On Gradient-like Explanation under a Black-box Setting: When Black-box Explanations Become as Good as White-box

Yi Cai [1] **Gerhard Wunder** [1]

Abstract

Attribution methods shed light on the explainability of data-driven approaches such as deep learning models by uncovering the most influential features in a to-be-explained decision. While determining feature attributions via gradients delivers promising results, the internal access required for acquiring gradients can be impractical under safety concerns, thus limiting the applicability of gradient-based approaches. In response to such limited flexibility, this paper presents GEEX (gradient-estimation-based explanation), a method that produces gradient-like explanations through only query-level access. The proposed approach holds a set of fundamental properties for attribution methods, which are mathematically rigorously proved, ensuring the quality of its explanations. In addition to the theoretical analysis, with a focus on image data, the experimental results empirically demonstrate the superiority of the proposed method over state-of-the-art black-box methods and its competitive performance compared to methods with full access.

1. Introduction

Explainability is an increasingly important research topic due to the breakthroughs led by rapidly developing deep learning. Owing to the growth of hardware computational powers, deep learning models with growing capacities are able to handle tasks in real-world scenarios, which even outperform human experts in certain domains. As data-driven models, deep learning solutions at the current stage are distinguished from traditional approaches based on expert systems (Russell, 2010). Data-driven approaches learn their decision rules implicitly from some given data distribution, which conceals decision reasoning from humans. The shortage of knowledge about the decision-making process puts the deployment of AI models at risk. A typical example of machine failures is the Clever-Hans-Effect (Johnson, 1911) exposed by previous research (Lapuschkin et al., 2019; Geirhos et al., 2020). In the context of machine learning, the effect refers to the case that data-driven models learn to use irrelevant features as shortcuts for classification due to an imbalanced data distribution (e.g., watermarks only contained in certain classes of instances because of different data sources). Models suffering from the Clever-Hans-Effect may perform well in laboratories, but their outcomes can be misleading and totally unreliable in practice. Apart from the unintentional failure, it has been shown that data-driven models are fragile under adversarial attacks. These attacks can steer model outcomes by adding artifacts not recognizable by bare human eyes to the targeted input, which makes counteracting adversaries a challenging task. Given the potential risks, employing these black boxes in crucial application scenarios, such as medical image classification and autonomous driving, can cause unpredictable consequences as they may fail accidentally or intentionally. Explainability, a key to the mysterious box of AI and a potential shield against adversarial attacks (Fidel et al., 2020; Watson & Al Moubayed, 2021), is particularly interested in how models make up their minds.

One way of delivering explanatory information is the feature attribution method, which is the focus of this paper. The goal of such attribution methods is to determine the contributions of input features to an inquired model outcome, and thus uncover the observations that support its decision. Existing attribution methods can be categorized as either white-box or black-box approaches depending on their assumption about model accessibility. As the name implies, white-box explanation methods assume full access to the target model. Given more details about the inference procedure, they produce precise explanations by investigating the gradient/information flow throughout the target model. In practice, however, there is no guarantee of detailed internal access to models due to safety and security concerns, which limits the applicability of white-box approaches in real-world scenarios. Flexibility is another concern. Modifications are needed when a white-box approach is applied

[1]Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany. Correspondence to: Yi Cai <yi.cai@fu-berlin.de>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

to explain other models that its original design does not consider. One should not expect a gradient-based approach examining neural networks with backward propagation to uncover the inference process of a tree-based model without adjustments. Contrary to the full access assumption, black-box explainers require only query-level access, meaning that a to-be-explained model can only be accessed via its input and output interfaces. As direct investigation into inference procedures becomes unfeasible under a black-box setting, methods of this kind raise queries and explain model decisions indirectly by analyzing the correlation between input features and model outputs. The loosened accessibility assumption, coupled with the less specific explanation procedure (no prior knowledge about model structure considered), improves the applicability of black-box explainers. On the other hand, the restricted access poses a challenge in deriving precise explanations, especially when dealing with models handling high-dimensional inputs, such as images.

Aiming at combining the strengths of both categories, this paper presents Gradient-Estimation-based EXplanation (GEEX)[1], an explanation method producing gradient-like explanations under a black-box setting. By employing gradient estimation, GEEX circumvents the necessity for the full access assumption and, in principle, is applicable to arbitrary models. This positions GEEX as an alternative for explainability under circumstances where internal details about the target are inaccessible. In comparison to other black-box explainers, the proposed method produces fine-grained and precise feature attributions rather than fuzzy hot regions. The qualitative analysis in the experiment demonstrates that the resulting explanations capture homologous structures when compared to explanations derived from actual gradients. Most importantly, we theoretically show that GEEX fulfills a set of fundamental properties of attribution methods, ensuring the usefulness and meaningfulness of the resultant explanations.

2. Related Work

Black-box explanation methods are widely adopted in practice due to their flexibility and applicability. They treat the to-be-explained model as a black box with its internal functions left out. The general ideas behind methods of this kind are similar: creating a set of queries by altering the feature values of x, then deriving the explanation for the decision $f(x)$ through analysis of the correlation between changes in the inputs and outputs. LIME (Ribeiro et al., 2016) is one of the most representative methods from this category. It generates the queries by randomly switching on and off features in the original input and observes the corresponding predictions from the target model. Based on the observations, LIME then fits a self-explainable surrogate model (typically in the form of linear regression), a proxy for extracting explanatory information. For image data, an additional step conducted by LIME is clustering pixels as superpixels according to the similarity of pixel values and their spatial distances (Vedaldi & Soatto, 2008), which reduces the search space to a user-defined size. Simplifying the search space enables LIME to highlight broader regions containing evidential features.

Apparently, grouping pixels can negatively affect explanation quality. For example, low-level features such as edges and contours are informative to deep learning models when solving classification tasks (Zeiler & Fergus, 2014). Superpixel techniques that segment pixels along edges inevitably fragment these low-level features into diverse components. Consequently, the explainer may overlook (parts of) the divided features or include irrelevant pixels. RISE (Petsiuk et al., 2018) overcomes the issue with mask resizing, which generates smaller initial masks and upsamples them to the target size through bilinear interpolation. This approach empowers RISE to handle low-level features of any shape without significantly expanding the search space. With the enhanced quality of derived explanations, mask resizing also extends the applicability of RISE to more complicated scenarios, such as explaining object detectors (Petsiuk et al., 2021) and image generators (Park et al., 2024).

Compared to black-box methods, more efforts have been invested in white-box approaches for explaining image classifiers, as they sharpen resultant attribution maps owing to the detailed access to the target model. The most straightforward white-box approach interprets vanilla gradients directly as explanations, tracing the decision function's partial derivatives with respect to the input backward through the model (Simonyan et al., 2014). However, subsequent research shows that vanilla gradients can be excessively noisy (Smilkov et al., 2017). With rapid gradient fluctuations (Balduzzi et al., 2017) identified as a possible cause, SMOOTHGRAD (Smilkov et al., 2017) smooths explanations by applying Gaussian noises to the input and averaging the resulting gradients. Gradient averaging yields more robust outcomes, with the denoising effect positively correlating to the number of Gaussian-noised samples. On the other hand, IG (integrated gradients) (Sundararajan et al., 2017) promotes the quality of gradient-based explanations with an incorporated baseline, which models feature absence. By integrating the gradients over a straightline path from the target instance to the pre-defined baseline, IG fulfills a set of fundamental properties that ensure explanation quality.

Alternative to gradient-based solutions, propagation-based methods (Montavon et al., 2017; Selvaraju et al., 2020; Achtibat et al., 2023) determine attributions by redistributing prediction scores back to input features by means of propaga-

[1]Full code for reproducibility can be found at: https://github.com/caiy0220/GEEX

tion rules. These approaches explicitly leverage model structures and complete the explanation process with one round-trip throughout the model. While being a non-negligible group of explainers, we selectively exclude propagation-based approaches from the discussion to focus on reproducing gradient-based explanations under a black-box setting.

3. Gradient Estimation for Explanation

Denoting a model function as $f(\cdot)$ and a target input (the explicand) as $\boldsymbol{x} = (x_1, x_2, ..., x_p)$, the goal of attribution methods is to determine a vector $\boldsymbol{\xi} \in \mathbb{R}^p$ that decomposes the total contribution $f(\boldsymbol{x})$ into feature attributions, so that the attribution scores satisfy *Completeness*, i.e. $b + \sum_k \xi_k = f(\boldsymbol{x})$. The bias b is a scalar representing model activation status given the full absence of input features. Ideally, the bias will have a zero value for properly defined feature absence. Here, as in the rest of the paper, bold symbols denote vectors, while plain variables signify scalars. The target function $f(\cdot)$ produces a scalar indicating the model's confidence in its decision-making process. In the context of a multi-class setting, the scalar value can be interpreted as the confidence in determining one class, and the resulting explanation reveals the reasoning of the model in deciding whether an explicand belongs to the specific class. If the internal access of the model is available, the gradient of $f(\cdot)$ with respect to the input features can be readily acquired, facilitating subsequent processing for the derivation of explanations. Although such a convenience is unfeasible under a black-box setting, luckily, it is still possible to estimate gradients through queries and observations.

3.1. Gradient Estimation

Gradient estimation is a group of algorithms designed to approximate the gradient of a function (Mohamed et al., 2020), which is widely utilized in black-box optimization problems (Wierstra et al., 2014). It offers an alternative in situations where the acquisition of exact gradients is impractical or computationally expensive. Different from the attempt to compute gradients through backward propagation of losses, gradient estimation approximates gradients with a search distribution determined by some parameters of interest. More specifically, it defines gradients as the direction towards lower expected loss with respect to the analyzing target, which is the input features $\boldsymbol{x}$ in the context of explainability. Denoting the loss function by $\mathcal{L}(\cdot)$ and the set of parameters by $\boldsymbol{x}$, the expected loss over the search distribution is defined as:

$$J(\boldsymbol{x}) := \mathbb{E}_{\pi(\boldsymbol{z}|\boldsymbol{x})}[\mathcal{L}(\boldsymbol{z})] = \int \mathcal{L}(\boldsymbol{z})\pi(\boldsymbol{z}|\boldsymbol{x})\,\mathrm{d}\boldsymbol{z}$$

where $\pi(\cdot|\boldsymbol{x})$ indicates the probability density function of the search distribution parameterized by $\boldsymbol{x}$ and $\boldsymbol{z}$ denotes samples drawn from the distribution. Here, the parameter set is denoted as $\boldsymbol{x}$ to emphasize that the analysis target is input features rather than model parameters considered in common settings, which are not available in this case. The search gradient can be written as follows through simplification using the log-likelihood trick under the assumption that both the loss and the search distribution are continuously differentiable (Mohamed et al., 2020):

$$\begin{aligned}\nabla_{\boldsymbol{x}} J(\boldsymbol{x}) &= \nabla_{\boldsymbol{x}} \int \mathcal{L}(\boldsymbol{z})\pi(\boldsymbol{z}|\boldsymbol{x})\,\mathrm{d}\boldsymbol{z} \\ &= \int [\mathcal{L}(\boldsymbol{z})\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{z}|\boldsymbol{x})]\pi(\boldsymbol{z}|\boldsymbol{x})\,\mathrm{d}\boldsymbol{z} \\ &= \mathbb{E}_{\pi(\boldsymbol{z}|\boldsymbol{x})}[\mathcal{L}(\boldsymbol{z})\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{z}|\boldsymbol{x})]\end{aligned}$$

The value of the integral above can be empirically approximated with a Monte Carlo estimator given n samples $\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, ..., \boldsymbol{z}^{(n)}\} \sim \pi(\cdot|\boldsymbol{x})$:

$$\boldsymbol{\eta}(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} J(\boldsymbol{x}) \approx \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(\boldsymbol{z}^{(i)})\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{z}^{(i)}|\boldsymbol{x}) \quad (1)$$

Once substituting the loss in Equation 1 with the target function, η produces the estimated gradient for $f(\cdot)$. The estimation demonstrates model sensitivities to input features, which, to a certain extent, uncovers the reasoning behind the target decision. However, the direct usage of gradient estimation is unsatisfactory as it shares several shortcomings with the actual gradient (Sundararajan et al., 2017). Using estimated gradients as explanations violates *Sensitivity*, a fundamental axiom of attribution methods stating that a feature should receive a non-zero attribution if modifying its value induces a change in the model outcome. The counterexample in Fig. 1 shows that estimated gradients can be trapped by a locally flattened segment of a function, resulting in an underestimation of feature contribution. Underrating or even overlooking relevant features causes the violation of Sensitivity. Furthermore, along the x-axis of the example, the attribution barely aligns with the total feature contribution represented by $f(x)$. The fact that local sensitivities, as indicated by gradients, do not necessarily correlate to actual feature contributions undermines the interpretability of these values in their raw form.

3.2. Gradient-estimation-based Explanation

The failure when employing raw gradient estimation stems from the lack of a reference point that models the absence of features, to which the impact of feature presence can be compared. To overcome the aforementioned limitations, we present GEEX (gradient-estimation-based explanation), an attribution method that introduces a baseline and integrates estimations over a straightline path from the baseline to the explicand. Denoting a baseline point as $\mathring{\boldsymbol{x}}$, the straightline path can be written as an interpolation:

$$\boldsymbol{x}(\alpha) = \mathring{\boldsymbol{x}} - \alpha(\boldsymbol{x} - \mathring{\boldsymbol{x}})$$

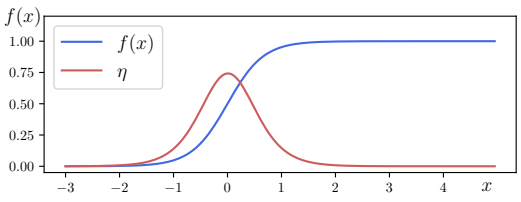

Figure 1. A simple case shows that considering the estimated gradient as an explanation can lead to misleading results. Suffering from gradient saturation, the attribution of x converges to 0 as its value increases, conflicting with the truth that the value of the sigmoid function $f(\cdot)$ relies solely on x.

An intuitive implementation of GEEX can be achieved by replacing the actual gradient in IG (Sundararajan et al., 2017) with the estimation kernel:

$$\boldsymbol{\xi} := \frac{(\boldsymbol{x} - \mathring{\boldsymbol{x}})}{s} \circ \sum_{j=1}^{s} \boldsymbol{\eta}(\boldsymbol{x}(\frac{j}{s}))$$

Replacing the loss with $f(\cdot)$ and expanding $\boldsymbol{\eta}$ yield:

$$\boldsymbol{\xi} = \frac{(\boldsymbol{x} - \mathring{\boldsymbol{x}})}{n \cdot s} \circ \sum_{j=1}^{s} \sum_{i=1}^{n} f(\boldsymbol{z}^{(i)}) \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{z}^{(i)} | \boldsymbol{x}(\frac{j}{s})) \quad (2)$$

where s denotes the number of interpolation steps for integral approximation. Although the search distribution $\pi(\cdot|\boldsymbol{x})$ is left unspecified here, we restrict $\boldsymbol{x}$ to be the location parameter, and $\pi(\cdot|\boldsymbol{x})$ is required to have a mean at its location parameter, which is a necessity for unbiased estimation (see Appendix A for theoretical details). A surprising fact is that GEEX inhabits more or less all of the properties of IG. In fact, the resultant method complies with a set of four fundamental axioms as stated in Theorem 3.1 (see the proof and further details in Appendix B). Please note that the fulfillment of the axioms comes true when enough samples have been drawn, so the statement should be interpreted in a probability sense.

Theorem 3.1. *GEEX, a path method built upon estimated gradients, satisfies Sensitivity, Insensitivity, Implementation Invariance, and Linearity.*

As the significance of *Sensitivity* has been shown before, the three remaining axioms also hold practical meanings for the proposed method. *Insensitivity* (called *Dummy* in (Friedman, 2004)) is a property that measures attributions to features having no impact. Opposite to Sensitivity, a violation of Insensitivity results in an overestimation of feature importance; failure to fulfill either of the two results in misleading explanation outcomes. Meanwhile, *Implementation Invariance*, a key concept especially for black-box explainers, ensures the applicability of GEEX. *Linearity* appears trivial among the four axioms as it does not directly relate to explanation quality. However, Linearity enables the decomposition of non-interacting features (features having no interaction with each other in the target function). Such a decomposition divides a high-dimensional feature space into subspaces with lower dimensionalities. For a function consisting of m terms, the variance of the gradient estimator deployed by GEEX is of the order $O(m^2)$ (Mohamed et al., 2020). Feature space decomposition that linearly reduces the number of terms results in a quadratic reduction of estimation variance. Although developing a detailed decomposition strategy exceeds the scope of this paper, we argue that the fulfillment of Linearity holds the potential to enhance estimation precision and computational efficiency, which indirectly contribute to explanation quality.

Having covered various axioms, of particular note is the *Completeness* of GEEX stated in Theorem 3.2. Being an approximation of the path integral, the sum of feature attributions determined by GEEX converges in probability to the prediction difference between the baseline and the explicand, i.e. $f(\boldsymbol{x}) - f(\mathring{\boldsymbol{x}})$. Viewing the prediction at $\mathring{\boldsymbol{x}}$ as the bias b, the Completeness will become evident with sufficient observations (for the detailed proof, see Appendix A). Satisfying Completeness is fundamental, although many do not, for attribution methods. This property upholds the practical meaning of attribution score – a value indicating the proportion of feature contribution to model outcome.

Theorem 3.2. *(Completeness) The explanation derived by GEEX is complete regarding the model outcome $f(\boldsymbol{x})$.*

Again, given the explanation by GEEX as an empirical approximation, the statement should be interpreted with a probability perspective. It is noteworthy that the approximation error has two sources: the error associated with the line integral approximation and the error of the gradient estimator. Although Completeness is desired, the iterative estimation process poses a challenge in deploying GEEX, as both sources contribute to the overall error of the explainer. Under limited computational resources, managing the allocation of efforts between the two estimators – the interpolation steps s and the sample set size n – for optimizing the ultimate precision of the explainer can be demanding. Moreover, even with the puzzle of hyperparameter selection solved, the performance of the proposed approach is bounded by IG from above. To address the challenges, the next section introduces the way of improving the practical usefulness of GEEX, enabling it to go beyond IG.

3.3. Noise Sampling and Computational Efficiency

First, we clarify the role of $\boldsymbol{x}$ in the search distribution. In principle, $\boldsymbol{x}$ can represent any parameter of any distribution, on condition that $\pi(\cdot|\boldsymbol{x})$ is continuously differentiable in $\boldsymbol{x}$. However, in addition to ensuring unbiased estimation as stated in the last section, considering $\boldsymbol{x}$ as the location

parameter of the search distribution brings practical convenience to the explanation process. This is particularly advantageous when handling different explicands, as it allows the usage of an identical pre-generated mask set during the explanation procedure.

For a standard distribution $\pi_{\boldsymbol{\theta}}(\boldsymbol{z}|\mathbf{0})$, the search distribution for a concrete explicand $\boldsymbol{x}$ from the location family holds $\pi_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) = \pi_{\boldsymbol{\theta}}(\boldsymbol{z} - \boldsymbol{x}|\mathbf{0})$. Here, $\boldsymbol{\theta}$ represents the remaining parameters, distinct from $\boldsymbol{x}$, that describe the distribution. The parameter set $\boldsymbol{\theta}$ can be a hyperparameter of the explainer, but we omit it for simplicity as it is irrelevant to the following discussion. Designating $\boldsymbol{x}$ as the location parameter allows a pre-construction of the sample set and pre-computation of the log derivative with the standard distribution according to the revision of Equation 2:

$$\begin{aligned}\boldsymbol{\xi} &= \frac{(\boldsymbol{x} - \mathring{\boldsymbol{x}})}{n \cdot s} \circ \sum_{j=1}^{s} \sum_{i=1}^{n} f(\boldsymbol{z}^{(i)}) \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{z}^{(i)} - \boldsymbol{x}(\frac{j}{s})|\mathbf{0}) \\ &= \frac{(\boldsymbol{x} - \mathring{\boldsymbol{x}})}{n \cdot s} \circ \sum_{j=1}^{s} \sum_{i=1}^{n} f(\boldsymbol{x}(\frac{j}{s}) + \boldsymbol{\epsilon}^{(i)}) \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}^{(i)}|\mathbf{0})\end{aligned}$$

where $\boldsymbol{\epsilon} = \boldsymbol{z} - \boldsymbol{x}$ denotes the pre-generated mask. The construction of the mask set $\{\boldsymbol{\epsilon}^{(i)}\}$ is a one-time effort, and it can be applied to arbitrary explicand-baseline pairs. This is possible because sampling for gradient estimation is decoupled from the concrete value of $\boldsymbol{x}$. This convenience facilitates the application of more complicated sampling strategies aiming at variance reduction, such as mirror sampling (Brockhoff et al., 2010) considered in this work, or potentially other computationally more expensive ones like orthogonal coupling (Choromanski et al., 2019).

Moreover, recognizing the decoupling of sampling from the location parameter, the sums from the two levels of approximation can be merged:

$$\boldsymbol{\xi} = \frac{(\boldsymbol{x} - \mathring{\boldsymbol{x}})}{n^*} \circ \sum_{\substack{\boldsymbol{\epsilon} \sim \pi(\cdot|\mathbf{0}) \\ \alpha \sim \mathcal{U}_{[0,1]}}} f(\boldsymbol{x}(\alpha) + \boldsymbol{\epsilon}) \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0}) \quad (3)$$

where n^* denotes the number of queries generated by $(\boldsymbol{\epsilon}, \alpha)$ pairs. Although the straightforward rewriting does not alter any underlying concepts, it streamlines hyperparameter selection. The form can be interpreted as the cumulative sum of dense *one-sample gradient estimators* along the integral path. More importantly, merging the terms for integral approximation and gradient estimation improves explanation quality by providing a "smoother" approximation of the path integral without compromising the precision of gradient estimations. Fig. 2 provides an illustrative example of the smoothed approximation. The key factor, allowing the improvement, is that observations from estimators $\boldsymbol{\eta}(\boldsymbol{x}(\alpha))$ for neighboring instances on the path share information, which positively contributes to the estimation precision of each other. The proof of the claim is given in Appendix C. The nature that GEEX can smoothly approximate the path integral allows it to even surpass white-box explainers under certain circumstances (see further discussions in Section 4.3). Figure 3 gives an overview of the explanation procedure.

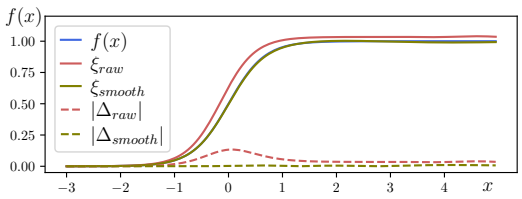

Figure 2. Given a baseline $f(-3) \approx 0$, the smoothed version of GEEX better approximates the actual contribution of the input feature with the same amount of observations. While the red solid line corresponds to explanations from the interpolation-based GEEX, the green line represents the results from the "smoothed" version, almost overlapping the actual contribution depicted by the blue line. The dashed line indicates the error of the derived explanation compared to the ground truth given by the total contribution $f(x)$.

Though the construction of a mask set $\{\boldsymbol{\epsilon}^{(i)}\}$ is straightforward in most cases, we suggest mask smoothing for the sampling when dealing with high-dimensional inputs, particularly referring to high-resolution images. Denoting the initial masks as $\hat{\boldsymbol{\epsilon}}$ and a blurring kernel as $\boldsymbol{w}$, the post-processing that finalizes the masks can be described as $\boldsymbol{\epsilon} = \frac{\boldsymbol{w}}{||\boldsymbol{w}||_F} * \hat{\boldsymbol{\epsilon}}$, where $||\boldsymbol{w}||_F$ is the Frobenius norm of the filter that normalizes the amplitude of perturbation, ensuring it is invariant to the convolution operator. In addition to the denoising effect by mitigating artifacts in adjacent pixels of masks, the filter softly groups spatially close pixels following the prior knowledge that adjacent pixels form low-level features. By applying similar changes to adjacent pixels simultaneously, soft grouping increases the possibility of removing a local pattern compared to conducting pixel-wise perturbation. In the case of high-dimensional explicands, such a convenience helps expose model sensitivities to the absence of local patterns, thus facilitating the identification of relevant pixels. However, it should be noted that the grouping does not stick to the assumption that feature values should be sampled independently for gradient estimation. Therefore, the application of mask smoothing raises a trade-off between the usefulness and correctness of resultant explanations and is preferred only when explaining high-dimensional explicands.

4. Experiments

With a focus on image data, we test the proposed method with models trained on three popular and publicly available image datasets. The experimental environment (including hardware and software settings) is described in Appendix D.

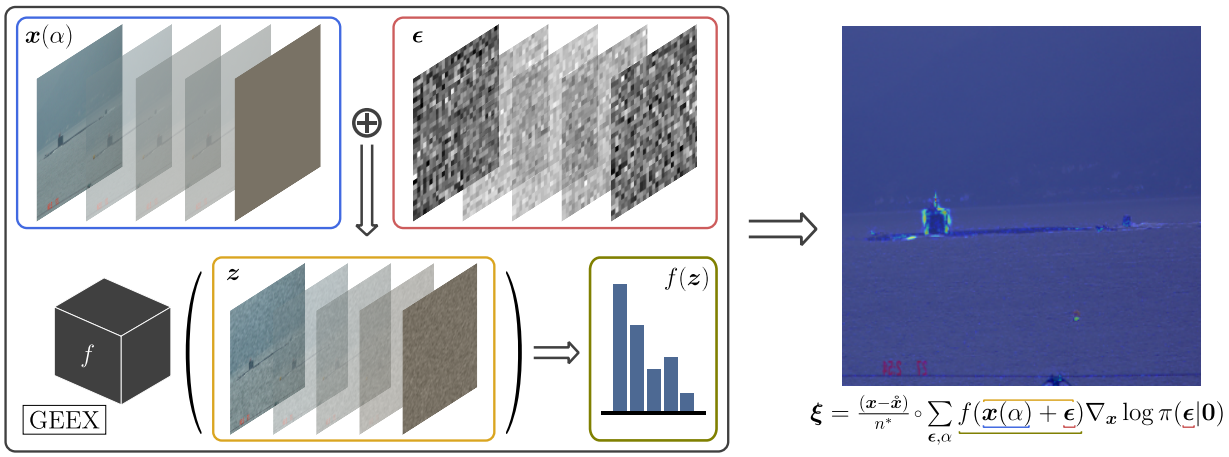

Figure 3. Overview of GEEX. A query $\boldsymbol{z}$ is determined by the sampled noise $\boldsymbol{\epsilon}$ and the position α on the path. The final explanation $\boldsymbol{\xi}$ (on the right, overlaid with the original input) is derived through the observations $\{f(\boldsymbol{z})\}$ and the pre-computed log derivatives.

4.1. Experimental Details

Dataset: The datasets considered during the experiments are: MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), and ImageNet (Russakovsky et al., 2015). The selection includes two grayscale datasets and one full-colored dataset with a notably larger input size. For each dataset, we train a classifier with its training set and evaluate explanations for model decisions on the test set.

Classifier: For MNIST and Fashion-MNIST, a simple CNN is trained. The model comprises two convolutional layers with a kernel size of 5, concatenated by three dense layers with sizes of 120, 84, and 10, respectively. The inputs from the two datasets have a shape of 28×28. For ImageNet, Inception V3[2] (Szegedy et al., 2016) is adopted, which takes inputs of size 299×299. Considering various configurations of the to-be-explained system enables the comparison of explainer performances across explanation tasks with different levels of complexity.

GEEX: A Gaussian distribution serves as the search distribution for GEEX, and the number of queries n^* is fixed to $5k$ across all test settings. The deviation σ, which determines the spread of the Gaussian, is configured as 1.0 for MNIST and Fashion-MNIST observing the polarized distribution of their pixel values. For ImageNet, where pixel values are more evenly distributed, σ is set to 0.3. Regarding the baseline $\hat{\boldsymbol{x}}$, a zero matrix is employed when explaining decisions on grayscale images, whereas the baseline for ImageNet is explicand-specific. For each explicand from ImageNet, the baseline is a blurred version of itself as suggested by (Sturmfels et al., 2020). To ensure a fair comparison, these baseline choices also apply to the competitors that incorporate a baseline during their explanation procedures. Besides, to mitigate the estimation noises caused by feature space expansion, mask smoothing is implemented through a Gaussian filter with a kernel size of 5 and a deviation of 0.7 when tested on ImageNet. Further details about hyperparameter selection are reported in Appendix F.

Competitor: We consider the gradient estimator (GE) as a competitor by interpreting gradient estimations directly as explanations. Its comparison to GEEX illustrates the importance of fulfilling the named properties. The remaining competitors, including two white-box and two black-box approaches, are listed below. For black-box explainers, the number of queries is identical to the setting for GEEX.

- SG (SmoothGrad) (Smilkov et al., 2017): a white-box approach interprets raw gradients as explanations.
- IG (Sundararajan et al., 2017): a white-box approach integrates gradients of entries over a straightline path from the explicand to a baseline.
- LIME (Ribeiro et al., 2016): a black-box approach explains a target model through a linear surrogate trained to mimic the observed behaviors of the target.
- RISE (Petsiuk et al., 2018): a black-box explainer determines feature attribution according to the expected impacts of input features on prediction outcomes.

4.2. Comparison to White-box Explanation

The evaluation of the proposed method commences with a qualitative assessment of the derived explanations. Sample explanations from various test settings are listed in Fig. 4 (additional examples can be found in Appendix E), where

[2]A pre-trained version from ImageNet is used without additional training, publicly available at: https://pytorch.org/vision/stable/models/inception.html

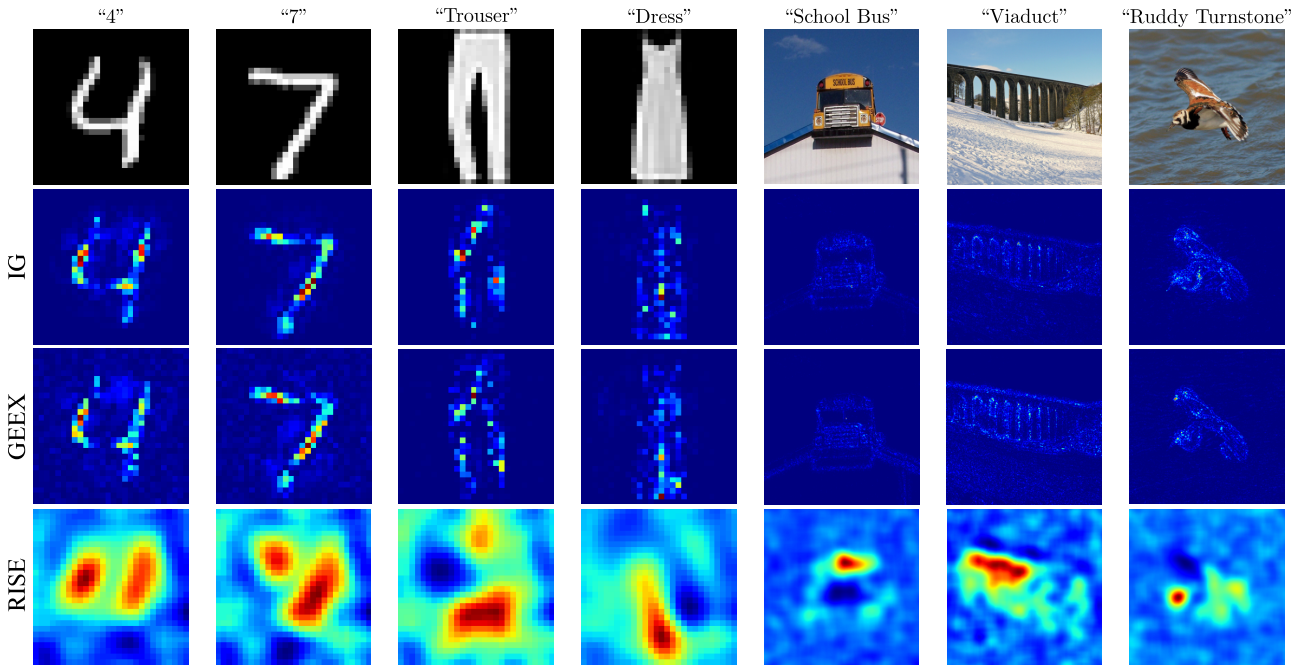

Figure 4. Sample explanations from GEEX (the second row) and two selected competitors

each column is an example set. The first image in an example set presents the explicand with the model prediction at the top, followed by the corresponding explanations from three chosen explainers visualized through saliency maps. The explainers comprise IG, GEEX, and RISE. The inclusion of IG and RISE serves the purpose of showcasing GEEX's capability in producing gradient-like explanations under a black-box setting and emphasizing its improvements compared to a state-of-the-art black-box explainer.

The provided examples contain explanations for four explicands with a smaller input size and three entries from a significantly higher-dimensional feature space. Across all sample sets, proximity between explanations from GEEX and IG is observed. Specifically, both GEEX and IG attribute the model's decision to the presence of pixels on the pant legs for the third explicand, predicted as "trouser", while pinpointing the bottom-middle located pixels as the most influential for predicting a "dress" in the fourth example. The insights garnered from the two explainers suggest that the classifier trained on Fashion-MNIST considers the presence of a gap in the bottom-middle region between two bright areas as a distinguishing feature between a trouser and a dress. However, the explanations from RISE fall short of reaching such a conclusion. Similar observations can be found in the sample explanations for InceptionV3. While the outcomes from RISE highlight fuzzy hot regions with noticeable background noises, the delivery of the proposed approach demonstrates fine-grained feature structures that steer the model's predictions. These include details such as the texture of the intake grille for the classified "school bus", the contour of the "viaduct", and the highlighted face as well as wing areas of the "ruddy turnstone". Again, the explanations by GEEX capture homologous attribution structures to the results from IG when dealing with more challenging tasks, consistent with the previously discussed performance in simpler test cases.

The qualitative examples intuitively show the performance of the proposed method. Nevertheless, human assessments for explanation evaluation are unscalable. Exhaustively evaluating explanation approaches with human efforts can be more than just expensive in practice. In fact, doing so can introduce human-sourced biases into the evaluation process as individuals may interpret explanations differently. For instance, given a model suffering from the Clever-Hans effect, which occasionally uses irrelevant features for prediction, an explainer correctly locating such mistakes can be underrated during human assessments due to mismatches between human domain knowledge and the actual model behaviors.

4.3. Quantitative Evaluation for Effectiveness

To quantify the performance of explainers objectively, this section evaluates explanation quality via a widely adopted scheme – evaluation via deletion (Samek et al., 2016). The evaluation process follows an intuitive yet effective idea: the removal of relevant features should induce larger drops in prediction confidence. More specifically, evaluation via deletion removes pixels sequentially in descending order according to their attribution scores. The changing

Table 1. The normalized AOPC scores by evaluation via deletion, higher is better.

Classifier	Replacement	SG	IG	RISE	LIME	GE	GEEX	Random
CNN (MNIST)	Baseline	0.8838	0.9434	0.9101	0.8653	0.8833	**0.9466**	0.3085
	Gaussian	0.8452	0.9415	0.8896	0.8692	0.8519	**0.9486**	0.3695
CNN (F-MNIST)	Baseline	0.8399	0.9275	0.8931	0.8167	0.8379	**0.9350**	0.3567
	Gaussian	0.8341	0.9219	0.8708	0.8085	0.8341	**0.9362**	0.4293
InceptionV3 (ImageNet)	Baseline	0.3781	**0.8805**	0.7659	0.6928	0.3806	0.7952	0.4003
	Gaussian	0.8557	**0.9155**	0.8699	0.8837	0.7289	0.9058	0.7434

*The overall best performances are in **bold** and the highest scores among black-box explainers are underlined.

trend of prediction confidence draws a curve throughout the deletion process, and the area over perturbation curve (AOPC) is considered as a metric to quantify the effectiveness of an explanation. To make the metric independent from the scale of model outcomes, the normalized AOPC is computed, i.e. the cumulative sum of dropping ratios: $AOPC = \frac{1}{l}\sum_{i=1}^{l}(1 - \frac{f(x^{(i)})}{f(x)})$, where $x^{(i)}$ denotes a variant of x with its top-i pixels masked out. One last thing to be clarified is the deletion operation. Following the discussion in Section 3.2, replacing a feature value with a corresponding baseline value is a natural way of defining the deletion. As some competitors do not actively use the chosen baselines, for a fair comparison, we also sample the replacement value from Gaussian as an alternative defining the deletion.

Table 1 reports the AOPC scores of competitors in all test settings, considering both definitions of the replacement value. The table groups explainers according to their accessibility assumptions. Alongside the named competitors, the AOPC scores of removing pixels in purely random order are also reported as a reference, illustrating the effectiveness of the derived explanations. Ideally, explanations delivering useful information are expected to achieve higher scores than random deletion. However, this is not the case for SG and GE in explaining decisions from InceptionV3. Their explanations, directly using either actual or estimated gradients, suffer from gradient saturation, leading to the overlooking of relevant features and subsequently limited performance. On the contrary, the fulfillment of Completeness and Sensitivity results in the competitive performance of GEEX. According to the AOPC scores, the proposed method consistently surpasses other black-box explainers across all test settings. The higher scores indicate that the assigned feature attributions correctly reflect to their actual contributions.

Compared to IG, GEEX achieves similar scores, aligning with the observations of their visually similar saliency maps in the qualitative assessment. For the simpler test cases, our approach even achieves better performances, which should be interpreted as an improvement brought by the smoother approximation of the path integral. While the white-box explainer approximates the path integral with sparse interpolation relying on accurate but expensive gradients (depending on model complexity), GEEX achieves a superior approximation with dense one-sample estimators that are computationally more efficient. Regarding the results on ImageNet, the larger feature space poses a challenge to all black-box approaches. As a result of higher gradient estimator variance caused by feature space expansion, GEEX falls behind IG. In this case, more observations are required to maintain the same level of estimation precision. Figure 5 illustrates the convergence of GEEX's performance towards IG as the number of queries increases.

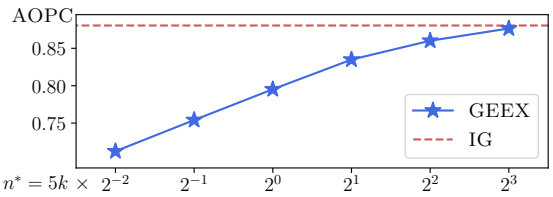

Figure 5. For InceptionV3, GEEX achieves an AOPC score converging to IG when the number of queries n^* increases.

5. Conclusion

In this work, we propose GEEX, an approach deriving gradient-like explanations under a black-box setting. It fulfills a set of fundamental properties of attribution methods, including *Completeness* and *Sensitivity*, thereby settling a theoretical guarantee for explanation quality. Alongside the theoretical analysis, the experimental results on three public datasets empirically show the competitive performance of the proposed method. In addition to surpassing all competitors in the simple test cases, the performance of GEEX also converges to the best score achieved in a white-box setting when acquiring sufficient observations. Although the computational expense can be a concern as for other black-box approaches, the computations in GEEX are highly parallelized, which allows it to meet any potential real-time requirement by distributing the workloads to distributed

agents. Moreover, we believe that reducing computational complexity through feature space decomposition, guided by the *Linearity* property, addresses the last piece of the puzzle and should be considered a direction of future works.

Acknowledgements

Yi Cai and Gerhard Wunder were supported by the Federal Ministry of Education and Research of Germany (BMBF) in the program of "Souverän. Digital. Vernetzt.", joint project "AIgenCY: Chances and Risks of Generative AI in Cybersecurity", project identification number 16KIS2013. Gerhard Wunder was also supported by BMBF joint project "6G-RIC: 6G Research and Innovation Cluster", project identification number 16KISK020K.

Impact Statement

This paper introduces an explanation method for data-driven models that grant only query-level access. By loosening the full-accessibility assumption, our approach is applicable in high-stakes environments where detailed access to models is typically restricted due to safety and security concerns. We believe that the presented method holds the potential to verify and supervise AI-based decisions, thereby promoting the transparency and trustworthiness of automated systems.

References

Achtibat, R., Dreyer, M., Eisenbraun, I., Bosse, S., Wiegand, T., Samek, W., and Lapuschkin, S. From attribution maps to human-understandable explanations through concept relevance propagation. *Nature Machine Intelligence*, 5 (9):1006–1019, 2023.

Balduzzi, D., Frean, M., Leary, L., Lewis, J., Ma, K. W.-D., and McWilliams, B. The shattered gradients problem: If resnets are the answer, then what is the question? In *International Conference on Machine Learning*, pp. 342–350. PMLR, 2017.

Brockhoff, D., Auger, A., Hansen, N., Arnold, D. V., and Hohm, T. Mirrored sampling and sequential selection for evolution strategies. In *Parallel Problem Solving from Nature, PPSN XI: 11th International Conference, Kraków, Poland, September 11-15, 2010, Proceedings, Part I 11*, pp. 11–21. Springer, 2010.

Choromanski, K., Rowland, M., Chen, W., and Weller, A. Unifying orthogonal monte carlo methods. In *International Conference on Machine Learning*, pp. 1203–1212. PMLR, 2019.

Fidel, G., Bitton, R., and Shabtai, A. When explainability meets adversarial learning: Detecting adversarial examples using shap signatures. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Friedman, E. J. Paths and consistency in additive cost sharing. *International Journal of Game Theory*, 32:501–518, 2004.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Johnson, H. M. Clever hans (the horse of mr. von osten): A contribution to experimental, animal, and human psychology. *The Journal of Philosophy, Psychology and Scientific Methods*, 8(24):663–666, 1911.

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1096, 2019.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Mohamed, S., Rosca, M., Figurnov, M., and Mnih, A. Monte carlo gradient estimation in machine learning. *The Journal of Machine Learning Research*, 21(1):5183–5244, 2020.

Montavon, G., Lapuschkin, S., Binder, A., Samek, W., and Müller, K.-R. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

Park, J.-H., Ju, Y.-J., and Lee, S.-W. Explaining generative diffusion models via visual analysis for interpretable decision-making process. *Expert Systems with Applications*, 248:123231, 2024.

Petsiuk, V., Das, A., and Saenko, K. RISE: Randomized input sampling for explanation of black-box models. In *Proceeedings of the British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK*, 2018.

Petsiuk, V., Jain, R., Manjunatha, V., Morariu, V. I., Mehra, A., Ordonez, V., and Saenko, K. Black-box explanation of object detectors via saliency maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11443–11452, 2021.

Ribeiro, M. T., Singh, S., and Guestrin, C. "Why should I trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115: 211–252, 2015.

Russell, S. J. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K.-R. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, 2016.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128:336–359, 2020.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations*. ICLR, 2014.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. SmoothGrad: Removing noise by adding noise. In *Proceedings of the ICML Workshop on Visualization for Deep Learning, Sydney, Australia, 10 August 2017*, 2017.

Sturmfels, P., Lundberg, S., and Lee, S.-I. Visualizing the impact of feature attribution baselines. *Distill*, 5(1):e22, 2020.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Vedaldi, A. and Soatto, S. Quick shift and kernel methods for mode seeking. In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV 10*, pp. 705–718. Springer, 2008.

Watson, M. and Al Moubayed, N. Attack-agnostic adversarial detection on medical data using explainable machine learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 8180–8187. IEEE, 2021.

Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J., and Schmidhuber, J. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833. Springer, 2014.

A. Proof of Completeness

Differing from the order of introducing the properties as presented in Section 3, we first show the *Completeness* of GEEX, which is stated in Theorem 3.2. Proving Completeness brings significant convenience to the proof of the remaining axioms. Defining the full absence with the baseline, the Completeness regarding the prediction outcome $f(\dot{\boldsymbol{x}})$ can be proved by showing:

$$f(\mathring{\boldsymbol{x}}) + \sum_{k=1}^{p} \xi_k = f(\dot{\boldsymbol{x}})$$

where $\dot{\boldsymbol{x}}$ denotes the explicand to avoid the potential confusion between the variable $\boldsymbol{x}$ and its concrete value given by $\dot{\boldsymbol{x}}$. The premise for GEEX satisfying Completeness is that the search distribution for the gradient estimator has a mean at its location parameter determined by the explicand:

$$\mathbb{E}_{\pi(\boldsymbol{z}|\boldsymbol{x})}[\boldsymbol{z}] = \boldsymbol{x} \tag{4}$$

Through the proof of Completeness, we also show that an unbiased search distribution is a necessity for an unbiased gradient estimation.

Proof. In GEEX, the attribution of the l-th feature is given by:

$$\begin{aligned}
\xi_l &= \frac{(\dot{x}_l - \mathring{x}_l)}{s} \cdot \sum_{j=1}^{s} \eta_l(\dot{\boldsymbol{x}}(\frac{j}{s})) \\
&= \frac{(\dot{x}_l - \mathring{x}_l)}{s} \cdot \sum_{j=1}^{s} \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{z}^{(i)}) \frac{\partial \log \pi(z_l^{(i)}|x_l)}{\partial x_l}\Big|_{x_l=\dot{x}_l(\frac{j}{s})} \\
&\overset{\boldsymbol{z}=\dot{\boldsymbol{x}}(\frac{j}{s})+\boldsymbol{\epsilon}}{=} \frac{(\dot{x}_l - \mathring{x}_l)}{n \cdot s} \cdot \sum_{j=1}^{s} \sum_{i=1}^{n} f(\dot{\boldsymbol{x}}(\frac{j}{s}) + \boldsymbol{\epsilon}^{(i)}) \frac{\partial \log \pi(\dot{x}_l(\frac{j}{s}) + \epsilon_l^{(i)}|x_l)}{\partial x_l}\Big|_{x_l=\dot{x}_l(\frac{j}{s})}
\end{aligned}$$

The third equation simply rewrites the symbol for a sample from a distribution whose location is described by the explicand as $\dot{\boldsymbol{x}}$ plus some noise $\boldsymbol{\epsilon}$. The model outcome in the above formula can be expanded with the Taylor series:

$$f(\dot{\boldsymbol{x}}(\frac{j}{s}) + \boldsymbol{\epsilon}^{(i)}) = f(\dot{\boldsymbol{x}}(\frac{j}{s})) + \nabla f(\dot{\boldsymbol{x}}(\frac{j}{s}))^T \cdot \boldsymbol{\epsilon}^{(i)} + O(||\boldsymbol{\epsilon}^{(i)}||^2)$$

Substituting the expansion and using $\frac{\partial}{\partial x_l} \log \pi$ as a shorthand for $\frac{\partial \log \pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|x_l)}{\partial x_l}\Big|_{x_l=\dot{x}_l(\frac{j}{s})}$:

$$\begin{aligned}
\xi_l &= \frac{(\dot{x}_l - \mathring{x}_l)}{n \cdot s} \sum_{j=1}^{s} \sum_{i=1}^{n} [f(\dot{\boldsymbol{x}}(\frac{j}{s})) + \nabla f(\dot{\boldsymbol{x}}(\frac{j}{s}))^T \cdot \boldsymbol{\epsilon}^{(i)} + O(||\boldsymbol{\epsilon}^{(i)}||^2)] \frac{\partial}{\partial x_l} \log \pi \\
&= \frac{(\dot{x}_l - \mathring{x}_l)}{n \cdot s} \cdot \Bigg[\underbrace{\sum_{j=1}^{s} \sum_{i=1}^{n} f(\dot{\boldsymbol{x}}(\frac{j}{s})) \frac{\partial}{\partial x_l} \log \pi}_{①} + \underbrace{\sum_{j=1}^{s} \sum_{i=1}^{n} \nabla f(\dot{\boldsymbol{x}}(\frac{j}{s}))^T \cdot \boldsymbol{\epsilon}^{(i)} \cdot \frac{\partial}{\partial x_l} \log \pi}_{②} + \underbrace{\sum_{j=1}^{s} \sum_{i=1}^{n} O(||\boldsymbol{\epsilon}^{(i)}||^2) \frac{\partial}{\partial x_l} \log \pi}_{③} \Bigg]
\end{aligned}$$

When the number of samples n for the gradient estimator increases, the term ① converges in probability to 0 as:

$$\begin{aligned}
① &= \sum_{j=1}^{s} \frac{1}{n} \sum_{i=1}^{n} f(\dot{\boldsymbol{x}}(\frac{j}{s})) \frac{\partial}{\partial x_l} \log \pi = \sum_{j=1}^{s} \Bigg[f(\dot{\boldsymbol{x}}(\frac{j}{s})) \cdot \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial x_l} \log \pi \Bigg] \\
&\xrightarrow{\mathbb{P}} \sum_{j=1}^{s} \Bigg[f(\dot{\boldsymbol{x}}(\frac{j}{s})) \cdot \mathbb{E}_{\pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|\dot{x}_l(\frac{j}{s}))} \Big[\frac{\partial}{\partial x_l} \log \pi \Big] \Bigg] = 0
\end{aligned}$$

This is because the expectation takes a zero value:

$$
\begin{aligned}
\mathbb{E}_{\pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|\dot{x}_l(\frac{j}{s}))}\Big[\frac{\partial}{\partial x_l}\log\pi\Big] &= \int_{-\infty}^{+\infty}\pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|\dot{x}_l(\frac{j}{s}))\cdot\frac{\partial\log\pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|x_l)}{\partial x_l}\Big|_{x_l=\dot{x}_l(\frac{j}{s})}\,\mathrm{d}\epsilon_l \\
&\overset{z_l=\dot{x}_l(\frac{j}{s})+\epsilon_l}{=} \int_{-\infty}^{+\infty}\pi(z_l|\dot{x}_l(\frac{j}{s}))\cdot\frac{\partial\log\pi(z_l|x_l)}{\partial x_l}\Big|_{x_l=\dot{x}_l(\frac{j}{s})}\,\mathrm{d}z_l \\
&= \int_{-\infty}^{+\infty}\frac{\partial\pi(z_l|x_l)}{\partial x_l}\Big|_{x_l=\dot{x}_l(\frac{j}{s})}\,\mathrm{d}z_l \\
&= \Big[\frac{\partial}{\partial x_l}\int_{-\infty}^{+\infty}\pi(z_l|x_l)\,\mathrm{d}z_l\Big]\Big|_{x_l=\dot{x}_l(\frac{j}{s})} \\
&= \frac{\partial}{\partial x_l}1\Big|_{x_l=\dot{x}_l(\frac{j}{s})} \\
\Rightarrow \mathbb{E}_{\pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|\dot{x}_l(\frac{j}{s}))}\Big[\frac{\partial}{\partial x_l}\log\pi\Big] &= 0
\end{aligned}
\tag{5}
$$

The interchange of derivatives and integrals is possible because $\pi(\boldsymbol{z}|\boldsymbol{x})$ is continuously differentiable in $\boldsymbol{x}$, a fundamental prerequisite when choosing the search distribution for gradient estimation. Now switching to the term ②:

$$
\begin{aligned}
② &= \sum_{j=1}^{s}\frac{1}{n}\sum_{i=1}^{n}\nabla f(\dot{\boldsymbol{x}}(\frac{j}{s}))^{T}\cdot\boldsymbol{\epsilon}^{(i)}\cdot\frac{\partial}{\partial x_l}\log\pi \\
&= \sum_{j=1}^{s}\frac{1}{n}\sum_{i=1}^{n}\Big[\Big[\sum_{k=1}^{p}\frac{\partial f}{\partial x_k}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}\cdot\epsilon_k^{(i)}\Big]\frac{\partial}{\partial x_l}\log\pi\Big] \\
&= \sum_{j=1}^{s}\frac{1}{n}\sum_{i=1}^{n}\Big[\frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}\cdot\epsilon_l^{(i)}+\sum_{\substack{k=1\\k\neq l}}^{p}\frac{\partial f}{\partial x_k}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}\cdot\epsilon_k^{(i)}\Big]\cdot\frac{\partial}{\partial x_l}\log\pi \\
&= \underbrace{\sum_{j=1}^{s}\frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}\cdot\frac{1}{n}\sum_{i=1}^{n}\Big[\epsilon_l^{(i)}\cdot\frac{\partial}{\partial x_l}\log\pi\Big]}_{ⓐ}+\underbrace{\sum_{j=1}^{s}\sum_{\substack{k=1\\k\neq l}}^{p}\frac{\partial f}{\partial x_k}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}\cdot\frac{1}{n}\sum_{i=1}^{n}\Big[\epsilon_k^{(i)}\cdot\frac{\partial}{\partial x_l}\log\pi\Big]}_{ⓑ}
\end{aligned}
$$

The first term ⓐ converges to $\frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}$ as observations for the gradient estimator expand:

$$
\begin{aligned}
ⓐ &\xrightarrow{\mathbb{P}} \frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}\cdot\mathbb{E}_{\pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|\dot{x}_l(\frac{j}{s}))}\Big[\epsilon_l\cdot\frac{\partial}{\partial x_l}\log\pi\Big] \\
&= \frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}\cdot\int_{-\infty}^{+\infty}\epsilon_l\cdot\pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|\dot{x}_l(\frac{j}{s}))\cdot\frac{\partial\log\pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|x_l)}{\partial x_l}\Big|_{x_l=\dot{x}_l(\frac{j}{s})}\,\mathrm{d}\epsilon_l \\
&\overset{z_l=\dot{x}_l(\frac{j}{s})+\epsilon_l}{=} \frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}\cdot\int_{-\infty}^{+\infty}(z_l-\dot{x}_l(\frac{j}{s}))\cdot\pi(z_l|\dot{x}_l(\frac{j}{s}))\cdot\frac{\partial\log\pi(z_l|x_l)}{\partial x_l}\Big|_{x_l=\dot{x}_l(\frac{j}{s})}\,\mathrm{d}z_l \\
&= \frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}\cdot\int_{-\infty}^{+\infty}(z_l-\dot{x}_l(\frac{j}{s}))\cdot\frac{\partial\pi(z_l|x_l)}{\partial x_l}\Big|_{x_l=\dot{x}_l(\frac{j}{s})}\,\mathrm{d}z_l \\
&= \frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}\cdot\Big[\frac{\partial}{\partial x_l}\int_{-\infty}^{+\infty}(z_l-\dot{x}_l(\frac{j}{s}))\cdot\pi(z_l|x_l)\,\mathrm{d}z_l\Big]\Big|_{x_l=\dot{x}_l(\frac{j}{s})} \\
&= \frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}\cdot\Big[\Big[\frac{\partial}{\partial x_l}\int_{-\infty}^{+\infty}z_l\cdot\pi(z_l|x_l)\,\mathrm{d}z_l\Big]\Big|_{x_l=\dot{x}_l(\frac{j}{s})}-\dot{x}_l(\frac{j}{s})\cdot\Big[\frac{\partial}{\partial x_l}\int_{-\infty}^{+\infty}\pi(z_l|x_l)\,\mathrm{d}z_l\Big]\Big|_{x_l=\dot{x}_l(\frac{j}{s})}\Big] \\
&\overset{(5)}{=} \frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}\cdot\Big[\Big[\frac{\partial}{\partial x_l}\mathbb{E}_{\pi(z_l|\dot{x}_l(\frac{j}{s}))}\Big[z_l\Big]\Big]\Big|_{x_l=\dot{x}_l(\frac{j}{s})}-\dot{x}_l(\frac{j}{s})\cdot 0\Big]
\end{aligned}
$$

Applying the premise stated in Equation 4 yields:

$$\text{ⓐ} \xrightarrow{\mathbb{P}} \frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})} \cdot \left[\frac{\partial}{\partial x_l} x_l\right]\Big|_{x_l=\dot{x}_l(\frac{j}{s})} = \frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}$$

The second term ⓑ produces 0 because of the independent sampling for different features:

$$\begin{aligned}
\text{ⓑ} &\xrightarrow{\mathbb{P}} \frac{\partial f}{\partial x_k}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})} \cdot \mathbb{E}_{\pi(\dot{\boldsymbol{x}}(\frac{j}{s})+\boldsymbol{\epsilon}|\dot{\boldsymbol{x}}(\frac{j}{s}))}\left[\epsilon_k \cdot \frac{\partial}{\partial x_l}\log\pi\right] \\
&\overset{\text{Independency}}{=} \frac{\partial f}{\partial x_k}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})} \cdot \mathbb{E}_{\pi(\dot{x}_k(\frac{j}{s})+\epsilon_k|\dot{x}_k(\frac{j}{s}))}\left[\epsilon_k\right] \cdot \mathbb{E}_{\pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|\dot{x}_l(\frac{j}{s}))}\left[\frac{\partial}{\partial x_l}\log\pi\right] \\
&\overset{(5)}{=} \frac{\partial f}{\partial x_k}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})} \cdot \mathbb{E}_{\pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|\dot{x}_l(\frac{j}{s}))}\left[\epsilon_k\right] \cdot 0 \\
&= 0
\end{aligned}$$

Replacing the terms in ② with the derived values:

$$\text{②} = \sum_{j=1}^{s} \frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}$$

Lastly, the element $O(||\boldsymbol{\epsilon}^{(i)}||^2)$ of the term ③ is bounded by $c_j^+||\boldsymbol{\epsilon}^{(i)}||^2$ and $c_j^-||\boldsymbol{\epsilon}^{(i)}||^2$, indicating:

$$c_j^-||\boldsymbol{\epsilon}^{(i)}||^2 \leq O(||\boldsymbol{\epsilon}^{(i)}||^2) \leq c_j^+||\boldsymbol{\epsilon}^{(i)}||^2$$

where c_j^-/c_j^+ is a negative/positive constant. The upper and lower bounds for ③ can then be written as:

$$\sum_{j=1}^{s}\frac{c_j^-}{n}\sum_{i=1}^{n}||\boldsymbol{\epsilon}^{(i)}||^2 \cdot \frac{\partial}{\partial x_l}\log\pi \leq \sum_{j=1}^{s}\frac{1}{n}\sum_{i=1}^{n}O(||\boldsymbol{\epsilon}^{(i)}||^2) \cdot \frac{\partial}{\partial x_l}\log\pi \leq \sum_{j=1}^{s}\frac{c_j^+}{n}\sum_{i=1}^{n}||\boldsymbol{\epsilon}^{(i)}||^2 \cdot \frac{\partial}{\partial x_l}\log\pi$$

Expanding the norm in the bound:

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}||\boldsymbol{\epsilon}^{(i)}||^2\frac{\partial}{\partial x_l}\log\pi &= \frac{1}{n}\sum_{i=1}^{n}\left[\epsilon_l^{(i)}\right]^2 \cdot \frac{\partial}{\partial x_l}\log\pi + \sum_{\substack{k=1\\k\neq l}}^{p}\frac{1}{n}\sum_{i=1}^{n}\left[\epsilon_k^{(i)}\right]^2 \cdot \frac{\partial}{\partial x_l}\log\pi \\
&\xrightarrow{\mathbb{P}} \mathbb{E}_{\pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|\dot{x}_l(\frac{j}{s}))}\left[\epsilon_l^2 \cdot \frac{\partial}{\partial x_l}\log\pi\right] + \sum_{\substack{k=1\\k\neq l}}^{p}\mathbb{E}_{\pi(\dot{\boldsymbol{x}}(\frac{j}{s})+\boldsymbol{\epsilon}|\dot{\boldsymbol{x}}(\frac{j}{s}))}\left[\epsilon_k^2 \cdot \frac{\partial}{\partial x_l}\log\pi\right] \\
&\overset{\text{Independency}}{=} \mathbb{E}_{\pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|\dot{x}_l(\frac{j}{s}))}\left[\epsilon_l^2 \cdot \frac{\partial}{\partial x_l}\log\pi\right] + \sum_{\substack{k=1\\k\neq l}}^{p}\mathbb{E}_{\pi(\dot{x}_k(\frac{j}{s})+\epsilon_k|\dot{x}_k(\frac{j}{s}))}\left[\epsilon_k^2\right] \cdot 0 \\
&= \int_{-\infty}^{+\infty}\epsilon_l^2 \cdot \pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|\dot{x}_l(\frac{j}{s})) \cdot \frac{\partial\log\pi(\dot{x}_l(\frac{j}{s})+\epsilon_l|x_l)}{\partial x_l}\Big|_{x_l=\dot{x}_l(\frac{j}{s})}\mathrm{d}\epsilon_l \\
&\overset{z_l=\dot{x}_l(\frac{j}{s})+\epsilon_l}{=} \int_{-\infty}^{+\infty}\left(z_l-\dot{x}_l(\frac{j}{s})\right)^2 \cdot \pi(z_l|\dot{x}_l(\frac{j}{s})) \cdot \frac{\partial\log\pi(z_l|x_l)}{\partial x_l}\Big|_{x_l=\dot{x}_l(\frac{j}{s})}\mathrm{d}z_l \\
&= \int_{-\infty}^{+\infty}(z_l^2 - 2\cdot z_l\cdot\dot{x}_l(\frac{j}{s}) + \dot{x}_l(\frac{j}{s})^2) \cdot \frac{\partial\pi(z_l|x_l)}{\partial x_l}\Big|_{x_l=\dot{x}_l(\frac{j}{s})}\mathrm{d}z_l \\
&= \frac{\partial}{\partial x_l}\left[\int_{-\infty}^{+\infty}z_l^2\pi(z_l|x_l)\,\mathrm{d}z_l - 2\dot{x}_l(\frac{j}{s})\int_{-\infty}^{+\infty}z_l\pi(z_l|x_l)\,\mathrm{d}z_l + \dot{x}_l(\frac{j}{s})^2\int_{-\infty}^{+\infty}\pi(z_l|x_l)\,\mathrm{d}z_l\right]\Big|_{x_l=\dot{x}_l(\frac{j}{s})} \\
&= \frac{\partial}{\partial x_l}\left[\mathbb{E}_{\pi(z_l|x_l)}\left[z_l^2\right] - 2\dot{x}_l(\frac{j}{s})\mathbb{E}_{\pi(z_l|x_l)}\left[z_l\right] + \dot{x}_l(\frac{j}{s})^2\cdot 1\right]\Big|_{x_l=\dot{x}_l(\frac{j}{s})} \qquad (6)
\end{aligned}$$

Denoting the variance of z_l with σ_l^2, the first term in Equation 6 is:

$$\frac{\partial}{\partial x_l}\mathbb{E}_{\pi(z_l|\dot{x}_l(\frac{j}{s}))}\left[z_l^2\right] = \frac{\partial}{\partial x_l}\left[\mathbb{E}^2_{\pi(z_l|\dot{x}_l(\frac{j}{s}))}\left[z_l\right] + \sigma_l^2\right] = \frac{\partial}{\partial x_l}\left[(x_l+\delta_l)^2 + \sigma_l^2\right]$$

where δ denotes any bias of the distribution mean $\mathbb{E}_{\pi(z_l|\dot{x}_l(\frac{j}{s}))}[z_l]$ to its location parameter x_l. Given that x_l, as a location parameter, has no control over the spread σ_l of the distribution, Equation 6 then becomes:

$$\begin{aligned}(6) &= \frac{\partial}{\partial x_l}\left[(x_l+\delta_l)^2 + \sigma_l^2 - 2\dot{x}_l(x_l+\delta_l) + \dot{x}^2\right]\Big|_{x_l=\dot{x}_l(\frac{j}{s})} \\ &= \left[2x_l + 2\delta_l - 2\dot{x}_l + 0\right]\Big|_{x_l=\dot{x}_l(\frac{j}{s})} = 2\cdot\delta_l \\ &\overset{(4)}{=} 0\end{aligned}$$

where the unbiasedness of the search distribution ensures the vanishing of the higher order residual. The upper and lower bounds of ③ can be updated as:

$$0 \le ③ \le 0 \Rightarrow ③ = 0$$

Combining ①, ②, and ③ we show that the gradient estimator converges to the actual gradient without bias, which allows rewriting the explanation as follows:

$$\xi_l = \frac{(\dot{x}_l - \mathring{x}_l)}{s}\cdot\sum_{j=1}^{s}\frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})} \tag{7}$$

The aggregation of gradient estimations over the interpolation $\dot{\boldsymbol{x}}(\frac{j}{s})$ approaches the true path integral as the interpolation interval $\frac{1}{s}$ becomes small:

$$\xi_l \overset{s\to\infty}{\to} \int_0^1 \frac{\partial f}{\partial x_l}\frac{\partial x_l}{\partial \alpha}\,\mathrm{d}\alpha$$

Applying the fundamental theorem for line integrals yields:

$$\begin{aligned}\sum_{k=1}^{p}\xi_k &= \int_0^1 \Big(\frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial \alpha} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial \alpha} + \ldots + \frac{\partial f}{\partial x_p}\frac{\partial x_p}{\partial \alpha}\Big)\,\mathrm{d}\alpha \\ &= \int_0^1 \nabla_{\boldsymbol{x}} f(\boldsymbol{x}(\alpha))\,\mathrm{d}\alpha \\ &= f(\dot{\boldsymbol{x}}) - f(\mathring{\boldsymbol{x}}) \\ \Rightarrow f(\mathring{\boldsymbol{x}}) + \sum_{k=1}^{p}\xi_k &= f(\dot{\boldsymbol{x}})\end{aligned}$$

□

In the proof, we show that the total attribution sum of GEEX converges in probability to the prediction difference between the baseline and the explicand. The error of the explainer arises from two sources: the error of the gradient estimator and the error associated with the approximation of the line integral. Optimizing the explainer's performance requires minimization of both errors.

B. Proof of Satisfaction on the Four Axioms

Theorem 3.1 states that GEEX fulfills the four fundamental axioms of attribution methods. This section gives the proof of these properties one by one.

B.1. Axiom: Insensitivity

Insensitivity (Dummy) states that the attribution to a feature on which the target model does not functionally depend should be zero. Formally, for a feature x_l, Insensitivity requires:

$$\xi_l = 0, \text{ if } \frac{\partial f}{\partial x_l} = 0 \text{ for } \boldsymbol{x} \in \mathbb{R}^p$$

Proof. Focusing on the *l*-th feature that should receive a zero attribution score, its attribution determined by GEEX can be written as follows according to Equation 7:

$$\xi_l = \frac{(x_l - \mathring{x}_l)}{s} \cdot \sum_{j=1}^{s} \frac{\partial f}{\partial x_l}\Big|_{\boldsymbol{x}=\dot{\boldsymbol{x}}(\frac{j}{s})}$$

Then applying the definition of Insensitivity reaches the end of the proof:

$$\xi_l = \frac{(x_l - \mathring{x}_l)}{s} \cdot \sum_{j=1}^{s} 0 = 0$$

□

Compared to Insensitivity, a similar but still slightly different property is *Missingness*. With the absence defined by some baseline value $\mathring{\boldsymbol{x}}$, this property requires attribution methods to distribute a zero value to the contribution ξ_l of an absent feature x_l, namely:

$$\xi_l = 0, \text{ if } x_l = \mathring{x}_l$$

Missingness differs from Insensitvitiy due to its reliance on the definition of a baseline. However, despite this difference, both properties are carried by GEEX. The proof of Missingness can be done in a single line:

$$\xi_l = \frac{(x_l - \mathring{x}_l)}{s} \cdot \sum_{j=1}^{s} \eta_l(\dot{\boldsymbol{x}}(\frac{j}{s})) \overset{x_l=\mathring{x}_l}{=} 0 \cdot \sum_{j=1}^{s} \eta_l(\dot{\boldsymbol{x}}(\frac{j}{s})) = 0$$

B.2. Axiom: Sensitivity

Sensitivity states that if the explicand and the baseline differing in one feature receive different predictions, then the differing feature should be assigned a non-zero importance score. The proof of Sensitivity is readily accessible with the help of the proof of Completeness.

Proof. Denoting the only different feature by x_l, the explanation outcome takes the following value:

$$\boldsymbol{\xi} = (0, \ldots, \xi_l, \ldots, 0)$$

because the other terms are canceled out according to Missingness given $x_k = \mathring{x}_k,\ \forall x_k \neq x_l$. Reusing the conclusion in the proof of Completeness yields:

$$\begin{aligned} \xi_l &= \xi_l + \sum_{\substack{k=1 \\ k\neq l}}^{p} 0 = \sum_{k=1}^{p} \xi_k \\ &\overset{\mathbb{P}}{\to} \int_0^1 \nabla_{\boldsymbol{x}} f(\boldsymbol{x}(\alpha)) \mathrm{d}\alpha \\ &= f(\boldsymbol{x}) - f(\mathring{\boldsymbol{x}}) \neq 0 \\ \Rightarrow \xi_l &\neq 0 \end{aligned}$$

□

B.3. Axiom: Implementation Invariance

For any two functionally equivalent models, *Implementation Invariance* indicates that the explanations for the decisions made by the two functionally equivalent models ought to be identical despite the different implementations. Intuitively, black-box explainers naturally satisfy Implementation Invariance as their explanation procedures do not consider or utilize any details about model implementations. However, we still give the formal proof for GEEX, which shows that our method aligns with the intuition.

Proof. Given two models $f_{\phi_1}(\cdot)$ and $f_{\phi_2}(\cdot)$, functional equivalence indicates:

$$f_{\phi_1}(\boldsymbol{x}) = f_{\phi_2}(\boldsymbol{x}),\ \forall \boldsymbol{x} \in \mathbb{R}^p$$

where ϕ denotes some learnable parameters in a model, which is used to indicate the implementation difference. The explanations $\boldsymbol{\xi}^{(f_{\phi_1})}$ and $\boldsymbol{\xi}^{(f_{\phi_2})}$ for the two models at an arbitrary point $\boldsymbol{x}$ hold:

$$\begin{aligned}
\boldsymbol{\xi}^{(f_{\phi_1})} &= \frac{(\boldsymbol{x} - \mathring{\boldsymbol{x}})}{n \cdot s} \circ \sum_{j=1}^{s} \sum_{i=1}^{n} f_{\phi_1}(\boldsymbol{z}^{(i)}) \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{z}^{(i)} | \boldsymbol{x}(\frac{j}{s})) \\
&\overset{f_{\phi_1}(\boldsymbol{x}) = f_{\phi_2}(\boldsymbol{x})}{=} \frac{(\boldsymbol{x} - \mathring{\boldsymbol{x}})}{n \cdot s} \circ \sum_{j=1}^{s} \sum_{i=1}^{n} f_{\phi_2}(\boldsymbol{z}^{(i)}) \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{z}^{(i)} | \boldsymbol{x}(\frac{j}{s})) \\
&= \boldsymbol{\xi}^{(f_{\phi_2})}
\end{aligned}$$

□

B.4. Axiom: Linearity

For any two functions $f_{\phi_1}(\cdot)$ and $f_{\phi_2}(\cdot)$, *Linearity* requires the explanation for the linear composition of the two functions $af_{\phi_1} + bf_{\phi_2}$ equaling the weighted sum of the separate explanations for them, namely:

$$\boldsymbol{\xi}^{(af_{\phi_1} + bf_{\phi_2})} = a \cdot \boldsymbol{\xi}^{(f_{\phi_1})} + b \cdot \boldsymbol{\xi}^{(f_{\phi_2})}$$

The Linearity of GEEX is proved below.

Proof.

$$\begin{aligned}
\boldsymbol{\xi}^{(af_{\phi_1} + bf_{\phi_2})} &= \frac{(\boldsymbol{x} - \mathring{\boldsymbol{x}})}{n \cdot s} \circ \sum_{j=1}^{s} \sum_{i=1}^{n} \Big[af_{\phi_1}(\boldsymbol{z}^{(i)}) + bf_{\phi_2}(\boldsymbol{z}^{(i)}) \Big] \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{z}^{(i)} | \boldsymbol{x}(\frac{j}{s})) \\
&= \frac{a(\boldsymbol{x} - \mathring{\boldsymbol{x}})}{n \cdot s} \circ \sum_{j=1}^{s} \sum_{i=1}^{n} f_{\phi_1}(\boldsymbol{z}^{(i)}) \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{z}^{(i)} | \boldsymbol{x}(\frac{j}{s})) + \frac{b(\boldsymbol{x} - \mathring{\boldsymbol{x}})}{n \cdot s} \circ \sum_{j=1}^{s} \sum_{i=1}^{n} f_{\phi_2}(\boldsymbol{z}^{(i)}) \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{z}^{(i)} | \boldsymbol{x}(\frac{j}{s})) \\
&= a \cdot \boldsymbol{\xi}^{(f_{\phi_1})} + b \cdot \boldsymbol{\xi}^{(f_{\phi_2})}
\end{aligned}$$

□

C. Complementary Information from Neighbors on the Path

In the last part of Section 3, we finalize the explainer as a dense approximation of the path integral with one-sample gradient estimators:

$$\boldsymbol{\xi} = \frac{(\boldsymbol{x} - \mathring{\boldsymbol{x}})}{n^*} \circ \sum_{\substack{\boldsymbol{\epsilon} \sim \pi(\cdot|\mathbf{0}) \\ \alpha \sim \mathcal{U}_{[0,1]}}} f(\boldsymbol{x}(\alpha) + \boldsymbol{\epsilon}) \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})$$

Intuitively, reducing the capacity of the gradient estimator may seem contradictory to the conclusion drawn from the proof of Completeness, which suggests that optimizing explanation quality requires minimization of estimator errors at both levels. However, this is not necessarily true because neighboring estimators can share complementary information and thus promote

their estimations. Before showing this complementary information, we first provide a formal definition of "neighboring estimators". The definition relies on the assumption of continuous differentiability of the target function $f(\cdot)$, a fundamental requirement for applying any gradient-based explainers, regardless of the accessibility setting.

Definition C.1. For an interval $[a, b]$ on a path $\boldsymbol{x}(\alpha)$ in which $f(\cdot)$ is locally linear, the one-sample estimators $\boldsymbol{\eta}_{n=1}(\boldsymbol{x}(\dot{\alpha}))$ for arbitrary points $\dot{\alpha} \in [a, b]$ are **neighboring** estimators.

Given the assumption that $f(\cdot)$ is continuously differentiable, it is always possible to find such an interval for any estimator on the path that determines its neighboring estimators. Next, we prove that the set of neighboring estimators with a size of m achieves the same level of precision as a m-sample estimator at some point on the interval $\alpha^* \in [a, b]$ denoted by $\boldsymbol{\eta}_{n=m}(\boldsymbol{x}(\alpha^*))$.

Proof. The collaborative estimation of neighbors $\{\boldsymbol{\eta}(\boldsymbol{x}(\dot{\alpha})) \mid \dot{\alpha} \in [a, b]\}$ can be written as:

$$\begin{aligned}\frac{1}{m}\sum_{\dot{\alpha}\sim\mathcal{U}_{[a,b]}} \boldsymbol{\eta}_{n=1}(\boldsymbol{x}(\dot{\alpha})) &\overset{(7)}{=} \frac{1}{m}\sum_{\dot{\alpha}\sim\mathcal{U}_{[a,b]}} \Big[\nabla_{\boldsymbol{x}} f(\boldsymbol{x}(\dot{\alpha})) + \boldsymbol{\sigma}^{(\dot{\alpha})}\Big] \\ &\overset{\text{Local linearity}}{=} \nabla_{\boldsymbol{x}} f(\boldsymbol{x}(a)) + \frac{1}{m}\cdot\sum_{\dot{\alpha}\sim\mathcal{U}_{[a,b]}} \boldsymbol{\sigma}^{(\dot{\alpha})}\end{aligned}$$

where $\boldsymbol{\sigma}^{(\dot{\alpha})}$ denotes the estimation error of one estimator. As an unbiased estimator, the error term follows some distribution $\mathcal{D}_{(0,\boldsymbol{\sigma}^{(\dot{\alpha})})}$, which has a mean at 0 and a variance of value $\left[\boldsymbol{\sigma}^{(\dot{\alpha})}\right]^2$. Recalling that $\boldsymbol{x}$ is the location parameter of the search distribution, the variance of an estimator is:

$$\begin{aligned}\left[\boldsymbol{\sigma}^{(\dot{\alpha})}\right]^2 &= \mathbb{V}\Big[f(\boldsymbol{x}(\dot{\alpha}) + \boldsymbol{\epsilon}) \cdot \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\Big] \\ &\overset{\text{Local linearity}}{=} \mathbb{V}\Big[(f(\boldsymbol{x}(a) + \boldsymbol{\epsilon}) + \delta(\dot{\alpha})) \cdot \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\Big]\end{aligned}$$

where $\delta(\dot{\alpha}) = f(\boldsymbol{x}(\dot{\alpha})) - f(\boldsymbol{x}(a))$. Expanding the form above yields:

$$\begin{aligned}\left[\boldsymbol{\sigma}^{(\dot{\alpha})}\right]^2 &= \mathbb{V}\Big[f(\boldsymbol{x}(a) + \boldsymbol{\epsilon}) \cdot \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\Big] + \mathbb{V}\Big[\delta(\dot{\alpha})\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\Big] + 2(\mathbb{E}\Big[\delta(\dot{\alpha}) f(\boldsymbol{x}(a) + \boldsymbol{\epsilon}) \cdot \left[\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\right]^2\Big] - 0) \\ &= \mathbb{V}\Big[f(\boldsymbol{x}(a) + \boldsymbol{\epsilon}) \cdot \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\Big] + \mathbb{E}\Big[\left[\delta(\dot{\alpha})\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\right]^2\Big] - 0 + 2\mathbb{E}\Big[\delta(\dot{\alpha}) f(\boldsymbol{x}(a) + \boldsymbol{\epsilon}) \cdot \left[\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\right]^2\Big] \\ &= \mathbb{V}\Big[f(\boldsymbol{x}(a) + \boldsymbol{\epsilon}) \cdot \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\Big] + \delta(\dot{\alpha})^2\mathbb{E}\Big[\left[\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\right]^2\Big] + 2\delta(\dot{\alpha})\mathbb{E}\Big[f(\boldsymbol{x}(a) + \boldsymbol{\epsilon}) \cdot \left[\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\right]^2\Big]\end{aligned}$$

For a fixed search distribution, the expectations in the last two terms are constant. Denoting them as $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ yields:

$$\left[\boldsymbol{\sigma}^{(\dot{\alpha})}\right]^2 = \mathbb{V}\Big[f(\boldsymbol{x}(a) + \boldsymbol{\epsilon}) \cdot \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\Big] + \delta(\dot{\alpha})^2\boldsymbol{c}_1 + 2\delta(\dot{\alpha})\boldsymbol{c}_2$$

where $\boldsymbol{c}_1$ is a constant matrix, and the elements in $\boldsymbol{c}_2$ are not necessarily identical. However, mirror sampling adopted in this work ensures the isotropicity of the search distribution, which brings convenience to the proof of complementary information. The isotropic distribution, in combination with local linearity, simplifies $\boldsymbol{c}_2$:

$$\begin{aligned}\boldsymbol{c}_2 &= \mathbb{E}\Big[f(\boldsymbol{x}(a) + \boldsymbol{\epsilon}) \cdot \left[\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\right]^2\Big] \\ &\overset{\text{Local linearity}}{=} f(\boldsymbol{x}(a)) \cdot \mathbb{E}\Big[\left[\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\right]^2\Big] + \mathbb{E}\Big[\nabla f(\boldsymbol{x}(\alpha))^T \cdot \boldsymbol{\epsilon} \cdot \left[\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\right]^2\Big] \\ &= f(\boldsymbol{x}(a)) \cdot \mathbb{E}\Big[\left[\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\right]^2\Big] + \sum_{k=1}^{p} \frac{\partial f}{\partial x_k} \cdot \mathbb{E}\Big[\epsilon_k \cdot \left[\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\right]^2\Big] \\ &\overset{\text{Isotropicity}}{=} f(\boldsymbol{x}(a)) \cdot \mathbb{E}\Big[\left[\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{\epsilon}|\mathbf{0})\right]^2\Big] + \sum_{k=1}^{p} \frac{\partial f}{\partial x_k} \cdot \mathbf{0} \\ &= f(\boldsymbol{x}(a))\boldsymbol{c}_1\end{aligned}$$

The variance of a one-sample estimator can be updated:

$$[\boldsymbol{\sigma}^{(\dot{\alpha})}]^2 = \mathbb{V}\Big[f(\boldsymbol{x}(a)+\boldsymbol{\epsilon})\cdot\nabla_{\boldsymbol{x}}\log\pi(\boldsymbol{\epsilon}|\mathbf{0})\Big] + \Big[\delta(\dot{\alpha})^2 + 2\delta(\dot{\alpha})\cdot f(\boldsymbol{x}(a))\Big]\boldsymbol{c}_1$$

The variance of the collaborative estimation denoted by $[\boldsymbol{\sigma}^{[a,b]}_{N=m}]^2$ is:

$$\begin{aligned}
[\boldsymbol{\sigma}^{[a,b]}_{N=m}]^2 &= \frac{1}{m^2}\sum_{\dot{\alpha}\sim\mathcal{U}_{[a,b]}}[\boldsymbol{\sigma}^{(\dot{\alpha})}]^2 \\
&= \frac{1}{m^2}\sum_{\dot{\alpha}\sim\mathcal{U}_{[a,b]}}\mathbb{V}\Big[f(\boldsymbol{x}(a)+\boldsymbol{\epsilon})\cdot\nabla_{\boldsymbol{x}}\log\pi(\boldsymbol{\epsilon}|\mathbf{0})\Big] + \Big[\delta(\dot{\alpha})^2 + 2\delta(\dot{\alpha})\cdot f(\boldsymbol{x}(a))\Big]\boldsymbol{c}_1 \\
&= \frac{1}{m}\mathbb{V}\Big[f(\boldsymbol{x}(a)+\boldsymbol{\epsilon})\cdot\nabla_{\boldsymbol{x}}\log\pi(\boldsymbol{\epsilon}|\mathbf{0})\Big] + \frac{1}{m^2}\sum_{\dot{\alpha}\sim\mathcal{U}_{[a,b]}}\Big[\delta(\dot{\alpha})^2 + 2\delta(\dot{\alpha})\cdot f(\boldsymbol{x}(a))\Big]\boldsymbol{c}_1 \\
&= \frac{1}{m}\mathbb{V}\Big[f(\boldsymbol{x}(a)+\boldsymbol{\epsilon})\cdot\nabla_{\boldsymbol{x}}\log\pi(\boldsymbol{\epsilon}|\mathbf{0})\Big] + \frac{1}{m^2}\sum_{\dot{\alpha}\sim\mathcal{U}_{[a,b]}}\Big[f(\boldsymbol{x}(\dot{\alpha}))^2 - f(\boldsymbol{x}(a))^2\Big]\boldsymbol{c}_1
\end{aligned}$$

Similarly, the variance of estimation error for an m-sample estimator is:

$$[\boldsymbol{\sigma}^{(\alpha)}_{N=m}]^2 = \frac{1}{m}\mathbb{V}\Big[f(\boldsymbol{x}(a)+\boldsymbol{\epsilon})\cdot\nabla_{\boldsymbol{x}}\log\pi(\boldsymbol{\epsilon}|\mathbf{0})\Big] + \frac{1}{m}\Big[f(\boldsymbol{x}(\alpha))^2 - f(\boldsymbol{x}(a))^2\Big]\boldsymbol{c}_1$$

Depending on the value of $f(\cdot)$ on the interval, the standard deviation of the collaborative estimator is bounded either by both endpoints of the segment when the minimum of $f(\boldsymbol{x}(\alpha))^2 - f(\boldsymbol{x}(a))^2$ is not on $[a,b]$:

$$\min(\boldsymbol{\sigma}^{(a)}_{N=m}, \boldsymbol{\sigma}^{(b)}_{N=m}) \leq \boldsymbol{\sigma}^{[a,b]}_{N=m} \leq \max(\boldsymbol{\sigma}^{(a)}_{N=m}, \boldsymbol{\sigma}^{(b)}_{N=m})$$

or by the minimum $\alpha_{min} \in [a,b]$ and the endpoint with a higher function outcome otherwise:

$$\boldsymbol{\sigma}^{(\alpha_{min})}_{N=m} \leq \boldsymbol{\sigma}^{[a,b]}_{N=m} \leq \max(\boldsymbol{\sigma}^{(a)}_{N=m}, \boldsymbol{\sigma}^{(b)}_{N=m})$$

Given the continuous differentiability and the bounded variance on the interval, there always exists a point $\alpha^* \in [a,b]$ such that:

$$\begin{aligned}
&\Big[f(\boldsymbol{x}(\alpha^*))^2 - f(\boldsymbol{x}(a))^2\Big]\boldsymbol{c}_1 = \frac{1}{m}\sum_{\dot{\alpha}\sim\mathcal{U}_{[a,b]}}\Big[f(\boldsymbol{x}(\alpha))^2 - f(\boldsymbol{x}(a))^2\Big]\boldsymbol{c}_1 \\
\Rightarrow\ &[\boldsymbol{\sigma}^{(\alpha^*)}_{N=m}]^2 = [\boldsymbol{\sigma}^{[a,b]}_{N=m}]^2
\end{aligned}$$

In other words, the collaborative estimator achieves the same level of precision as the m-sample estimator. □

D. Experimental Environment

The proposed approach and the designed experiments are implemented using *Python* 3.10.9 with standard packages. Specifically, these include *Numpy* 1.26.3, *Pytorch* of version 2.0.0, and *torchvision* 0.15.1. The *CUDA* version is 11.4 for GPU support. The experiments are conducted on a machine operated by Debian 11 with 32GB RAM. The machine possesses an Intel *i9-10980XE* CPU and an Nvidia *RTX A5500* GPU of 24GB VRAM.

E. Sample Explanations

Figures 6, 7, and 8 list more example sets containing explanations from all competitors. With an organization slightly different from the examples in the main body of the paper, each row in the three figures represents an example set, with the first *column* presenting explicands, followed by explanations derived through SG, IG, GEEX, RISE, and LIME in columns 2 to 6, accordingly. In the last column of the figures, the plots visualize the perturbation curves acquired by conducting the deletion process on an explicand guided by the corresponding explanations. The legend of the plot gives the normalized

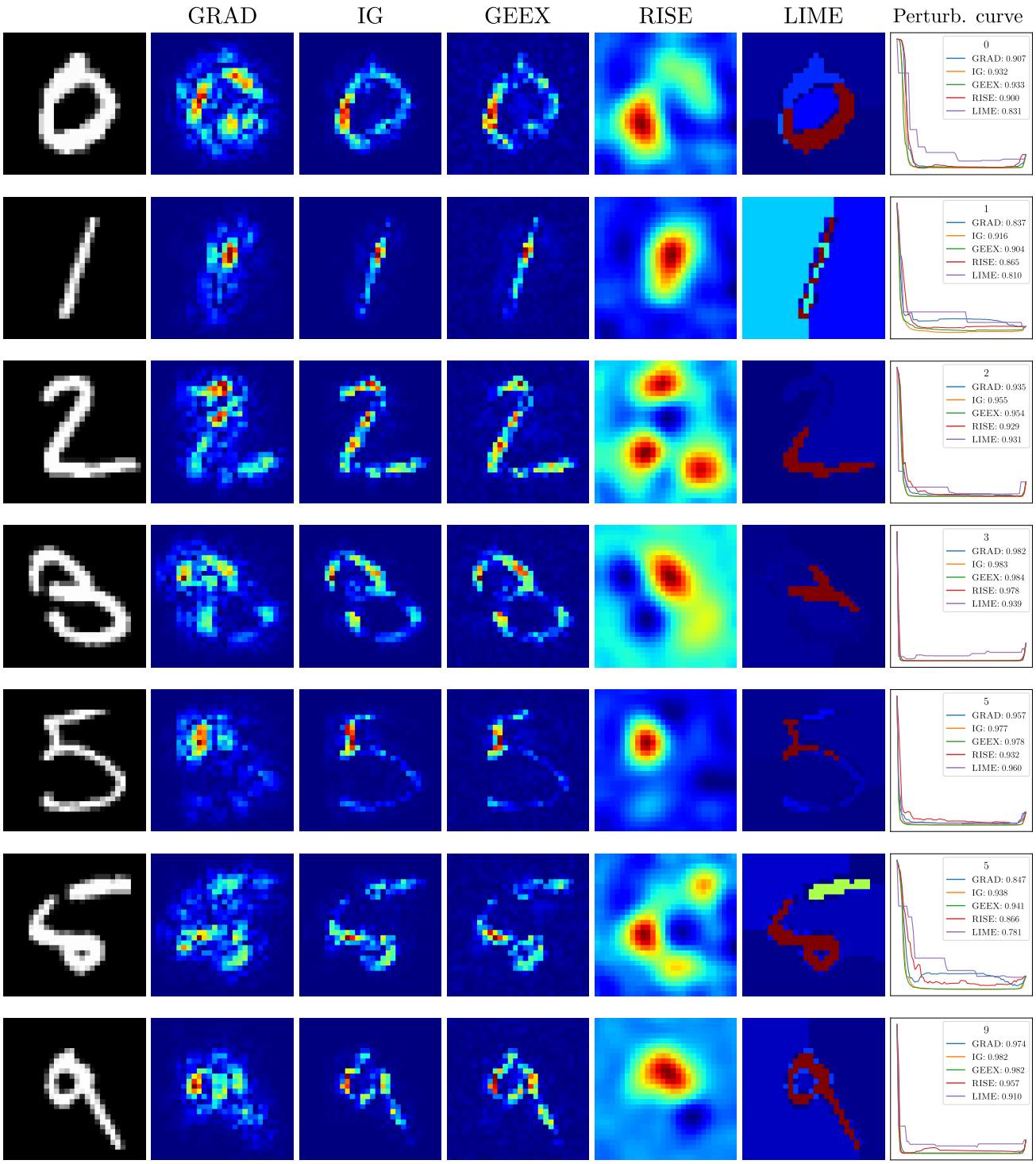

Figure 6. Explanations by all competitors for decisions by the CNN trained on MNIST, the last column visualizes the perturbation curves following corresponding explanations until all pixels are masked out. Legend titles refer to model predictions on corresponding explicands.

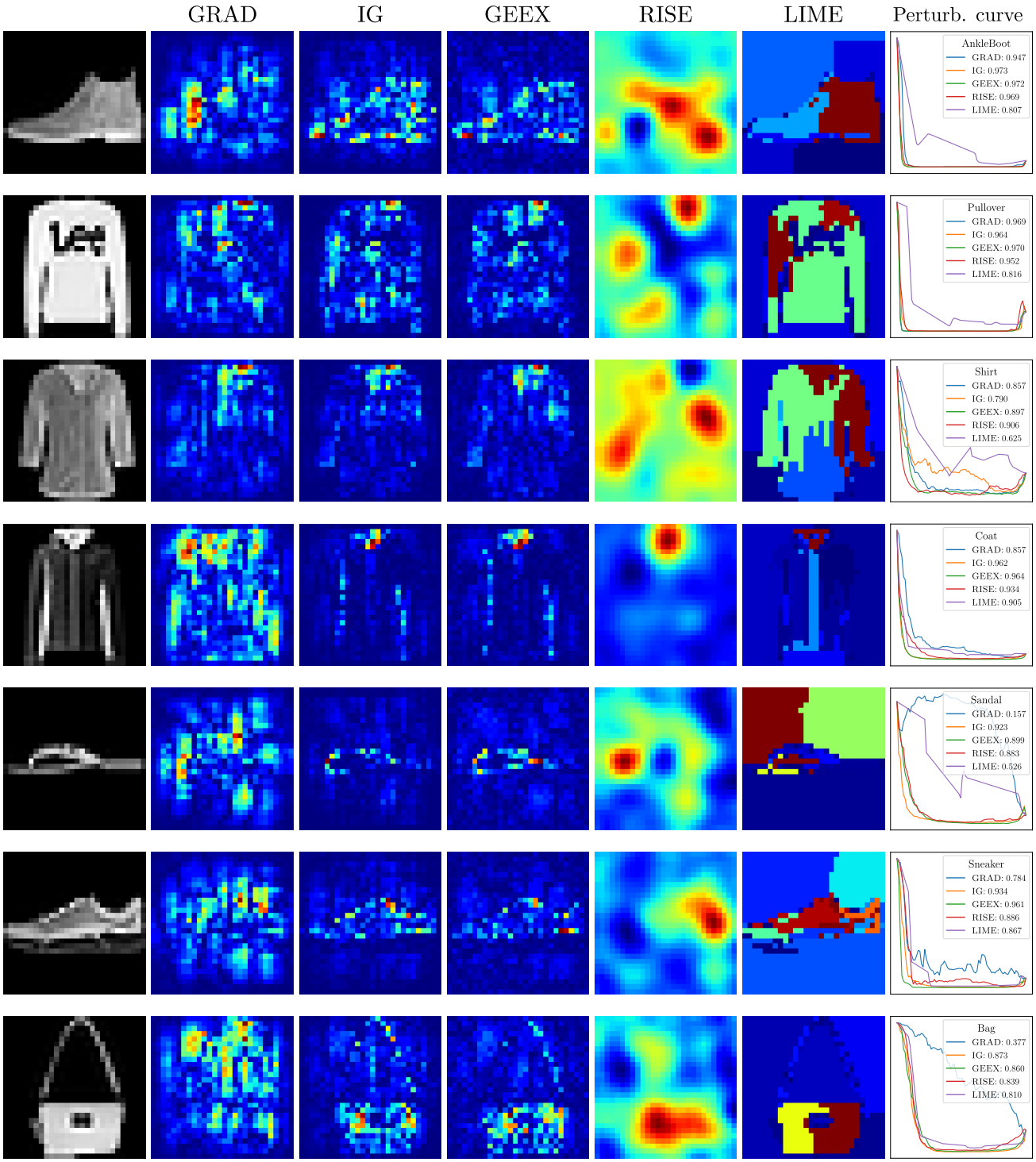

Figure 7. Explanations by all competitors for decisions by the CNN trained on Fashion-MNIST, the last column visualizes the perturbation curves following corresponding explanations until all pixels are masked out. Legend titles refer to model predictions on corresponding explicands.

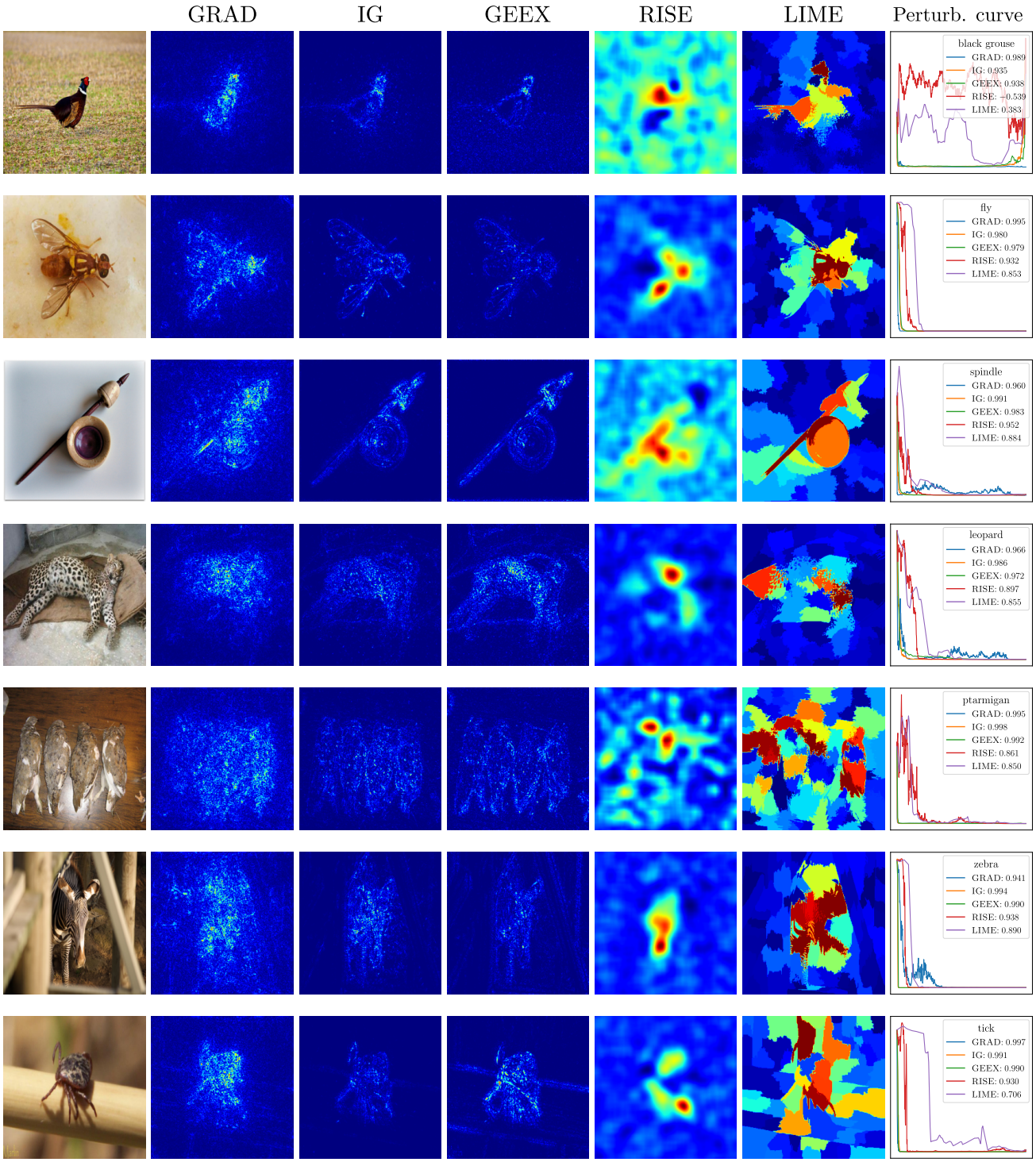

Figure 8. Explanations by all competitors for decisions by Inception V3 pre-trained on ImageNet, the last column visualizes the perturbation curves following corresponding explanations until all pixels are masked out. Legend titles refer to model predictions on corresponding explicands.

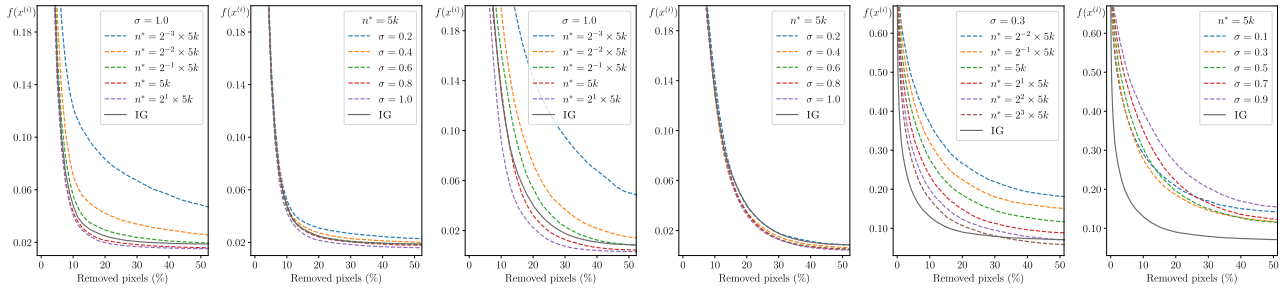

Figure 9. Effects of n^* and σ reported on (a) MNIST, (b) Fashion-MNIST, and (c) ImageNet. For each group of charts, the left plot refers to the changes in perturbation curves while increasing the sample set size m with fixed σ, the right plot presents the impact of changing the search range σ with m set to 5k. The solid line is a reference derived by IG.

AOPC scores achieved by the listed explainers, which are the float numbers following the competitor's name. The title of the legend indicates the original prediction of the target model on the explicand.

Similar to the observations in the previous examples, GEEX and IG identify homologous attribution structures in their explanations across the three test cases. The mostly identical AOPC scores achieved by the two methods agree with the intuitive visual similarity. The overlaps between their perturbation curves suggest that pixels are ranked similarly in the explanations by GEEX and IG, thereby steering similar deletion processes. Compared to the other two black-box approaches, GEEX delivers fine-grained explanations that improve both visual interpretability and explanation quality in terms of AOPC score. More specifically, when dealing with explicands containing more complicated patterns, GEEX points out low-level features that contribute to a decision rather than a few hot-regions as done by the black-box competitors. For example, in the fifth row of Fig.8, GEEX assigns attributions to pixels representing the relevant objects – "ptarmigan", whereas the explanations from RISE and LIME are more difficult to comprehend.

F. Effect of Hyperparameters

This section studies the effects of the most important hyperparameters in GEEX, namely the sample set size n^*, and the search distribution spread σ, to guide the selection of these values. During the test, we assume the two hyperparameters affect explanation quality independently, and study the impact of each individually while the other is set to a fixed value. For the deviation σ, we set the candidate values $\{0.2,\ 0.4,\ 0.6,\ 0.8,\ 1.0\}$ for the grayscale datasets, and $\{0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.9\}$ for ImageNet with n^* set fixed to $5k$ in all three cases. Regarding the test for the impact of n^*, we consider the candidate values in exponential growth with a base of $5k$, which is the value considered during the quantitative evaluation. More specifically, n^* takes value from $\{625,\ 1.25k,\ 2.5k,\ 5k,\ 10k\}$ for the smaller explicands with σ set to 1.0, and from $\{1.25k,\ 2.5k,\ 5k,\ 10k,\ 20k\}$ for higher-dimensional explicands with $\sigma = 0.3$.

The charts in Fig. 9 plot the perturbation curves of GEEX under different configurations. These plots are grouped according to the test scenarios with each of them zoomed into the first half of the deletion process to better visualize the differences among the curves. For each group, the plot on the left demonstrates the change of explanation quality when altering the sample set size n^*, and the right shows the impact of the search range σ. The dashed line represents the perturbation curves drawn by GEEX while the solid gray line derived by IG is considered as a reference.

Aligning with the expectation, n^* positively correlates to explanation quality in all scenarios. With the expanding sample set, the perturbation curve of GEEX converges to the reference as a result of the reduced estimation error, and it surpasses the reference when the sample set reaches a certain size. For the MNIST and Fashion-MNIST, the growing trend of areas above a perturbation curve slows down when the sample set reaches a size of roughly 5k in both cases. By contrast, enlarging the sample set improves explanation quality constantly on ImageNet due to the higher estimation variance caused by feature space expansion, indicating that GEEX has not yet reached its upper bound. However, it is noteworthy that the number of observations is in quadratic growth for maintaining the same level of estimation precision when the feature space expands, which means that matching or overcoming the white-box solution is more expensive in a higher-dimensional feature space.

The deviation σ has minor impacts while tested on MNIST and Fashion-MNIST. We explain the observation as a result of

the relatively simple target functions. For solving the simple classification tasks, the learned function $f(\cdot)$ has a lower level of non-linearity than the more complicated case, simplifying the choice of σ. The reason is that σ has less chance to exceed the local linearity range for a smoother $f(\cdot)$, and can derive reliable estimation even if it takes a relatively higher value. On the other hand, the value of σ plays a significant role in explaining decisions by InceptionV3. The higher non-linearity of $f(\cdot)$ brings the challenge of determining the optimal σ. The value of σ should be large enough to expose differences in model outcomes for gradient estimators, but at the same time, σ should be small enough to ensure the local linearity within the search range. Although locating the optimum under the named constraints sounds formidable, finding a proper choice for σ is trivial in practice with a binary search. Moreover, as the number of observations almost surely contributes to the results positively, the choice of n^* is more about the trade-off between explanation quality and computational expense. As the only parameter left to be determined in GEEX, selecting the value for σ is an affordable task, especially when compared to other approaches that have a large set of tunable hyperparameters.