========== TO REVIEWER No.2 ==========

[Q1.1] On similar approaches.

[A] Other than the discussions in the related works section, we wish to highlight that, the difference between our work and the usual "data poisoning" works is as follows: In data poisoning setting, the attacker's goal is to add or modify as few samples as possible (i.e., "an attacker usually cannot directly access an existing training database but may provide new training data" [2]), whereas our proposed framework put the constraint on the perturbation levels (as human imperceivable as possible). In addition, to our best knowledge, this is the first scalable solution that can run on big datasets (thanks to the optimization method we designed). Other SOTA methods for poisoning was only tested on datasets of small sizes (e.g., [20] used only 1,000 samples). In other words, a direct comparison is less fair for the standard poisoning algorithms. Nevertheless, we will include some more actual quantitative assessments for such comparison to make the points clearer in the revised version. Thank you for the suggestions. (Please also refer to the similar response to [Q2.1] and [Q3.2].)

========== TO REVIEWER No.4 ==========

[Q2.1] On comparable baseline methods.

[A] Please also see [Q1.1]. We would also like to emphasize that, our work is not about poisoning only but rather on high-dimensional adversarial training examples with bounded perturbation. Therefore, we designed several different experiments with different angles toward this goal. Other existing methods for poisoning seems to be less aligned with our propose, making such comparisons less attractive. Meanwhile, as reviewer 2 also pointed out, we actually did conducted some baseline method such as Random Flip (Figure 5) for compare but only for validating purposes.

[Q2.2] "Is modifying 60% a practical setting?"

[A] Actually, yes. Unlike the traditional poisoning task, adversarial training data can be used in a good way. For instance, in some applications an agent may agree to release some internal data for academic research, but does not like to enable the data receiver to build a model which performs well on real test data; this can be realized by applying DeepConfuse before the data release. For motivations on privacy and why not just using synthetic data for these tasks, please also see our response to [Q3.2]. We will make the above points clearer in the revised version and to revise the typos/add the omitted related works as suggested. Thank you for the suggestions.

========== TO REVIEWER No.6 ==========

[Q3.1] How about some visualizations and interpretations for the corresponding models?

[A] Thank you for the suggestion. For linear SVMs, we did a quick visualization on the weights trained on clean and adversarial training data as shown below:

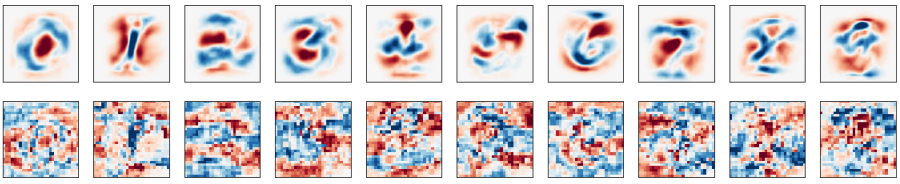

Figure 1: LinearSVM weights visualization for MNIST. Top row: Weights trained on clean training data. Bottom Row: Weights trained on adversarial training data.

It can be shown that the weights for the SVM trained on adversarial data indeed went to the opposite direction compared with the corresponding clean model and trend to overfits on image corners. We will definitely add some more experimental results with a more detailed discussion in the revised version as suggested.

[Q3.2] On some more motivations.

[A] Please also see [Q1.1] and [Q2.1]. For data privacy aspect, our proposed approach is quite different from releasing synthetic data via GANs. Consider a company selling surveillance cameras and the user(usually the police) will store all the photos been taken. These photos cannot be synthetic for obvious reasons. On the other hand, such a company does not want any other third parties to train a classifier using their data. Then DeepConfuse is suitable for this kind of task and the camera company can just send the modified data in real-time. We will elaborate more on the motivation side in the revised version.

Thank you all for your thorough and insightful comments again.