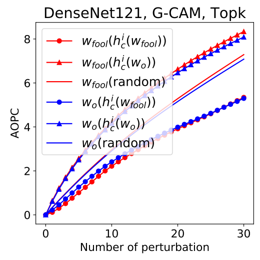
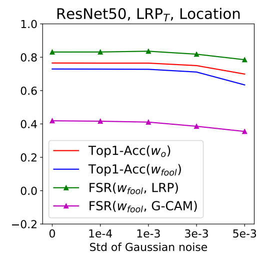
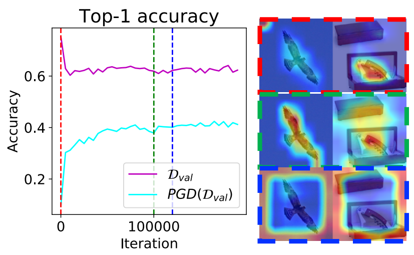


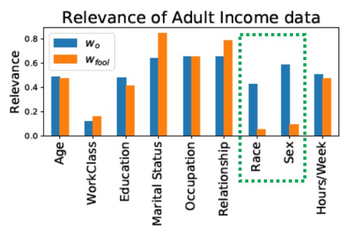
(a) AOPC curve



(b) Weight+Gaussian



(c) AMM for adversarial training



(d) Disguising the bias of a model

AOPC [R1,R2] We employ Area Over Prediction Curve (AOPC) [Samek *et.al*, `arXiv:1509.06321`], a principled way of quantitatively evaluating the validity of neural network interpretations, to check whether the manipulated model also has been significantly *changed* by fooling the interpretation. From the definition of AOPC, we can conclude that the interpretation is closely tied to the actual prediction procedure of the model if AOPC rapidly increases with the number of perturbation of input pixels. Fig. (a) shows the average AOPC curves on 10K validation images for the original and manipulated DenseNet121 models, $\boldsymbol{w}_o$ and $\boldsymbol{w}^*_{fool}$, with three different perturbation orders; i.e., with respect to the $\mathbf{h}_c^{\mathcal{I}}(\boldsymbol{w}_o)$ scores, $\mathbf{h}_c^{\mathcal{I}}(\boldsymbol{w}^*_{fool})$ scores, and a random order. We did the Top-k fooling with $\mathcal{I}$ being Grad-CAM. From the figure, we observe that $\boldsymbol{w}_o(\mathbf{h}_c^{\mathcal{I}}(\boldsymbol{w}_o))$ and $\boldsymbol{w}^*_{fool}(\mathbf{h}_c^{\mathcal{I}}(\boldsymbol{w}_o))$ show almost identical AOPC, which suggests that $\boldsymbol{w}^*_{fool}$ has *not changed much* from $\boldsymbol{w}_o$ and is making its prediction by focusing on similar parts that $\boldsymbol{w}_o$ bases its prediction. In contrast, the AOPC of both $\boldsymbol{w}_o(\mathbf{h}_c^{\mathcal{I}}(\boldsymbol{w}^*_{fool}))$ and $\boldsymbol{w}^*_{fool}(\mathbf{h}_c^{\mathcal{I}}(\boldsymbol{w}^*_{fool}))$ lie significantly lower, even lower than the case of random perturbation. From this, we can deduce that $\mathbf{h}_c^{\mathcal{I}}(\boldsymbol{w}^*_{fool})$ is highlighting parts that are less helpful than random pixels for making predictions, hence, is a "wrong" interpretation. We believe the notion of "good/bad/wrong" interpretation (Re:**[R2]**) can be defined by examining the AOPC of $\mathbf{h}_c^{\mathcal{I}}(\cdot)$ and that of random perturbation. Furthermore, the [Basic question] (Re:**[R1]**) says that [Ross *et.al*, 2017] can be contradicting to our result, but depending on the training objective and model capacity, we believe the two phenomenon can both exist.

Robustness of/detecting our fooling [R1,R3] Detecting or undoing our model manipulation would not be easy as we cannot easily access the original interpretation results or training data. Fig. (b) shows a feasible attempt, inspired by [Roth *et al.*, `arXiv:1902.04818`], fails for detecting our manipulation. Namely, as the adversarial input examples can be detected by adding small Gaussian perturbation to the input, we may also suspect that adding small Gaussian noise to the parameters might reveal our fooling. However, Fig. (b) shows that $\boldsymbol{w}_o$ and $\boldsymbol{w}^*_{fool}$ behave very similarly in terms Top-1 accuracy as we increase the Gaussian noise level, and FSR do not change radically, either. Another possible detection scheme would be to use a black-box interpreter and compare its AOPC with that of the fooled model.

[R1] ① We will modify the acronym for Simple Gradient and correct the error regarding LRP passing the sanity check. ② We found fooling SmoothGrad / Integrated Gradients while maintaining the accuracy is much harder, as the reviewer has expected. (LIME is a black-box interpreter and is out of scope.) We will discuss this result in our final version. ③ Fig. (c), which shows the accuracy of a ResNet50 model on $\mathcal{D}_{val}$ and PGD($\mathcal{D}_{val}$) (i.e., the PGD-attacked $\mathcal{D}_{val}$), demonstrates that adversarially trained model can be also adversarially manipulated by our method. Namely, starting from a pre-trained $\boldsymbol{w}_o$ (dashed red), we adversarially train the model with PGD attacks ($\epsilon = 1.5$) [Shafahi *et.al*, `arXiv:1904.12843`] to obtain $\boldsymbol{w}_{adv}$ (dashed green), then started our adversarial model manipulation with Location fooling with Grad-CAM. Note the Top-1 accuracy on $\mathcal{D}_{val}$ drops while that on PGD($\mathcal{D}_{val}$) increases during adversarial training phase, and they are maintained during our model manipulation phase (e.g., at blue dashed line). The right panel shows the Grad-CAM interpretations of two images at three distinct phases (see the color-coded boundaries), and we clearly see the success of the Location fooling (blue dashed, third row). ④ We plan to release the code after acceptance.

[R2] ① We will carry out more thorough investigation on active fooling in the future work. Moreover, LRP$_T$ is our novel variation of LRP, and the local characteristic of LRP$_T$ is maintained to LRP since the relevance values of LRP$_T$ are propagated down to the input pixel level. Directly fooling with LRP did not work well. ② We have randomly selected the visualization examples, and have not cherry-picked. ③ We argue the correlation metric also is not enough to accurately measure the success of fooling; e.g., in Location fooling, the mere correlation cannot tell whether the fooling was successful. Instead, we devised FSR to more directly measure the success of intended fooling. For computing FSR, we simply randomly selected 10K images from the ImageNet validation, and ℓ_2, a common measure of distance in 2-D images, is used in FSR for Center-mass fooling even though ℓ_1 and ℓ_2 give almost the same result. ④ For L248/L255 comments, please refer to the AOPC results. ⑤ We will emphasize the contrast between the random parametrization.

[R3] ① Our method does share some similarities with backdooring; e.g., manipulating model parameters to hide some intention while maintaining the accuracy. However, the setting/objective are radically different. Namely, backdooring has nothing to do with the *interpretation* of a model, and it could even be detected by the interpretation methods. But, with our model manipulation, we can even make the backdooring stronger to be not caught by the interpretation methods. ② To support the motivation of our work, we carried out additional experiment on the 'Adult income' classification data in [`UCI ML Repository`]. We trained a classifier with 8 conv layers, $\boldsymbol{w}_o$, and the LRP result in Fig. (d) (blue) shows it assigns high importance on *sensitive* features like 'Race' and 'Sex'. Now, we (or a lazy developer) can manipulate the model with Location fooling that zero-masks the two features and obtain $\boldsymbol{w}_{fool}$ that essentially has the same accuracy of 76.8% as $\boldsymbol{w}_o$ but with a new interpretation that disguises the bias (orange). Obviously, from our discussions on AOPC etc. above, the lazy developer has not resolved the bias in the model but simply fooled the interpretation result.