We thank all of the reviewers for their thoughtful feedback, and will incorporate their suggestions into the next version of our paper. We detail our responses to their comments below.

**R1.** We thank R1 for their comments and will emphasize the broader implications of our work on model explainability.

**R2.** R2 asked to contrast using (i) influence functions to measure the importance of training points with (ii) existing techniques for measuring feature importance, namely Datta, Sen, & Zick, 2016; Adler et al., 2016; and Adebayo & Kagal, 2016. These papers address a different problem setting from ours and their methods are correspondingly distinct.

The main difference is that the papers above seek to explain a *fixed* model $\theta$, whereas we examine how the *learned* model $\hat{\theta}$ changes as a function of its training data. Our central issue is therefore reasoning about retraining the model, which is not a concern shared by the papers above. Concretely, those papers consider the question: given a fixed model $\theta$, how do its predictions on a test set $\mathcal{D}_{\text{test}}$ depend on the values of some feature $x_k$ in $\mathcal{D}_{\text{test}}$? They investigate this by perturbing the value of $x_k$ in various ways (e.g., by randomizing $x_k$ for each test point $x$ in $\mathcal{D}_{\text{test}}$). In contrast, we are given the training set $\mathcal{D}_{\text{train}}$ and our goal is determining the effect of removing groups of points in $\mathcal{D}_{\text{train}}$ on the learned $\hat{\theta}$.

Despite their differences, these methods could be complementary, as R2 suggested. For instance, if we find (with feature importance methods) that a model depends heavily on some feature, we could use influence functions to identify the training data most responsible for that dependence. We will include this discussion and we thank R2 for pointing it out.

**R3: Non-convex objectives.** R3 asked if our empirical findings hold for non-convex models. Our initial experiments are promising and suggest that this can be true; we will discuss this question in our next revision and plan to conduct a more extensive study. We are grateful to R3 for highlighting this question. To properly respond, let us first provide context on why influence functions and actual effects are classically only defined for convex models.

Recall that the actual effect $\mathcal{I}^*_f(w)$ of a subset $w$ measures the difference between (i) the original model $\hat{\theta}(\mathbf{1})$, which minimizes the loss on the training data $\mathcal{D}_{\text{train}}$, and (ii) the new model $\hat{\theta}(\mathbf{1}-w)$, which minimizes the loss on $\mathcal{D}_{\text{train}}$ with $w$ removed. Thus, for $\mathcal{I}^*_f(w)$ to be well-defined, there must be a unique model $\hat{\theta}(\mathbf{1})$ that globally minimizes the training loss on $\mathcal{D}_{\text{train}}$, and likewise for $\hat{\theta}(\mathbf{1}-w)$. This condition is satisfied when the model is strongly convex. Similarly, the influence $\mathcal{I}_f(w)$ is only well-defined when $\hat{\theta}(\mathbf{1})$ is unique and the model is strongly convex around it. Finally, to measure the actual effect and influence, the models $\hat{\theta}(\mathbf{1})$ and $\hat{\theta}(\mathbf{1}-w)$ must not only be well-defined but computable.

Non-convex models unfortunately violate all of these requirements: the global minimizer $\hat{\theta}(\mathbf{1})$ may not be unique for a given $\mathcal{D}_{\text{train}}$, and even if it were, we may not find it. For instance, neural networks are typically trained with SGD-based methods that only guarantee convergence to a local minimum, so in general we cannot compute $\hat{\theta}(\mathbf{1})$ nor $\hat{\theta}(\mathbf{1}-w)$.

To address these issues, we propose augmenting the classical definitions as follows. Let the actual effect $\mathcal{I}^*_f(w, \theta_0, r)$ of a subset $w$ given an initial trained model $\theta_0$ and a random seed $r$ to be the change in the model after removing $w$ and retraining the model by starting from $\theta_0$ and running SGD with randomness $r$. Specifying the initial model $\theta_0$ sidesteps the issue of $\hat{\theta}(\mathbf{1})$ being non-unique or impossible to compute, while specifying $r$ resolves the issue of the retrained model $\hat{\theta}(\mathbf{1}-w)$ being ill-defined. Similarly, we augment the predicted effect $\mathcal{I}_f(w, \theta_0)$ to be the influence of $w$ around the initial model $\theta_0$. Our goal is then to see if $\mathcal{I}_f(w, \theta_0) \approx \mathcal{I}^*_f(w, \theta_0, r)$ for all $r$.

We tested this with a multi-layer perceptron (MLP) with 2 hidden layers (128 and 32 nodes) on 10% of the MNIST 10-class dataset, choosing $\theta_0$ as a model trained from scratch with an arbitrary random seed. As in our submission, we generated 70 coherent groups of training points. For each group, we tried 50 values of $r$ but found negligible variation between the actual effects (max difference of $3 \times 10^{-4}$ between $r$'s). Figure S1 shows that influence is highly correlated with actual effects (for an arbitrary $r$) on the test point with highest loss (Spearman $\rho = 0.96$), even in this non-convex setting.

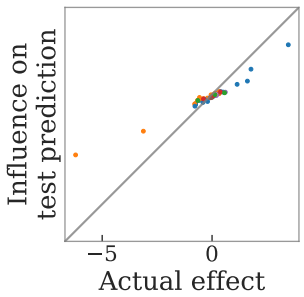


Figure S1: Predicted vs. actual effects on an MLP. Colors are as in our submission.

**R3: Case studies.** R3 is right to point out the distinction between removing certain labeling functions (LFs) and crowdworkers from the training data, which we do in our work, and actually changing the LFs or the crowdworker recruiting policy. For example, it is possible that explicitly encouraging LF programmers to create LFs with higher coverage could result in spurious LFs that lower model performance, even though the naturally-obtained high-coverage LFs are the most helpful for the model. Determining the actual effect of manipulating data collection will require systematic user experiments, and we will clarify and emphasize this distinction. We thank R3 for bringing this point up.