We thank all reviewers for their thoughtful feedback, and for recognizing that the work addresses an important question (**R2**, **R3**, & **R5**), is well-written (**R2** & **R3**), and considers a breadth of attribution methods (**R2** , **R4**, & **R5**). Below we address concerns raised as part of the reviews. We believe your insights and comments will make the paper stronger.

Additional Clarification & Insights (R3 & R5). Several interpretation approaches have been proposed (see arXiv:2003.07631 for 35 methods); however, it is unclear when, and for what debugging tasks, an interpretation method is effective. Our empirical assessment is a step towards addressing this issue, and provides initial evidence that future work, theoretical and empirical, can build upon. For example, surprisingly, we find that methods that modify back-propagation to compute relevance are able to detect the spurious background bug despite being essentially invariant to higher layer parameters. We will clarify in the paper, the contributions that differentiate our work, and provide concrete recommendations for practitioners. We will highlight where future work can build upon these findings.

Why only attributions? Influence Function and Concept Methods (R2 & R4). We focused on attribution methods to keep our inquiry thematically focused. However, we have tested: i) influence functions (IF) (Koh et. al. 2017) for training point ranking, and ii) the concept activation (TCAV) approach (Kim et. al. 2018). We assess IF under spurious correlation, mislabelled examples, and domain shift. We test TCAV under the spurious correlation and model contamination setting (see Figure 1). TCAV identifies reliance on the spurious background concept and shows sensitivity to parameter weights under the model contamination setting. We find that IF shows association regarding spurious correlation. For example, for a given input with the spurious background, we measure the fraction of the top 1 percent of training points (86) that are spurious. We find, on 50 examples, that 91 percent (2.4 standard error) of the top 1 percent of training points are spurious. Similarly, we use the self-influence metric to assess mislabelling (see arXiv:2002.08484), and find that for 50 examples, we need to check an average of 11 percent of the training set (5.9 standard error). These results suggest that both methods might be effective for model debugging.

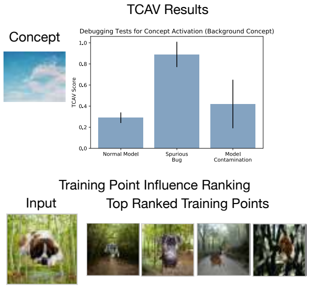

Figure 1: Assessing TCAV and influence functions.

Structure of the Paper (R4 & R5). We will streamline the exposition to minimize fragmentation and reduce excessive reliance on subheading & bolded paragraphs. For example, the discussion of the user study results can be summarized in a single section instead of across sections 2-5 as it currently stands.

@**R2**: *(Fig 5b):* Users were asked to chose one option, however a free form answer was provided when the need to select both or other motivations occurred. Participants selected the free form option in 51 out of 1130 responses (4.5%). Among these responses only 6 reported that a combination of both label or explanation was used to make their choice. The remainder contained other comments focusing on surprising and unexpected results associated with either the explanation or the label. *(Only Correct Predictions):* Since the focus of our work is model debugging, it is necessary to test settings where the predicted class is incorrect. *(Visual Similarity vs. Feature Ranking (212-215).* These findings indicate that similar portions of the input is important under the two different settings, but that the ranking of the importance of these dimensions is different. *(Class Prediction).* We always explain the predicted class even when the predicted class is incorrect. *(Test-Time Debugging).* We explain the 'erroneous' prediction on the out-of-domain model. Eg., the fashion MNIST input in Fig. 8 is classified as an '8' for an MNIST model, so we explain this target on the fashion MNIST input for the MNIST model. We will clarify these details in the updated manuscript.

@**R3**: *(Lai & Tan FAccT 2019).* Thanks for the pointer, we will include and discuss the paper. A key difference is that under our setting, the end user's *sole* task is to assess the model's reliability; however, Lai & Tan consider a setting where the human and model combine to perform a task. *(Bug Categorization):* We agree that the bug categorization addresses a simple supervised learning setting. Adversarial examples, and **R3**'s example on unsupervised pre-training do not fit, cleanly, under the current categorization. We will clarify and state the setting more clearly in the manuscript. *(User Study Details):* 80% of the participants had either trained a model, taken an ML class, or are ML researchers (Fig.50 Appendix). We also provided training on model attributions, training data, and the task. We will include this discussion in the main body of the paper rather than the appendix.

@**R4**: *(Image classification):* We will make it clear that we focus on image classification, and that our results might not translate to tabular settings with semantically meaningful features. *(Loss):* We performed all mislabeled experiments again, but this time explain the *loss*. For methods that modify back-propagation, our findings do not change (SSIM & Rank Correlation ≥ 0.85). For the other methods, we observe a slight drop: SSIM 0.72-0.81, and rank correlation 0.69-0.83. We agree that this work opens up discussion about other ways of using attribution methods to diagnose bugs.

@**R5**: *(SSIM):* Our use of SSIM to measure visual similarity follows previous work (Adebayo et. al. 2018 & Sixt et. al. 2020) that used it for a similar purpose.