We thank all four reviewers for their great reviews. We provide our feedback for each reviewer as follows.

General concerns on experiment designs from R1 and R3:

The main goal of the empirical study in our paper is 1) to validate the proposed influence function is a good approximation to the ground truth influences and 2) to show the scalability. To validate the correctness, one needs to **train the pretraining model from scratch with each training sample removed** and retrain the finetuning task to get the ground truth influence. Due to the difficulty of evaluation, we can only use small scale datasets (e.g. MNIST and CIFAR) to generate figures like Figure 1 (a-c) in our paper to show the correctness. In fact, small datasets such as MNIST and CIFAR are standard datasets in influence function or data cleansing literature. In related works such as Koh and Liang, ICML'17, Koh et al. NeurIPS'19, and Hara et al. NeurIPS'19, the researchers also evaluated their methods on these datasets or similar scale datasets.

In this work, in addition to validate the correctness, we further demonstrate that scalability is not a problem for our algorithm – the computation can scale to a large and complex NLP task: Elmo model+sentiment analysis task. The Elmo model has 93.6 million parameters and is trained using 30 million sentences; however it's almost impossible to plot Figure 1 for Elmo to validate the correctness since to do so we need to train the Elmo model from scratch with each sample removed which takes months.

We thank the reviewer R1 for pointing out a potential large scale experiment on ImageNet with interesting applications, but we were not able to finish that in the limited rebuttal time as for this task we need to remove one sample in ImageNet at a time for all the images, and train a ResNet model from scratch to validate the correctness. We will add that in a revised version.

R1. We thank the reviewer for the insightful review. please see the above 'General concerns on experiment designs' section for details about our empirical study. We hope this can address your concern.

R2. We thank the reviewer for the positive comments.

High-level discussion: We will add a section in the revised version on higher-level discussion about. From the experimental results in Section 4.1 we can see that, removing the pretrain examples with high positive influence function values will decrease the model's total loss values on the test set and thus improve model's performance in testing. Hence, deleting those low-quality pretraining examples or replacing them with newly collected examples would be a good strategy.

Generalization to other finetune tasks: Our proposed multi-stage influence function is finetune task specific (see the influence score computation in Eq(11)). Please refer to Section 4.2 for finetune tasks' accuracy after removing low quality pretrain examples identified by our influence function. However, if we use one finetune task's influence function to clean pretrain data for another finetune task, it is hard to tell the performance. We design a new experiment on that: similar to the settings in Section 4.1, we use a 4-class classification (0, 1, 2, and 3) task from MNIST as pretrain task and a 3-class classification (4, 5, 6) task as finetune task to calculate influence function. Then we remove the pretrain data and go through the same process to get the finetune loss difference of another classification task (7, 8, 9). The Pearson r value of this task is only 0.13. This experiment illustrates that the low quality examples identified by influence function for one finetune task may not be low quality for another finetune task. While it is still interesting as future work to see whether that are some pretrain examples that are task-agnostic with high/low influence scores and removing/adding them can improve performance for various finetune tasks.

R3. We thank the reviewer for the insightful review.

Latent baseline: We thank the reviewer for proposing this baseline scenario. We implemented the reviewer's idea and use the pretrain loss value as the score and followed the procedures in Section 4.1 to evaluate its effectiveness. The Pearson r values are 0.09 for CIFAR and 0.10 for MNIST, which are much smaller than our proposed method. We will add this baseline in the revised version. We think that the main reasons are that: (1) Training examples with large training loss or large entropy values of the predicted distribution are not necessarily low quality examples. For example, in a binary soft SVM, samples in between two margins are support vectors that with high uncertainty, but it is uncertain whether the impact is positive or negative. (2) Even if we can identify low quality data in the pretrain task, because pretrain and finetune are different tasks, it is possible that those examples are actually helpful if we want the embedding to perform better in the finetune task. Please see our response to R2's "Generalization to other finetune tasks" for a relevant experiment we added.

Larger datasets: please see the above 'General concerns on experiment designs' section for details about our empirical study. We hope this can address your concern.