greater influence of H on E through Y (optimal). The third column shows the delta correlation deviation from the ideal (Identity) for each method, reflecting divergence from optimal as model performance improves.

Direct vs. indirect influences: To understand how much of the explanation is reflecting the prediction, we can tease apart the effect of H on E that flows *directly* vs. *indirectly* through the prediction Y.[3] Intuitively, if explanations were only sensitive to Y, one would observe a *low direct effect* and a *high indirect effect*. Conversely, a *high direct effect* of H on E hints at the sensitivity of explanations to *factors* not related to the prediction. Unlike all ITE_E values we discussed so far that measures the *total effect* of H on E (arising both directly and indirectly through Y), we "sever" the influence that H has on Y while retaining its effect on E. We compare H's treatment effects on E when Y is and is not randomly permuted.

In the first column in Figure 4, we first observe that none of explanations seem to follow the 'ideal case' (Identity, E is maximally informative of Y). The second column simply plots the difference between total and direct effects by subtracting direct effect from total effect (dotted line $-$ solid line in the first column). This quantity roughly corresponds to the effect of H on E mediated through Y (ideally, this value should be high in higher-performing buckets).

What is even more concerning is *how much* the difference between ideal case v.s., actual case *worsens* in higher performing models. The third column plots this value: the difference between the ideal case (blue dotted line in the second column) and others. In other words, the higher a model performs, the more information for E comes from something *other than* Y. This is particularly concerning because these are models that are more likely to be deployed. For the case of SG and Grad-CAM, the influence of H on E mostly comes from H, not from the trained model or the prediction from it Y. Putting it together, our comparison of direct and indirect influence reveals that the pattern of how Y mediates the total influence of H on E is surprising and undesirable at times.

4 Discussion and Conclusions

Our work investigates the relationship between E and Y using tools from causal inference. In analyzing the treatment effect of a causal ancestor (i.e., H, determined prior to model training) of E and Y on them, the patterns observed for the direct and indirect influence reveals an undesirably high direct influence of H on E relative to influce of Y on E. Our results suggest that the relationships between E and Y is far from ideal. In fact, the gap between 'ideal' case only increases in higher-performing models–models that are likely to be deployed. This means that there are *other* factors that influence E more than the prediction of the model, Y, and their influence becomes bigger and bigger as a model performs better. If the users' goal is to understand the model's prediction, then most of the influence of H on E should be through Y (note that *which* H should not influence E is a decision by a user). The goal of our work is to first show that such influence exists in current models and present methods to perform quantitative analysis via the lens of the causal inference framework.

One can view our analysis as a more extensive, causal edition of Adebayo et al. (2018); we measure the treatment effect of H on E and Y across $30{,}000$ models, while they quantitatively measure *visual* similarities of Es as varying the quality of Y in a single pair of models (trained

[3]Since the individual for which E is sought is fixed throughout (i.e., X does not change; see discussion on identifiability at Appendix A.2), we disregard the effect of X on E in this study.

and untrained). Furthermore, our analysis reveals that Grad-CAM (which arguably 'passed' the sanity check in Adebayo et al. (2018)) shows a worse correlation between the two ITEs across the buckets, meaning that the hyperparameters affect Y and E differently, hinting that no methods concretely outperform others. Our results should be taken as a strong encouragement for practitioners to review other evidence instead of taking explanations at face value in their final decision-making.

References

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Adebayo, J., Muelly, M., Abelson, H., and Kim, B. Post hoc explanations may be ineffective for detecting unknown spurious correlation. *ICLR*, 2022.

Alqaraawi, A., Schuessler, M., Weiß, P., Costanza, E., and Berthouze, N. Evaluating saliency map explanations for convolutional neural networks: a user study. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pp. 275–285, 2020.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. How to explain individual classification decisions. *arXiv preprint arXiv:0912.1128*, 2009.

Bau, D., Liu, S., Wang, T., Zhu, J., and Torralba, A. Rewriting a deep generative model. *CoRR*, abs/2007.15646, 2020. URL https://arxiv.org/abs/2007.15646.

Chu, E., Roy, D., and Andreas, J. Are visual explanations useful? a case study in model-in-the-loop prediction. *arXiv preprint arXiv:2007.12248*, 2020.

Erhan, D., Bengio, Y., Courville, A., and Vincent, P. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

Jiang, Y., Krishnan, D., Mobahi, H., and Bengio, S. Predicting the generalization gap in deep networks with margin distributions. In *International Conference on Learning Representations*, 2019.

Kapishnikov, A., Venugopalan, S., Avci, B., Wedin, B., Terry, M., and Bolukbasi, T. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5050–5058, 2021.

Kindermans, P.-J., Hooker, S., Adebayo, J., Alber, M., Schütt, K. T., Dähne, S., Erhan, D., and Kim, B. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 267–280. Springer, 2019.

Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. Concept bottleneck models. In *International Conference on Machine Learning*, pp. 5338–5348. PMLR, 2020.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual knowledge in gpt. *arXiv preprint arXiv:2202.05262*, 2022.

Parliament and of the European Union, C. General data protection regulation. 2016.

Pearl, J. *Causality*. Cambridge university press, 2009.

Poursabzi-Sangdeh, F., Goldstein, D. G., Hofman, J. M., Vaughan, J. W., and Wallach, H. Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810*, 2018.

Reichenbach, H. *The Direction of Time*. University of California Press, Berkeley, CA, 1956.

Rieger, L., Singh, C., Murdoch, W. J., and Yu, B. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *ICML*, 2020.

Rubin, D. B. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005.

Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., and Batra, D. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Srinivas, S. and Fleuret, F. Rethinking the role of gradient-based attribution methods for model interpretability. In *International Conference on Learning Representations*, 2021.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

Unterthiner, T., Keysers, D., Gelly, S., Bousquet, O., and Tolstikhin, I. Predicting neural network accuracy from weights, 2020.

Wachter, S., Mittelstadt, B., and Russell, C. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

Wang, Z., Fredrikson, M., and Datta, A. Robust models are more interpretable because attributions look normal. *arXiv preprint arXiv:2103.11257*, 2021.

Xu, S., Venugopalan, S., and Sundararajan, M. Attribution in scale and space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9680–9689, 2020.

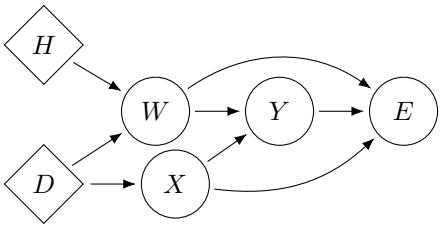

Figure 5: Extended version of explanation generating process from Figure 1b, now with weights W and dataset D made explicit.

A Additional background material

A.1 The explanation generating process

To ease understandability, we refer to Figure 5 as the extended graph of Figure 1b which makes the weights W and data D explicit variables. Similar to Figure 1, diamond nodes are considered factors whose effect we study, and circle nodes are random variables. In this extended graph, we clarify that H is *not* the model or trained weights. In other words, what we call hyperparameters (H) are sets like "*method of optimization*: SGD or AdaGrad" or "*regularizer coefficients*: 0.1 or 0.01 etc". All Hs can be assigned a value before we train any model and before observing any data. Note that we do not have weights (denoted by W) in Figure 1b, as they are not the focus of our study; instead, we are interested in whether and how decisions made prior to training a model (i.e., assignments of H) influence downstream Y and E.

Furthermore, considering the manner in which the model zoo was constructed whereby hyperparameters are sampled independently from some domain, there are no edges (no backdoors) from X (or D) to H. On the other hand, W may be affected by the data distribution D, directly and/or through the training samples, but W is not the focus of our work. Since we focus on the causal effect of hyperparameters H on Y and E (not the weights W on Y and E), the formulations in Section 2.2 remain unchanged.

A.2 On the identifiability and computability of treatment effects

An astute reader may notice that evaluating the treatment effects above as the difference between counterfactual contrasts bears a resemblance to another common explainability method, namely *counterfactual explanations* (Wachter et al., 2017). This parallel is evident when thinking of Figure 1 in a coarser manner, i.e., $\mathcal{H}, \mathcal{X} \rightarrow \mathcal{Y}$, whereby the hyperparameters and dataset instance enter a *potentially blackbox but queriable procedure* and yield a prediction. Whereas the counterfactual explanations of Wachter et al. (2017) aim to identify minimal feature perturbations of the dataset instance under a fixed model (i.e., the hyperparameters do not change; procedure: *model prediction*), evaluating treatment effects as in Equation (1) is done by iterating over values of hyperparameters to contrast resulting predictions given a fixed dataset instance (procedure: *model training*).

Due to our mechanical setup, a number of interesting observations arise. Although the *training* (T), *predicting* (P), and *explaining* (E) procedures may not be expressible in closed-form, the prediction Y_h in Equation (1) is exactly computable on a computer through *forward simulation*. In other words, upon selecting a set of hyperparameters, $H = h$, and under a fixed seed, all sources of randomness are controlled for and the procedures $\mathtt{T}, \mathtt{P}, \mathtt{E}$ deterministically yield a trained model, a prediction for a given instance, and the explanation for the said instance and model. This is significant as it allows for the *exact computation* of both $Y_{\text{TREATMENT}}$ and Y_{CONTROL} which is all that is needed to yield the value of the ITE exactly. In other words, we can view both $Y_{\text{TREATMENT}}$ and Y_{CONTROL} as *factual* outcomes. Therefore, unlike real-world settings (e.g., taking a headache medication) where one cannot measure the ITE exactly (due to the impossibility of observing both *factual* and *counterfactual* outcomes simultaneously; whereby in such cases, the ITE is either approximated or the ATE is used instead.) the effect of all treatments, on both individual-level or population-level, are identifiable.

Table 1: Comparison of the classical and mechanical (our) setting for computing ITE values.

(a) In the classical setting for computing treatment effects, only one of the potential outcomes for each individual, i, is observable. The average treatment effect is defined as the average difference between individual treatment effects $ATE = \mathbb{E}[Y_1^{(i)}] - \mathbb{E}[Y_0^{(i)}]$.

i	Y_0	Y_1	Y_2
1	a	-	-
2	-	f	-
3	-	-	k
4	-	h	-
⋮	⋮	⋮	⋮

(b) In our mechanical setting, given a model, $\hat{f}_h$, the potential outcome for any and all instances is computable (i.e., $\exists\, Y_h(X_i), i \in \mathcal{I} \implies \exists\, Y_h(X_k)\ \forall\ k \in \mathcal{I}$). Instead, one asks how to compute the treatment effect for h' when no data is available for this hyperparameter.

i	Y_0	Y_1	Y_2
1	a	e	-
2	b	f	-
3	c	g	-
4	c	h	-
⋮	⋮	⋮	⋮

Although the treatment effects are *identifable*, evaluating them is *computationally expensive*. To understand why, it helps to illustrate a parallel with the setting of counterfactual explanations (Wachter et al., 2017). Whereas the treatment effects in our setting (see Equation (1)) contrasts $Y_h^*(x)$ and $Y_{h'}^*(x)$, the work of Wachter et al. (2017) contrasts $Y_h^*(x)$ and $Y_h^*(x')$. Unlike the latter which only requires the invocation of the *predicting procedure* given a new instance x (e.g., a forward pass through a neural network), the former invokes the *training procedure* given a new hyperparameter setting (i.e., a full re-training). In practice, computing power is limited and we may only have access to the predictions under a single model, say, $Y_h^*(x)$ and it can be prohibitively expensive to produce the prediction under a different model, $Y_{h'}^*(x)$, especially for large neural networks.

In order to reason about $Y_{h'}^*(x)$, one is compelled to instead ask a *counterfactual* question: "*What would the prediction have been, had the optimizer been ν'*?" which can be answered through causal modeling without conducting real-world experiments, i.e., retraining with optimizer ν'. Metaphorically, there would have been no need for counterfactuals had one been able to simulate the entire universe (limited by either identification or computation). It is the physical constraints that call for these counterfactuals. Unfortunately, the procedures in Figure 1 (left) are not available in closed form. We clarify that unlike the classical randomized control trial (RCT) setting of evaluating ATE by contrasting average ITE values (where instances are randomly assigned to control or treatment), the mechanical nature of our setting allows for the target evaluation of all instances under control (h) or any treatment regime (h'); the challenge lies in the fact that applying a treatment to any one individual is as expensive as applying it to all individuals (see Table 1a and Table 1b for comparison). In this case, future research may explore the question of whether one can learn approximate procedures (i.e., approximate structural equations) to *predict the predictions of an untrained classifier, given only its hyperparameters*. In this regard, our preliminary results suggest a promising alternative to training individual models: developing meta-models that estimate a base model's prediction and explanation for an instance using only its hyperparameters, without actual training. This idea is derived from AutoML research, which predicts model accuracy based solely on hyperparameters, without training Unterthiner et al. (2020). As this issue rapidly evolves into a complex and multifaceted problem, we only briefly present the preliminary results here: a simple 3-layer MLP (namely, "meta-model") trained using X and H from a 10% sample of models in the repository (i.e., 10% of 30,000 "base-models"), can estimate the predictions Y for the rest of the base-models with an accuracy of approximately 45%. It is important to note that the input features do not have trained weights and rely on hyperparameters instead, therefore saving compute. Furthermore, when the training is conducted on a subset comprising 10% of the top-15% performing models rather than on all models (with a mix of highly and poorly performing base models; refer to Table 2), the meta-model can predict the predictions Y for the remaining base-models with an accuracy of around 80%. Not only would this be a fascinating follow-up research project, but it would also hold substantial practical value for our framework.

An implicit assumption made in (4) was that of mutual independence between hyperparameters, i.e., $h_i \perp\!\!\!\perp h_j\ \forall\ j \neq i \implies h_{\setminus i} \sim \prod_{j \neq i} \mathbb{P}(h_j)$. This assumption yields an *unconditional*

Table 2: Test accuracy boundaries for each performance bucket for each dataset in the model zoo Unterthiner et al. (2020).

percentile	0-20	20-40	40-60	60-80	80-90	90-95	95-99	99-100
CIFAR10	5-15	15-25	25-33	33-38	38-46	46-50	50-52	50-57
SVHN	7-17	17-19.5	19.5-19.6	19.6-33	33-51	51-59	59-65	65-78
MNIST	4-11	11-35	35-73	73-89	89-95	95-96	96-97	97-98
FASHION	1-11	11-47	47-68	68-76	76-82	82-84	84-85	85-88

treatment effect, whereby the causal effect of $h_i = \text{TREATMENT}$ vs $h_i = \text{CONTROL}$ is averaged over all possible combinations of other hyperparameters, even if the combination rarely occurs in high-performing models. In practice, however, it is conceivable that the hyperparameters are selected carefully by the system designer and may be interpreted as being sampled from a distribution over hyperparameters, $\mathcal{H}$, internalized by the designer through *prior* experience in training desirable models (e.g., accuracy, fairness). Such down-stream criteria may act as a common child of the hyperparameters, inducing complex inter-dependencies (cf. Berkson's paradox, (Pearl, 2009)). In this case (i.e., $h_{\setminus i} \not\perp \prod_{j \neq i} \mathbb{P}(h_j)$), the treatment effect answers such a query as "among the set of hyperparameters that yield models with at least γ performance, what is the treatment effect of optimizer choice ν_1 as opposed to ν_2 on the local prediction of x?" Therefore, whether or not we assume hyperparameters to be mutually independent depends on the query being asked and assumptions made of the prediction/explanation generative process.

Finally, one could consider straightforward extensions of (3) and (4) to support distributions over baseline control groups by adding an outer expectation that weights over the probability control group occurrence.

A.3 Model zoo details

For each of the 4 datasets (CIFAR10, SVHN, MNIST, FASHION) we consider $30{,}000$ pre-trained models, with diverse test accuracies resulting from the combinations of hyperparameters considered in the zoo (Unterthiner et al., 2020, Fig. 6). We optionally analyze models stratified by their test performance, over 8 performance buckets; Table 2 shows the boundaries of these buckets.

As a demonstration, Figure 6 shows the diversity in predictions of $30{,}000$ base models for a subset of CIFAR10 images for 1 randomly sampled datapoint from each class. It is noteworthy that the non-kernelized ITE values of (4) can be read directly from the figure, by contrasting the mean (shown in diamond) of each pair of nested bar plots (via application of linearity of expectations to (4)).

Pre-processing explanations and other details To study the effect of hyperparameters on explanations, we generate explanations, $E_h(x)$, via saliency-based methods. In particular, the Gradient (Simonyan et al., 2013; Erhan et al., 2009; Baehrens et al., 2009) and its smooth counterpart, SmoothGrad (Smilkov et al., 2017), Integrated Gradient (IG) (Sundararajan et al., 2017), and Grad-CAM (Selvaraju et al., 2016) methods are used due to their commonplace deployment[4] (Adebayo et al., 2018). Note that many other widely used methods are based on these four methods Kapishnikov et al. (2021); Xu et al. (2020); Wang et al. (2021); Simonyan et al. (2013). The generated explanation maps $E_h(x)$ are then processed to first remove outliers (via percentile clipping the values above 99th percentile), following by normalizing all attributions to fall in $[0, 1]$. For Grad-CAM which only generates positive attributes, this is straightforward; for other methods that give positive and negative attributes (as each carry different semantics; contributing towards/against the prediciton), we first normalize to $[-1, 1]$ and then clip any value below 0.

The set of hyperparameters considered include the choice of optimizer, w_0 type, w_0 std., b_0 type, choice of activation function, learning rate, ℓ_2 regularization, dropout strength, and dataset split (see Unterthiner et al., 2020, Appendix A.2). To evaluate treatment effects as per (4),

[4]All methods are openly accessible here: https://github.com/PAIR-code/saliency.

continuous features are discretized by (log-)rounding to the nearest predetermined marker from within the range of the feature.[5]

Relation to other explainability metrics

There are many such heuristics for rating explainability, and we recognize the absence of such comparisons in our research study. At the same time, we emphasize that our proposed metric assesses "how much of the explanation is actually explaining the prediction," which, at least from an intuitive standpoint, is neither implied by nor implies other such metrics as *intelligibility*, *transparency*, *complexity*, or *user-friendliness*. We also recognize that relying solely on the suggested metric may lead to misleading results and should not be considered adequate for endorsing an explanation approach. As demonstrated in footnote 1, we provide an instance where the Identity explanation implies an ideal correlation between ITE_E and ITE_Y, even though it does not offer a meaningful explanation. We encourage further investigation in this direction for future research.

B Additional experimental results

In this section, we present additional experimental results to complement those in the main body across different data dimensions or on new datasets.

As a demonstration, Figure 6 shows the diversity in predictions of $30{,}000$ base models for a subset of CIFAR10 (top) and SVHN (bottom) images for 1 randomly sampled datapoint from each class. It is noteworthy that the non-kernelized ITE values of (4) can be read directly from the figure, by contrasting the mean (shown in diamond) of each pair of nested bar plots (via application of linearity of expectations to (4)).

[5]The following markers are used for (log-)rounding continuous features: ℓ_2 reg.: $[1e^{-8}, 1e^{-6}, 1e^{-4}, 1e^{-2}]$, dropout: $[0, 0.2, 0.45, 0.7]$, w_0 std.: $[1e^{-3}, 1e^{-2}, 1e^{-1}, 0.5]$, learning rate: $[5e^{-4}, 5e^{-3}, 5e^{-2}]$.

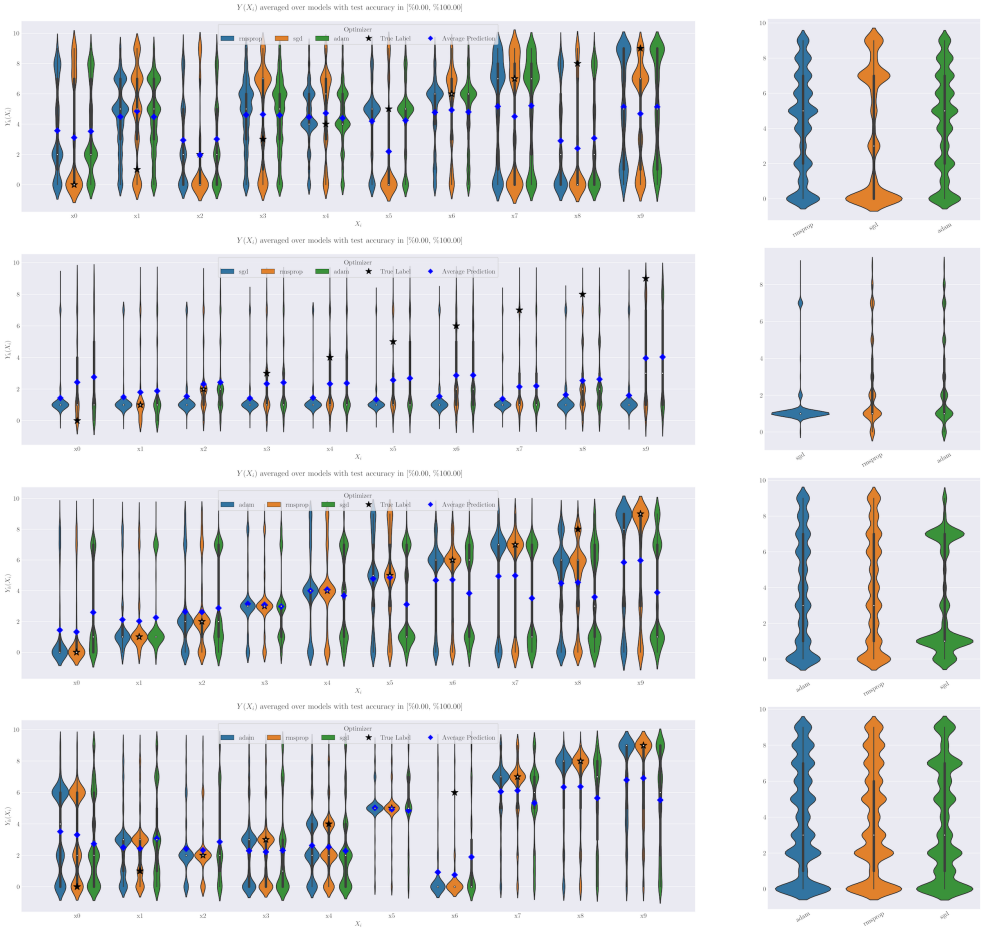

Figure 6: The distribution of $Y_h(x_i)$ for a subset of 10 random instances(1 per class) on $30{,}000$ base models (row 1: CIFAR10; row 2: SVHN; row 3: MNIST; row 4: FASHION). For each instance, each column holds the value of $h_{\text{optimizer}}$ fixed at one of m unique values pertaining to this hyperparameter, while unconditionally iterating over other hyperparameters. In this manner, the difference in predictions across values of the hyperparameter, both at an individual (left) and aggregate level (right) can be attribute to, and only to, changes in this hyperparameter.

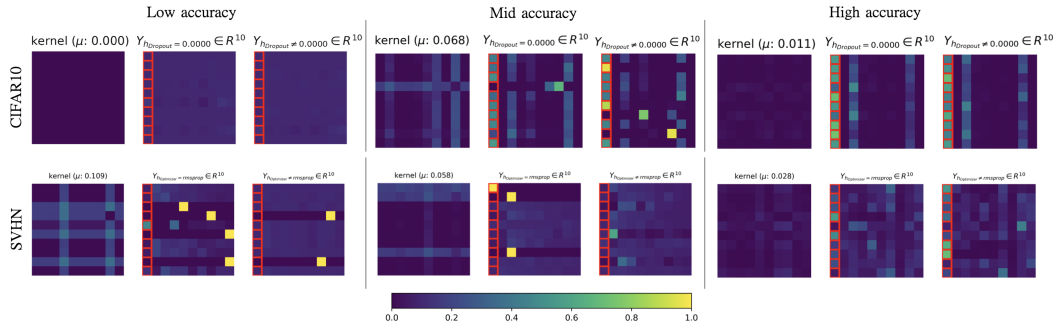

Figure 7: Examples of class predictions ($Y_{h=n}(x)$ and $Y_{h\neq n}(x)$) and their dissimilarities ($\|\phi(Y_{h=n}(x)) - \phi(Y_{h\neq n}(x))\|^2_{\mathcal{G}}$) for different accuracy buckets for CIFAR10 (top) and SVHN (bottom). Each row shows 10 random predictions from 3 models in the low- (left), mid- (center), and top- (right) performance buckets, under two different treatment groups for the dropout value ($= 0$ and $\neq 0$). In each performance bucket, there are three subplots. Each subplot is showing 10 randomly selected samples (each row) and their post-softmax values for one of the 10 classes (hence a 10×10 grid). The first plot in each trio shows the RBF kernel evaluation of the center and right predictions. The center and right plots show these treatment/control groups. This figure is intended to complement Figure 2 to explain why ITE for Y is large for mid-accuracy buckets and small for high-accuracy buckets. For CIFAR10, the values are small for low-performing models (most models in this bucket predicting similarly) but for SVHN the values are large due to different diverse predictions.

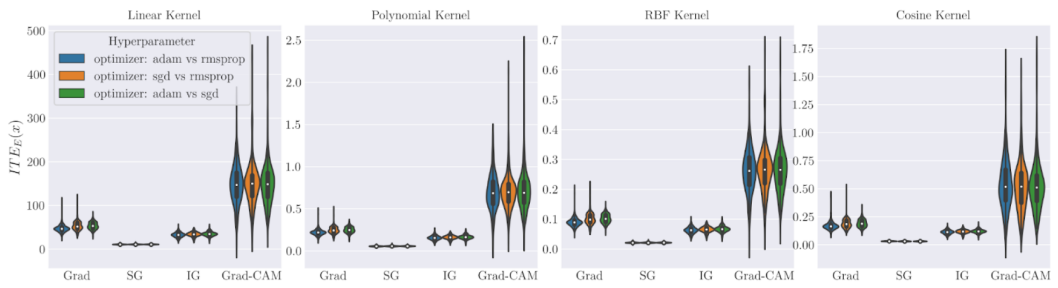

Figure 8: Comparison of the ITE values with kernelized version of (4) for $E_h(x)$ obtained for 100 instances from CIFAR10 for different choices of the kernel (each column) shows that KTE is not sensitive to the choice of kernels. Contrast this figure with **??**; we conclude that the choice of baseline (i.e., where we contrast *optimizer: adam* against all other optimizers as in **??** or against other individual values) does not affect the overall trend and should be chosen according to the question in mind: to compare the effect of a hyperparameter value against all other possible values, or against a particular value.

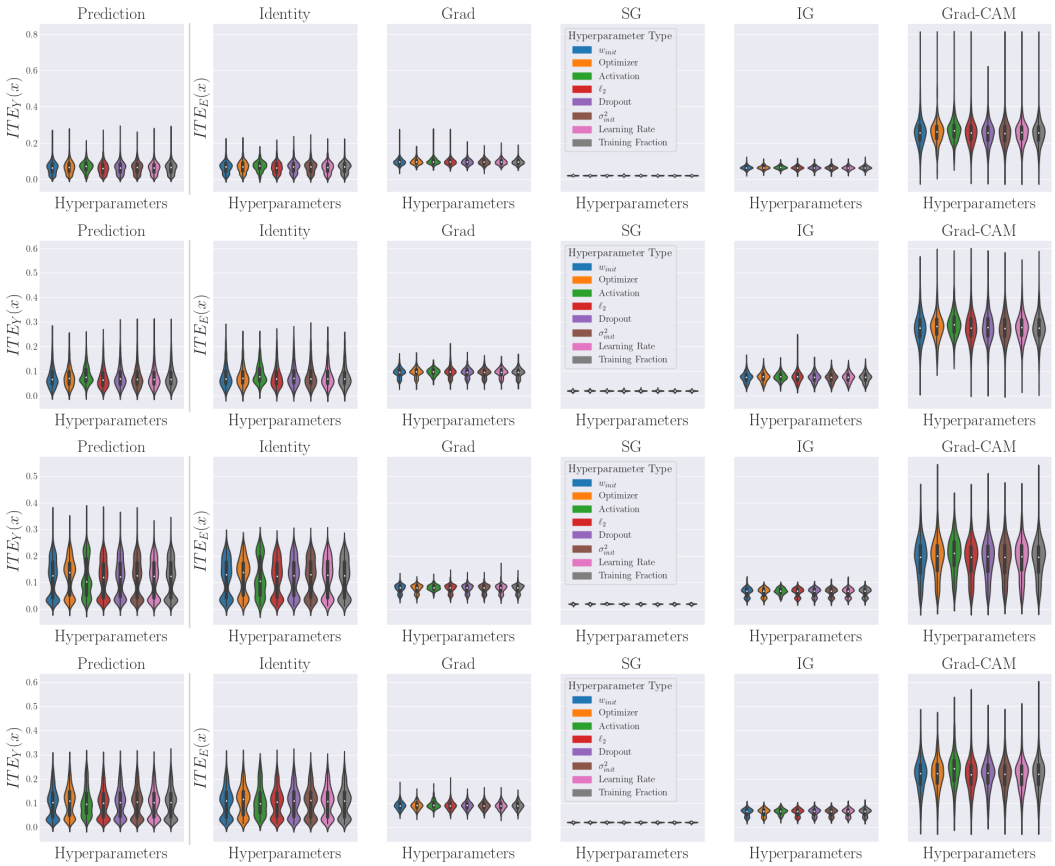

Figure 9: ITE values for Y (left) and E (right) show similar effect for different *types* of H across CIFAR10 (row 1), SVHN (row 2), MNIST (row 3), FASHION (row 4).

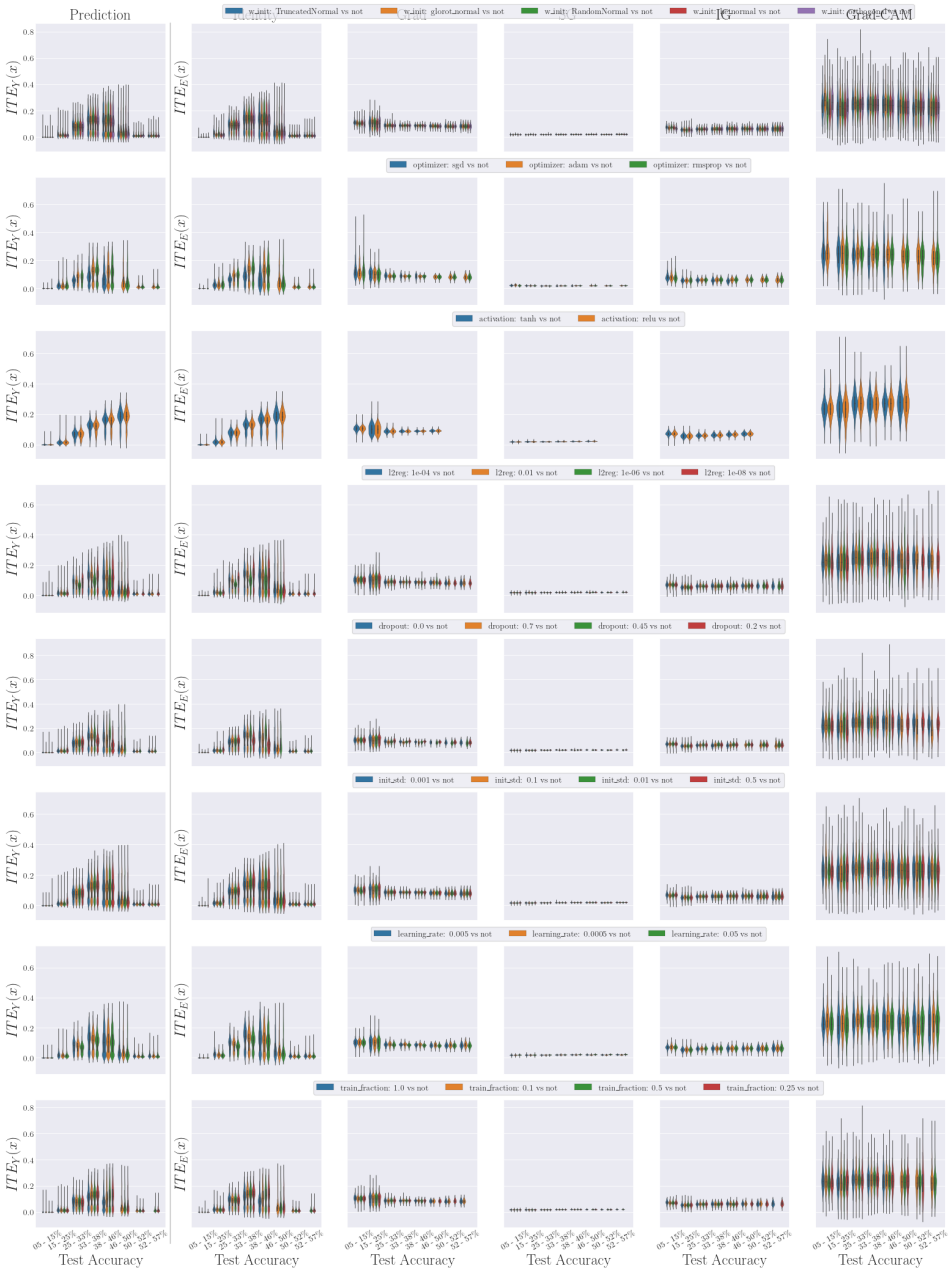

Figure 10: Comparison of ITE values of all hyperparameters (each row)on Y (left) and E (right) for models trained on CIFAR10 across different performance buckets, showing the discrepancy in the effect of H on Y vs. that on E.

Figure 11: Comparison of ITE values of all hyperparameters (each row)on Y (left) and E (right) for models trained on SVHN across different performance buckets, showing the discrepancy in the effect of H on Y vs. that on E.

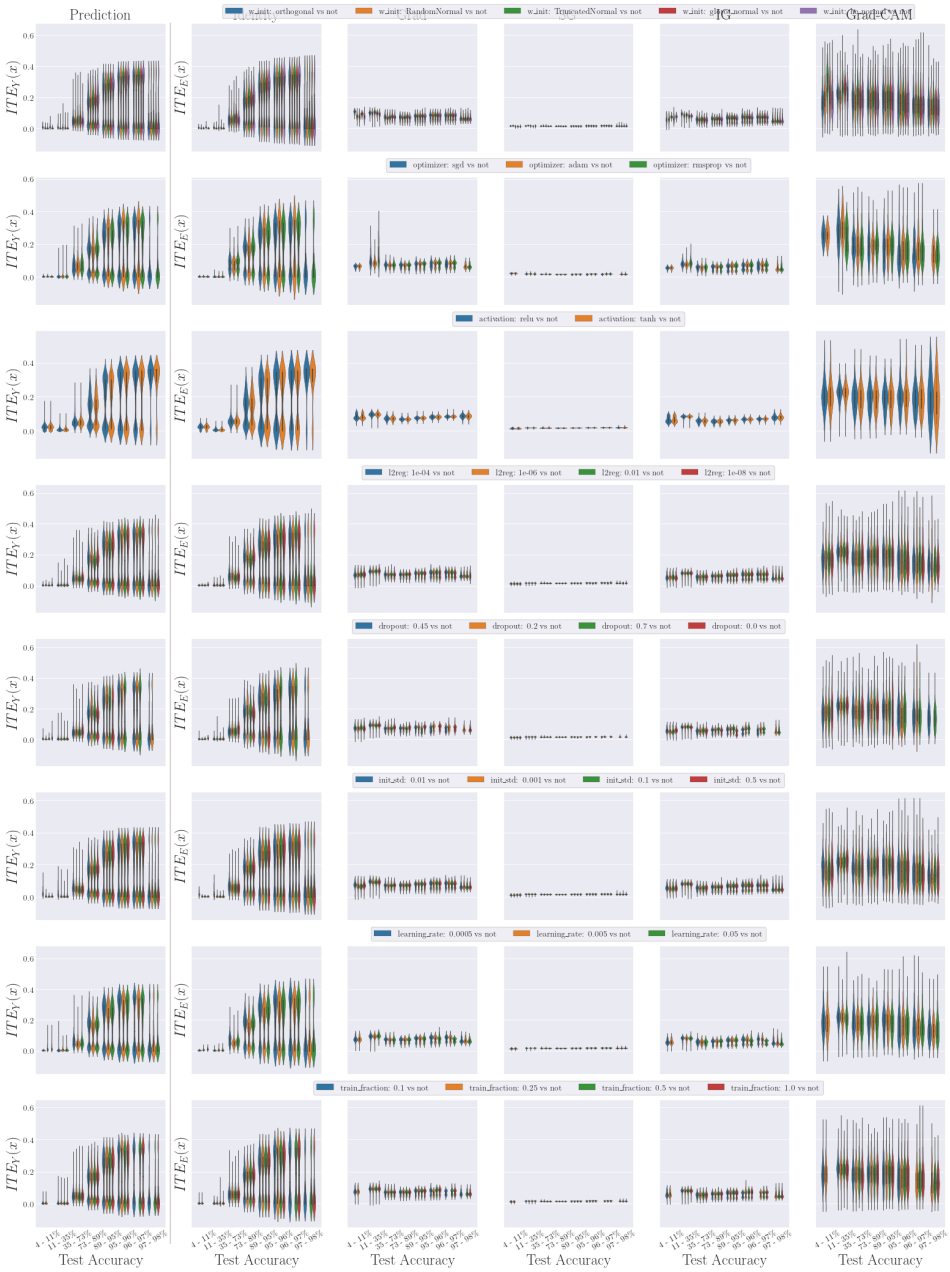

Figure 12: Comparison of ITE values of all hyperparameters (each row)on Y (left) and E (right) for models trained on MNIST across different performance buckets, showing the discrepancy in the effect of H on Y vs. that on E.

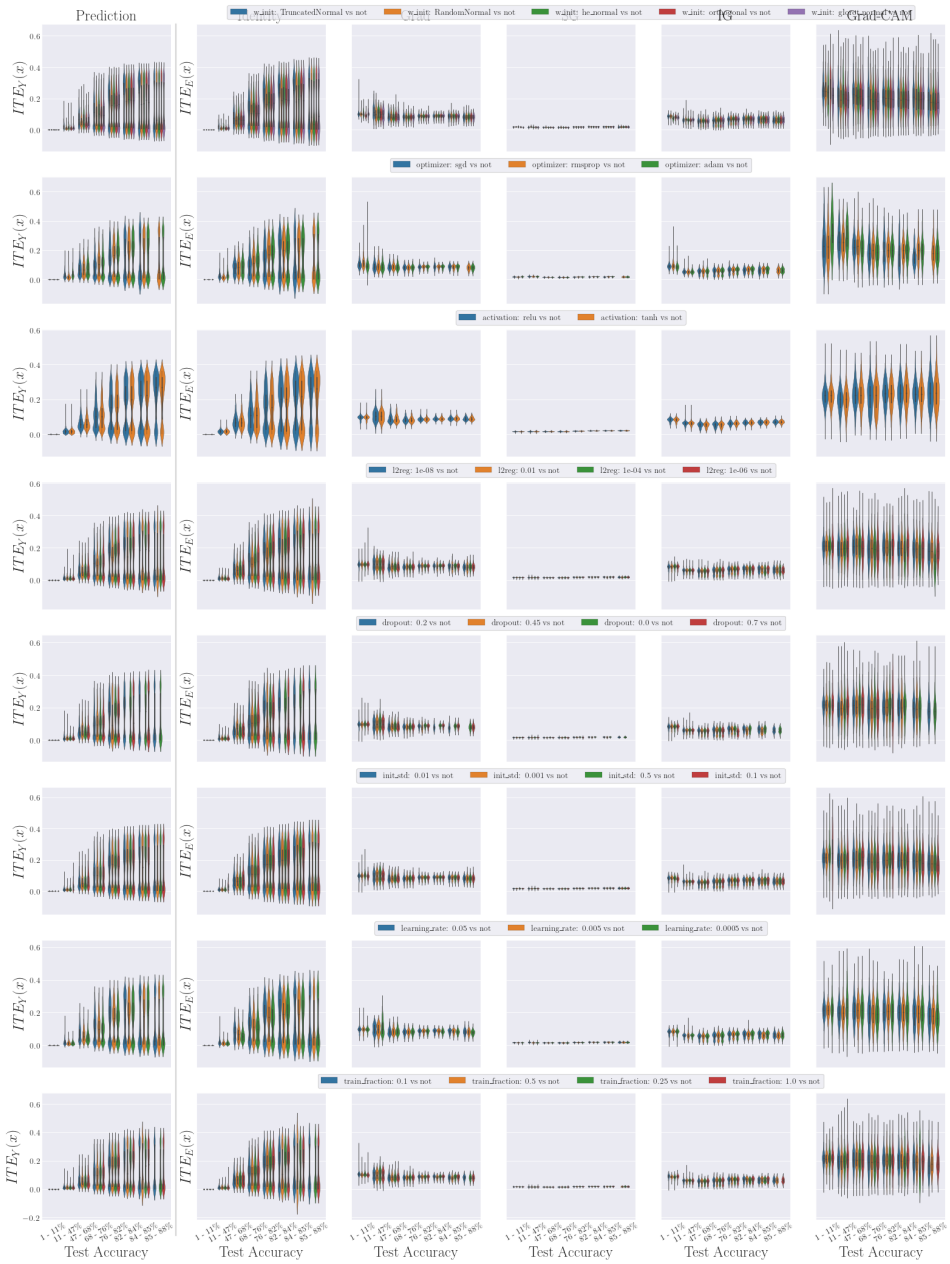

Figure 13: Comparison of ITE values of all hyperparameters (each row)on Y (left) and E (right) for models trained on FASHION across different performance buckets, showing the discrepancy in the effect of H on Y vs. that on E.

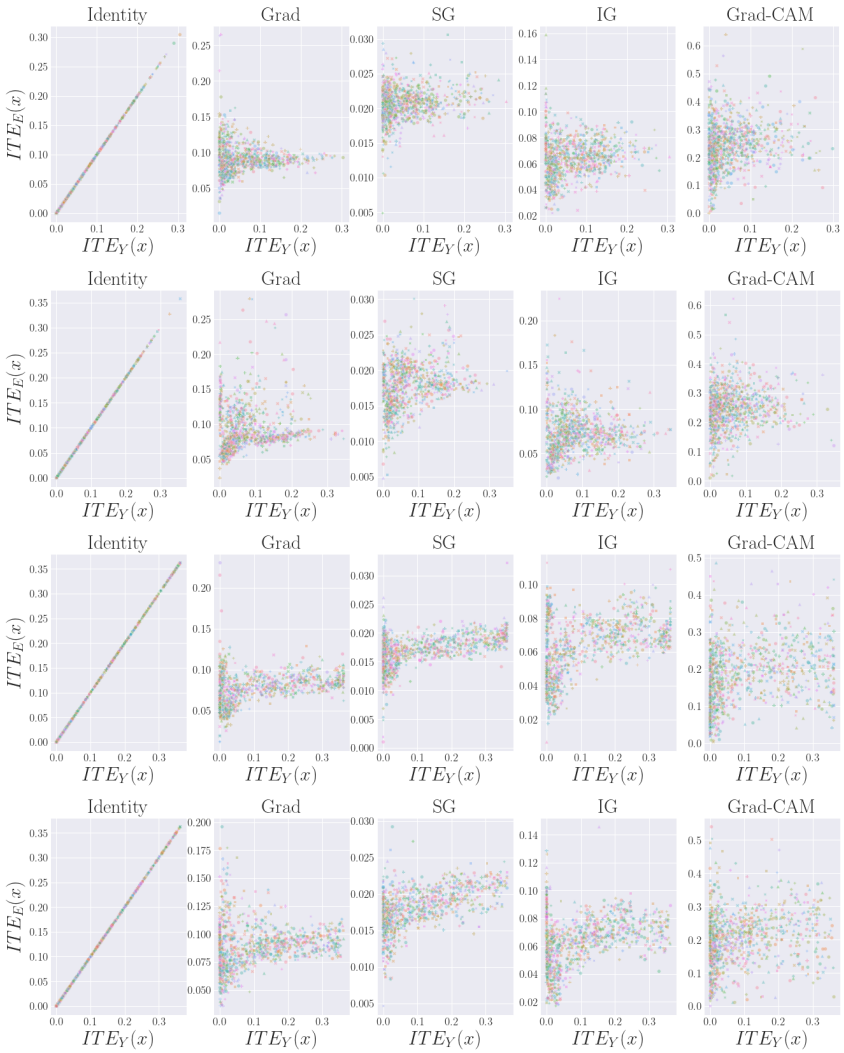

Figure 14: Scatter plot of ITE values for Y and E (row 1: CIFAR10; row 2: SVHN; row 3: MNIST; row 4: FASHION) across explanation methods reveals no apparent patterns.

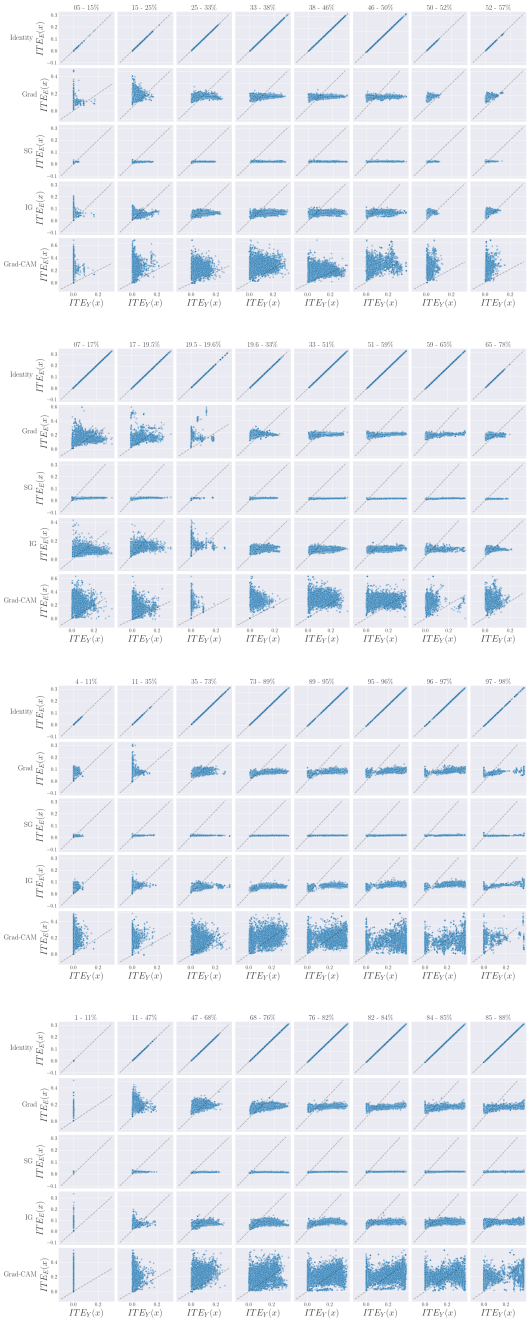

Figure 15: Each column is a subset of models at each accuracy bucket, each row is different explanation methods (row 1: CIFAR10; row 2: SVHN; row 3: MNIST; row 4: FASHION). Whereas low-performing models (first column) show little change in predictions as their explanations differ, top-performing models show the reverse of this trend.

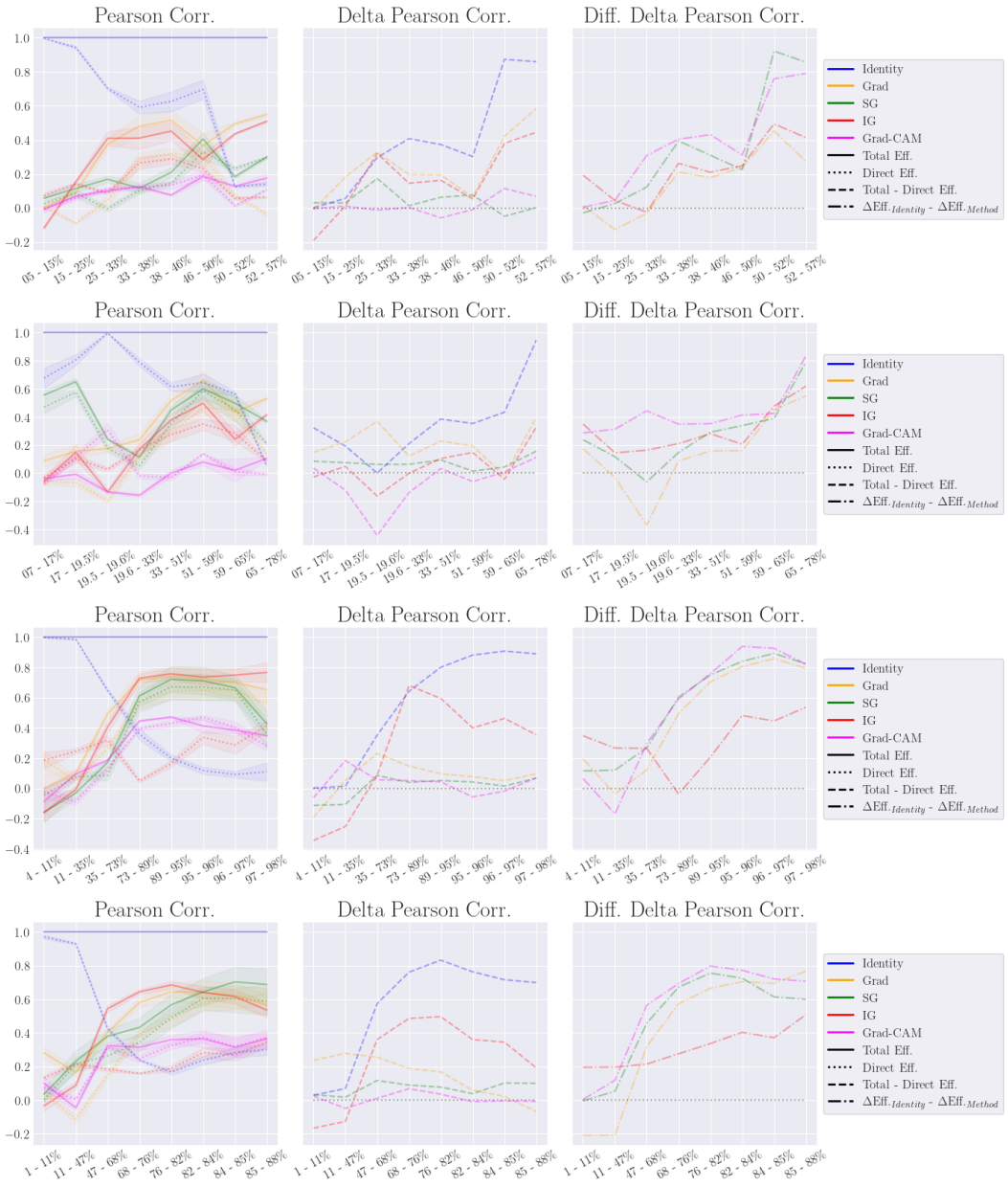

Figure 16: Pearson correlation and Spearman's Rank correlation for ITE of Y and ITE of E across different explanation methods and model performance buckets, for mediated and unmediated Y (row 1: CIFAR10; row 2: SVHN; row 3: MNIST; row 4: FASHION). Absolute values of correlation values are smaller across both datasets (max around 0.5), suggesting that E takes influence from H that does not necessarily pass through Y. The final absolute correlation is going down for top-performing models in both datasets. The increase in delta correlation between mediated and unmediated Y suggests that the direct impact of Y on E is becoming even more important in top-performing models, even more so for SVHN than for CIFAR10.