Monte Carlo Multi-Feature Baseline Shapley (MMBS): An axiomatic attribution method for fine-grained explanations of image classification networks

Anonymous authors
Paper under double-blind review

Abstract

This paper presents the Multi-Feature Baseline Shapley (MBS) attribution method for explaining the outcome of a neural network for a given input. MBS generalizes the Integrated Gradients (IG) and Baseline Shapley (BShap) methods by introducing a step size parameter. When the step size is set to one, MBS equals BShap, and when it is set to the number of features, MBS equals IG. MBS is an axiomatic method, which means that it was designed to satisfy certain axioms (mathematical properties). These axioms ensure that the attribution maps relate to the neural network in appealing ways, for example, by preserving linearity or symmetry. We prove that MBS satisfies eight axioms that are also satisfied by IG and BShap. To quickly approximate MBS, this paper presents the Monte Carlo Multi-Feature Baseline Shapley (MMBS) method, which is an unbiased estimator of MBS. On image classification tasks, we show that MMBS also approximates a Monte Carlo estimate for BShap while being up to 20,000 times faster to compute. Furthermore, we compare MMBS to nine configurations of existing attribution methods on three image classification networks trained on either the Fashion MNIST or ImageNet1k dataset. MMBS has the best area under the deletion curve score on all three networks.

1 Introduction

What makes a dog a Dalmatian? That is one of the many distinctions that an image classification neural network must learn to achieve a good score on the popular ImageNet1k benchmarking dataset (Russakovsky et al., 2015). Image classification networks trained on ImageNet1k aim to classify images into one thousand classes. For a given input image, they compute a probability for each of these classes. To explain how an image classification network came to its decision, an attribution map (also called a heatmap, importance map, or saliency map) can be calculated. An attribution map provides an indication of how much each feature (pixel or color subpixel) of a given input image contributed to the calculated probability for a given class. There are many methods for calculating attribution maps, and Figure 1 shows the results of several popular methods on images of Dalmatian dogs. ImageNet1k contains more than one hundred classes of dog breeds, so a fine-grained attribution method is necessary to highlight the subtle differences between these classes. Dalmatian dogs can be distinguished from other dog breeds by their black spots, so you would expect attribution maps to highlight the regions that contain spots. However, multiple attribution map methods in Figure 1 produced results that are too smooth or too noisy to highlight the spots.

The weights of a neural network fully specify the network's behavior, so they can be considered an explanation of the network. However, a typical image classification network has millions of weights that are combined in a complex layered structure, making it almost impossible for a human to interpret the weights directly. An attribution map is much simpler for a human to comprehend, because it can be shown as an image. However, an attribution map cannot fully capture a neural network's behavior because it assigns only one score per feature. It is challenging to specify how an attribution map should approximate a neural network's behavior, since it should also explain the network's incorrect or unexpected decisions. Early attribution map methods relied on heuristics that were convenient to calculate (Simonyan et al., 2013; Zhou et al., 2016; Selvaraju et al., 2017). Recently, there has been growing interest in axiomatic attribution methods (Sundararajan et al., 2017; Lundstrom & Razaviyayn, 2025; Sundararajan & Najmi, 2020), which are methods that are

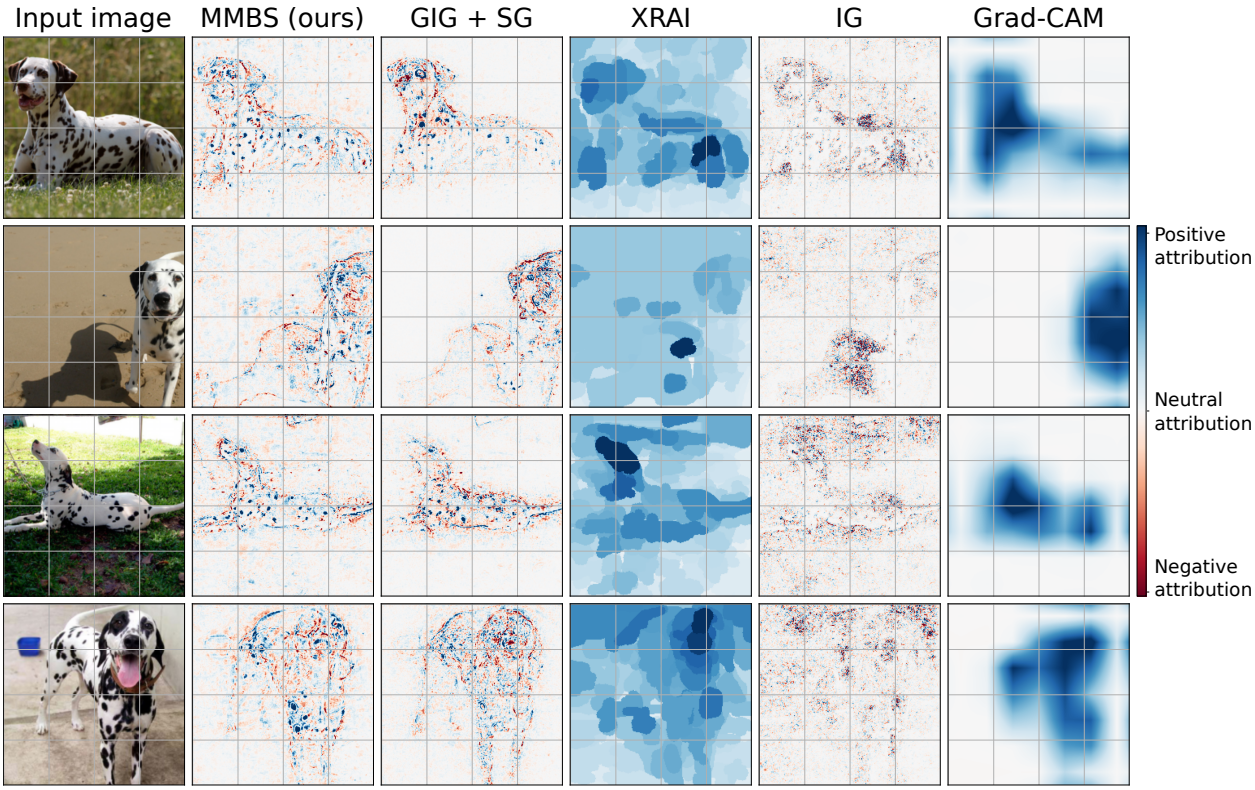

Figure 1: Attribution maps calculated with different methods for the class Dalmation on images of Dalmations (IG = Integrated Gradients, GIG + SG = Guided Integrated Gradients with SmoothGrad). There are large differences between the results of different methods. Dalmatians can be distinguished from other dogs by their black spots, and MMBS highlights this most clearly.

designed to satisfy certain axioms (mathematical properties). The axioms define the attribution problem more precisely, and they are chosen so that the attribution map relates to the neural network in an appealing way, for example, by preserving properties of the neural network, such as linearity or symmetry.

Integrated Gradients (IG) is an axiomatic attribution method based on several appealing axioms (Sundararajan et al., 2017; Lundstrom et al., 2022; Lundstrom & Razaviyayn, 2025). However, when applying IG to image classification problems, the resulting attribution maps often have a noisy appearance, as shown in Figure 1. Several papers have proposed modifications of IG to obtain less visually-noisy attribution maps, such as the XRAI (Kapishnikov et al., 2019) and the Guided Integrated Gradients + Smoothgrad (Kapishnikov et al., 2021; Smilkov et al., 2017) methods, which are also displayed in Figure 1. Baseline Shapley (BShap) is another axiomatic attribution method that satisfies many of the same axioms as IG (Sundararajan & Najmi, 2020). However, the computational cost of BShap grows in the order of $O(n!n)$ with the number of input features n, making it impossible in practice to use BShap on images. It is possible to approximate BShap using a Monte Carlo approach (Castro et al., 2009; Mitchell et al., 2022), but that is still computationally expensive, so, to the knowledge of the authors, it has only been used on very small images (Ancona et al., 2019) or with a very low number of samples (10 samples) (Yeh et al., 2022).

In this paper, we present the Multi-Feature Baseline Shapley (MBS) method. MBS generalizes IG and BShap by introducing a step size parameter (Section 3.2). When the step size is set to one, MBS equals BShap, and when it is set to the number of features, MBS equals IG. We prove that MBS satisfies eight axioms that are satisfied by both IG and BShap (Section 3.3 and Appendix B). We also present the Monte Carlo Multi-Feature Baseline Shapley (MMBS) method, which is an unbiased estimator of MBS (Section 3.4). For image classification on the Fashion MNIST and ImageNet1k datasets, we show that even when the step size is large, MMBS attribution maps are very similar to a Monte Carlo estimate of BShap (Castro et al., 2009; Yeh

et al., 2022), but they can be computed up to 20,000 times faster (Section 4.2). MMBS also produces results that are significantly less visually noisy than IG (Figure 1). Moreover, we show that for small, medium (ResNet50), and large (Vision Transformer) neural networks trained on image classification tasks (Fashion MNIST or ImageNet), the MMBS attribution maps achieve lower area under deletion curve (AUDC) scores than all other state-of-the-art attribution methods we compared it with (Section 4.3).

2 Related work

Some of the earlier attribution methods proposed in the literature approximate the behavior of neural networks using heuristics that are convenient to calculate. Simonyan et al. (2013) used the gradient as an attribution map because features with a large gradient should affect the neural network's output most strongly. However, the gradient at a single point is insufficient to capture the full decision-making process due to the highly non-linear nature of neural networks. The Class Activation Map (CAM) (Zhou et al., 2016) and Grad-CAM (Selvaraju et al., 2017) methods use the fact that in convolutional and pooling layers, the inputs and outputs are strongly spatially correlated. This approach can therefore only be applied to convolutional neural networks consisting of many convolutional and pooling layers, with a fully connected block at the end. Moreover, the image size of a Grad-CAM attribution map is equal to the image size of the activations just before the fully connected block, which is typically very small, resulting in a blurry attribution map.

The concept of axiomatic attribution methods was introduced together with the Integrated Gradients (IG) method (Sundararajan et al., 2017). The idea of axiomatic attribution was derived from similar methods in game theory. Specifically, IG is based on the Aumann-Shapley (AS) (Aumann & Shapley, 1974) method for solving the cost-sharing problem in game theory. IG attribution maps are calculated by integrating the gradients over the straight-line path between a baseline and the neural network input. More rigorous proofs, and proofs for additional axioms have been derived by Lundstrom et al. (2022); Lundstrom & Razaviyayn (2025); Sundararajan & Taly (2018).

IG often produces attribution maps with a noisy appearance. Typically, the absolute attribution values are higher on the regions of interest; however, pixels with positive and negative attribution values may be very close to each other, even within seemingly even regions. To obtain attribution maps with less visual noise, several modifications have been proposed. Many of these methods modify the integration path. Guided Integrated gradients (GIG) (Kapishnikov et al., 2021) aims to avoid integrating over regions with large unrelated changes in the gradient. To achieve this, the integration path is chosen adaptively so that in each step, only the features with the lowest partial derivatives are modified. SAMP (Zhang et al., 2024) aims to concentrate the attributions to a small number of features. It optimizes the integration path to maximize the variance of the attributions under a Brownian motion assumption. In BlurIG (Xu et al., 2020), the integration path is chosen so that the input image is increasingly blurred in each integration step. In Adversarial Integration (Pan et al., 2021), the integration path is equal to a path that was used to generate an adversarial example (Goodfellow et al., 2014), and the results of multiple of these paths are averaged to obtain the final result. Important Direction Integrated Gradients (IDGI) (Yang et al., 2023) modifies the integration approach instead of the path, so that in every step along the path, a component in the direction of the local gradient is added. However, none of these modifications of IG have been proven to satisfy new axioms.

The BShap method is another axiomatic attribution method (Sundararajan & Najmi, 2020). It is based on the Shapley-Shubik method (Shubik, 1962) for solving the cost-sharing problem in game theory. BShap satisfies many of the same axioms as IG, but there are also some differences (Sundararajan & Najmi, 2020; Friedman & Moulin, 1999; Sprumont, 1998). The use of the BShap method on images has been fairly limited due to the high computational cost (Ancona et al., 2019; Yeh et al., 2022). However, similar methods have been used to calculate attributions for neural networks with fewer input features (Lundberg & Lee, 2017), and variants have been derived for several specific problems, such as trees or linear models (Chen et al., 2023).

Another category of methods takes the output of an existing attribution method (e.g., IG, GradCAM, or IDGI) and modifies it to obtain additional properties, much like adding a regularizer to an optimization problem. XRAI (Kapishnikov et al., 2019) segments locally similar regions into superpixels, and then ranks the attribution of the superpixels based on a (sub)pixel-wise attribution method. SmoothGrad (Smilkov

et al., 2017) adds noise to the input image before calculating the attribution map. This is done multiple times with independently sampled noise, and the results are averaged. For many attribution map methods, this results in an attribution map that is visually less noisy. Gildenblat (2021) applied multiple random rotations, scalings (90-110%), and flips, and calculated an attribution map for each transformed version of the input image. The inverse transformations were applied to the results, and these results are averaged to obtain an attribution map that is invariant to these transformations.

3 Methods

In this section, we will first introduce the notation and a formal definition of baseline attribution methods (Section 3.1). After that, we will introduce MBS (Section 3.2) and describe its axioms (Section 3.3). We will then introduce MMBS, a Monte Carlo estimator of MBS (Section 3.4), and discuss its computational cost relative to Monte Carlo estimators of IG and BShap (Section 3.5). Finally, we will describe how the AUDC metric can be used to evaluate attribution map methods (Section 3.6).

3.1 Notation and problem formulation

For ease of comparison, we will mostly follow the notation of Lundstrom & Razaviyayn (2025). We denote functions as uppercase letters (A), integers and vectors as lowercase letters (a), scalars as lowercase Greek letters (α), and sets as calligraphic uppercase letters ($\mathcal{A}$). For vectors $a, b \in \mathbb{R}^n$ with $a_i < b_i \, \forall i$ we use the notation $[a, b]$ to denote a subset of $\mathbb{R}^n$ so that $x \in [a, b] \iff a_i \leq x_i \leq b_i \, \forall i$. $\mathcal{F}^2(a, b)$ is the set of neural networks that are composed of real analytic functions and ReLU nonlinearities, and that have $[a, b]$ as their domain. For any given $F \in \mathcal{F}^2(a, b)$, its domain is denoted as $\mathcal{D}_F$. We aim to explain a neural network $F \in \mathcal{F}^2(a, b)$ on a certain input (the explicant) $\bar{x} \in \mathcal{D}_F$, relative to some baseline $x' \in \mathcal{D}_F$. Baseline attribution methods are any function in the form $A : [a, b] \times [a, b] \times \mathcal{F}^2(a, b) \to \mathbb{R}^n$. The domain of a given A is denoted as $\mathcal{D}_A$. We also follow some notation from (Friedman & Moulin, 1999): We denote a combination of elements of vectors $x, y \in \mathbb{R}^n$ based on a set $\mathcal{S}$, by $z = x\backslash^{\mathcal{S}} y$ where $z_i = x_i$ if $i \in \mathcal{S}$ and $z_i = y_i$ otherwise. Moreover, we denote the partial derivative of F at $x \in \mathcal{D}_F$ in the i-th coordinate as $\partial_i F(x)$. Finally, $\mathcal{R}$ is the set of all possible orderings of the numbers $[1, 2, \ldots, n]$, so it contains $n!$ orderings.

3.2 Multi-Feature Baseline Shapley (MBS)

The Multi-feature Baseline Shapley (MBS) method is calculated by dividing all features in $\bar{x}$ and x' into steps of size $m \in \mathbb{R}$ for every possible ordering $r \in \mathcal{R}$. The number of steps is $\frac{n}{m}$ rounded up to the nearest integer $\lceil \frac{n}{m} \rceil$. In every step, an IG attribution map A^{IG} is calculated, and the values of this attribution map are only non-zero for the features included in that step:

$$A^{\mathrm{MBS}}(\bar{x}, x', F) = \frac{1}{n!} \sum_{r \in \mathcal{R}} \sum_{k=1}^{\lceil \frac{n}{m} \rceil} A^{\mathrm{IG}}(\bar{x}\backslash^{\mathcal{I}(r,k)} x', \bar{x}\backslash^{\mathcal{I}(r,k-1)} x', F), \tag{1}$$
$$\mathcal{I}(r, k) = \{l : r_l \leq km\}.$$

A^{IG} is the IG method, and it is defined as follows (Sundararajan et al., 2017):

$$A_i^{\mathrm{IG}}(\bar{x}, x', F) = (\bar{x}_i - x'_i) \int_{\zeta=0}^{1} \partial_i F((1-\zeta)x' + \zeta\bar{x}) d\zeta. \tag{2}$$

MBS is a generalization of the IG method, because MBS is equal to IG when $m = n$.

MBS is also a generalization of the BShap method, because MBS equals BShap when $m = 1$. BShap is defined as follows (Sundararajan & Najmi, 2020):

$$A_i^{\mathrm{BShap}}(\bar{x}, x', F) = \frac{1}{n!} \sum_{r \in \mathcal{R}} F(\bar{x}\backslash^{\{l|r_l \leq i\}} x') - F(\bar{x}\backslash^{\{l|r_l \leq (i-1)\}} x'). \tag{3}$$

3.3 Axioms

Eight axioms that are satisfied by MBS are defined below. The proofs that MBS satisfies these axioms are provided in Appendix B. For the axioms of Completeness, Dummy/Sensitivity(b), Linearity, Symmetry-Preserving, and Affine Scale Invariance, it has been proven that they are satisfied by both IG and BShap

(Sundararajan & Najmi, 2020). For the remaining three axioms of Implementation invariance (Sundararajan et al., 2017), Sensitivity(a) (Lundstrom et al., 2022), and Non-Decreasing Positivity (Lundstrom et al., 2022), it has so far only been proven that they are satisfied by IG. We say that MBS satisfies an axiom if it satisfies it for every step size m. Therefore, the proofs for MBS also prove that BShap satisfies all eight axioms. We use the same names and definitions for the axioms as Lundstrom & Razaviyayn (2025) and Lundstrom et al. (2022), but we have slightly modified the wording for consistency. In all definitions, we use A to denote any baseline attribution method that satisfies the axiom, such as A^{MBS}, A^{IG}, or A^{BShap}.

Definition of Implementation Invariance: A is not a function of model implementation, but solely a function of the mathematical mapping of the model's domain to the range.

Implementation invariance is appealing because it ensures that networks that behave in the same way are also explained in the same way.

Definition of Completeness: If $(\bar{x}, x', F) \in \mathcal{D}_A$, then $\sum_{i=1}^{n} A_i(\bar{x}, x', F) = F(\bar{x}) - F(x')$.

Completeness provides an intuitive scale to the attribution values: The attribution value of a feature can be interpreted as how much that feature contributed to the change in neural network outcome.

Definition of Sensitivity(a): If $(\bar{x}, x', F) \in \mathcal{D}_A$, $F(\bar{x}) \neq F(x')$, and $\bar{x}, x'$ only vary in the i-th component, i.e. $\bar{x}_i \neq x'_i$, and $\bar{x}_j = x'_j \ \forall j \neq i$, then $A_i(\bar{x}, x', F) \neq 0$.

Definition of Dummy/Sensitivity(b): If $(\bar{x}, x', F) \in \mathcal{D}_A$ and $\partial_i F \equiv 0$, then $A_i(\bar{x}, x', F) = 0$.

Sensitivity(a) ensures that when only one feature is different between the baseline and the explicant, and this difference results in a difference in neural network outcome, that feature is not assigned a zero value in the attribution map. Dummy/Sensitivity(b) ensures that features that do not affect the neural network outcome are assigned a value of zero.

Definition of Linearity: If $(\bar{x}, x', F), (\bar{x}, x', G) \in \mathcal{D}_A$ and $\alpha, \beta \in \mathbb{R}$, then $(\bar{x}, x', \alpha F + \beta G) \in \mathcal{D}_A$ and $A(\bar{x}, x', \alpha F + \beta G) = \alpha A(\bar{x}, x', F) + \beta A(\bar{x}, x', G)$.

Definition of Symmetry-Preserving: Suppose that $(\bar{x}, x', F) \in \mathcal{D}_A$ and i and j are indices. Let $S_{ij}(x)$ be the function that swaps the values of x_i and x_j. Then if $F(x) = F(S_{ij}(x))$ for any $x \in \mathcal{D}_F$, and $\bar{x} = S_{ij}(\bar{x})$ and $x' = S_{ij}(x')$, we have $A_i(\bar{x}, x', F) = A_j(\bar{x}, x', F)$.

Linearity and Symmetry-Preserving ensure that the attribution map preserves certain properties of the network.

Definition of Non-Decreasing Positivity: If $(\bar{x}, x', F) \in \mathcal{D}_A$ and F is non-decreasing[1] from x' to $\bar{x}$ then $A_i(\bar{x}, x', F) \geq 0$ for every index i.

Non-Decreasing Positivity avoids assigning negative attribution values when the neural network is non-decreasing from the baseline to the explicant.

Definition of Affine Scale Invariance (ASI): Suppose that $(\bar{x}, x', F) \in \mathcal{D}_A$, $c, d \in \mathbb{R}$ with $c \neq 0$, and i is an index. Let T be an affine transformation of element i, so that $T(x) := (x_1, \cdots, cx_i + d, \cdots, x_n)$. Then we have $A(\bar{x}, x', F) = A(T(\bar{x}), T(x'), F \circ T^{-1})$.

Affine Scale Invariance (ASI) is appealing because it makes the attribution map invariant to affine unit changes; for example, when two networks perform equivalent computations but one expects an input in degrees Celsius and the other in degrees Fahrenheit, they will have the same attribution maps.

3.4 Monte Carlo Multi-Feature Baseline Shapley (MMBS)

The number of orderings in $\mathcal{R}$ grows factorially with the number of features, which rapidly grows to an infeasible amount to compute. For example, for the very small 28×28 pixel images of the MNIST (LeCun et al., 2010) dataset, the number of orderings is already approximately 3.2×10^{1930}.

To approximate the BShap method in a feasible amount of time, Yeh et al. (2022) proposed a Monte-Carlo approach based on the ApproShapley method of Castro et al. (2009). We will call this method Monte Carlo

[1] A definition for non-decreasing based on (Lundstrom et al., 2022) is provided in Appendix B.7.

BShap (MBShap). Instead of averaging over all possible orderings, it averages over a randomly sampled set of orderings $\widetilde{\mathcal{R}}$:

$$A_i^{\text{MBShap}}(\bar{x}, x', F) = \frac{1}{|\widetilde{\mathcal{R}}|} \sum_{r \in \widetilde{\mathcal{R}}} F(\bar{x}\setminus^{\{l|r_l \leq i\}} x') - F(\bar{x}\setminus^{\{l|r_l \leq (i-1)\}} x'). \tag{4}$$

In the standard approach of calculating integrated gradients, the integral is approximated by a constant number of equally spaced steps (Sundararajan et al., 2017). We introduce an alternative approach using Monte Carlo estimation, which we call Monte Carlo Integrated Gradients (MIG). Here $\widetilde{\mathcal{Z}}$ is a set of randomly sampled values ζ from a uniform distribution over the range $[0, 1]$. The value of the integral is estimated by evaluating the gradient of F for all values ζ in $\widetilde{\mathcal{Z}}$:

$$A_i^{\text{MIG}}(\bar{x}, x', F) = (\bar{x}_i - x'_i) \frac{1}{|\widetilde{\mathcal{Z}}|} \sum_{\zeta \in \widetilde{\mathcal{Z}}} \partial_i F((1-\zeta)x' + \zeta \bar{x}). \tag{5}$$

By combining the ideas from MBShap and MIG, we can create a Monte Carlo estimator for MBS, which we call Monte Carlo Multi-feature Baseline Shapley (MMBS). Here $\widetilde{\mathcal{S}}$ is a set of randomly sampled orderings r and vectors z of length $\lceil \frac{n}{m} \rceil$ with each element z_k sampled from a uniform distributon over the range $[0, 1]$:

$$\begin{aligned} A_i^{\text{MMBS}}(\bar{x}, x', F) = \frac{1}{|\widetilde{\mathcal{S}}|} \sum_{(r,z) \in \widetilde{\mathcal{S}}} \sum_{k=1}^{\lceil \frac{n}{m} \rceil} ((\bar{x}\setminus^{\mathcal{I}(r,k)} x' - \bar{x}\setminus^{\mathcal{I}(r,k-1)} x') \\ \partial_i F((1-z_k)(\bar{x}\setminus^{\mathcal{I}(r,k-1)} x') + z_k(\bar{x}\setminus^{\mathcal{I}(r,k)} x'))). \end{aligned} \tag{6}$$

MMBS is an unbiased estimator of MBS. The proof is provided in Appendix A.

3.5 Computational cost per sample

The definitions of MBShap, MIG, and MMBS in the previous section are all given for one feature. However, they are typically calculated for the entire image for every sample, which makes it possible to reuse neural network evaluations or gradient calculations. In MBShap, $n + 1$ evaluations of F have to be calculated per sample. MIG only requires one gradient calculation per sample. In MMBS, the gradient of F has to be calculated $\lceil \frac{n}{m} \rceil$ times per sample. Calculating a gradient is more computationally expensive than evaluating a neural network. However, when the step size m is large, MMBS can be significantly faster than MBShap, because of the lower number of calls.

3.6 The area under the deletion curve (AUDC) evaluation metric

The area under the deletion curve (AUDC) is a common metric to evaluate attribution maps (Petsiuk et al., 2018; Kapishnikov et al., 2019; 2021). A deletion curve shows how the outcome of the neural network changes when the features of $\bar{x}$ are replaced by corresponding features in a baseline $x'' \in \mathcal{D}_F$ in the order of their ranking in the attribution map. To calculate a deletion curve, first, the ranking q of the features in the attribution map is determined: q_i is 1 if feature i is the feature with the highest attribution value, q_i is 2 if feature i has the second highest attribution, et cetera. The deletion curve $C_k(\bar{x}, x'', F, q)$ is defined as follows, where k is the number of removed features:

$$C_k(\bar{x}, x'', F, q) = F(\bar{x}\setminus^{\{l|q_l \geq k\}} x''). \tag{7}$$

The area under the deletion curve (AUDC) is defined as follows:

$$M(\bar{x}, x'', F, q) = \frac{1}{n+1} \sum_{k=0}^{n} C_k(\bar{x}, x'', F, q). \tag{8}$$

When $F(\bar{x})$ is high, and $F(x'')$ is low, the AUDC can be used as a performance metric, and a lower AUDC is considered a better performance. The reasoning behind this is that higher attributed features in $\bar{x}$ should play

a large role in the decision of the network. Setting these features to the value of the corresponding baseline feature should lower the outcome of the neural network more than changing a lower-ranked feature.

To lower the computation time, we approximated Equation 8 by sampling 200 equally spaced points along the deletion curve and assuming linear changes between these points. We implemented this using the `trapz` function in the Numpy Python library.

4 Experiments

In this section, we will first describe the datasets and neural networks that were used (Section 4.1). Then, we will show how the step size and number of samples affect the outcome of MMBS, and that MMBS can be used to approximate BShap (Section 4.2). After that, we will compare AUDC scores of MMBS and several competing methods on three neural networks (Section 4.3). Finally, we will investigate the influence of the attribution and evaluation baselines on attribution maps and their AUDC scores (Section 4.4).

4.1 Datasets and neural networks

Evaluating an attribution method requires a dataset and a network trained on this dataset. We used the Fashion MNIST (Xiao et al., 2017) and ImageNet1k (Russakovsky et al., 2015) datasets. To enable fast prototyping on the Fashion MNIST dataset, we used a very lightweight convolutional neural network (CNN) architecture consisting of two convolutional layers and two fully connected layers (Oikarinen, 2021). On ImageNet1k, we used two neural network architectures: A ResNet with a depth of 50 layers (ResNet50) (He et al., 2016), and a Vision Transformer(ViT) with a patch size of 16x16 pixels (ViT-B/16) (Dosovitskiy et al., 2021). For both networks on ImageNet1K, we used pretrained weights from the torchvision library (Marcel & Rodriguez, 2010). On the grayscale images of Fashion MNIST, each pixel is a feature. In the color images of ImageNet1K, each color subpixel is a feature. When displaying the attribution maps of color images, we show the sum of the different color subpixels.

4.2 MMBS as a fast approximation of BShap

We performed a parameter sweep over the number of iterations $|\widetilde{\mathcal{S}}|$ and the number of steps $\frac{n}{m}$, on one image of the Fashion MNIST dataset. The results are shown in Figure 2. These results suggest that MBShap converges at around 256 iterations. This is a massive reduction in iterations compared to the $n!$ iterations of BShap. Moreover, MMBS appears to be a close approximation of BSHap at a low number of steps (around 8 or 16), further reducing the computational costs.

We also compared the convergence of MBShap and MMBS on the larger ImageNet images, using both the ResNet and the ViT architectures. Given the long computation time of MBShap, we again only used one image per network, and the results are shown in Figure 3. MMBS was used with 8 steps, and was significantly faster to compute than MBShap: 3516 times faster on the ResNet50 and 20251 times faster on the ViT-B/16. Nevertheless, at 160 iterations, both methods converge to very similar images.

4.3 Comparison with other methods

We compared MMBS to the existing methods of IG (Sundararajan et al., 2017), GIG (Kapishnikov et al., 2021), XRAI (Kapishnikov et al., 2019), and Grad-CAM (Selvaraju et al., 2017). MMBS, IG, and GIG were also paired with Smoothgrad (Smilkov et al., 2017). For MMBS, we used 8 steps, 1024 samples, and an all-zero baseline. For IG, we used 256 steps and an all-zero baseline. For GIG, we used the implementation from the Saliency library, which is developed by the same group that wrote the GIG paper, but it uses a slightly different approach to bound the maximum step size, and by default, it uses a higher fraction of features that can be modified in each step (25% instead of 10%). We calculated the unbounded version of GIG as described in the paper (GIG (paper)), and the default values of the Saliency library (GIG (Saliency)). In both versions, we used 256 steps and an all-zero baseline. For SmoothGrad, we used a noise standard deviation of $0.15(\max(\bar{x}) - \min(\bar{x}))$, and 25 samples. For MMBS + SG, we used 128 instead of 1024 samples in each MMBS calculation to reduce the total computation time. For XRAI, we tried two variants: XRAI (B + W) is based on the average of IG with a black baseline and IG with a white baseline, which is how XRAI was used in its original publication (Kapishnikov et al., 2019). XRAI (zero) is based on IG with an all-zero baseline. Grad-CAM was not applied to the ViT model because it did not have a suitable CNN architecture.

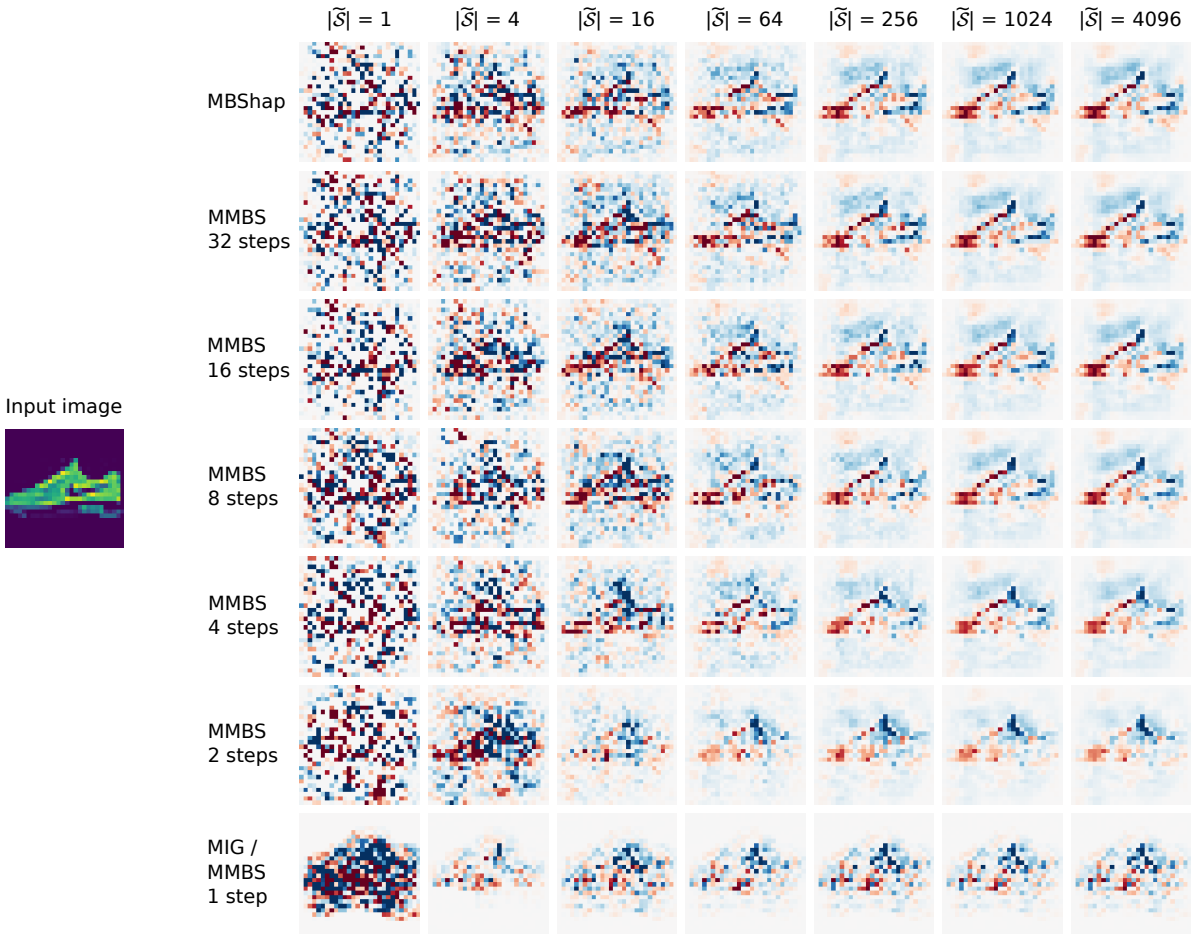

Figure 2: This figure shows heatmaps of an image of a sneaker calculated using the IG, MBShap, and MMBS methods for a small CNN trained on Fashion MNIST. The same orderings r were used when calculating the MBShap and MMBS attribution maps. All images use the same colormap where blue denotes positive attributions and red denotes negative attributions. MBShap converged at a fairly low number of samples (around 256). MMBS closely approximated MBSHap at a fairly low number of steps (around 8).

Each method was applied to all three neural networks, and AUDC scores were calculated using a baseline of all zeros. From the Fashion MNIST dataset, 1000 randomly sampled images from the test set were used, and from the ImageNet dataset, the first image from each class in the validation set was used, also resulting in 1000 images. AUDC scores were only included when the network outcome on the input image was at least 0.2 and on the baseline image was at most 0.05, which was the case for 839, 747, and 893 images, respectively, on the Fashion MNIST, ResNet, and ViT networks.

The average deletion curves are shown in Figure 4 and the AUDC scores are shown in Table 1. MMBS had the lowest AUDC on all three networks. Another interesting observation is that IG and both versions of GIG had a lower AUDC on all networks when combined with SmoothGrad. However, when MMBS was combined with SmoothGrad, the AUDC scores became slightly higher.

4.4 The effect of the baselines on the attribution and deletion curve

Like IG (Sturmfels et al., 2020), the result of MMBS depends on the chosen baseline. To illustrate this, we calculated IG and MMBS attribution maps of the ResNet50 network for five different baselines on an image of a toucan, which has a black and an almost white region. The baselines are: zeros before normalization (Black), ones before normalization (White), zeros after normalization (Zero), uniform noise over the full range of possible colors (Noise), and a Gaussian blurred version of the input image with a σ of 25 pixels (Blur).

Figure 3: The convergence of MBShap and MMBS attribution maps on a ResNet and a Vision Transformer network. MMBS produces results similar to MBShap at a much lower computation time.

Table 1: Comparison of AUDC scores on different datasets. The best score for each network is shown in boldface.

Method	Fashion MNIST AUDC	ImageNet Resnet AUDC	ImageNet ViT AUDC
MMBS (ours)	**0.051** ± 0.052	**0.018** ± 0.041	**0.025** ± 0.039
MMBS + SG	0.054 ± 0.049	0.029 ± 0.043	0.031 ± 0.049
IG	0.182 ± 0.182	0.076 ± 0.083	0.130 ± 0.162
IG + SG	0.081 ± 0.071	0.042 ± 0.066	0.083 ± 0.152
GIG (paper)	0.314 ± 0.290	0.130 ± 0.143	0.078 ± 0.164
GIG (paper) + SG	0.135 ± 0.184	0.041 ± 0.070	0.027 ± 0.082
GIG (Saliency)	0.175 ± 0.182	0.056 ± 0.080	0.117 ± 0.155
GIG (Saliency) + SG	0.078 ± 0.067	0.037 ± 0.060	0.074 ± 0.140
XRAI (B + W)	0.342 ± 0.226	0.200 ± 0.126	0.425 ± 0.243
XRAI (zero)	0.261 ± 0.185	0.210 ± 0.128	0.416 ± 0.240
GradCAM	0.488 ± 0.187	0.160 ± 0.107	N.A

Neural networks are often normalized so that the inputs of the training set have a mean of zero, which was also done in this paper. Therefore, a black image does not correspond to an input value of all zeros. The results are shown in Figure 5. When applying the neural network on the baseline, the output probability on the class toucan was less than 0.5% on all baselines. Nevertheless, the attribution maps look different for each baseline.

When evaluating attribution methods using the AUDC, there are two baselines: the attribution baseline x', and the evaluation baseline x''. We calculated AUDC values for IG and MMBS using the ResNet50 network for every combination of the five baseline choices from the previous paragraph (and Figure 5). We used the first image from each class in the validation set of ImageNet1k, and AUDC scores were included only when the network output on the input image was at least 0.2 and on the baseline image was at most 0.05. This was the case for 747, 747, 747, 746, and 743 images for the black, white, zero, noise, and blur evaluation baselines, respectively. Table 2 shows the results of this experiment. For each evaluation baseline, the lowest AUDC was achieved by MMBS using the same attribution baseline.

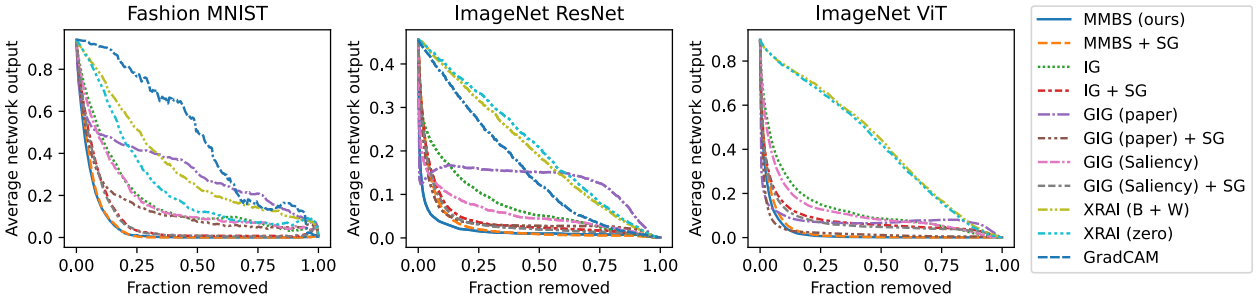

Figure 4: The average deletion curves for different attribution map methods on three networks.

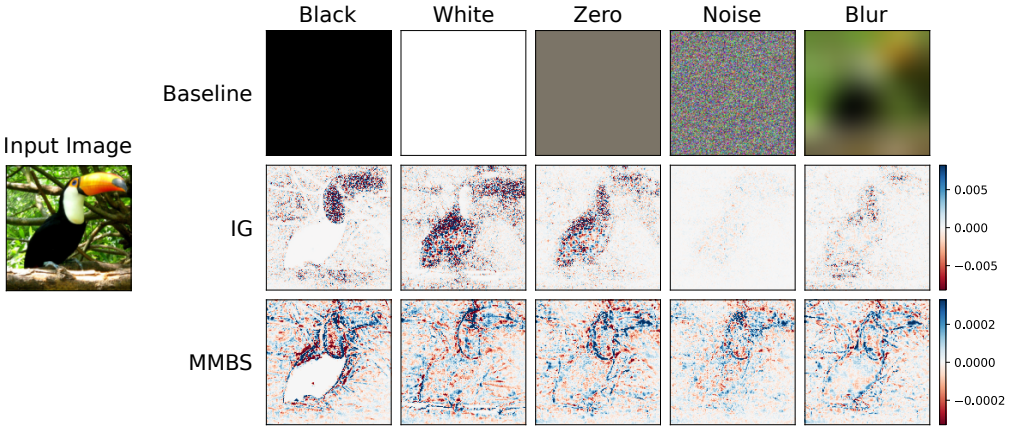

Figure 5: IG and MMBS Attribution map results with different baselines. MMBS was configured with 8 steps and 1024 samples, and IG with 256 steps. With both methods, the outcome is affected by the baseline.

5 Discussion and future work

In this section, we will first discuss how the differences between IG and BShap may be characterized by their different mathematical axioms (Section 5.1). We will then briefly discuss the potential of MMBS in other applications than image classification (Section 5.2). After that, we will discuss how to choose a baseline, and whether distributions of baselines may be a better choice for modeling neutrality (Section 5.3).

Table 2: Area under deletion curve (AUDC) metrics for different combinations of baselines used in the calculation (rows) and evaluation (columns) of IG and MMBS attribution maps. The lowest score for each AUDC metric is shown in bold numbers.

Baseline	Method	Black AUDC	White AUDC	Zero AUDC	Noise AUDC	Blur AUDC
Black	IG	0.038 ± 0.050	0.096 ± 0.075	0.142 ± 0.102	0.079 ± 0.064	0.181 ± 0.124
	MMBS	**0.014** ± 0.031	0.218 ± 0.170	0.160 ± 0.145	0.070 ± 0.085	0.203 ± 0.158
White	IG	0.107 ± 0.082	0.035 ± 0.040	0.128 ± 0.104	0.069 ± 0.062	0.165 ± 0.115
	MMBS	0.205 ± 0.162	**0.014** ± 0.025	0.121 ± 0.128	0.033 ± 0.050	0.152 ± 0.136
Zero	IG	0.094 ± 0.079	0.070 ± 0.065	0.076 ± 0.084	0.057 ± 0.051	0.117 ± 0.106
	MMBS	0.110 ± 0.113	0.085 ± 0.099	**0.018** ± 0.039	0.016 ± 0.023	0.045 ± 0.077
Noise	IG	0.062 ± 0.080	0.034 ± 0.052	0.063 ± 0.083	0.011 ± 0.012	0.098 ± 0.105
	MMBS	0.057 ± 0.094	0.024 ± 0.050	0.022 ± 0.046	**0.006** ± 0.012	0.047 ± 0.074
Blur	IG	0.071 ± 0.072	0.053 ± 0.061	0.085 ± 0.093	0.048 ± 0.048	0.090 ± 0.107
	MMBS	0.091 ± 0.093	0.074 ± 0.081	0.035 ± 0.062	0.021 ± 0.030	**0.021** ± 0.040

5.1 Additional axioms of IG and BShap

In Section 3.3, we proved that IG, BShap, and MMBS have several axioms in common. However, the experiments in Section 4 make it apparent that IG and MMBS/BShap can provide significantly different attributions. This implies that IG and BShap each satisfy additional axioms that the other method does not.

Lundstrom & Razaviyayn (2025) proved several additional axioms of IG. One way to distinguish a method from all other methods is to prove that it is the only method that satisfies a certain combination of axioms. This is called a unique characterization. Lundstrom & Razaviyayn (2025) also presented four ways to uniquely characterize IG. For many of the axioms used in these unique characterizations, we know that MBS and BShap satisfy them too, so MBS and BShap have to not satisfy at least one axiom of the remaining axioms in each characterization. Using this reasoning, we can derive that MBS and BShap do not satisfy the axioms of Proportionality and Symmetric Monotonicity.

In the cost-sharing literature, several additional axioms of the Shapley-Shubik (SS) method (Shubik, 1962) have been studied. SS is equal to BShap when you assume that the baseline x' consists of all zeros, that $F(x') = 0$, and that F is non-decreasing, positive, and continuously differentiable. Friedman & Moulin (1999) showed that SS has the axiom of demand monotonicity, and the Aumann-Shapley method (and therefore also IG) does not have this axiom. Moreover, they provided a unique characterization of SS using this axiom. Sprumont (1998) showed that SS satisfies the axiom of Ordinality, which is a generalization of the Affine Scale Invariance axiom. They also provided a unique characterization of SS using this axiom. It would be interesting future work to investigate whether these results can be generalized to BShap on the broader function space $\mathcal{F}^2$ used in this paper and by Lundstrom & Razaviyayn (2025). Sundararajan & Najmi (2020) showed that BShap maintains the axioms of SS when the baseline is not zero, but they did not address that neural networks are often not nondecreasing, positive, and continuously differentiable.

5.2 Other applications than image classification

In this paper, we focused on experiments with image classification problems. This was done to limit the scope of the paper and because image classification is a very popular problem to explain using attribution methods. However, the MMBS method can also be used in the attribution of other models. It would be interesting future work to investigate whether MMBS is also a close approximation of MBShap for other types of inputs, such as the problems surveyed by Chen et al. (2023).

5.3 Defining a neutral baseline

Similar to much of the existing literature (Sundararajan et al., 2017; Sundararajan & Najmi, 2020; Kapishnikov et al., 2021), we assume in this paper that a single image can serve as a neutral baseline. This is the case in the original cost-sharing problem, where the outcome of F is guaranteed to be zero on an all-zero input (Shubik, 1962). Moreover, in certain cases, a specific baseline may be used to obtain a specific explanation of a neural network (Mamalakis et al., 2023), or to explain only the changes over time (Schut et al., 2024). However, in many cases, one baseline does not accurately capture the concept of neutrality (Sturmfels et al., 2020). In image classification models, the outputs always sum to one, so it's impossible to have a baseline that has a value of zero for all classes. In models trained on ImageNet1k, this is alleviated by the large number of classes, so any baseline is likely to have a network output close to zero for almost all classes. Still, in Section 4.4, we showed that different baselines may lead to strongly different attribution maps even when, for all these baselines, the network output on the target class is low.

A potentially better definition of a neutral baseline is the data distribution conditioned on the features that haven't been removed (Lundberg & Lee, 2017). BShap, MMBS, and deletion curves could be adapted to this definition of neutrality by averaging in every step over many random samples from the conditional data distribution. In practice, the conditional data distribution is often unavailable, but it may be interesting future work to approximate it by conditional sampling using a diffusion model (Lugmayr et al., 2022) or another generative AI model. Some existing approaches that are used in the calculation of attribution maps or deletion curves can be considered to approximate the conditional data distribution: The Expected Gradients method (Erion et al., 2021) averaged over IG attribution maps with baselines that were randomly sampled from the training set, which closely approximates sampling from the unconditional data distribution. Rong

et al. (2022) used inpainting of the missing pixels to calculate deletion maps, so the results are conditioned on the data, but not random. MMBS showed a good AUDC performance (Table 2) when the attribution and evaluation baselines were the same. It would be interesting future work to investigate whether AUDC scores are also low when neutrality is modeled in other ways, as long as it is the same in the calculation of the attribution map and the AUDC.

6 Conclusion

In this paper, we presented the MBS and MMBS methods for calculating attribution maps. MBS generalizes the IG and BShap methods and satisfies eight axioms that IG and BShap also satisfy. MMBS is an unbiased estimator of MBS. On image classification tasks, we showed that MMBS can not only serve as a fast approximation to MBS but also to BShap, yielding results similar to MBShap with a speedup of up to 20,000 times. Moreover, we compared MMBS with existing methods across three image classification networks, and it achieved the lowest AUDC metric on all three networks. All in all, MMBS is an attribution method with a strong theoretical foundation and an acceptable computation time that yields state-of-the-art AUDC scores.

Code and data availability

All code used in the experiments of this paper is included in the supplementary material. The ImageNet1k (Russakovsky et al., 2015) and Fashion MNIST (Xiao et al., 2017) datasets are already publicly available.

References

Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International conference on machine learning*, pp. 272--281. PMLR, 2019.

Robert J Aumann and Lloyd S Shapley. *Values of non-atomic games*. Princeton University Press, 1974.

Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *Computers & operations research*, 36(5):1726--1730, 2009.

Hugh Chen, Ian C Covert, Scott M Lundberg, and Su-In Lee. Algorithms to estimate shapley value feature attributions. *Nature Machine Intelligence*, 5(6):590--601, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott M Lundberg, and Su-In Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature machine intelligence*, 3(7):620--631, 2021.

Eric Friedman and Herve Moulin. Three methods to share joint costs or surplus. *Journal of economic Theory*, 87(2):275--312, 1999.

Jacob Gildenblat. Pytorch library for cam methods. https://github.com/jacobgil/pytorch-grad-cam, 2021.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770--778, 2016.

Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Viégas, and Michael Terry. Xrai: Better attributions through regions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4948--4957, 2019. doi: https://doi.org/10.1109/iccv.2019.00505.

Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5050--5058, 2021.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs*, 2010.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461--11471, 2022.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Daniel Lundstrom and Meisam Razaviyayn. Four axiomatic characterizations of the integrated gradients attribution method. *Journal of Machine Learning Research*, 26(177):1--31, 2025.

Daniel Lundstrom, Tianjian Huang, and Meisam Razaviyayn. A rigorous study of integrated gradients method and extensions to internal neuron attributions. In *International Conference on Machine Learning*, pp. 14485--14508. PMLR, 2022.

Antonios Mamalakis, Elizabeth A Barnes, and Imme Ebert-Uphoff. Carefully choose the baseline: Lessons learned from applying xai attribution methods for regression tasks in geoscience. *Artificial Intelligence for the Earth Systems*, 2(1):e220058, 2023. doi: https://doi.org/10.1175/aies-d-22-0058.1.

Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, pp. 1485--1488, 2010. doi: https://doi.org/10.1145/1873951.1874254.

Rory Mitchell, Joshua Cooper, Eibe Frank, and Geoffrey Holmes. Sampling permutations for shapley value estimation. *Journal of Machine Learning Research*, 23(43):1--46, 2022.

Tuomas Oikarinen. Training a nn to 99% accuracy on mnist in 0.76 seconds. https://github.com/tuomaso/train_mnist_fast, 2021.

Deng Pan, Xin Li, and Dongxiao Zhu. Explaining deep neural network models with adversarial gradient integration. In *Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: randomized input sampling for explanation of black-box models. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, 2018.

Yao Rong, Tobias Leemann, Vadim Borisov, Gjergji Kasneci, and Enkelejda Kasneci. A consistent and efficient evaluation strategy for attribution methods. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18770--18795. PMLR, 2022.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211--252, 2015. doi: https://doi.org/10.1007/s11263-015-0816-y.

Dirk Elias Schut, Rachael Maree Wood, Rob Schouten, Robert van Liere, Tristan van Leeuwen, and Kees Joost Batenburg. Longitudinal ct scanning for explainable early detection of postharvest disorders: The 'braeburn'browning case. *Available at SSRN 4924886*, 2024.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618--626, 2017.

Martin Shubik. Incentives, decentralized control, the assignment of joint costs and internal pricing. *Management science*, 8(3):325--343, 1962.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. doi: https://doi.org/10.48550/arXiv.1706.03825.

Yves Sprumont. Ordinal cost sharing. *Journal of Economic Theory*, 81(1):126--162, 1998.

Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Visualizing the impact of feature attribution baselines. *Distill*, 5(1), 2020. doi: https://doi.org/10.23915/distill.00022.

Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International conference on machine learning*, pp. 9269--9278. PMLR, 2020.

Mukund Sundararajan and Ankur Taly. A note about: Local explanation methods for deep neural networks lack sensitivity to parameter values, 2018. URL https://arxiv.org/abs/1806.04205.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319--3328. PMLR, 2017.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Shawn Xu, Subhashini Venugopalan, and Mukund Sundararajan. Attribution in scale and space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9680--9689, 2020.

Ruo Yang, Binghui Wang, and Mustafa Bilgic. Idgi: A framework to eliminate explanation noise from integrated gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23725--23734, 2023.

Chih-Kuan Yeh, Kuan-Yun Lee, Frederick Liu, and Pradeep Ravikumar. Threading the needle of on and off-manifold value functions for shapley explanations. In *International Conference on Artificial Intelligence and Statistics*, pp. 1485--1502. PMLR, 2022.

Borui Zhang, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Path choice matters for clear attributions in path methods. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=gzYgsZgwXa.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921--2929, 2016.

A Proof that MMBS is an unbiased estimator of MBS

Proof *MMBS is an unbiased estimator of MBS*
We will first prove that MMBS with one sample ($\widetilde{\mathcal{S}} = \{(\tilde{r}, \tilde{z})\}$) is an unbiased estimator of MBS. $\tilde{r}$ and every element $\tilde{z}_k$ of $\tilde{z}$ are independent random variables, so the expected values can be split. $\tilde{r}$ is sampled from a distribution that has a probability of $\frac{1}{n!}$ for each ordering in $\mathcal{R}$. Each $\tilde{z}_k$ is sampled from a uniform

distribution over the range $[0, 1]$.

$$
\begin{aligned}
&\mathbb{E}_{\tilde{r},\tilde{z}}\left[\sum_{k=1}^{\lceil \frac{n}{m} \rceil}((\bar{x}\backslash^{\mathcal{I}(\tilde{r},k)}x' - \bar{x}\backslash^{\mathcal{I}(\tilde{r},k-1)}x')\partial_i F((1-\tilde{z}_k)(\bar{x}\backslash^{\mathcal{I}(\tilde{r},k-1)}x') + \tilde{z}_k(\bar{x}\backslash^{\mathcal{I}(\tilde{r},k)}x')))\right] \\
&= \mathbb{E}_{\tilde{r}}\left[\sum_{k=1}^{\lceil \frac{n}{m} \rceil}(\bar{x}\backslash^{\mathcal{I}(\tilde{r},k)}x' - \bar{x}\backslash^{\mathcal{I}(\tilde{r},k-1)}x')\,\mathbb{E}_{\tilde{z}_k}\left[\partial_i F((1-\tilde{z}_k)(\bar{x}\backslash^{\mathcal{I}(\tilde{r},k-1)}x') + \tilde{z}_k(\bar{x}\backslash^{\mathcal{I}(\tilde{r},k)}x'))\right]\right] \\
&= \mathbb{E}_{\tilde{r}}\left[\sum_{k=1}^{\lceil \frac{n}{m} \rceil}(\bar{x}\backslash^{\mathcal{I}(\tilde{r},k)}x' - \bar{x}\backslash^{\mathcal{I}(\tilde{r},k-1)}x')\int_{\zeta=0}^{1}\partial_i F((1-\zeta)(\bar{x}\backslash^{\mathcal{I}(\tilde{r},k-1)}x') + \zeta(\bar{x}\backslash^{\mathcal{I}(\tilde{r},k)}x'))d\zeta\right] \\
&= \mathbb{E}_{\tilde{r}}\left[\sum_{k=1}^{\lceil \frac{n}{m} \rceil}A^{\mathrm{IG}}(\bar{x}\backslash^{\mathcal{I}(\tilde{r},k)}x', \bar{x}\backslash^{\mathcal{I}(\tilde{r},k-1)}x', F)\right] \\
&= \frac{1}{n!}\sum_{r\in\mathcal{R}}\sum_{k=1}^{\lceil \frac{n}{m} \rceil}A^{\mathrm{IG}}(\bar{x}\backslash^{\mathcal{I}(r,k)}x', \bar{x}\backslash^{\mathcal{I}(r,k-1)}x', F) \\
&= A_i^{\mathrm{MBS}}(\bar{x}, x', F)
\end{aligned}
\tag{9}
$$

MMBS with multiple samples is the average of multiple independent single-sample MMBS calls. Since these are all unbiased estimators, their average is also unbiased. □

B Proofs that MBS satisfies the axioms

B.1 Implementation invariance

Definition of Implementation Invariance: A is not a function of model implementation, but solely a function of the mathematical mapping of the model's domain to the range.

Proof *MBS satisfies implementation invariance*
MBS only uses the output or the gradient of the full model (F) in its calculations. Therefore, it satisfies implementation invariance. □

B.2 Completeness

Definition of Completeness: If $(\bar{x}, x', F) \in \mathcal{D}_A$, then $\sum_{i=1}^{n} A_i(\bar{x}, x', F) = F(\bar{x}) - F(x')$.

Proof *MBS satisfies completeness.*
Because IG satisfies completeness (Lundstrom & Razaviyayn, 2025), every IG call within MBS can be written as the difference between two calls to F. By summing all IG calls with the same ordering r, most of these calls will cancel out, except the first startpoint $F(x')$ and the last endpoint $F(\bar{x})$. For every $r \in \mathcal{R}$, this sum satisfies completeness, so MBS satisfies completeness.

$$
\begin{aligned}
\sum_{i=1}^{n} A_i^{\mathrm{MBS}}(\bar{x}, x', F) &= \sum_{i=1}^{n}\frac{1}{n!}\sum_{r\in\mathcal{R}}\sum_{k=1}^{\lceil \frac{n}{m} \rceil}A_i^{\mathrm{IG}}(\bar{x}\backslash^{\mathcal{I}(r,k)}x', \bar{x}\backslash^{\mathcal{I}(r,k-1)}x', F) \\
&= \frac{1}{n!}\sum_{r\in\mathcal{R}}\sum_{k=1}^{\lceil \frac{n}{m} \rceil}\sum_{i=1}^{n}A_i^{\mathrm{IG}}(\bar{x}\backslash^{\mathcal{I}(r,k)}x', \bar{x}\backslash^{\mathcal{I}(r,k-1)}x', F) \\
&= \frac{1}{n!}\sum_{r\in\mathcal{R}}\sum_{k=1}^{\lceil \frac{n}{m} \rceil}F(\bar{x}\backslash^{\mathcal{I}(r,k)}x') - F(\bar{x}\backslash^{\mathcal{I}(r,k-1)}x') \\
&= \frac{1}{n!}\sum_{r\in\mathcal{R}}F(\bar{x}) - F(x') \\
&= F(\bar{x}) - F(x')
\end{aligned}
\tag{10}
$$

□

B.3 Sensitivity(a)

Definition of Sensitivity(a): If $(\bar{x}, x', F) \in \mathcal{D}_A$, $F(\bar{x}) \neq F(x')$, and $\bar{x}, x'$ only vary in the i-th component, i.e. $\bar{x}_i \neq x'_i$, and $\bar{x}_j = x'_j \; \forall j \neq i$, then $A_i(\bar{x}, x', F) \neq 0$.

Proof *MBS satisfies Sensitivity(a)*
The value of $(\bar{x}\backslash^{\mathcal{I}(r,k)} x')_j$ is always equal to either $\bar{x}_j$ or x'_j. Therefore, if $\bar{x}_j = x'_j$, then $(\bar{x}\backslash^{\mathcal{I}(r,k)} x')_j = \bar{x}_j = x'_j$ for all values of k, which in turn causes $A_j^{\text{MBS}}(\bar{x}, x', F)$ to be zero:

$$
\begin{aligned}
&A_j^{\text{MBS}}(\bar{x}, x', F) \\
&= \frac{1}{n!} \sum_{r \in \mathcal{R}} \sum_{k=1}^{\lceil \frac{n}{m} \rceil} A_j^{\text{IG}}(\bar{x}\backslash^{\mathcal{I}(r,k)} x', \bar{x}\backslash^{\mathcal{I}(r,k-1)} x', F) \\
&= \frac{1}{n!} \sum_{r \in \mathcal{R}} \sum_{k=1}^{\lceil \frac{n}{m} \rceil} ((\bar{x}\backslash^{\mathcal{I}(r,k)} x')_j - (\bar{x}\backslash^{\mathcal{I}(r,k-1)} x')_j) \int_{\zeta=0}^{1} \partial_j F((1-\zeta)(\bar{x}\backslash^{\mathcal{I}(r,k-1)} x') + \zeta(\bar{x}\backslash^{\mathcal{I}(r,k)} x'))d\zeta \\
&= \frac{1}{n!} \sum_{r \in \mathcal{R}} \sum_{k=1}^{\lceil \frac{n}{m} \rceil} (\bar{x}_j - \bar{x}_j) \int_{\zeta=0}^{1} \partial_j F((1-\zeta)(\bar{x}\backslash^{\mathcal{I}(r,k-1)} x') + \zeta(\bar{x}\backslash^{\mathcal{I}(r,k)} x'))d\zeta \\
&= 0
\end{aligned}
\tag{11}
$$

If $\bar{x}_j = x'_j$ for all values of j except $j = i$, then $A_i^{\text{MBS}}(\bar{x}, x', F)$ is the only value that may not be zero. In that case, because MBS satisfies the completeness axiom, $A_i^{\text{MBS}}(\bar{x}, x', F)$ has to be equal to $F(\bar{x}) - F(x')$, so, when also $F(\bar{x}) \neq F(x')$, then $A_i^{\text{MBS}}(\bar{x}, x', F) \neq 0$ □

B.4 Dummy/Sensitivity(b)

Definition of Dummy/Sensitivity(b): If $(\bar{x}, x', F) \in \mathcal{D}_A$ and $\partial_i F \equiv 0$, then $A_i(\bar{x}, x', F) = 0$.

Proof *MBS satisfies dummy.*
The i-th element of MBS is only calculated from the i-th components of the IG steps. Therefore, the dummy axiom of MBS can be proven from the fact that IG satisfies dummy (Lundstrom & Razaviyayn, 2025). When $\partial_i F \equiv 0$ then:

$$
\begin{aligned}
A_i^{\text{MBS}}(\bar{x}, x', F) &= \frac{1}{n!} \sum_{r \in \mathcal{R}} \sum_{k=1}^{\lceil \frac{n}{m} \rceil} A_i^{\text{IG}}(\bar{x}\backslash^{\mathcal{I}(r,k)} x', \bar{x}\backslash^{\mathcal{I}(r,k-1)} x', F) \\
&= \frac{1}{n!} \sum_{r \in \mathcal{R}} \sum_{k=1}^{\lceil \frac{n}{m} \rceil} 0 \\
&= 0
\end{aligned}
\tag{12}
$$

□

B.5 Linearity

Definition of Linearity: If $(\bar{x}, x', F), (\bar{x}, x', G) \in \mathcal{D}_A$ and $\alpha, \beta \in \mathbb{R}$, then $(\bar{x}, x', \alpha F + \beta G) \in \mathcal{D}_A$ and $A(\bar{x}, x', \alpha F + \beta G) = \alpha A(\bar{x}, x', F) + \beta A(\bar{x}, x', G)$.

Proof *MBS satisfies linearity.*
The linearity axiom of MBS can be proven from the linearity axiom of IG (Lundstrom & Razaviyayn,

2025).

$$
\begin{aligned}
A^{\mathrm{MBS}}(\bar{x}, x', \alpha F + \beta G) &= \frac{1}{n!} \sum_{r \in \mathcal{R}} \sum_{k=1}^{\lceil \frac{n}{m} \rceil} A^{\mathrm{IG}}(\bar{x}\backslash^{\mathcal{I}(r,k)} x', \bar{x}\backslash^{\mathcal{I}(r,k-1)} x', \alpha F + \beta G) \\
&= \frac{1}{n!} \sum_{r \in \mathcal{R}} \sum_{k=1}^{\lceil \frac{n}{m} \rceil} (\alpha A^{\mathrm{IG}}(\bar{x}\backslash^{\mathcal{I}(r,k)} x', \bar{x}\backslash^{\mathcal{I}(r,k-1)} x', F) \\
&\qquad + \beta A^{\mathrm{IG}}(\bar{x}\backslash^{\mathcal{I}(r,k)} x', \bar{x}\backslash^{\mathcal{I}(r,k-1)} x', G)) \\
&= \alpha A^{\mathrm{MBS}}(\bar{x}, x', F) + \beta A^{\mathrm{MBS}}(\bar{x}, x', G)
\end{aligned} \tag{13}
$$

□

B.6 Symmetry preserving

Definition of Symmetry-Preserving: Suppose that $(\bar{x}, x', F) \in \mathcal{D}_A$ and i and j are indices. Let $S_{ij}(x)$ be the function that swaps the values of x_i and x_j. Then if $F(x) = F(S_{ij}(x))$ for any $x \in \mathcal{D}_F$, and $\bar{x} = S_{ij}(\bar{x})$ and $x' = S_{ij}(x')$, we have $A_i(\bar{x}, x', F) = A_j(\bar{x}, x', F)$.

Proof *MBS satisfies symmetry preserving.*
Let $e^{(i)}$ be the i-th standard basis vector of $\mathbb{R}^n$. If you assume $\forall x \in \mathcal{D}_F, F(x) = F(S_{ij}(x))$, then:

$$
\partial_i F(x) = \lim_{\beta \to 0} \frac{F(x + \beta e^{(i)})) - F(x)}{\beta} = \lim_{\beta \to 0} \frac{F(S_{ij}(x) + \beta e^{(j)}) - F(S_{ij}(x))}{\beta} = \partial_j F(S_{ij}(x)) \tag{14}
$$

By using that $\bar{x} = S_{ij}(\bar{x})$ and $x' = S_{ij}(x')$ we can derive:

$$
(\bar{x}\backslash^{\mathcal{I}(r,k)} x')_i = (\bar{x}\backslash^{\mathcal{I}(S_{ij}(r),k)} x')_j \tag{15}
$$

$$
\bar{x}\backslash^{\mathcal{I}(S_{ij}(r),k)} x' = S_{ij}(\bar{x}\backslash^{\mathcal{I}(r,k)} x') \tag{16}
$$

We define $P(r, \bar{x}, x', F)$ as the sum over all IG calls for a given ordering $r \in \mathcal{R}$:

$$
P(r, \bar{x}, x', F) = \sum_{k=1}^{\lceil \frac{n}{m} \rceil} A^{\mathrm{IG}}(\bar{x}\backslash^{\mathcal{I}(r,k)} x', \bar{x}\backslash^{\mathcal{I}(r,k-1)} x', F) \tag{17}
$$

By using Equations 14, 15, and 16, we can prove that $P_i(r, \bar{x}, x', F) = P_j(S_{ij}(r), \bar{x}, x', F)$:

$$
\begin{aligned}
&P_i(r, \bar{x}, x', F) \\
&= \sum_{k=1}^{\lceil \frac{n}{m} \rceil} A_i^{\mathrm{IG}}(\bar{x}\backslash^{\mathcal{I}(r,k)} x', \bar{x}\backslash^{\mathcal{I}(r,k-1)} x', F) \\
&= \sum_{k=1}^{\lceil \frac{n}{m} \rceil} \left((\bar{x}\backslash^{\mathcal{I}(r,k)} x' - \bar{x}\backslash^{\mathcal{I}(r,k-1)} x')_i \int_{\zeta=0}^{1} \partial_i F \left((1-\zeta)(\bar{x}\backslash^{\mathcal{I}(r,k-1)} x') + \zeta(\bar{x}\backslash^{\mathcal{I}(r,k)} x') \right) d\zeta \right) \\
&= \sum_{k=1}^{\lceil \frac{n}{m} \rceil} \left((\bar{x}\backslash^{\mathcal{I}(r,k)} x' - \bar{x}\backslash^{\mathcal{I}(r,k-1)} x')_i \int_{\zeta=0}^{1} \partial_j F \left(S_{ij} \left((1-\zeta)(\bar{x}\backslash^{\mathcal{I}(r,k-1)} x') + \zeta(\bar{x}\backslash^{\mathcal{I}(r,k)} x') \right) \right) d\zeta \right) \\
&= \sum_{k=1}^{\lceil \frac{n}{m} \rceil} \left((\bar{x}\backslash^{\mathcal{I}(S_{ij}(r),k)} x' - \bar{x}\backslash^{\mathcal{I}(S_{ij}(r),k-1)} x')_j \int_{\zeta=0}^{1} \partial_j F \left((1-\zeta)(\bar{x}\backslash^{\mathcal{I}(S_{ij}(r),k-1)} x') + \zeta(\bar{x}\backslash^{\mathcal{I}(S_{ij}(r),k)} x') \right) d\zeta \right) \\
&= \sum_{k=1}^{\lceil \frac{n}{m} \rceil} A_j^{\mathrm{IG}}(\bar{x}\backslash^{\mathcal{I}(S_{ij}(r),k)} x', \bar{x}\backslash^{\mathcal{I}(S_{ij}(r),k-1)} x', F) \\
&= P_j(S_{ij}(r), \bar{x}, x', F)
\end{aligned} \tag{18}
$$

The set $\mathcal{R}$ of all possible orderings of n features can be divided into two disjoint sets $\mathcal{R}_{r_i<r_j}$ and $\mathcal{R}_{r_i>r_j}$, where $\mathcal{R}_{r_i<r_j} \cup \mathcal{R}_{r_i>r_j} = \mathcal{R}$. $\mathcal{R}_{r_i<r_j}$ contains all orderings where $r_i < r_j$, and $\mathcal{R}_{r_i>r_j}$ contains all orderings where $r_i > r_j$. For every ordering $r \in \mathcal{R}_{r_i<r_j}$ there is a unique ordering $S_{ij}(r) \in \mathcal{R}_{r_i>r_j}$. Because $\mathcal{R}_{r_i<r_j} \cup \mathcal{R}_{r_i>r_j} = \mathcal{R}$, we can rewrite MBS in a way that proves that it is symmetry preserving:

$$\begin{aligned} A_i^{\text{MBS}}(\bar{x}, x', F) &= \frac{1}{n!} \sum_{r \in \mathcal{R}_{r_i<r_j}} P_i(r, \bar{x}, x', F) + P_i(S_{ij}(r), \bar{x}, x', F) \\ &= \frac{1}{n!} \sum_{r \in \mathcal{R}_{r_i<r_j}} P_j(S_{ij}(r), \bar{x}, x', F) + P_j(r, \bar{x}, x', F) = A_j^{\text{MBS}}(\bar{x}, x', F) \end{aligned} \tag{19}$$

□

B.7 Non-Decreasing Positivity

Definition of Non-Decreasing Positivity: If $(\bar{x}, x', F) \in \mathcal{D}_A$ and F is non-decreasing from x' to $\bar{x}$ then $A_i(\bar{x}, x', F) \geq 0$ for every index i.
Definition of Non-Decreasing (Lundstrom et al., 2022): F is non-decreasing from x' to x if $F(\gamma(t))$ is non-decreasing for every monotone path $\gamma(t)$ from x' to x.
Definition of monotone path function (Lundstrom et al., 2022; Lundstrom & Razaviyayn, 2025): A function $\gamma(t) : [0, 1] \to \mathcal{D}_F$ is a monotone path function from $x' \in \mathbb{R}^n$ to $x \in \mathbb{R}^n$ if $\gamma(t)$ is a continuous, piecewise smooth curve from x' to x, and $|x'_i - \gamma_i(t_1)| \leq |x'_i - \gamma_i(t_2)|$ for all indices i and all $t_1, t_2 \in [0, 1]$ where $t_1 < t_2$.

Proof *MBS satisfies non-decreasing positivity*
Because of the definition of non-decreasing, $F(\gamma(t))$ is non-decreasing for every monotone path $\gamma(t)$ from x' to $\bar{x}$. It is also non-decreasing for every subsection of these paths. For every k, r, a monotone path can be constructed by connecting the points $[x', \bar{x}\backslash^{\mathcal{I}(r,k-1)}x', \bar{x}\backslash^{\mathcal{I}(r,k)}x', \bar{x}]$ using straight line segments. Therefore, F is also non-decreasing from $\bar{x}\backslash^{\mathcal{I}(r,k-1)}x'$ to $\bar{x}\backslash^{\mathcal{I}(r,k)}x'$. Because of this, and because IG satisfies non-decreasing positivity (Lundstrom & Razaviyayn, 2025), $A^{\text{IG}}(\bar{x}\backslash^{\mathcal{I}(r,k)}x', \bar{x}\backslash^{\mathcal{I}(r,k-1)}x', F)$ is non-negative for all k, r. MBS is a sum of these terms, so MBS is also non-negative. □

B.8 Affine scale invariance

Definition of Affine Scale Invariance (ASI): Suppose that $(\bar{x}, x', F) \in \mathcal{D}_A$, $c, d \in \mathbb{R}$ with $c \neq 0$, and i is an index. Let T be an affine transformation of element i, so that $T(x) := (x_1, \cdots, cx_i + d, \cdots, x_n)$. Then we have $A(\bar{x}, x', F) = A(T(\bar{x}), T(x'), F \circ T^{-1})$.

Proof *MBS satisfies affine scale invariance.*
The *affine scale invariance* of MBS can be proven by using the *affine scale invariance* of IG (Lundstrom & Razaviyayn, 2025):

$$\begin{aligned} &A^{\text{MBS}}(T(\bar{x}), T(x'), F \circ T^{-1}) \\ &= \frac{1}{n!} \sum_{r \in \mathcal{R}} \sum_{k=1}^{\lceil \frac{n}{m} \rceil} A^{\text{IG}}(T(\bar{x})\backslash^{\mathcal{I}(r,k)}T(x'), T(\bar{x})\backslash^{\mathcal{I}(r,k-1)}T(x'), F \circ T^{-1}) \\ &= \frac{1}{n!} \sum_{r \in \mathcal{R}} \sum_{k=1}^{\lceil \frac{n}{m} \rceil} A^{\text{IG}}(T(\bar{x}\backslash^{\mathcal{I}(r,k)}x')), T(\bar{x}\backslash^{\mathcal{I}(r,k-1)}x'), F \circ T^{-1}) \\ &= \frac{1}{n!} \sum_{r \in \mathcal{R}} \sum_{k=1}^{\lceil \frac{n}{m} \rceil} A^{\text{IG}}(\bar{x}\backslash^{\mathcal{I}(r,k)}x', \bar{x}\backslash^{\mathcal{I}(r,k-1)}x', F) \\ &= A^{\text{MBS}}(\bar{x}, x', F) \end{aligned} \tag{20}$$

□