RETHINKING SHAPLEY VALUE FOR NEGATIVE INTERACTIONS IN NON-CONVEX GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

We study causal interaction for payoff allocation in cooperative game theory, including quantifying feature attribution for deep learning models. Most feature attribution methods mainly stem from the criteria from the Shapley value, which provides a unique payoff vector for players by marginalizing contributions in a cooperative game. However, interactions between players in the game do not exactly appear in the original formulation of the Shapley value. In this work, we clarify the role of interactions in computing the Shapley value by reformulation and discuss implicit assumptions from a game-theoretical perspective. Our theoretical analysis demonstrates that, when negative interactions exist—common in deep learning models—attributions or payoffs can be underrated by the efficiency axiom in this classical setup. We suggest a new allocation rule that decomposes contributions into interactions and aggregates positive parts for non-convex games. Furthermore, we propose an approximation algorithm to reduce the cost of interaction computation which can be applied for differentiable functions such as deep learning models. Our approach mitigates counter-intuitive phenomena where even features highly relevant to the decision are assigned low attribution in the previous approaches.

1 INTRODUCTION

As black-box models like Deep Neural Networks (DNNs) become increasingly prevalent, providing the cause of their decision-making process is crucial for model reliability and interpretation. One approach to understanding such prediction models is to quantify the attributions of individual features [1] for decisions. Theoretically, feature attribution methods are usually grounded in the Shapley value (Shapley, 1953), which provides a fair payoff allocation rule in cooperative game theory. Shapley (1953) introduced a set of axioms—*linearity*, *dummy*, *symmetry*, and *efficiency*—that uniquely defines this payoff allocation. This axiomatic approach has been widely extended to machine learning research and attribution methods (Montavon et al., 2017; Sundararajan et al., 2017; Rozemberczki et al., 2022).

Despite this game-theoretic foundation, there has been limited exploration of *interaction* between features and the relationship with axioms in computing attributions. Causal interaction measures the discrepancy of a player's effect on the game output when another player participates or not (VanderWeele, 2015; Keele & Stevenson, 2021). Understanding the impact of interactions can be crucial when assigning reasonable payoffs or attributions in complex games or functions. However, since the original Shapley value is formulated as the marginalized causal effect of each feature, it does not give any information about conditions or assumptions on interactions (Grabisch & Roubens, 1999; Procaccia et al., 2014).

In this work, we study the role of interactions in computing the Shapley value by reformulation in interaction terms. Our derivation shows that the Shapley value can be interpreted as the sum of a feature's single effect and the weighted interactions with other players. Based on the equation, we demonstrate that the Shapley value implicitly assumes non-negative interactions between players. It is connected to the fact that the candidates for payoff allocation are designed under the efficiency axiom and the players' rational behavior on the grand coalition from a game-theoretical perspective (Von Neumann & Morgenstern, 1944; Roth, 1988; Peters & Peters, 2015). Consequently, the

[1] In this paper, features refer to input features, rather than learned representations.

Shapley value becomes a reasonable and well-defined allocation when a game is convex, and we prove that the convexity is equivalent to non-negative interactions.

When negative interactions exist, i.e. for non-convex games, the Shapley value is no longer a reasonable allocation resulted from the player's efficient and rational behaviors. Most axiomatic approaches for feature attribution, including extensions of the Shapley value, conventionally adhere to the efficiency, even though many black-box models like DNNs do not satisfy the convexity (Montavon et al., 2017; Sundararajan et al., 2017). Enforcing the efficiency axiom for non-convex games leads to the undervaluation of payoffs or feature attributions even though the players can potentially obtain larger payoffs in the specific subset of players. In the case study, we provide examples of negative interactions for simple non-convex games, such as a max function and a sigmoid function, and their connection to deep learning cases. As a result, this non-convexity causes counter-intuitive phenomena where even features that seem highly relevant to the decision are assigned low attribution.

Building on our theoretical analysis of interaction, we extend the Shapley value to non-convex games while keeping its philosophy that assigns payoffs by measuring synergistic interactions between players. We propose a new allocation rule that decomposes each contribution into interactions and aggregates only the positive components for non-convex games, namely, Aggregated Positive Interactions (API). We provide an unbiased estimation for API to enable computation by sampling approach. Furthermore, to reduce the cost of interaction computation, we propose an approximation algorithm that can be applied to differentiable functions such as deep learning models. Our method effectively resolves the issue of undervaluing feature attributions and uncovers potential contributions that were overlooked under previous assumptions.

2 PRELIMINARIES

2.1 COOPERATIVE GAME

A *cooperative game* consists of a set of players $N = \{1, \cdots, n\}$, called the *grand coalition*, and a *characteristic function* $v : 2^N \to \mathbb{R}$, which maps a coalition $S \subseteq N$ to the utility $v(S)$ players in S achieve. The goal is to determine the payoff vector $\phi(v) \in \mathbb{R}^n$ where i-th element $\phi_i(v)$ indicates the payoff allocated to player i from the total utility $v(N)$. This game is sometimes called a *transferable utility* game allowing the utility to be fully transferred to the players as their payoffs (Von Neumann & Morgenstern, 1944; Roth, 1988; Peters & Peters, 2015). In classic literature, for convenience, the function v itself is often referred to as the game, and the cardinality of each set of players $(N, S, T, \cdots)$ is denoted by the corresponding lower-case letter $(n, s, t, \cdots)$. We follow this convention in this work.

A game is *convex* if

$$v(S) + v(T) \leq v(S \cup T) + v(S \cap T), \quad \forall S, T \subseteq N. \tag{1}$$

If Equation (1) holds under the more relaxed condition $S \cap T = \emptyset$, then the game is *super-additive*. This indicates that a convex game is a special case of a super-additive game.

Remark. *The implicit assumption in a cooperative game is that players strategically form the grand coalition to maximize their payoffs (Roth, 1988; Fujimoto et al., 2006; Peters & Peters, 2015). Convex games indicate that the benefits of joining a coalition increase as the coalition size grows, ensuring that cooperation always leads to higher utility and that forming the grand coalition N is an optimal strategy (Shapley, 1971).*

2.2 SHAPLEY VALUE

The Shapley value (Shapley, 1953) is the payoff vector $\phi(v) \in \mathbb{R}^n$ that allocates the total utility $v(N)$ among the players $i \in N$ with , which is given by:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{1}{n} \binom{n-1}{s}^{-1} [v(S \cup \{i\}) - v(S)]. \tag{2}$$

It is a unique form that satisfies four axioms designed for a fair allocation: *linearity*, *dummy*, *symmetry*, and *efficiency* (Weber, 1988). Notably, Equation (2) can be interpreted as the weighted average

of each player's contribution when joining all possible coalitions. Furthermore, this can be viewed as the expectation of the causal effects since it evaluates the expected change of values while putting or removing a player (Janzing et al., 2020; Pearl, 2022). Using the Shapley value or following its philosophy for feature attribution is akin to evaluating the attribution as the causality of a feature's contribution to the function's outcome. From this perspective, $\Delta_i v(S) \coloneqq v(S \cup \{i\}) - v(S)$ is referred to by various terms, such as *(causal) effect* or *contribution*.

2.3 Core & Efficiency

The Shapley value is one of *solution* candidates, which denote design choices of payoff vectors, for fair allocation in cooperative games. To understand the meaning of the Shapley value as a solution concept, we review the notion of *core* and *efficiency* in game theory and explain how convexity is related to these notions.

A *core* (Shapley, 1971; Dehez, 2017) is a set of payoff vectors $a \in \mathbb{R}^n$ that satisfies the following two conditions:

$$\{a \in \mathbb{R}^n \mid \sum_{i \in N} a_i = v(N),\ \sum_{i \in S} a_i \geq v(S)\ \forall S \subset N\}. \tag{3}$$

The core represents the set of socially stable outcomes where no partial coalition can achieve a result better than the grand coalition. The first equality condition denotes the *efficiency* axiom that states the payoff allocations of individual players must sum to the total utility $v(N)$. This equality condition is the same as the efficiency axiom of the Shapely value. The second inequality condition, namely the *coalitional rationality*, implies that the sum of payoffs allocated to any subset coalition S is at least as large as the utility $v(S)$ the coalition S can achieve on its own. This setup implies situations where it is possible to sustain cooperation among all players by their rational behaviors, i.e., the grand coalition.

Therefore, the existence of the core is a natural assumption in a cooperative game when we take into account full efficiency and grand coalition for payoff allocation. In addition, a core of a game is a polytope and when the game is convex, the Shapley value is a solution positioned at the center of the core, which is a reasonable choice for fair allocation (Shapley, 1971). However, in non-convex games, the core may not exist or the Shapley value may not lie within the core.

3 Interaction in Shapley value

3.1 Interaction in Games

In this work, *interaction* [2] between two players in the cooperative game is defined as follows:

Definition 1 (interaction)**.** For a cooperative game v on a player set N, an interaction between two player $i, j \in N$ given a subset of players T is defined as follows:

$$\begin{aligned} I_{ij}(T) &= v(T \cup \{i,j\}) - v(T \cup \{i\}) - v(T \cup \{j\}) + v(T) \\ &= \Delta_i v(T \cup \{j\}) - \Delta_i v(T). \end{aligned} \tag{4}$$

This definition comes from the notion of causal interaction in causal inference (VanderWeele, 2015; Keele & Stevenson, 2021). It quantifies the discrepancy of player i's effect on the output when player j participates or not. If there are no interactions between players in a cooperative game, each player's effect would be constant regardless of other players' conditions, making payoff allocation trivial. Therefore, cooperative games naturally assume synergistic interactions among players, leading to a grand coalition through their rational behavior.

Generally, the Shapley value is interpreted as the marginal contribution of each player, i.e., the average effect of the player with all combinations of other players. The reason for marginalization is the possibility that the player's effect changes depending on the combination of other players. However, the original formulation of the Shapley value gives limited information on how such interactions are connected to payoff allocations. To understand the interactions in the Shapley value, we need to represent it with interaction terms. We derive the following representation of the Shapley value.

[2] Note that this definition of interaction is different from that in other works (Tsai et al., 2023).

Theorem 1. *The Shapley value is a weighted sum of interactions:*

$$\phi_i(v) = \Delta_i v(\emptyset) + \sum_{t=0}^{n-2} \frac{1}{n} \binom{n-1}{t}^{-1} \sum_{\substack{j \in N \\ j \neq i}} \sum_{\substack{T \subseteq N \setminus \{i,j\} \\ |T|=t}} I_{ij}(T) \tag{5}$$

Proof. See Appendix. □

The theorem says that the Shapley value measures the sum of each player's effect and the interactions resulting from cooperation.

3.2 Relationship with Convexity

In Section 2.3, we introduced the concept of core representing a set of feasible solutions assuming players' rational behaviors on their grand coalition. When the game is convex, the core is guaranteed to exist, while in non-convex games, it may not.

The connection between convexity and interaction can be revealed from the following theorem that the convexity of the game is equivalent to non-negative interactions:

Theorem 2. *A game v is convex if and only if $I_{ij}(R) \geq 0 \; \forall i, j \in N, \forall R \subseteq N \setminus \{i, j\}$.*

Proof. See Appendix. □

It implies that finding a solution within the core implicitly assumes non-negative interactions. When a game has negative interactions, it becomes non-convex, leading to an empty core. In such cases, the Shapley value no longer satisfies the core's properties and does not reflect players' efficient and rational behaviors. As a result, for non-convex games, calculating the Shapley value becomes nothing but taking an expectation formula rather than a meaningful or reasonable solution based on a grand coalition.

3.3 Problems in Non-convex Games

For non-convex games, the efficiency axiom becomes meaningless since the player's grand coalition is no longer the result of the player's efficient and rational behaviors. However, the efficiency axiom is conventionally used in the applications of the Shapley value and attribution methods (Lundberg, 2017; Sundararajan et al., 2017). Enforcing the efficiency axiom for non-convex games can lead to the undervaluation of payoffs or feature attributions even though the players have a potential role in improving the output.

The issue of undervaluation can be intuitively observed from Equation (5). Even though there are coalitions where the player provides a potential power to increase the output of the game, the payoff can become lower by negative interactions. In the feature attribution task, the feature may get low attribution even if it is crucial evidence for the decision. This is not a desirable situation considering why we want to evaluate attributions or payoffs. In practice, most black-box models with high expressive power like deep learning models are non-convex. This has been linked to incorrect experimental results in some Shapley-based feature attribution methods, where features that are likely to be relevant to a decision receive low (or even negative) attribution scores. Traditional methods often address this by simply adjusting the results with post-processing, such as taking the absolute value of the attributions. In Section 4, we explain some simple examples that have negative interactions and the connection to deep learning models. In Section 5, we suggest a new solution to solve this issue from a game-theoretical perspective.

4 Case Study

In this section, we will explore examples of non-convex games where negative interactions occur. The problem emerges when the grand coalition is no longer a result of rational behavior. If the grand coalition is inefficient—meaning that some players make significant contributions to the outcome

but are not necessary to effectively improve the final output—they experience negative interactions. Conceptually, this can happen due to role redundancy or output saturation, which we will illustrate using examples like the max function and sigmoidal functions. We will also explain how negative interactions can practically arise in deep learning models that incorporate such functions.

Max Function. The max function returns the highest value that the participating players can yield. A player's causal effect on the max function output diminishes when another player with a similar contribution joins. In this case, interaction becomes negative, and indeed the max function is non-convex. Furthermore, consider the following function

$$f(x_1, \cdots, x_8) = \max(x_1, x_2, x_3, 4x_4) + \max(6x_5, 6x_6, 6x_7, 7x_8).$$

We set each variable x_i to be binary (0 or 1) to represent the participation keeping the setup of cooperative game theory. We can generally expect that player 8 may get more payoffs than player 4. However, there are negative interactions in this function, for example, $I_{14}(\emptyset) = -1, I_{58}(\emptyset) = -6$, which lead to counter-intuitive results. The Shapley value of player 8 is $10/4$ while that of player 4 is $13/4$. This phenomenon occurs conceptually because of redundancy in the roles of individual players. Even if some player has a large impact on the outcome, if their roles are similar, their interactions become negative leading to underrated payoffs.

Sigmoid Function. Another example of non-convex functions that yield problematic negative interactions is a sigmoid function that takes the sum of features as input. There exist negative interactions when input is sufficiently large so that it lies in the near-saturation region. Consider the following function

$$f(x_1, x_2, x_3, x_4) = 10 \cdot \sigma(7x_1 + 6x_2) + 10 \cdot \sigma(2x_3 + 0.1x_4)$$

where σ denotes a sigmoid function $1/(1+e^{-x})$. Similar to the max function case, players 1, 2 have stronger negative interaction ($I_{12}(\emptyset) = -4.96$) than that of player 3, 4 ($I_{34}(\emptyset) = -0.15$). Consequently, the Shapley value of player 1 (2.51) is lower than that of player 3 (3.73). This phenomenon arises due to the presence of saturation or an upper bound. This example can be generalized to any squashing function that exhibits saturation or is upper-bounded, e.g., softmax.

Deep Learning model. Generally, the deep learning model is a non-convex function and it has various non-convex components in its internal mechanism. Consider a *max pooling* and an *attention* module in a deep learning structure. CNN classifiers commonly utilize the *max pooling* as a key operation and Transformer models consists of the *attention* modules, which is the weighted sum of the values, where the weights are the softmax outputs of the similarities between the queries and keys (Vaswani et al., 2017). Therefore, one can expect the similar problems as before can happen when dealing with CNNs and Transformers, and we show the counter-intuitive results of the Shapley value with such models in Figure 1 and 3.

5 Aggregated Positive Interactions

We have discussed the issue of utilizing the Shapley value and its variants for non-convex games. Equation (5) shows that the Shapley value measures a player's attribution by aggregating its positive interaction with other players for convex games. To extend this philosophy to non-convex games, we propose a solution that decomposes each contribution into interactions and aggregates the positive parts, which represent the player's potential influence on the game by constructing synergistic coalitions.

$$\phi_i(v) = \Delta_i v(\emptyset) + \sum_{t=0}^{n-2} \frac{1}{n} \binom{n-1}{t}^{-1} \sum_{\substack{j \in N \\ j \neq i}} \sum_{\substack{T \subseteq N \setminus \{i,j\} \\ |T| = t}} \max(I_{ij}(T), 0). \tag{6}$$

We call this solution *Aggregated Positive Interactions (API)*. It is an extension of the Shapley value to non-convex games. In convex games, API follows the axioms of the Shapley value, as all interactions are preserved. It can also be interpreted as the Shapley value of the approximated convex game where positive interactions of the original game are preserved. Since negative interactions are removed in aggregation, a player's payoff is no longer underrated from irrational coalitions. Notably, API effectively addresses the undervaluation demonstrated in Section 4. For the Max Function case, API assigns values of 8 and 4 to players 8 and 4, respectively, and for the Sigmoid Function case, it assigns values of 4.99 and 3.81 to players 1 and 3, respectively.

Remark. *The Dummy and Sensitivity axioms have been discussed on how a feature attribution corresponds to its causal effect on the function output (Roth, 1988; Sundararajan et al., 2017). In previous attribution methods, zero attribution does not imply that the feature has no effect on the function. This is because negative interactions can cancel out the feature's effect. In our solution, a feature that has no effect on the function output is equivalent to zero attribution.*

Since dealing with the interaction terms requires high computational cost for applying complex black-box models like DNNs, we first propose the estimation of Equation (5), (6). Then, we introduce an approximation algorithm to reduce the computational cost for interaction terms using gradient information.

5.1 Estimation through Permutation Sampling

We reformulate Equation (5) into the expectation form to enable sample estimation with vectorized representations. Let $i \sim u_1(S)$ be the uniform distribution of player i from set S, and $T \sim u_2(t, S)$ be the uniform distribution of subset S with cardinality $|T| = t$. $\phi(v)$ and $\Delta v(\emptyset)$ denote the Shapley value and the discrete derivatives of all players in a vector form, respectively. We define an interaction vector $I_{\cdot j}(T) \in \mathbb{R}^n$ where the i-th element is $I_{ij}(T)$ for given $T \subseteq N \setminus \{j\}$. We set $I_{ij}(T) = 0$ when $i = j$ or $i \in T$, then $I_{\cdot j}(T)$ is well-defined for $T \subseteq N \setminus \{j\}$. Then, we obtain the following vectorized form to estimate the Shapley value with interaction vectors.

Theorem 3. *The vector of the Shapley value $\phi(v)$ is represented as follows:*

$$\phi(v) = \Delta v(\emptyset) + \sum_{t=0}^{n-2} \mathbb{E}_{j \sim u_1(N), T \sim u_2(t, N \setminus \{j\})}[I_{\cdot j}(T)]. \tag{7}$$

Proof. See Appendix. □

We can apply sample estimation for expected interaction at each cardinality and sum over interactions. The estimation can be conducted by permutation sampling. For each permutation, we sequentially add features, and compute interactions at each level of cardinality. In this setup, we need to reformulate Equation (7) with the uniform distribution of permutation.

Corollary 1. *The vector of the Shapley value $\phi(v)$ is represented as follows:*

$$\phi(v) = \Delta v(\emptyset) + \sum_{t=0}^{n-2} \mathbb{E}_{\pi \sim \mathrm{Unif}(\Pi(N))}[I_{\cdot \pi_{t+1}}([\pi]_t)]. \tag{8}$$

where $\Pi(N)$ is a collection of all possible permutations of players in N. π_t represents the t-th element in π, and $[\pi]_t$ represents the subset of players up to the t-th player in the ordering π.

It can be easily proved by Theorem 3. From the result, we can apply permutation sampling to estimate the Shapley value with interactions. It can be interpreted as the extension of permutation sampling on the original Shapley value (Castro et al., 2009). Simply, we obtain Aggregated Positive Interactions by maintaining only non-negative interactions during sampling.

5.2 Approximation Algorithm

The major computational burden comes from interaction terms. Since the computational cost for interactions grows with the number of player combinations, it requires n-times more computations compared to computing individual effects in the original Shapley value. We propose an algorithm to reduce the cost of interaction computation that can be applied to differentiable functions. In differentiable functions like DNNs, this approach leverages backpropagation to reduce computational complexity effectively.

Notations. We first explain notations for our algorithm described in Algorithm 1. v is a differentiable function to assess attributions, such as a deep learning predictor. x is a target instance with n features, and each feature is treated as a player in N. A baseline value in $\bar{x}$ refers to the value assigned when a feature does not participate in the game, and it is typically set to the mean value

from the data or zero. $\nabla v(S)$ denotes a gradient of v evaluated when features in S are set to values in x and the others are set to values in $\bar{x}$.

The original definition of interaction is expressed as the difference between two contributions, represented by discrete derivatives. We replace this discrete derivative with a partial derivative using a first-order Taylor approximation. This allows us to compute interactions quickly by leveraging gradients obtained through backpropagation.

$$I_{ij}(T) = \Delta_i v(T \cup \{j\}) - \Delta_i v(T) \approx \{\partial_i v(T \cup \{j\}) - \partial_i v(T)\} * (x_i - \bar{x}_i) \tag{9}$$

When calculating interactions for all features given a set T, the original computation requires $O(n^2)$ forward passes. However, with this approximation, all feature gradients can be computed simultaneously with a single backpropagation, reducing the computation to $O(n)$ backward passes. By utilizing this approximation and permutation sampling, our algorithm computes API as described in Algorithm 1.

Algorithm 1 Approximation for Aggregated Positive Interactions

Input: Differentiable function v, Instance x, Baseline $\bar{x}$
Parameter: # of permutations k
Output: Attribution ϕ
Initialize $\mathrm{Cnt} \leftarrow 0$, $\phi \leftarrow \Delta v(\emptyset)$, $g_1 \leftarrow \nabla v(\emptyset)$
while $\mathrm{Cnt} < k$ **do**
  sample $\pi \sim \mathrm{Unif}(\Pi(N))$
  for $t = 0, \cdots, n-2$ **do**
    $g_2 \leftarrow \nabla v([\pi]_{t+1})$
    $\hat{I} \leftarrow \max((g_2 - g_1) \circ (x - \bar{x}), \mathbf{0}_n)$
    $\hat{I}_i \leftarrow 0 \quad \forall i \in [\pi]_{t+1}$
    $\phi \leftarrow \phi + \hat{I}/k$
    $g_1 \leftarrow g_2$
    $\mathrm{Cnt} \leftarrow \mathrm{Cnt} + 1$
  end for
end while
return ϕ

6 Evaluation

In Section 4, we introduced some simple non-convex examples to identify the impact of negative interactions on the Shapley value. In this section, we identify such phenomenon in practical deep learning tasks and evaluate the results from Aggregated Positive Interactions (API). Our experiments are conducted on image classifiers (VGG19 (Simonyan & Zisserman, 2014), ResNet50 (He et al., 2016)) trained on the ImageNet dataset (Deng et al., 2009), and a sentence classifier (BERT (Devlin, 2018)) trained on the IMDB Review dataset (Maas et al., 2011). In experimental results, Approximated Shapley value (abbreviated as Approx. SV) denotes the approximation of the Shapley value by sampling permutation and aggregating all interactions. We samples 100 permutations for the image classifiers and 300 permutations for the sentence classifier. In the case of ImageNet data, we convert images into 20×20 patches for feasible computation.

6.1 Impact of Positive Interactions on Image Classifier

We identify the impact of positive interactions on VGG19. Figure 1 shows attribution results, high-attribution parts, and low-attribution parts when aggregating all interactions and only positive interactions. In attribution map, red indicates positive attribution, white represents zero attribution, and blue denotes negative attribution.

When all interactions are aggregated, the attributions tend to become more dispersed. This dispersion can be explained by negative interactions, as described in max pooling, where nearby inputs are grouped together, causing negative interactions within the same pooling operator. This makes it difficult to pinpoint the areas that contributed most to the model's decision using the Shapley value.

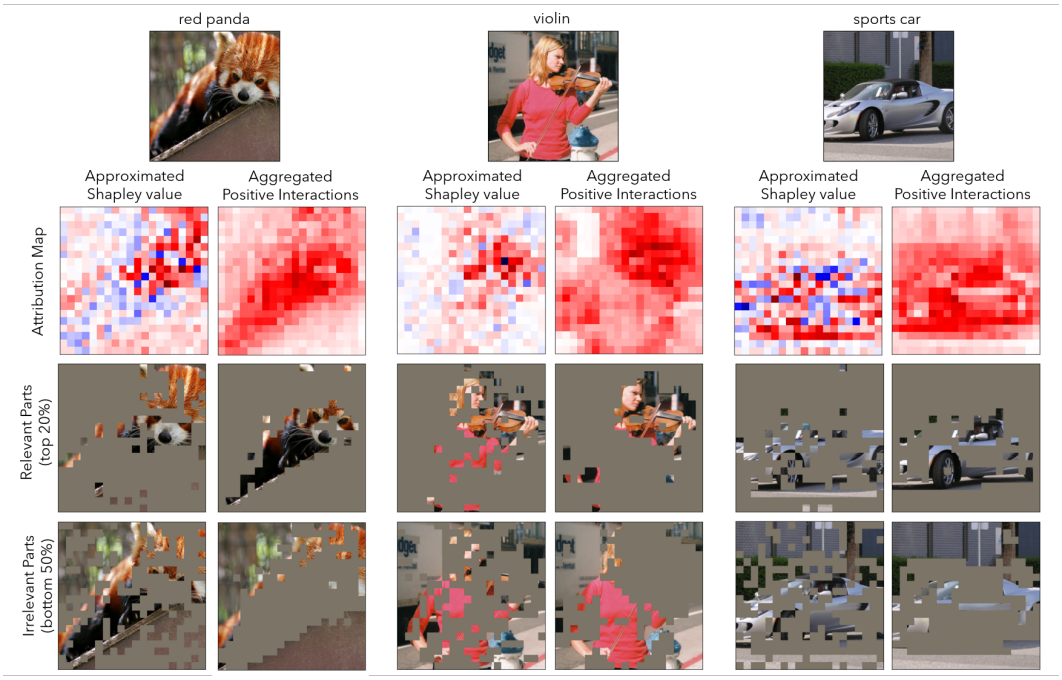

Figure 1: Aggregated Positive Interactions on the ImageNet dataset. While the approximated Shapley Value produces dispersed attributions across relevant regions, API more effectively captures these regions.

However, when we aggregate only positive interactions, key regions to the model decision become more distinct, such as the eyes, nose, and front paws of a red panda, or the wheels and windows of a sports car.

6.2 Underrated Attributions from Efficiency Axiom

To demonstrate the limitations of efficiency axiom-based methods in non-convex games, we compared our approach to related attribution methods using ResNet50 in Figure 2. For KernelSHAP (Lundberg, 2017), we use 40,000 samples, as our method approximates using 100 permutations across 400 features. Integrated Gradients (IG) (Sundararajan et al., 2017) is computed with 100 steps, and the attributions are summed for all pixels within each patch. We also evaluated performance using the Insertion and Deletion metrics. Insertion measures the logit value when the top 30% attribution patches are added, while Deletion measures the logit value when the bottom 30% attribution patches are removed. Higher values for both metrics indicate that the method more accurately identifies the regions that are crucial for the model's decision.

KernelSHAP and IG, constrained by the efficiency axiom, tend to disperse attributions across various regions, similar to the Approximated Shapley value. In contrast, our API assigns attributions more consistently based on the key features of the input, avoiding unnecessary attributions to the background. In practice, the absolute value of attributions has been generally utilized, since it yields plausible results even though it is heuristic choice. API solves this problem and provides more reasonable explanation with game-theoretical approach. In some observations, we have identified that KernelSHAP may perform better numerically than API, but it still suffered from the issue of dispersing attributions over irrelevant regions. We suspect this may be due to errors introduced during the first-order Taylor approximation used in API for calculating interactions.

6.3 Application on Language Model

We demonstrate the application of API on a language model, using a BERT (Devlin, 2018) trained on the IMDB dataset for sentence prediction (Maas et al., 2011). This model takes a movie review,

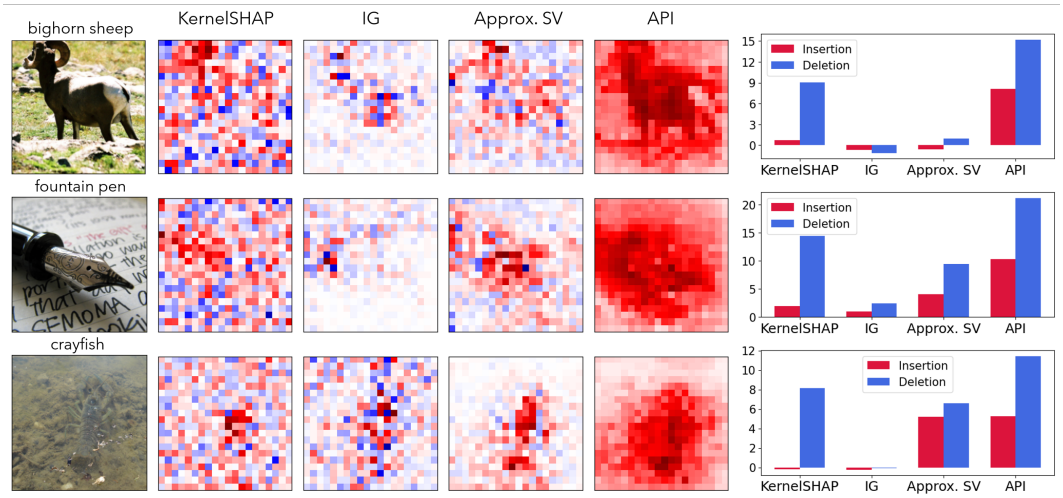

Figure 2: Comparison to other Attribution methods based on Efficiency axiom. Insertion measures the logit value when the top 30% attribution patches are added, while Deletion measures it with the bottom 30% removed. API achieves better results compared to the previous methods.

which is a sequence of tokens, and predicts whether it is positive or negative. Figure 3 shows a fraction of input texts where the top row is predicted as negative, while the bottom row is positive. For these texts, we compare the approximated Shapley value and API, estimated with 300 permutations, by highlighting the tokens with attribution in the top 20%.

For the negative review case, the approximated Shapley value only highlights 'hardly be' while API focuses on the entire phrase 'can hardly be taken seriously on any level'. This indicates that the portion from 'taken' to 'level' was underrated due to the strong negative interactions, even though that portion would have contributed significantly to the prediction if given with some subset of tokens in the input text. Specifically, 'hardly be' alone does not intuitively seem influential for the negative prediction, while it conveys a negative meaning when combined with the subsequent tokens from 'taken' to 'level'. The subsequent tokens have strong negative interactions with other tokens in the input text and these negative interactions significantly obscure their contributions to the prediction. Our proposed API addresses these problematic negative interactions and focuses on the meaningful combination of tokens that leads to the negative prediction, assigning high attributions to the portion from 'token' to 'level'.

A similar interpretation applies to the positive case in the bottom row. The approximated Shapley value underrates tokens 'i', liked', and 'movie' which seem relevant to the positive prediction, and this causes other irrelevant tokens to get higher attributions. In contrast, API effectively addresses the problem of negative interactions, reasonably attributing the entire phrase 'i really liked this movie'.

	Approximated Shapley value	Aggregated Positive Interactions
Negative Review	it doesn' t matter what one' s political views are because this film can hardly be taken seriously on any level.	it doesn' t matter what one' s political views are because this film can hardly be taken seriously on any level.
Positive Review	i really liked this movie. i' ve read a few of the other comments, and although i pity those who did not understand it, i do agree with some of the criticisms.	i really liked this movie. i' ve read a few of the other comments, and although i pity those who did not understand it, i do agree with some of the criticisms.

Figure 3: Application on a Language Model. While the approximated Shapley value assigns low attributions to the tokens that seem relevant due to the negative interactions, API effectively results in more reasonable attributions by focusing only on the positive interactions.

7 Related Work

Shapley Value. The payoff allocation for the utility function has been studied with axiomatic approaches in cooperative game theory literature (Roth, 1988; Fujimoto et al., 2006; Peters & Peters, 2015). The Shapley value (Shapley, 1953) is the payoff allocation that uniquely satisfies the *linearity*, *dummy*, *symmetry*, and *efficiency* axioms (Weber, 1988). It can also be interpreted as the expectation of the causal effects since the contribution of a player is measured by *do*-operator (Janzing et al., 2020; Pearl, 2022). Due to the well-defined axioms of the Shapley value, there have been a lot of approaches to apply the Shapley value to deep learning literature (Scott et al., 2017; Frye et al., 2020). Kumar et al. have pointed out the limitation of the Shapley value when the game is not inessential and proposed shapley residuals which quantifies the lost information when using the Shapley value. While some studies have identified problems of the Shapley-based axioms such as efficiency (Kwon & Zou, 2021; 2022), they have not focused on the underlying assumption of the convexity inherent in cooperative games nor utilized the interactions, which is the focus of this work.

Game-theoretic Interaction. The concept of interaction was firstly proposed in order to deal with the information of *cooperation* existing among players (Grabisch & Roubens, 1999), which cannot be taken into account by the Shapley value (Shapley, 1953) and its variants (Banzhaf III, 1964; Monderer & Samet, 2002). Since the natural extensions of the axioms of the Shapley value do not guarantee the uniqueness of interaction indices, previous studies have introduced additional axioms (Grabisch & Roubens, 1999; Sundararajan et al., 2020; Tsai et al., 2023). Fumagalli et al. proposed a general form of these interaction indices and an efficient sampling-based estimator SHAP-IQ. Some approaches have attempted to explain the underlying reasoning of the inference of DNNs through the game-theoretic interactions (Deng et al., 2021; Ren et al., 2023; Li & Zhang, 2023). However, none of these studies have considered the interactions to analyze the axioms of the Shapley value.

Attribution Methods. Attribution methods aim to measure the contributions of input features for the model output and there have been various approaches in the deep learning literature. One class of methods utilizes the gradient information of the model by aggregating gradients along a path in the input domain, satisfying certain axioms such as completeness (Sundararajan et al., 2017; Kapishnikov et al., 2021; Jeon et al., 2023). Another type of method sequentially redistributes the model output across the layers, adhering to specific conditions including the conservation property (Bach et al., 2015; Montavon et al., 2017; Shrikumar et al., 2017; Nam et al., 2020). Some works have established a unified attribution framework. Additive feature attribution methods unified some existing methods, with SHAP uniquely satisfying specific Shapley-based axioms (Lundberg, 2017). Taylor attribution framework leverages the Taylor decomposition to illustrate how the interactions between input features are distributed to individual features within existing attribution methods (Deng et al., 2023). Although most previous studies have utilized the Shapley-based axioms, such as efficiency—often referred to by other names like completeness or conservation properties, none have critically analyzed the appropriateness of the axioms from the perspective of interactions between input features which deemed important information when dealing with complex deep learning models.

8 Conclusion

This study has explored the impact of causal interactions on payoff allocation in cooperative game theory, particularly concerning feature attribution in deep learning models. From the game-theoretic perspective, we show that the Shapley value framework implicitly assumes non-negative interactions between players in the game, which is equivalent to convexity. In non-convex games, the Shapely value and its Efficiency axiom no longer provide a reasonable allocation and may lead to the undervaluation of payoffs or attributions. To extend the Shapley value to non-convex games while keeping its philosophy, we propose a new allocation rule, namely Aggregated Positive Interactions (API), that decomposes contributions into interactions and aggregates positive interactions. Additionally, we introduce an approximation algorithm to enhance computational efficiency in interaction computation for differentiable functions. Our experimental results show that API effectively resolves the counter-intuitive results from the Shapely value and feature attribution methods with the efficiency axiom, providing reasonable attributions.

ACKNOWLEDGMENTS

REFERENCES

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

John F Banzhaf III. Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev.*, 19: 317, 1964.

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

Pierre Dehez. On harsanyi dividends and asymmetric values. *International Game Theory Review*, 19(03):1750012, 2017.

Huiqi Deng, Qihan Ren, Hao Zhang, and Quanshi Zhang. Discovering and explaining the representation bottleneck of dnns. *arXiv preprint arXiv:2111.06236*, 2021.

Huiqi Deng, Na Zou, Mengnan Du, Weifu Chen, Guocan Feng, Ziwei Yang, Zheyang Li, and Quanshi Zhang. Understanding and unifying fourteen attribution methods with taylor interactions. *arXiv preprint arXiv:2303.01506*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley explainability on the data manifold. *arXiv preprint arXiv:2006.01272*, 2020.

Katsushige Fujimoto, Ivan Kojadinovic, and Jean-Luc Marichal. Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices. *Games and Economic Behavior*, 55 (1):72–99, 2006.

Fabian Fumagalli, Maximilian Muschalik, Patrick Kolpaczki, Eyke Hüllermeier, and Barbara Hammer. Shap-iq: Unified approximation of any-order shapley interactions. *Advances in Neural Information Processing Systems*, 36, 2024.

Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of game theory*, 28:547–565, 1999.

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.

Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. Feature relevance quantification in explainable ai: A causal problem. In *International Conference on artificial intelligence and statistics*, pp. 2907–2916. PMLR, 2020.

Giyoung Jeon, Haedong Jeong, and Jaesik Choi. Beyond single path integrated gradients for reliable input attribution via randomized path sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2052–2061, 2023.

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.

Luke Keele and Randolph T Stevenson. Causal interaction and effect modification: same model, different concepts. *Political Science Research and Methods*, 9(3):641–649, 2021.

Indra Kumar, Carlos Scheidegger, Suresh Venkatasubramanian, and Sorelle Friedler. Shapley residuals: Quantifying the limits of the shapley value for explanations. *Advances in Neural Information Processing Systems*, 34:26598–26608, 2021.

Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. *arXiv preprint arXiv:2110.14049*, 2021.

Yongchan Kwon and James Y Zou. Weightedshap: analyzing and improving shapley based feature attributions. *Advances in Neural Information Processing Systems*, 35:34363–34376, 2022.

Mingjie Li and Quanshi Zhang. Does a neural network really encode symbolic concepts? In *International conference on machine learning*, pp. 20452–20469. PMLR, 2023.

Scott Lundberg. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Dov Monderer and Dov Samet. Variations on the shapley value. *Handbook of game theory with economic applications*, 3:2055–2076, 2002.

Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222, 2017.

Woo-Jeoung Nam, Shir Gur, Jaesik Choi, Lior Wolf, and Seong-Whan Lee. Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 2501–2508, 2020.

Judea Pearl. Direct and indirect effects. In *Probabilistic and causal inference: the works of Judea Pearl*, pp. 373–392. 2022.

Hans Peters and Hans Peters. Cooperative games with transferable utility. *Game Theory: A Multi-Leveled Approach*, pp. 151–169, 2015.

Ariel Procaccia, Nisarg Shah, and Max Tucker. On the structure of synergies in cooperative games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

Jie Ren, Mingjie Li, Qirui Chen, Huiqi Deng, and Quanshi Zhang. Defining and quantifying the emergence of sparse concepts in dnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20280–20289, 2023.

Alvin E Roth. Introduction to the shapley value. *The Shapley value*, 1, 1988.

Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Olivér Kiss, Sebastian Nilsson, and Rik Sarkar. The shapley value in machine learning. *arXiv preprint arXiv:2202.05594*, 2022.

M Scott, Lee Su-In, et al. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30:4765–4774, 2017.

Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2, 1953.

Lloyd S Shapley. Cores of convex games. *International journal of game theory*, 1:11–26, 1971.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMlR, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The shapley taylor interaction index. In *International conference on machine learning*, pp. 9259–9268. PMLR, 2020.

Che-Ping Tsai, Chih-Kuan Yeh, and Pradeep Ravikumar. Faith-shap: The faithful shapley interaction index. *Journal of Machine Learning Research*, 24(94):1–42, 2023.

TJ VanderWeele. *Explanation in causal inference: methods for mediation and interaction*. Oxford University Press, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need.(nips), 2017. *arXiv preprint arXiv:1706.03762*, 10:S0140525X16001837, 2017.

J Von Neumann and O Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1944.

Robert J Weber. Probabilistic values for games. *The Shapley Value. Essays in Honor of Lloyd S. Shapley*, pp. 101–119, 1988.

A APPENDIX

For convenience, we represent the cardinality of N, S, T as n, s, t, respectively. $\Pi(S)$ denotes the set of permutations for a player set S.

A.1 PROOF OF THEOREM 1

Lemma 1. *Given a player i and $S \subseteq N \setminus \{i\}$, for any permutation $\pi \in \Pi(S)$ and $k \in \{1, \cdots s\}$,*

$$\Delta_i v(S) = \Delta_i v([\pi]_{k-1}) + \sum_{t=k}^{s} I_{i,\pi_t}([\pi]_{t-1}), \tag{10}$$

where $\pi = (\pi_1, \cdots, \pi_s)$, $[\pi]_k$ represents the subset of players up to the k-th player in the ordering π, $[\pi]_0 \coloneqq \emptyset$ and $[\pi]_s \coloneqq S$.

The proof of Lemma 1 is trivial.

Theorem 1. *The Shapley value is a weighted sum of interactions:*

$$\phi_i(v) = \Delta_i v(\emptyset) + \sum_{t=0}^{n-2} \frac{1}{n} \binom{n-1}{t}^{-1} \sum_{\substack{j \in N \\ j \neq i}} \sum_{\substack{T \subseteq N \setminus \{i,j\} \\ |T|=t}} I_{ij}(T) \tag{11}$$

Proof Sketch:

1. The Shaley value is a weighted sum of local contributions.
2. Each local contribution is decomposed into interactions.
3. Compute the coefficients of individual interactions on the Shapley value.

Proof. The Shapley value is a weighted sum of local contributions:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} p_S \Delta_i v(S) \tag{12}$$

where $p_S = \frac{1}{n}\binom{n-1}{s}^{-1}$.

Each local contribution $\Delta_i v(S)$ can be decomposed into the summation of interactions in varying given sets. Let $\pi \coloneqq (\pi_1, \cdots, \pi_s)$ be a fixed ordering for set S. By Lemma 1, we obtain

$$\begin{aligned} \Delta_i v(S) &= \Delta_i v([\pi]_s) \\ &= \{\Delta_i v([\pi]_s) - \Delta_i v([\pi]_{s-1})\} + \Delta_i v([\pi]_{s-1}) \\ &= I_{i,\pi_s}([\pi]_{s-1}) + \Delta_i v([\pi]_{s-1}) \\ &= \cdots \\ &= \sum_{t=1}^{s} I_{i,\pi_t}([\pi]_{t-1}) + \Delta_i v(\emptyset). \end{aligned} \tag{13}$$

From the decomposition, the Shapley value can be represented as a weighted sum of interactions with a local contribution on the empty set.

$$\phi_i(v) = \Delta_i v(\emptyset) + \sum_{\substack{j \in N \\ j \neq i}} \sum_{T \subseteq N \setminus \{i,j\}} w_T^{ij} I_{ij}(T) \tag{14}$$

To compute the coefficient w_T^{ij}, we transform the Shapley value into permutation forms.

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} p_S \Delta_i v(S) = \sum_{S \subseteq N \setminus \{i\}} \frac{1}{s!} \sum_{\pi \in \Pi(S)} p_S \Delta_i v([\pi]_s) \tag{15}$$

From (13) & (15), the coefficient is represented as follows:

$$\begin{aligned} w_T^{ij} &= \sum_{\pi' \in \Pi(T)} \sum_{S \subseteq N \setminus \{i\}} \frac{1}{s!} \sum_{\pi \in \Pi(S)} p_S \mathbb{1}[\pi' = [\pi]_t] \\ &= \sum_{\pi' \in \Pi(T)} \sum_{S \subseteq N \setminus \{i\}} \frac{1}{s!} \cdot \frac{1}{n} \binom{n-1}{s}^{-1} \cdot (s-t-1)! \\ &= t! \cdot \sum_{s=t+1}^{n-1} \binom{n-t-2}{s-t-1} \cdot \left\{ \frac{1}{s!} \cdot \frac{1}{n} \binom{n-1}{s}^{-1} \cdot (s-t-1)! \right\} \\ &= t! \cdot \frac{1}{n} \sum_{s=t+1}^{n-1} \frac{(n-t-2)!}{(n-s-1)!} \frac{(n-s-1)!}{(n-1)!} \\ &= \frac{1}{n} \sum_{s=t+1}^{n-1} \frac{1}{n-t-1} \binom{n-1}{t}^{-1} \\ &= \frac{1}{n} \binom{n-1}{t}^{-1} \end{aligned} \tag{16}$$

The coefficient w_T^{ij} only depends on the cardinality of T. Finally, the Shapley value is represented as follows:

$$\phi_i(v) = \Delta_i v(\emptyset) + \sum_{t=0}^{n-2} \frac{1}{n} \binom{n-1}{t}^{-1} \sum_{\substack{j \in N \\ j \neq i}} \sum_{\substack{T \subseteq N \setminus \{i,j\} \\ |T|=t}} I_{ij}(T) \tag{17}$$

□

A.2 Proof of Theorem 2

Theorem 2. *A game v is convex if and only if $I_{ij}(R) \geq 0 \; \forall i, j \in N, \forall R \subseteq N \setminus \{i, j\}$.*

Proof. ($\rightarrow$) Necessity

Show $I_{ij}(R) \geq 0$, for any $i, j \in N$ and $R \subseteq N \setminus \{i, j\}$. Set $S = R \cup \{i\}, T = R \cup \{j\}$.

By convexity, we obtain

$$v(R \cup \{i.j\}) + v(R) \geq v(R \cup \{i\}) + v(R \cup \{j\}) \tag{18}$$

$$\Delta_i v(R \cup \{j\}) \geq \Delta_i v(R) \tag{19}$$

Therefore, $I_{ij}(R) \geq 0$.

($\leftarrow$) Sufficiency

Given $S, T \subseteq N$, set $P = S \setminus T$, $Q = T \setminus S$, and $R = S \cap T$. p, q are the cardinality of P, Q, respectively. Choose permutations $\pi \in \Pi(P), \pi' \in \Pi(Q)$. By Lemma 1, the following equations hold for all $k \in \{1, \cdots, q\}$.

$$\begin{aligned} \Delta_{\pi'_k} v(S \cup [\pi']_{k-1}) &= \Delta_{\pi'_k} v(R \cup [\pi']_{k-1}) + \sum_{i=1}^{p} I_{\pi'_k, \pi_i}(R \cup [\pi]_{i-1} \cup [\pi']_{k-1}) \\ &\geq \Delta_{\pi'_k} v(R \cup [\pi']_{k-1}) \end{aligned} \tag{20}$$

Sum over all k, then we obtain

$$\begin{aligned}\sum_{k=1}^{q}\Big\{\Delta_{\pi'_k}v(S\cup[\pi']_{k-1})-\Delta_{\pi'_k}v(R\cup[\pi']_{k-1})\Big\}\geq 0\\ \Big\{v(S\cup[\pi']_q)-v(S)\Big\}-\Big\{v(R\cup[\pi']_q)-v(R)\Big\}\geq 0\\ v(S\cup T)-v(S)-v(T)+v(S\cap T)\geq 0.\end{aligned} \tag{21}$$

Therefore, a game v is convex.

□

A.3 Proof of Theorem 3

Let $i\sim u_1(S)$ be the uniform distribution of player i from set S, and $T\sim u_2(t,S)$ be the uniform distribution of subset S with cardinality $|T|=t$. $\phi(v)$ and $\Delta v(\emptyset)$ denote the Shapley value and the discrete derivatives of all players in a vector form, respectively. We define an interaction vector $I_{\cdot j}(T)\in\mathbb{R}^n$ where the i-th element is $I_{ij}(T)$ for given $T\subseteq N\setminus\{j\}$. We set $I_{ij}(T)=0$ when $i=j$ or $i\in T$, then $I_{\cdot j}(T)$ is well-defined for $T\subseteq N\setminus\{j\}$. Then, we obtain the following vectorized form to estimate the Shapley value with interaction vectors.

Theorem 3. *The vector of the Shapley value $\phi(v)$ is represented as follows:*

$$\phi(v)=\Delta v(\emptyset)+\sum_{t=0}^{n-2}\mathbb{E}_{j\sim u_1(N),T\sim u_2(t,N\setminus\{j\})}[I_{\cdot j}(T)].$$

Proof. Since we set $I_{ij}=0$ when $i=j$ or $i\in T$, the following equality holds for all $i\in N$:

$$\begin{aligned}\sum_{\substack{j\in N\\ j\neq i}}\sum_{\substack{T\subseteq N\setminus\{i,j\}\\ |T|=t}}I_{ij}(T)&=\sum_{j\in N}\sum_{\substack{T\subseteq N\setminus\{j\}\\ |T|=t}}I_{ij}(T)\\ &=n\cdot\binom{n-1}{t}\cdot\mathbb{E}_{j\sim u_1(N),T\sim u_2(t,N\setminus\{j\})}[I_{ij}(T)].\end{aligned} \tag{22}$$

By combining Theorem 1 and Equation (22) and vectorizing the result, we obtain the conclustion.

$$\phi(v)=\Delta v(\emptyset)+\sum_{t=0}^{n-2}\mathbb{E}_{j\sim u_1(N),T\sim u_2(t,N\setminus\{j\})}[I_{\cdot j}(T)].$$

□

A.4 An Illustrative example of the Max Function

We provide an illustrative example of the max function demonstrated in Section 4. Pattern scores for image classification are computed in each region, then max pooling is applied, and the results are summed for the final output. In this example, efficiency constrains the sum of attributions in each region to 7 and 4, respectively. Although features that highly contribute to the output are extracted in region 1 (the first max pooling), the Shapley Value of the feature '7' is lower than that of the feature '4' in region 2 (the second max pooling) due to strong negative interactions from an inefficient coalition as discussed in Section 4. The proposed API effectively alleviates this undervaluation, assigning a higher attribution to the feature '7' (7) than to the feature '4' (4), by removing the problematic negative interactions.

A.5 Experiment results for complexity

We additionally conducted quantitative experiments. We inserted the features (i.e., patches) in descending order of attribution scores and compute the output class probability distribution to measure the KL divergence with the original distribution (which includes all features). Then, we identified

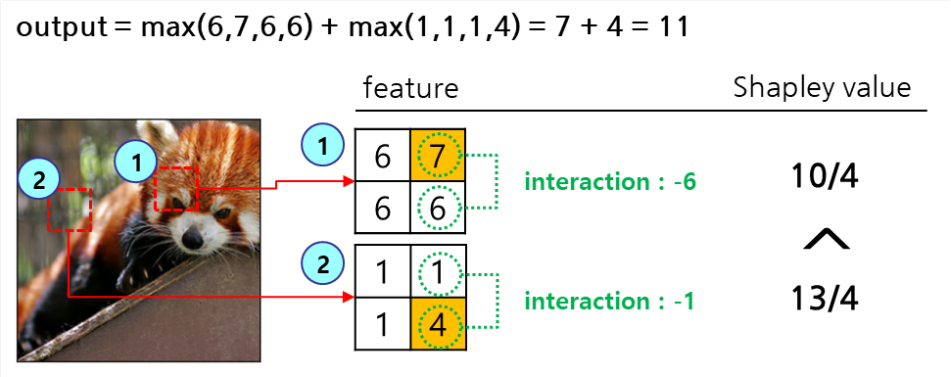

Figure 4: Illustrative example of the Max Function. Due to the negative interactions, the feature '7' receives a lower Shapley value than the feature '4', which is counter-intuitive.

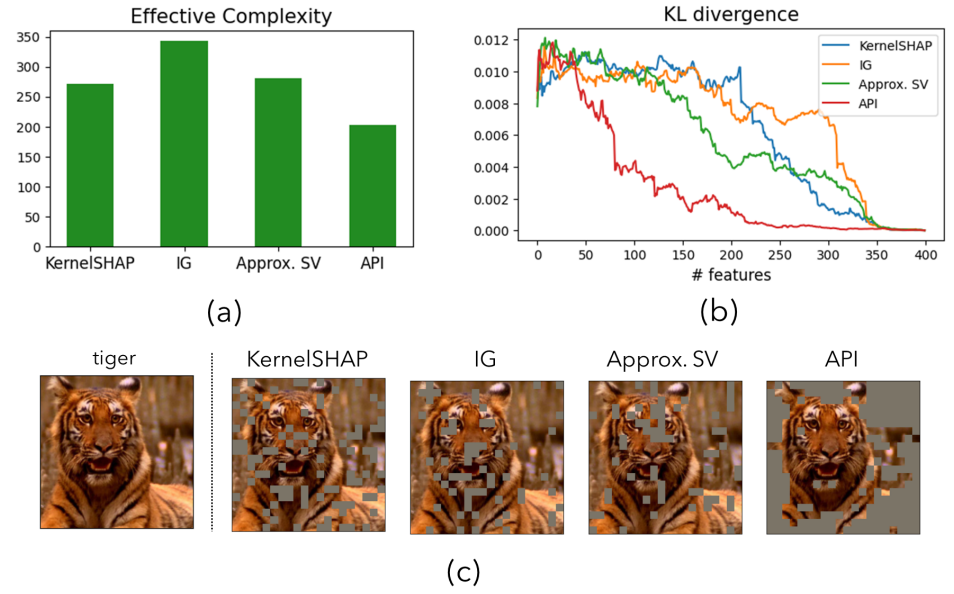

Figure 5: Experiment results for complexity. We measured the minimum number of features inserted (in descending order of attribution scores) to achieve the KL divergence between the model output and the original model output below 0.001. API requires fewer features to reproduce the original model output compared to other methods.

the minimum number of features required to reduce the KL divergence to below 0.001. Figure 5 (a) shows the average number of features required, with API achieving almost a 30% reduction compared to other methods. More specifically, the average numbers are 272.34, 342.94, 281.59, and 203.32 for KernelSHAP, IG, Approximated Shapley value, and API, respectively. This indicates API requires fewer features to reproduce the original model output, effectively capturing the relevant features for the model prediction.

Figure 5 (b) shows the KL divergence with respect to the number of features inserted for a specific image, as depicted in Figure 5 (c). We observe that API leads to a more rapid decrease in KL divergence with relatively fewer features inserted. Specifically, the number of features required to reach the KL divergence of 0.001 is 311, 339, 337, and 203 for KernelSHAP, IG, Approximated Shapley value, and API, respectively. Furthermore, Figure 5 (c) shows a visualization of the features inserted to achieve the KL divergence of 0.001. With the proposed API, we can reproduce the original decision with fewer irrelevant features inserted compared to other methods.

A.6 RUNTIME OF API

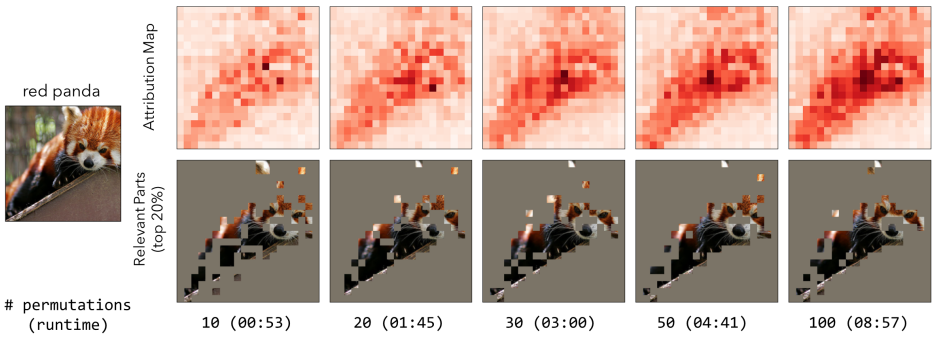

Figure 6: Runtime with respect to the number of permutation samples. API takes around 55 seconds for every 10 permutation samples. Notably, stable results can be achieved with as few as 30 permutations.

Figure 6 shows the runtime of API for different numbers of permutation samples, using an ImageNet classifier. We observed that stable results can be achieved with as few as 30 permutations. On an RTX 6000 GPU, this process took approximately 3 minutes. While this is slower compared to IG, it is comparable to other permutation-based methods grounded in game theory. Handling interactions inherently involves significant computational costs. Thus, accelerating this process through path-based approximations is one of out future research directions.