8. Appendix

8.1. Classification and Interpretation Robustness

Suppose the gradient based interpretation can be written as $g(x) = \nabla\ell(x)$, where ℓ can be the cross-entropy loss (or our defined prediction loss J). We leverage Lipschitz continuous gradient to hint the relation between classification robustness and interpretation robustness as what follows.

A differentiable function $\ell(x)$ is called smooth within local region $B(x, r)$ iff it has a Lipschitz continuous gradient, i.e., if $\exists K > 0$ such that

$$||\nabla\ell(x') - \nabla\ell(x)|| \leqslant K||x' - x||, \quad \forall x' \in B(x, r). \tag{17}$$

Proposition 1 *Lipschitz continuous gradient implies:*

$$||\ell(x') - \ell(x)|| \leqslant ||\nabla\ell(x)||r + \frac{K}{2}r^2 \tag{18}$$

Prop. 1 says, the change of classification is bounded by input gradient $||\nabla\ell(x)||$, as well as $\frac{K}{2}$. K can be chosen as the Frobenius norm of input hessian $||H||_F(x)$ [16]. Therefore, the regularisation of input gradient and input hessian can affect classification robustness and interpretation robustness.

Proof. We first show that for $K > 0$, $||\nabla\ell(x_1) - \nabla\ell(x_2)|| \leqslant K||x_1 - x_2||$ implies

$$\ell(x_1) - \ell(x_2) \leqslant \nabla\ell(x_2)^T(x_1 - x_2) + \frac{K}{2}||x_1 - x_2||^2$$

Recall from the integral calculus $\ell(a) - \ell(b) = \int_b^a \nabla\ell(\theta)\, d\theta$,

$$\ell(x_1) - \ell(x_2) = \int_0^1 \nabla\ell(x_2 + \tau(x_1 - x_2))^T(x_1 - x_2)\, d\tau = \int_0^1 (\nabla\ell(x_2 + \tau(x_1 - x_2))^T - \nabla\ell(x_2)^T + \nabla\ell(x_2)^T)(x_1 - x_2)\, d\tau$$

As $\nabla\ell(x_2)$ is independent of τ, it can be taken out from the integral

$$\ell(x_1) - \ell(x_2) = \nabla\ell(x_2)^T(x_1 - x_2) + \int_0^1 (\nabla\ell(x_2 + \tau(x_1 - x_2))^T - \nabla\ell(x_2)^T)(x_1 - x_2)\, d\tau$$

Then we move $\nabla\ell(x_2)^T(x_1 - x_2)$ to the left and get the absolute value

$$|\ell(x_1) - \ell(x_2) - \nabla\ell(x_2)^T(x_1 - x_2)| = |\int_0^1 (\nabla\ell(x_2 + \tau(x_1 - x_2))^T - \nabla\ell(x_2)^T)(x_1 - x_2)\, d\tau| \leqslant \int_0^1 |(\nabla\ell(x_2 + \tau(x_1 - x_2))^T - \nabla\ell(x_2)^T)(x_1 - x_2)|\, d\tau \leqslant_{c.s.} \int_0^1 ||(\nabla\ell(x_2 + \tau(x_1 - x_2)) - \nabla\ell(x_2))||||(x_1 - x_2)||\, d\tau$$

c.s. means Cauchy – Schwarz inequality. By applying lipschitz continuous gradient, we can get

$$\begin{aligned} ||(\nabla\ell(x_2 + \tau(x_1 - x_2)) - \nabla\ell(x_2))|| &\leqslant K||\tau(x_1 - x_2)|| \\ &\leqslant K\tau||x_1 - x_2|| \end{aligned}$$

Note $\tau \geqslant 0$, and the absolute sign of τ can be removed. Then, we can get

$$|\ell(x_1) - \ell(x_2) - \nabla\ell(x_2)^T(x_1 - x_2)| \leqslant \int_0^1 K\tau||x_1 - x_2||^2\, d\tau = \frac{K}{2}||x_1 - x_2||^2$$

Next, get the norm of two sides, and apply triangle inequality, we finally get

$$\begin{aligned} ||\ell(x') - \ell(x)|| &\leqslant ||\nabla\ell(x)^T(x' - x) + \frac{K}{2}||x' - x||^2|| \\ &\leqslant ||\nabla\ell(x)||||x' - x|| + \frac{K}{2}||x' - x||^2 \\ &\leqslant ||\nabla\ell(x)||r + \frac{K}{2}r^2 \end{aligned} \tag{19}$$

QED

8.2. Genetic Algorithm based Optimisation

Genetic Algorithm (GA) is a classic evolutionary algorithm for solving the either constrained or unconstrained optimisation problems. It mimics the biological evolution by selecting the most fitted individuals in the population, which will be the parents for the next generation. It consists of 4 steps: initialisation, selection, crossover, and mutation, the last three of which are repeated until the convergence of fitness values.

Initialisation The initialisation of population is crucial to the quick convergence. Diversity of initial population could promise approximate global optimal[27]. Normally, we use the Gaussian distribution with the mean at input seed x, or a uniform distribution to generate a set of diverse perturbed inputs within the norm ball $B(x, r)$.

Selection A fitness function is defined to select fitted individuals as parents for the latter operations. We use the fitness proportionate selection [28].

$$p_i = \frac{\mathcal{F}_i}{\sum_{i=1}^{n} \mathcal{F}_i} \tag{20}$$

The fitness value is used to associate a probability of selection p_i for each individuals to maintaining good diversity of population and avoid premature convergence. The fitness function is the objective function to be optimised. For example, previous paper applies GA to the perturbation optimisation to generate the high quality AEs [13]. In this paper, the explanation discrepancy is optimised to find the worst case adversarial explanations.

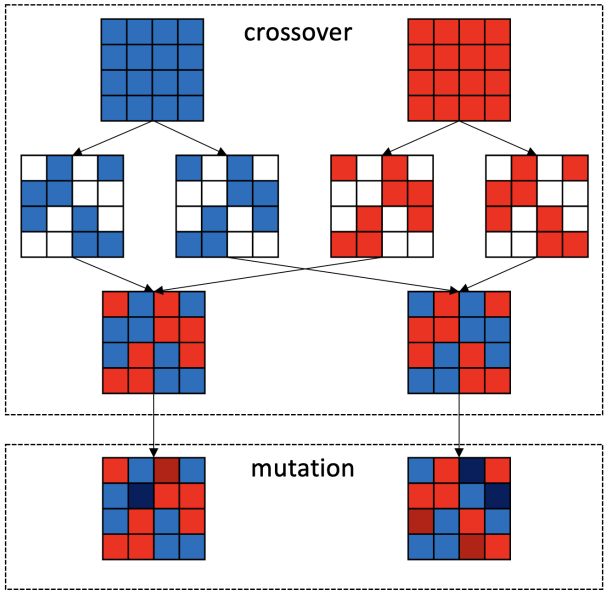

Figure 5: Illustration of crossover and mutation in GA

Crossover The crossover operator will combine a pair of parents from last step to generate a pair of children, which share many of the characteristics from the parents. The half elements of parents are randomly exchanged.

Mutation Some elements of children are randomly altered to add variance in the evolution. It should be noticed that the mutated samples should still fall into the norm ball $B(x, r)$. Finally, the children and parents will be the individuals for the next generation.

Termination The termination condition of GA is either maximum number of iterations is reached or the highest ranking of fitness reaches a plateau such that successive iterations no longer produce better results. In this paper, we fix the maximum iteration number for simplicity.

GA can be directly applied to the unconstrained optimisation when objective function equals to fitness function. The constraint optimisation is more challenging and different strategies are proposed to handle the non-linear constraint for GA [30]. One of the popular approaches is based on the superiority of feasible individuals to make distinction between feasible and infeasible solutions [34].

8.3. Subset Simulation

Subset Simulation (SS) is widely used in reliability engineering to compute the small failure probability. The main idea of SS is introducing intermediate failure events so that the failure probability can be expressed as the product of larger conditional failure probabilities [6].

Suppose the distribution of perturbed inputs with the norm ball is $q(x)$, and the failure event is denoted as F. let $F = F_m \subset F_{m-1} \subset \cdots \subset F_2 \subset F_1$ be a sequence of increasing events so that $F_m = \bigcap_{i=1}^{m} F_i$. By the definition of conditional probability, we get

$$\begin{aligned} P_F &= P(F_m) = P(\bigcap_{i=1}^{m} F_i) \\ &= P(F_m | \bigcap_{i=1}^{m-1} F_i) P(\bigcap_{i=1}^{m-1} F_i) \\ &= P(F_m|F_{m-1}) P(\bigcap_{i=1}^{m-1} F_i) \\ &= P(F_m|F_{m-1}) \cdots P(F_2|F_1) P(F_1) \\ &= P(F_1) \prod_{i=2}^{m} P(F_i|F_{i-1}) \end{aligned} \tag{21}$$

F_m is usually a rare event, which means a large amount of samples are required for the precise estimation by Simple Monte Carlo (SMC). SS decomposes the rare event with a series of intermediate events, which are more frequent. The conditional probabilities of intermediate events involved in Eq. (11) can be chosen sufficiently large so that they can be efficiently estimated. For example, $P(F_1) = 1$, $P(F_i|F_{i-1}) = 0.1$, $i = 2, 3, 4, 5, 6$, then $P_F \approx 10^{-5}$ is too small for the efficient estimation by SMC.

The keypoint of SS is estimating $P(F_1)$ and conditional probabilities $P(F_i|F_{i-1})$. On the one hand, F_1 can be chosen as the common event such that by SMC of N perturbed inputs within the norm ball $x'_k \sim q(x')$, all samples fall into F_1. On the other hand, computing the conditional probability

$$P(F_{i+1}|F_i) = \frac{1}{N} \sum_{k=1}^{N} \mathbb{1}_{F_{i+1}}(x'_k) \approx \rho \tag{22}$$

requires the simulation of $(1-\rho)N$ additional samples. For example, if we have N samples belonging to F_{i-1} with $i \geqslant 2$, and $P(F_i|F_{i-1}) = \rho$, which indicate ρN samples belongs to F_i. To estimate next conditional probability $P(F_{i+1}|F_i)$, $(1-\rho)N$ additional samples lying in F_i should be simulated to expand the population size to N. Given the conditional distribution $q(x'|F_i) = q(x')I_{F_1}(x')/P(F_i)$, on average $1/P(F_i)$ samples are simulated before one such sample occur. The Markov Chain Monte Carlo based on Metropolis-Hastings (MH) algorithm can be adopted to improve the efficiency.

At intermediate iteration i, we already obtain ρN samples lying in F_i, that is $x' \in F_i$. The target distribution is $q(\cdot|F_i)$. We can use MH algorithm to generate new samples x'' from the proposal distribution $g(x''|x')$. $g(x''|x')$ can be normal distribution or uniform distribution centred at x'. The MH algorithm can be written as below:

8.3.1 Initialisation

Pick up a sample x' belonging to F_i. Set step $t = 0$ and let $x_t = x'$.

8.3.2 Iteration

At step t, generate a random candidate sample x'' according to $g(x''|x_t)$. Calculate the acceptance probability

$$A(x'', x_t) = \min\{1, \frac{q(x_t|F_i)}{q(x''|F_i)} \frac{g(x_t|x'')}{g(x''|x_t)}\} \tag{23}$$

and accept the new sample x'' with probability $A(x'', x_t)$. Further check if $x'' \in F_i$, otherwise reject x''. In practice, we generate a uniform random number $u \in [0, 1]$, set x_{t+1} as

$$x_{t+1} = \begin{cases} x'' & \text{if } u \leqslant A(x'', x_t) \text{ and } x'' \in F_i \\ x_t & \text{Otherwise} \end{cases} \tag{24}$$

and increment $t = t + 1$.

We can run a large amount of Markov chains simultaneously to enlarge the set of i.i.d. samples falling into F_i. However, as discussed in [24, 36], MH becomes inefficient for high dimensional problems. The acceptance probability $A(x'', x')$ will rapidly decrease with increasing dimensions. It results in many repeated samples and high correlated Markov chains. It is recommended to adapt the proposal distribution $g(x''|x')$ after M steps of MH [33]. The mean acceptance probability should be kept around 0.234 [18].

The whole process of SS can be summarized as follows. First, we simulate N perturbed samples within the norm ball $B(x, r)$ (all belong to F_1) and use SMC to estimate $P(F_2|F_1)$. From these N samples, we already obtain ρN samples distributed from $q(\cdot|F_2)$. Start from each of these ρN samples falling in F_2, we can create a Markov chain and run MH M steps to generate new samples distributed from $q(\cdot|F_2)$. In initial SS method [6], ρN distinct Markov chains (with different start points) are created. $1/\rho$ new samples are drawn from each chain, and the covariance between new samples in same Markov chain should be considered for evaluating the coefficient of variation (c.o.v) of the final estimation on P_F. [11] modify the algorithm by firstly enlarge set to N samples with replacement from ρN. Then N Markov Chains are constructed and only one sample is drawn from each chain.

These new generated samples can be utilised to estimate $P(F_3|F_2)$. Repeating this process until the rare failure of interest. We get the final estimation of failure event probability by "assembling" the conditional probabilities with Eq. (11).

8.3.3 Statistical Property of SS Estimator

We present the analysis on statistical property of P_{F_i} (shortened notation for $P(F_1)$ and $P(F_i|F_{i-1})$) and P_F. They are based on the assumption that Markov chain generated by MH algorithm is theoretically ergodic. That is, the stationary distribution is unique and tend to the corresponding conditional probability distribution. Since we simulate samples from Markov chain to estimate P_{F_i} (ref. to Eq. (22)), The coefficient of variation of P_{F_i} (c.o.v) is

$$\delta_i = \sqrt{\frac{1 - P_{F_i}}{P_{F_i} N}(1 + \lambda_i)} \tag{25}$$

$\lambda_i > 0$ represents the dependency of samples drawn from Markov Chain. This is compared to case when we use SMC to simulate independent samples from the known distribution ($\lambda_i = 0$). As $N \to \infty$, the Central Limit Theorem (CLT) tells $\bar{P}_{F_1} \to P(F_1)$, and $\bar{P}_{F_i} \to P(F_i|F_{i-1})$. We can get almost surely $\bar{P}_F \to P(F_1) \prod_{i=2}^{m} P(F_i|F_{i-1}) = P_F$. It should be noted that $\bar{P}_F$ is biased for N, but asymptotically unbiased due to the fact that samples in F_i for computing $\bar{P}_{F_i}$ are utilised to start Markov chain for computing $\bar{P}_{F_{i+1}}$. This bias will asymptotically vanish when N goes to infinity.

Proposition 2 *$\bar{P}_F$ is biased for N, the fractional bias is bounded by:*

$$|E\left[\frac{\bar{P}_F - P_F}{P_F}\right]| \leqslant \sum_{i>j} \delta_i \delta_j + o(1/N) = O(1/N) \tag{26}$$

Proof. We define $Z_i = (\bar{P}_{F_i} - P_{F_i})/\sigma_i$, and get $\bar{P}_{F_i} = P_{F_i} + \sigma_i Z_i$. By CLT, it's clear that $E[Z_i] = 0$ and $E[Z_i^2] =$

1.

$$\begin{aligned}\frac{\bar{P}_F - P_F}{P_F} &= \prod_{i=1}^{m} \bar{P}_{F_i}/P_{F_i} - 1 \\ &= \prod_{i=1}^{m}(1+\delta_i Z_i) - 1 \\ &= \prod_{i=1}^{m}\delta_i Z_i + \sum_{i=1}^{m}\delta_i Z_i + \sum_{i>j}\delta_i\delta_j Z_i Z_j + \\ &\quad \sum_{i>j>k}\delta_i\delta_j\delta_k Z_i Z_j Z_k + ...\end{aligned}$$

Take expectation and use $E[Z_i] = 0$, we can further get

$$\begin{aligned}E\left[\frac{\bar{P}_F - P_F}{P_F}\right] = \left(\prod_{i=1}^{m}\delta_i\right) E\left[\prod_{i=1}^{m} Z_i\right] + \sum_{i>j}\delta_i\delta_j E[Z_i Z_j] \\ + \sum_{i>j>k}\delta_i\delta_j\delta_k E[Z_i Z_j Z_k] + ...\end{aligned}$$

Since $\{Z_i\}$ are correlated, $E[Z_i Z_j]$, $E[Z_i Z_j Z_k]$,.... are not zero, and $\bar{P}_{F_i}$ is biased for every N. δ_i is $O(1/\sqrt{N})$ according to the definition, which makes $\sum_{i>j}\delta_i\delta_j E[Z_i Z_j]$ have $O(1/N)$ and remaining items with higher product of δ_i have $o(1/N)$. Take absolute value of both sides and use Cauchy-Schwartz inequality to obtain $|E[Z_i Z_j]| \leqslant \sqrt{E[Z_i^2]E[Z_j^2]} = 1$. Finally, we can get the proof.

Proposition 3 *$\bar{P}_F$ is a consistent estimator and its c.o.v. δ is bounded by:*

$$\delta^2 = E\left[\frac{\bar{P}_F - P_F}{P_F}\right]^2 \leqslant \sum_{i,j=1}\delta_i\delta_j + o(1/N) = O(1/N) \tag{27}$$

Proof.

$$\begin{aligned}&E\left[\frac{\bar{P}_F - P_F}{P_F}\right]^2 \\ &= E\left[\prod_{i=1}^{m}\delta_i Z_i + \sum_{i=1}^{m}\delta_i Z_i + \sum_{i>j}\delta_i\delta_j Z_i Z_j + ...\right]^2 \\ &= \sum_{i,j=1}^{m}\delta_i\delta_j E[Z_i Z_j] + o(1/N) \\ &\leqslant \sum_{i,j=1}^{m}\delta_i\delta_j + o(1/N) = O(1/N)\end{aligned}$$

As $\delta_i = O(1/\sqrt{N})$ and $E[Z_i Z_j] \leqslant 1$. Note that the bias is accounted for when c.o.v. δ is defined as the deviation about P_F, instead of $E[\bar{P}_F]$. The upper bound corresponds to the case that conditional probability $\{P_{F_i}\}$ are all correlated. Although $\{P_{F_i}\}$ are generally correlated, δ can be well approximated by $\sum_{i=1}^{m}\delta_i^2$. For simplicity, we can also make the assumption that enough steps of MH algorithm are taken to eliminate the dependency of simulated samples from MCMC ($\lambda_i = 0$) [11]. Then we use sample mean $\bar{P}_{F_i}$ to approximate P_{F_i}, and finally get

$$\bar{\delta}^2 \approx \sum_{i=1}^{m}\delta_i^2 = \sum_{i=1}^{m}\frac{1-\bar{P}_{F_i}}{\bar{P}_{F_i}N}(1+\lambda_i) \approx \sum_{i=1}^{m}\frac{1-\bar{P}_{F_i}}{\bar{P}_{F_i}N} \tag{28}$$

To get an idea of how many samples are required by SS to achieve the estimation accuracy P_F, we assume the c.o.v δ, $\lambda_i = \lambda$ and $P(F_i|F_{i-1}) = \rho$ are fixed, then $m = logP_F/log\rho + 1$, and $\delta^2 = (m-1)\frac{1-\rho}{\rho N}(1+\lambda)$, We can get the number of simulated samples in SS is

$$N_{SS} \approx mN = (\frac{|logP_F|^2}{|log\rho|^2} + \frac{|logP_F|}{|log\rho|})\frac{(1-\rho)(1+\lambda)}{N\delta^2}$$

Thus, for a fixed δ and ρ, $N_{SS} \propto (|logP_F|^2 + |log\rho||logP_F|)$. Compared to the SMC, the required samples are $N_{SMC} \propto 1/P_F$. This indicates that SS is substantially efficient to estimate small failure probability.

8.4. Complexity Analysis of Genetic Algorithm and Subset Simulation Applied on XAI Methods

Although the proposed evaluation methods can be applied to all kinds of feature attribution based XAI techniques, the time complexity will be extremely high for perturbation based XAI methods, such as LIME and SHAP, which take random perturbation of input features to yield explanations.

The complexity of GA is $O(t \cdot N \cdot (c(fitness) + c(crossover) + c(mutation)))$, where t and N are evolution iterations and population size, respectively. When we choose different XAI methods, the evaluation time of fitness values $c(fitness)$ will change correspondingly.

The complexity of SS is related to the number of sub-events m, the number of MH steps M and number of simulated samples N. For estimating conditional probability of each sub-event, M MH steps are taken, and running each MH step requires the calculation of property function of N samples. Thus, the complexity of SS is approximately $O(m \cdot M \cdot N \cdot c(property))$. When we choose different XAI methods, the evaluation time of property function $c(property)$ will change correspondingly.

Table 4: Time counts of $N \cdot c(cal\_attr\_dis)$ in seconds across different dataset ($N = 1000$). Results are averaged over 10 runs.

Dataset	Gradient x Input	Integrated Gradients	GradCAM	DeepLift	LIME	SHAP
MNIST	0.0202	0.0512	0.0342	0.0382	99.21	25.80
CIFAR-10	0.0909	0.3329	0.1222	0.1307	293.72	255.95
CelebA	0.0620	0.2759	0.0887	0.1029	739.59	692.75

From the definition of fitness function in GA and property function in SS. both $c(fitness)$ and $c(property)$ can be approximated by the computation of interpretation discrepancy $c(cal\_attr\_dis)$. In practice, we can compute interpretation discrepancy in a batch, e.g. N samples can run simultaneously to generate the explanations. Therefore, we count the running time of $N \cdot c(cal\_attr\_dis)$ across different datasets and different XAI methods in Nvidia A100. Results are presented in Table 4. LIME and SHAP take much more time than gradient-based XAI methods for the batch computation of interpretation discrepancy. This will be amplified by iteration number t in GA or number of sub-events times number of MH steps $m \cdot M$ in SS for one time evaluation of interpretation robustness.

8.5. Details of DL models

The information of DL models under evaluation are presented in Table 5. All experiments were run on a machine of Ubuntu 18.04.5 LTS x86_64 with Nvidia A100 GPU and 40G RAM. The source code, DL models, datasets and all experiment results are available in Supplementary Material, and will be publicly accessible at GitHub after the double-blind review process.

8.6. Experiment on Interpretation Discrepancy Measures

We study the quality of three widely used metrics, i.e. Mean Square Error (MSE), Pearson Correlation Coefficient (PCC), and Structural Similarity Index Measure (SSIM) [15] to quantify the visual discrepancy between two attribution maps. The proposed evaluation methods can produce the adversarial interpretation with the guidance of different metrics. As shown in Fig. 6, the first row displays three seed inputs and corresponding attribution maps. The following groups separated by lines show the adversarial interpretation of perturbed input measured by different metrics. The value of PCC appears to be relatively more accurate in terms of reflecting the visual difference between original interpretation of seeds input and adversarial interpretations. Smaller PCC represents larger visual difference between two attribution maps. In addition, the value range of PCC is 0$\sim$1, with 0$\sim$0.3 indicating weak association, 0.5$\sim$1.0 indicating strong association. Therefore, it provides a uniform measurement across different seeds input and different dataset. In contrast, MSE can also precisely measure the visual difference but vary greatly with respect to seed inputs and image size. SSIM exhibits the worst performance in measuring difference between attribution maps.

8.7. Experiment on Parameter Sensitivity

Additional experiments on hyper-parameter settings in GA and SS are presented in Fig. 7 and Fig. 8. The objective function interpretation discrepancy $\mathfrak{D}$, measured by PCC, is optimised to converge with the increasing number of iterations while the prediction loss J as the constraint is gradually satisfied. The number of iterations in GA is more important than population size.

For hyper-parameters in SS, apart from the sensitivity of MH steps, we also discuss the impact of population size n and quantile ρ for conditional probability. As expected, increasing population size will improve the estimation precision, using SMC results with 10^8 samples as the ground truth. However, there is no exact answer for which ρ is better. In most cases, we find that $\rho = 0.5$ can reduce the estimation error, but will take more time for one estimation. Larger ρ represents more sub events are decomposed and additional estimation of conditional probability will obviously cost more time. Fortunately, we find SS estimation accuracy is more sensitive to the number of MH steps M and population size n, compared with ρ. Therefore, setting $\rho = 0.1$ but increasing MH steps and population size will get sufficiently accurate results. Finally, the rarity of failure events can determine the setting of these hyper-parameters. The estimating accuracy of more rare events, e.g. PCC < 0.2, is more sensitive to the theses parameters.

8.8. Experiments on Evaluating XAI methods

8.8.1 Evaluation for Gradient-based XAI Methods

We evaluate the robustness of more XAI methods on CIFAR10 and CelebA dataset, including "Deconvolution", "Guided Backpropagation", "Gradient$\times$Input", "Integrated Gradients", "GradCAM", and "DeepLift". Results are presented in Fig. 9. In terms of misinterpretation with preserved classification, Integrated Gradients is the most robust XAI method due to the integral of gradient of model's output with respect to the input. The integral averages the gradient-based attribution maps over several perturbed images instead of single point explanation. DeepLift has the similar smoothing mechanism by comparing the neuron activation with a reference point. Therefore, single point explanation like Deconvolution and GradCAM are vulnerable to this type of misinterpretation when DL model's loss surface is highly curved, leading to the great change of gradients. Gradient$\times$Input is slightly better by leveraging the input sign and strength.

These XAI methods in general show similar robustness against misinterpretation conditioned on misclassification, although we find the single point explanation is a litter better than explanation averaged over several points under this circumstance. We guess the rarity of misclassification and misinterpretation make it difficult to find the perturbed input which have different attribution map with input seeds. Therefore, the averaged interpretation of perturbed input tend to be consistent with original interpretation.

Table 5: Details of the datasets and DL models under evaluation.

Dataset	Image Size	r	DL Model	Org.		Grad. Reg.		Hess. Reg.		Adv. Train.	
				Train	Test	Train	Test	Train	Test	Train	Test
MNIST	$1 \times 32 \times 32$	0.1	LeNet5	1.000	0.991	0.993	0.989	0.993	0.989	0.994	0.989
CIFAR-10	$3 \times 32 \times 32$	0.03	ResNet20	0.927	0.878	0.910	0.876	0.786	0.779	0.715	0.703
CelebA	$3 \times 64 \times 64$	0.05	MobileNetV1	0.934	0.917	0.918	0.912	0.908	0.904	0.769	0.789

8.8.2 Evaluation for Perturbation-based XAI Methods

We also consider the robustness of interpretation for LIME and SHAP, the most popular perturbation-based XAI methods. In contrast to the gradient-based XAI methods, the robustness problem of which is thoroughly studied, perturbation-based XAI methods are difficult to be attacked by adversarial noise due to the model-agnostic settings. As far as we have known, the only adversarial attack on LIME/SHAP [39] requires to scaffold the biased DL model. That's conceptually different from the interpretation robustness mentioned in this paper, for which the internal structure of DL model should not be maliciously modified. Thanks to the black-box nature of our evaluation approaches, we can assess the robustness of LIME/SHAP. As is known, image feature segmentation is an important procedure in LIME/SHAP. LIME/SHAP will produce inconsistent interpretation at each run when the number of samples is smaller than the number of image segments [53]. Therefore, we record the evaluation results when using different number of samples. For simplicity, we use quickshift to segment the images into around 40 pieces of super-pixels, which is the default settings of LIME/SHAP tools.

Table 6: Robustness evaluation of perturbation-based XAI methods.

Dataset	XAI Method + Num_Samples	Worst Case Evaluation		Probabilistic Evaluation	
		$sol_{\hat{F}}$ (PCC)	$sol_{\tilde{F}}$ (PCC)	$\ln P_{\hat{F}}$	$\ln P_{\tilde{F}}$
MNIST	LIME+50	0.0002	0.9886	-0.46	-12.96
	LIME+200	6.88e-05	0.9350	-0.37	-14.59
	LIME+500	8.59e-06	0.8360	-0.31	-16.98
	SHAP+50	4.11e-05	0.9648	-0.36	-14.78
	SHAP+200	0.0011	0.9708	-0.39	-14.44
	SHAP+500	0.0005	0.9851	-0.34	-14.41
CIFAR-10	LIME+50	0.0002	0.9940	-3.58	-28.96
	LIME+200	0.0001	0.9986	-3.78	-30.28
	LIME+500	0.0001	0.9965	-4.29	-40.06
	SHAP+50	0.0014	0.9973	-3.75	-48.56
	SHAP+200	0.0016	0.9950	-3.94	-47.87
	SHAP+500	0.0001	0.9982	-3.84	-46.24
CelebA	LIME+50	0.0004	0.9571	-1.17	-39.63
	LIME+200	1.23e-05	0.9824	-4.06	-41.41
	LIME+500	0.0001	0.9739	-5.53	-48.55
	SHAP+50	0.0008	0.9568	-4.24	-49.21
	SHAP+200	0.0006	0.9520	-4.97	-50.69
	SHAP+500	0.0002	0.9543	-4.41	-58.18

The initial results in Table 6 give us the hints that perturbation-based XAI methods also suffer from the lack of interpretation robustness, especially when classification is preserved but interpretation is different. In addition, increasing the number of perturbed samples is not significant to improving interpretation robustness. In other words, even if we use enough number of perturbed samples for LIME/SHAP to produce precise interpretation results, they are still easily fooled by adversarial noise. In the second experiment, we further explore the influence of image segmentation on interpretation robustness. By making the assumption that image segmentation is fixed or not fixed after adding adversarial noise, we can check whether adversarial noise change the image segmentation and indirectly affect the interpretation robustness of perturbation-based XAI methods. Result in Table 7 shows that current image segmentation used by LIME/SHAP is sensitive to the pixel-level adversarial noise and will produce different feature masks, which may affect the interpretation robustness. Nevertheless, fixing image segmentation is not effective to defend second type of misinterpretation-wrong classification with persevered interpretation.

Table 7: Sensitivity of Image Segmentation to adversarial noise when evaluating interpretation robustness for LIME+200.

Dataset	Image Segmentation	Worst Case Evaluation		Probabilistic Evaluation	
		$sol_{\hat{F}}$ (PCC)	$sol_{\tilde{F}}$ (PCC)	$\ln P_{\hat{F}}$	$\ln P_{\tilde{F}}$
MNIST	Not Fixed	6.88e-05	0.9350	-0.37	-14.59
	Fixed	0.3632	0.8892	-34.22	-17.38
CIFAR-10	Not Fixed	0.0001	0.9986	-3.78	-30.28
	Fixed	0.0004	1.0000	-100	-41.33
CelebA	Not Fixed	1.23e-05	0.9824	-4.06	-41.41
	Fixed	0.3547	0.8289	-100	-38.72

The above observations align with the insight that interpretation robustness is attributed to the geometrical properties of DL model (i.e. large curvature of loss function), but not the XAI methods. Therefore, the most effective way to address the problem is to train a DL model, which is more robust to be interpreted.

8.8.3 Evaluation on Different NN Architectures

Apart from evaluation on different datasets, we do experiments on different neural network architectures for CIFAR10 dataset. Results in Table 8 shows that Integrated Gradients maintain the most robust XAI method to misinterpretation with preserved classification, invariant to the change of neural network architecture. However, the robustness to misinterpretation conditioned on misclassification

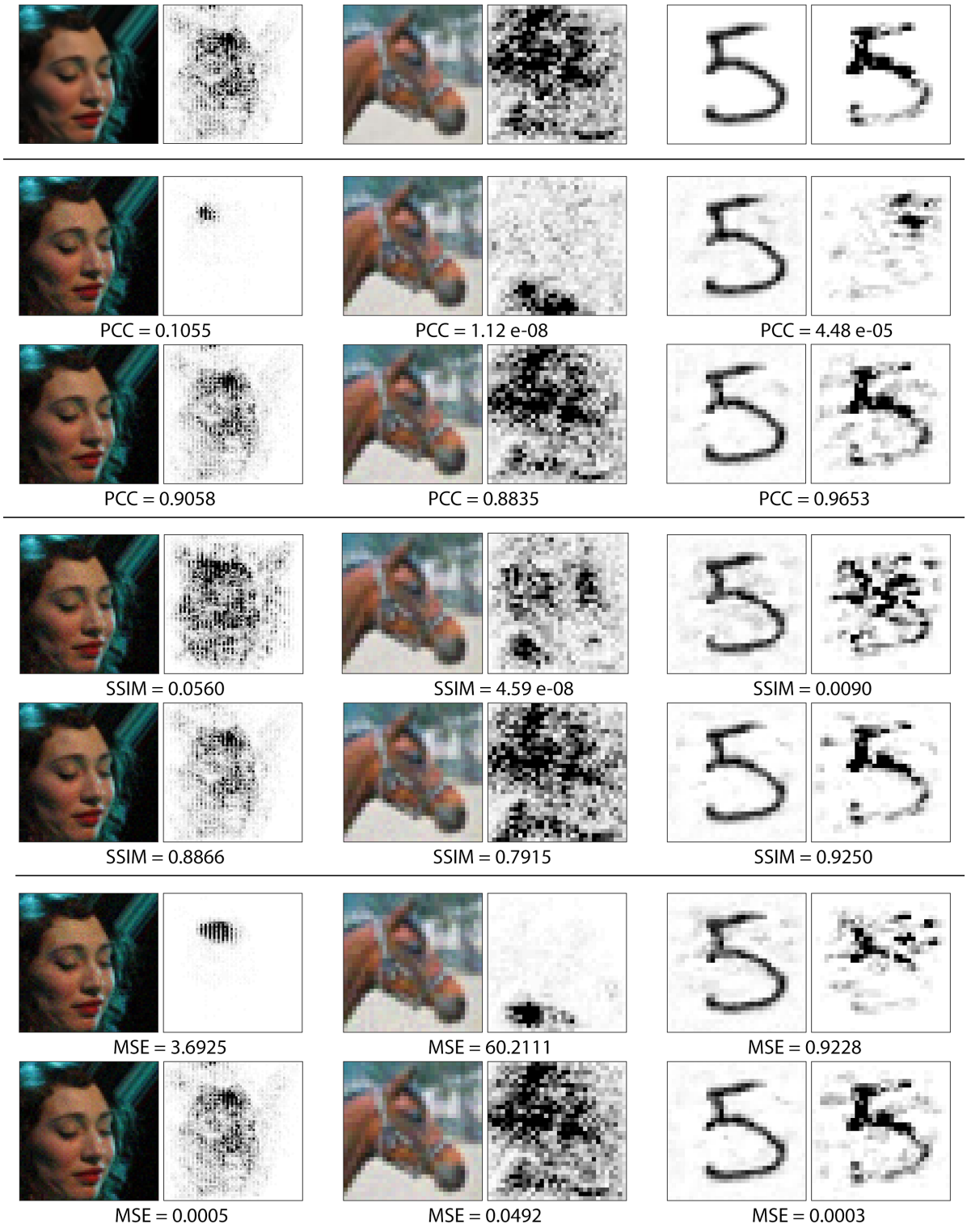

Figure 6: Comparison between PCC, SSIM and MSE as metrics of interpretation discrepancy between original interpretation and adversarial interpretation, generated by GA and SS. Smaller PCC, smaller SSIM, and larger MSE indicate greater difference. In this set of experiments, PCC is relatively the best to quantify the visual difference between attribution maps.

Table 8: Robustness evaluation of XAI methods on different neural network architecture for CIFAR-10 dataset.

Model Architecture	Eval Metrics	Gradient x Input	Integrated Gradients	GradCAM	DeepLift
ResNet20	$sol_{\widehat{F}}$	0.0166	**0.0375**	0.0044	0.0212
	$sol_{\widetilde{F}}$	0.8562	0.8308	**0.8079**	0.8551
	$\ln P_{\widehat{F}}$	-20.32	**-45.05**	-35.93	-21.22
	$\ln P_{\widetilde{F}}$	-80.73	**-87.64**	-68.27	-81.81
MobileNetV2	$sol_{\widehat{F}}$	0.0552	**0.1167**	0.0523	0.0712
	$sol_{\widetilde{F}}$	0.7689	0.7885	**0.7085**	0.7707
	$\ln P_{\widehat{F}}$	-12.75	**-34.99**	-16.01	-8.70
	$\ln P_{\widetilde{F}}$	-70.32	-62.19	**-82.17**	-68.38
VGG16	$sol_{\widehat{F}}$	0.0767	**0.1227**	0.1133	0.0206
	$sol_{\widetilde{F}}$	**0.7813**	0.8240	0.8637	0.8358
	$\ln P_{\widehat{F}}$	-14.42	**-53.48**	-47.52	-44.25
	$\ln P_{\widetilde{F}}$	-59.74	-54.155	-49.90	**-66.02**
DLA	$sol_{\widehat{F}}$	0.0737	**0.0953**	0.0078	0.0930
	$sol_{\widetilde{F}}$	0.7919	0.8111	**0.2113**	0.7983
	$\ln P_{\widehat{F}}$	-8.48	**-28.69**	-4.31	-9.77
	$\ln P_{\widetilde{F}}$	-39.57	-37.74	**-77.57**	-36.40

varies according to the internal structure of neural network. GradCAM seems to be robust in most cases.

8.8.4 Evaluation for Real-world Models

Table 9: Robustness evaluation for Wide ResNet-50-2 model trained on ImageNet dataset. Results are averaged over 20 samples.

XAI Methods	Worst Case Evaluation		Probabilistic Evaluation	
	$sol_{\widehat{F}}$ PCC	$sol_{\widetilde{F}}$ (PCC)	$\ln P_{\widehat{F}}$	$\ln P_{\widetilde{F}}$
Gradient x Input	0.159	0.463	-4.595	-100
Integrated Gradients	0.191	0.515	-39.235	-100
GradCAM	0.233	0.944	-98.725	-76.688
FullGrad	0.315	0.799	-100	-75.716
Extremal Perturbations	0.126	0.957	-4.321	-32.612
Accuracy	Top-1: 81.60%		Top-5: 95.76%	

We add additional experiments on wide_ResNet50_2 model trained on ImageNet-1K dataset in Table 9. We discover that FullGrad aggregates layer-wise gradient maps and thus combine the advantages of Gradient x Input and GradCAM. Extremal Perturbations seek to find the region of an input image that maximally excites a certain output, which is not robust to the adversarial perturbation.

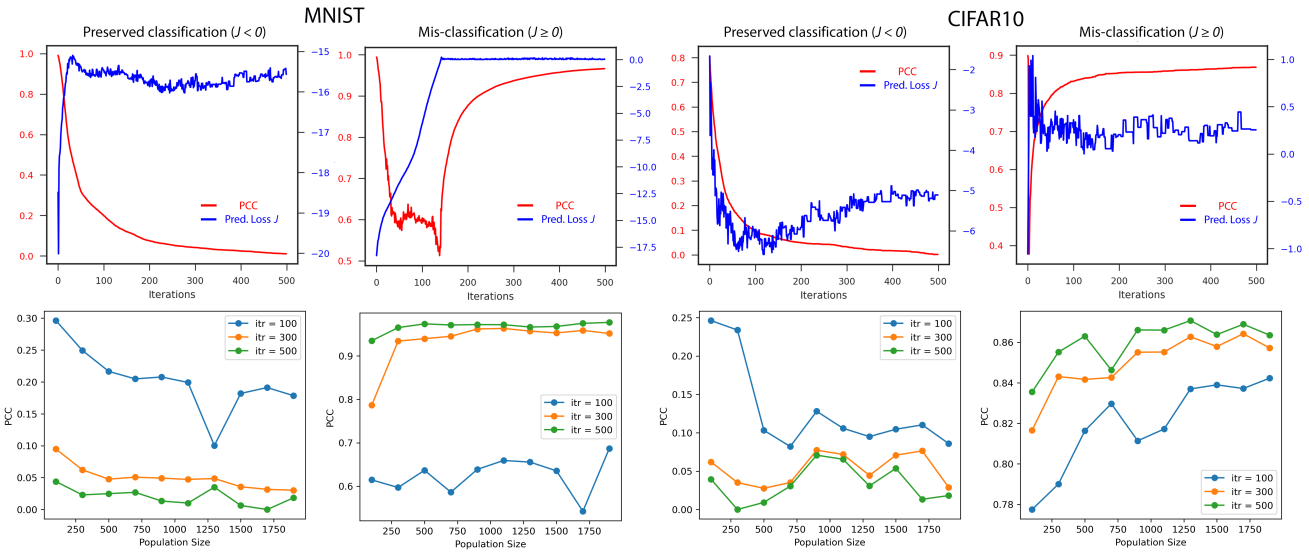

Figure 7: GA is applied to test seeds (norm balls) from MNIST and CIFAR10 dataset to find worst case interpretation discrepancy, measure by PCC. First row: fixed population size 1000, and varied iterations; Second row: fixed iterations, and varied population size. "Gradient×Input" interpretation method is considered.

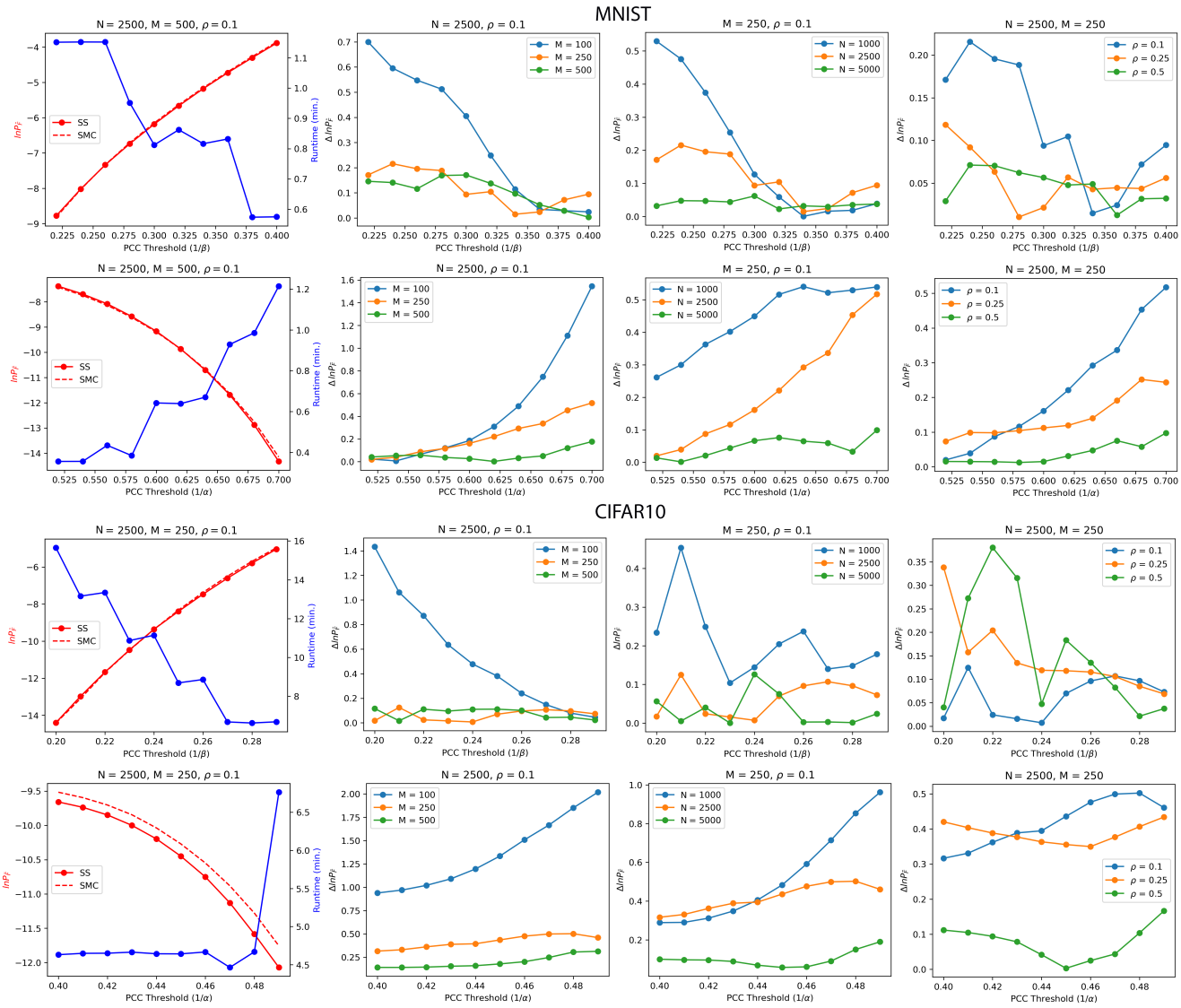

Figure 8: SS for estimating the probability of misinterpretation ($\ln P_F$) within a norm ball from MNIST, CIFAR10 dataset compared with SMC using 10^8 samples (22 minutes for each estimate for MNIST; 154 minutes for each estimate for CIFAR10). Results are averaged on 10 runs. "Gradient×Input" interpretation method is considered.

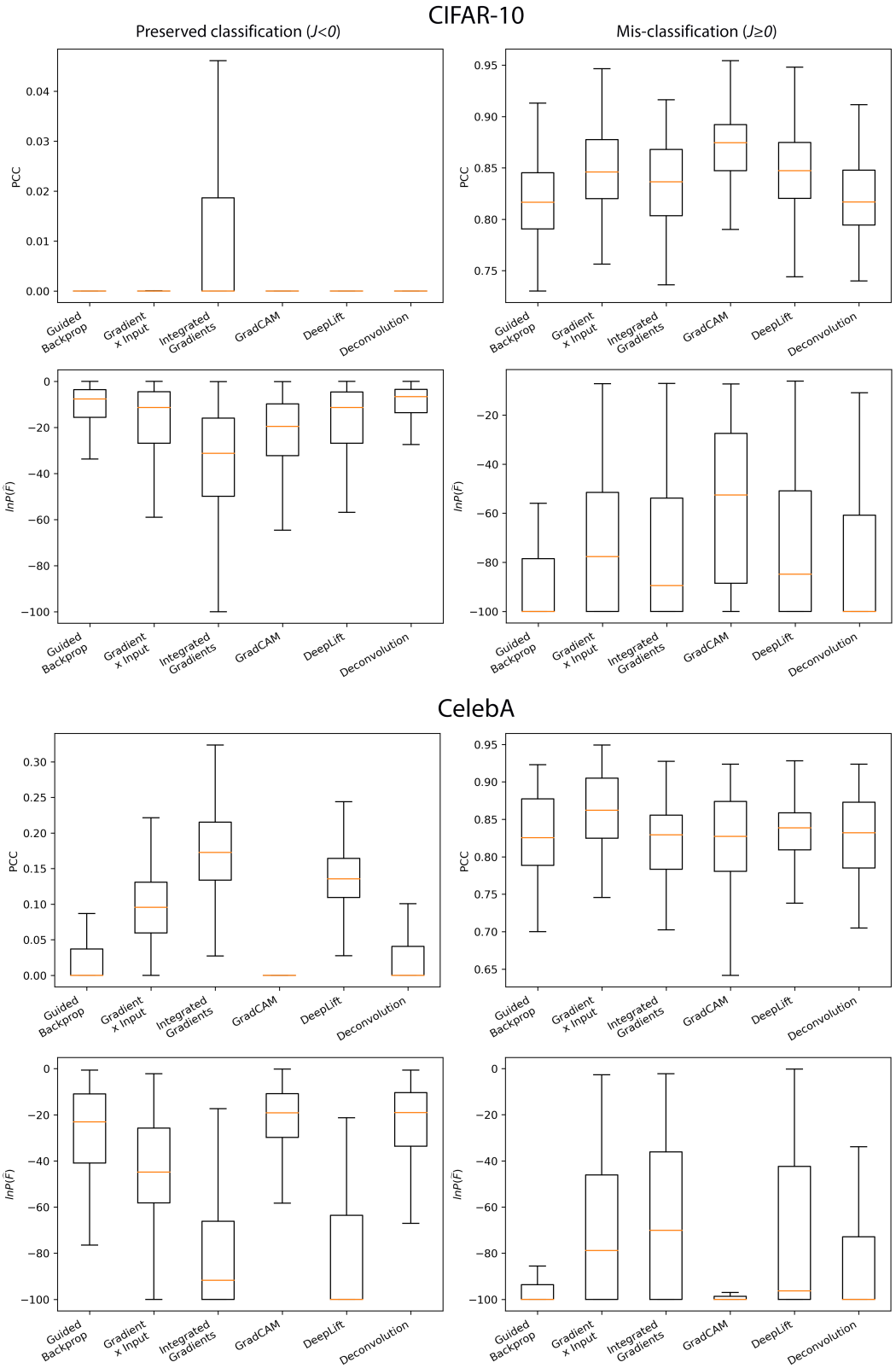

Figure 9: Robustness evaluation of different interpretation methods based on 100 randomly selected samples from CIFAR10 and CelebA test set. From top to bottom, first row (worst case evaluation) and second row (probabilistic evaluation). From left to right, first column (misinterpretation $\widehat{F}$) and second column (misinterpretation $\widetilde{F}$)