TOWARDS TRACTABLE, PROVABLY EXACT SHAPLEY INTERACTIONS AND BEYOND IN NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Shapley interactions and the broader family of game-theoretic explanations are widely used to characterise how features contribute to neural network predictions. Yet exact computation requires reasoning over exponentially many feature subsets, and these quantities are therefore typically estimated, without provable guarantees. We introduce a verification-based algorithm for discovering additive structures within neural networks, enabling the computation of *provable upper and lower bounds on the exact values* of a broad family of game-theoretic explanations, including Shapley interactions of any order and sparse Fourier explanations. We then analyse the computational limits of this problem, proving that computing Shapley interactions remains #P-hard even for networks with as few as two ReLU neurons or a single attention layer. To address this, we introduce a robustness-based training scheme that encourages additive structure in the learned model, drastically improving the scalability of our algorithm. Experimentally, our approach handles substantially larger models and search spaces and brings exact certified game-theoretic interactions to transformer architectures for the first time. Overall, our results provide game-theoretic attributions and interactions with provable guarantees where estimation would otherwise be required, while also providing a principled basis for evaluating such estimators.

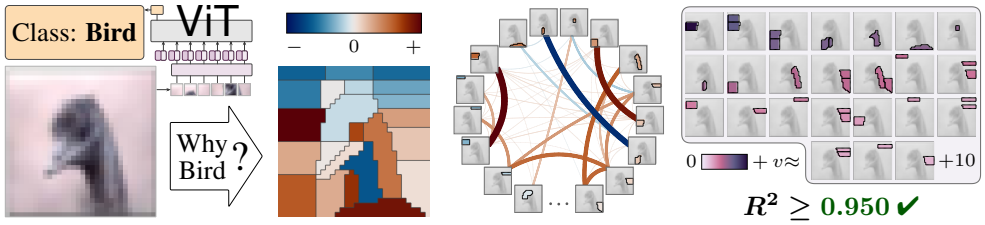
1 INTRODUCTION

Shapley additive explanations (SHAP) (Lundberg & Lee, 2017) are among the most widely used feature attribution methods, and they head a broader family of explanations grounded in cooperative game theory, including Shapley interaction indices, which capture joint effects between features, and sparse Fourier explanations (Kang et al., 2025; Butler et al., 2025). These explanations can be computed exactly for simple, already interpretable models such as decision trees and additive models (Muschalik et al., 2024b). For neural networks, however, where such explanations matter most, exact computation is intractable, since it ranges over exponentially many feature subsets. This difficulty compounds for higher-order interactions, where the number of quantities to compute grows with the order as well. Therefore, they are almost always estimated, which forfeits the axioms that characterise them, as these hold for the exact value alone. This is untenable wherever explanations inform consequential decisions, and it leaves the field without ground truth for evaluating estimators.

Recent work shows that neural network verification offers a way out. Rather than probing the value function one coalition at a time, affine bound propagation (ABP) (Zhang et al., 2018; Singh et al., 2019b; Xu et al., 2021) bounds it over entire regions of the coalition space at once, and branch and bound refines these regions until exact SHAP values are recovered (Boetius et al., 2026). However, the approach of Boetius et al. (2026) remains narrow in two respects: *(i)* it computes first-order Shapley values alone, thereby ignoring interactions between features, and *(ii)* it scales only to small models and search spaces.

Our contributions. We address both of these significant limitations by developing a general framework for obtaining certified explanations across a broad family of game-theoretic attribution methods, extending substantially beyond first-order interactions, and provide a theoretical analysis of the underlying computational problem. To contend with the considerable computational complexity of game-theoretic attribution, we propose a training-based approach that dramatically improves the scalability of provably exact explanations. More specifically, our contributions are as follows:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



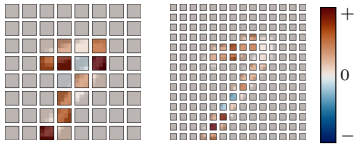
(a) Vision Transformer (b) Exact SHAP (c) Exact Order-2 SII (d) Certified PROXYSPEX

Figure 1: **Explaining a Vision Transformer.** Our work enables provably exact SHAP (b) and interaction indices (c) for transformers for the first time, and certifies Fourier explanations (d).

A general algorithm for certified higher-order explanations. We introduce Verified Decomposition into Additive Pieces (VERIDAP), a framework that leverages neural network verification to compute exact Shapley interactions of any order, along with provable bounds on them at every point before termination. VERIDAP decomposes a neural cooperative game into pieces on which the value function is tightly enclosed by two affine functions, and refines this decomposition until the exact values are recovered. Our framework certifies not one explanation but a whole family, covering many popular interaction indices of any order. This generality further lets us certify sparse Fourier explanations such as SPEX (Kang et al., 2025) and PROXYSPEX (Butler et al., 2025), providing formal proofs of interaction sparsity (Gorji et al., 2025) in practical neural networks for the first time.

A theoretical analysis of what verification can reach. With our framework in place, scale becomes a far more central question, due to the need to certify a rapidly growing number of interaction targets over ever finer pieces. Asking what determines scalability, we provide a theoretical computational complexity analysis that proves an asymmetry between the two natural parameters of the problem. While restricting the input dimension yields tractability at any interaction order, restricting the model size does not. Concretely, the problem remains hard for models with as few as two ReLU neurons, or a single attention layer. Therefore, changing the architecture does not help, and tractability must instead be obtained by shaping the input-space geometry of the trained neural network.

Lifting scalability with certified training. Motivated by these findings, we introduce a certified-training approach based on robust optimisation that directly reshapes the neural network's input geometry, thereby extending verified computation to substantially larger models and search spaces (Figure 2), at little cost in accuracy. This carries exact, certified explanations to transformers for the first time (Figure 1). Our approach computes exact Shapley interactions over search spaces and architectures far beyond the reach of prior exact methods. We use the resulting ground-truth interactions as a benchmark for evaluating Shapley estimators on neural networks, which has so far been infeasible.



(a) Previously (b) Our Work

Figure 2: **Scalability through Certified Training.** We compute exact Shapley values for search spaces of $2^{196} > 10^{59}$ coalitions.

Overall, we extend exact, certified explanations from first-order attributions to a substantially more general family of game-theoretic explanations, covering a broad class of any-order interactions, including the Fourier spectrum, and provide a theoretical analysis revealing what governs their hardness. Guided by this analysis, we introduce a training approach that drastically widens the range of models and search spaces for which we can provide explanations with provable guarantees.

2 PRELIMINARIES

We consider a neural network $f : \mathbb{R}^d \to \mathbb{R}$, $d \in \mathbb{N}$ as a composition of affine and non-linear layers, such as a ResNet (He et al., 2016) or a transformer (Vaswani et al., 2017). The output of f is the prediction we interpret, e.g. the logit or the predicted probability of one class.

Notation. Let $[d] := \{1, \ldots, d\}$ and $\mathbb{B}^d := \{0, 1\}^d \subset \mathbb{R}^d$, with the all-zero and all-one vectors $\mathbf{0}_d, \mathbf{1}_d \in \mathbb{B}^d$, and let $\odot$ denote the element-wise product. A *coalition* $\mathbf{s} \in \mathbb{B}^d$ is the indicator vector of its feature set $S := \{i : \mathbf{s}_i = 1\}$, with size $\|\mathbf{s}\|_1$. We identify $\mathbf{s}$ with S, writing $\mathbf{s} \cong S$, and use $\mathbf{s} \leq \mathbf{t}$, $\mathbf{s} \vee \mathbf{t} := \min(\mathbf{s} + \mathbf{t}, \mathbf{1}_d)$, $\mathbf{s} \setminus \mathbf{t} := \mathbf{s} \odot (\mathbf{1}_d - \mathbf{t})$ for subset, union, difference.

2.1 COOPERATIVE GAMES FOR INTERPRETING NEURAL NETWORKS

Let $\mathbf{x} \in \mathbb{R}^d$. For interpreting the prediction $f(\mathbf{x})$, we define a cooperative game among the input features. The *value function* $v : \mathbb{B}^d \to \mathbb{R}$ is the prediction when removing certain features. For simplicity, we consider the *baseline value function*, which sets the features with $\mathbf{s}_i = 0$ to a *baseline input* $\mathbf{y} \in \mathbb{R}^d$, keeping the features with $\mathbf{s}_i = 1$ at $\mathbf{x}$ (Sundararajan & Najmi, 2020), such that $v(\mathbf{s}) := f\big(\mathbf{s} \odot \mathbf{x} + (\mathbf{1}_d - \mathbf{s}) \odot \mathbf{y}\big)$. Appendix A.2 discusses the *marginal value function* (Lundberg & Lee, 2017). Interpreting the neural network now corresponds to determining how much a feature $i \in [d]$ or a coalition of features $\mathbf{s} \in \mathbb{B}^d$ contributes to the total value $v(\mathbf{1}_d) = f(\mathbf{x})$.

Features *interact* when their joint effect differs from the sum of their individual effects. The *discrete derivative* of v with respect to $\mathbf{s}$ in the presence of a disjoint coalition $\mathbf{t}$ measures this as the effect of adding $\mathbf{s}$ to $\mathbf{t}$ not explained by any sub-coalition $\mathbf{u}$ of $\mathbf{s}$, such that $\Delta v(\mathbf{s};\, \mathbf{t}) := \sum_{\mathbf{u} \leq \mathbf{s}} (-1)^{\|\mathbf{s} \setminus \mathbf{u}\|_1} v(\mathbf{u} \vee \mathbf{t})$, where $\mathbf{s} \odot \mathbf{t} = \mathbf{0}_d$. The value function is *additive*, $v(\mathbf{s}) = \mathbf{w}^\mathsf{T}\mathbf{s} + b$ with $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, exactly when no features interact, that is, $\Delta v(\mathbf{s};\, \mathbf{t}) = 0$ whenever $\|\mathbf{s}\|_1 \geq 2$.

Interaction Indices. Interaction indices (IIs) decompose the prediction of a neural network into the contributions of *coalitions* of features $\mathbf{s} \in \mathbb{B}^d$, by aggregating the discrete derivative of $\mathbf{s}$ over all *contexts* $\mathbf{t}$ in which its features can be added. Inserting $\Delta v(\mathbf{s};\, \mathbf{t})$ and collecting the coefficient of each evaluation $v(\mathbf{t})$ shows that every II is a *linear explanation* of the value function

$$\mathcal{I}(\mathbf{s}) := \sum_{\mathbf{t} \leq \mathbf{1}_d - \mathbf{s}} \lambda_{\mathbf{s}}(\|\mathbf{t}\|_1) \Delta v(\mathbf{s};\, \mathbf{t}) = \sum_{\mathbf{t} \in \mathbb{B}^d} \lambda_{\mathbf{s}}(\|\mathbf{t} \setminus \mathbf{s}\|_1)(-1)^{\|\mathbf{s} \setminus \mathbf{t}\|_1} v(\mathbf{t}), \tag{1}$$

where $\lambda_{\mathbf{s}} : \mathbb{N}_0 \to \mathbb{R}$ is the *II kernel*, which weights a context by its size. Equation (1) includes well-known interaction indices, which differ only in how they weight the contexts. The *Shapley II (SII)* (Grabisch & Roubens, 1999) uses $\lambda_{\mathbf{s}}^{\mathrm{SII}}(t) = (d - \|\mathbf{s}\|_1 - t)!\, t!/(d - \|\mathbf{s}\|_1 + 1)!$, and its values at singleton coalitions, $\|\mathbf{s}\|_1 = 1$, are the *Shapley values* (Shapley, 1952) computed by SHAP (Lundberg & Lee, 2017). The *Banzhaf II* (Penrose, 1946; Banzhaf, 1964) weights all contexts equally, $\lambda_{\mathbf{s}}^{\mathrm{BII}}(t) = 2^{-(d - \|\mathbf{s}\|_1)}$, and fixed order $k = \|\mathbf{s}\|_1$ IIs include the Shapley–Taylor II (Sundararajan et al., 2020), the Faithful Shapley II (Tsai et al., 2023), and the k-Shapley II (Bordt & von Luxburg, 2023). Appendix A.3 provides additional details.

Sparse Fourier Explanations. Sparse Fourier explanations (SFEs), such as SPEX (Kang et al., 2025) and PROXYSPEX (Butler et al., 2025), approximate the value function using a sparse expansion in the Fourier basis $\chi_{\mathbf{s}}(\mathbf{t}) := (-1)^{\mathbf{s}^\mathsf{T}\mathbf{t}}$ of the Boolean cube (O'Donnell, 2014). The *Fourier coefficient* of $\mathbf{s}$ is an II of Equation (1) with constant kernel, i.e. a rescaled Banzhaf II (Kang et al., 2025),

$$\widehat{v}(\mathbf{s}) := 2^{-d} \sum_{\mathbf{t} \in \mathbb{B}^d} \chi_{\mathbf{s}}(\mathbf{t}) v(\mathbf{t}) = 2^{-d}(-1)^{\|\mathbf{s}\|_1} \sum_{\mathbf{t} \leq \mathbf{1}_d - \mathbf{s}} \Delta v(\mathbf{s};\, \mathbf{t}) = (-2)^{-\|\mathbf{s}\|_1} \mathcal{I}^{\mathrm{BII}}(\mathbf{s}), \tag{2}$$

such that $v(\mathbf{s}) = \sum_{\mathbf{t} \in \mathbb{B}^d} \widehat{v}(\mathbf{t}) \chi_{\mathbf{t}}(\mathbf{s})$. Writing $\|g\|_2^2 := 2^{-d} \sum_{\mathbf{s} \in \mathbb{B}^d} g(\mathbf{s})^2$ for $g : \mathbb{B}^d \to \mathbb{R}$, Parseval's theorem relates the value function v to its Fourier coefficients via $\|v\|_2^2 = \sum_{\mathbf{s} \in \mathbb{B}^d} \widehat{v}(\mathbf{s})^2$. SFEs search for a small set of distinct coalitions $\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)}$, with estimates $\widetilde{v}_1, \ldots, \widetilde{v}_K$, such that the *surrogate* $h(\mathbf{t}) := \sum_{k \in [K]} \widetilde{v}_k \chi_{\mathbf{s}^{(k)}}(\mathbf{t})$ maximises the *faithfulness* $R^2 := 1 - \|h - v\|_2^2 / \|v - \widehat{v}(\mathbf{0}_d)\|_2^2$, where the denominator is the variance of the game, since $\widehat{v}(\mathbf{0}_d) = 2^{-d} \sum_{\mathbf{s} \in \mathbb{B}^d} v(\mathbf{s})$ is its mean.

2.2 AFFINE BOUNDS ON NEURAL NETWORKS

Unlike previous works that empirically estimate $\mathcal{I}(\mathbf{s})$ and R^2, we provide guarantees for both quantities leveraging ABP for neural networks (Singh et al., 2019b; Zhang et al., 2018; Xu et al., 2021). Bound propagation exploits that the neural network f is a composition of layers to compute guaranteed bounds on f. Given a *box* $[\underline{\mathbf{x}}, \overline{\mathbf{x}}] := \{\mathbf{x} \in \mathbb{R}^d : \underline{\mathbf{x}} \leq \mathbf{x} \leq \overline{\mathbf{x}}\}$, ABP propagates two affine functions layer by layer, producing affine bounds $\underline{f}, \overline{f} : \mathbb{R}^d \to \mathbb{R}$, $\underline{f}(\mathbf{x}) := \underline{\mathbf{w}}^\mathsf{T}\mathbf{x} + \underline{b}$ and $\overline{f}(\mathbf{x}) := \overline{\mathbf{w}}^\mathsf{T}\mathbf{x} + \overline{b}$ with $\underline{\mathbf{w}}, \overline{\mathbf{w}} \in \mathbb{R}^d$ and $\underline{b}, \overline{b} \in \mathbb{R}$. ABP guarantees three key properties: *(i)* the affine bounds satisfy $\underline{f}(\mathbf{x}) \leq f(\mathbf{x}) \leq \overline{f}(\mathbf{x})$ for every $\mathbf{x} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}]$, *(ii)* if all layers of f are affine, the affine bounds are *exact*, meaning $\underline{f}(\mathbf{x}) = \overline{f}(\mathbf{x}) = f(\mathbf{x})$ for every $\mathbf{x} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}]$ (Zhang et al., 2018), and *(iii)* the bounds are also exact for a degenerate box $[\mathbf{x}, \mathbf{x}]$ (Boetius et al., 2025). We refer to Zhang et al. (2018) for a detailed introduction to a representative ABP method.

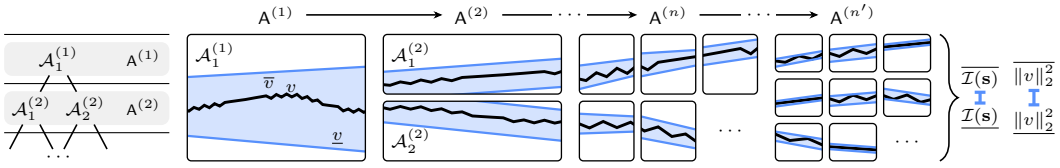


Figure 3: **Decompositions into Additive Pieces (DAPs)** partition the coalition space $\mathbb{B}^d$ to tightly bound the value function v using affine bounds $\underline{v}, \overline{v}$. Starting from a coarse partition $\mathsf{A}^{(1)}$, VERIDAP successively tightens the affine bounds by refining the partition into $\mathsf{A}^{(2)}, \ldots, \mathsf{A}^{(n')}$.

Algorithm 1 VERIDAP

Input: Value function $v : \mathbb{B}^d \to \mathbb{R}$, batch size $b \in \mathbb{N}$
$\mathcal{A}_1^{(1)} \leftarrow [\mathbf{0}_d, \mathbf{1}_d]$; $\mathsf{A}^{(1)} \leftarrow \{\mathcal{A}_1^{(1)}\}$
$(\underline{v}_1^{(1)}, \overline{v}_1^{(1)}) \leftarrow \mathrm{ABP}(v, \mathcal{A}_1^{(1)})$ *// See Section 2.2 for* ABP.
for $n \in \{2, \ldots, 2^d\}$ **do**
  $\mathsf{B}^{(n)} \leftarrow \textsc{Select}(\mathsf{A}^{(n-1)}, b)$; $\mathsf{C}^{(n)} \leftarrow \textsc{Split}(\mathsf{B}^{(n)})$ // $|\mathsf{B}^{(n)}| \leq b$, $|\mathsf{C}^{(n)}| = 2|\mathsf{B}^{(n)}|$
  $\{(\underline{v}_j^{(n)}, \overline{v}_j^{(n)})\}_{j \in [|\mathsf{C}^{(n)}|]} \leftarrow \mathrm{ABP}(v, \mathsf{C}^{(n)})$
  $\mathsf{A}^{(n)} \leftarrow (\mathsf{A}^{(n-1)} \setminus \mathsf{B}^{(n)}) \cup \mathsf{C}^{(n)}$
end for

Bounds on the Value Function. Since $\mathbb{B}^d$ is embedded in $[\mathbf{0}_d, \mathbf{1}_d] \subset \mathbb{R}^d$, ABP can also bound the value function v. In fact, since $\mathbf{s} \mapsto \mathbf{s} \odot \mathbf{x} + (\mathbf{1}_d - \mathbf{s}) \odot \mathbf{y}$ is affine, v is f with this map as a pre-processing layer. For any box $[\underline{\mathbf{s}}, \overline{\mathbf{s}}] \subseteq [\mathbf{0}_d, \mathbf{1}_d]$, ABP thus provides $\underline{\mathbf{w}}, \overline{\mathbf{w}} \in \mathbb{R}^d$ and $\underline{b}, \overline{b} \in \mathbb{R}$ with

$$\underline{v}(\mathbf{s}) := \underline{\mathbf{w}}^\mathsf{T}\mathbf{s} + \underline{b} \leq v(\mathbf{s}) \leq \overline{\mathbf{w}}^\mathsf{T}\mathbf{s} + \overline{b} =: \overline{v}(\mathbf{s}), \qquad \mathbf{s} \in [\underline{\mathbf{s}}, \overline{\mathbf{s}}]. \tag{3}$$

3 DECOMPOSING NEURAL COOPERATIVE GAMES INTO ADDITIVE PIECES

Computing an II by Equation (1) evaluates v on all 2^d coalitions, which is impractical beyond a few features. However, where v is additive, this cost disappears: features do not interact there and any linear explanation follows in closed form from the affine coefficients (Lundberg et al., 2020; Bordt & von Luxburg, 2023; Muschalik et al., 2024b). When the coalition space is decomposed into subcubes (*pieces*) on which v is additive, the cost of computing IIs scales with the number of pieces rather than the number of coalitions. Since neural networks rarely admit a small decomposition into *exactly* additive pieces, we consider pieces where ABP (Section 2.2) proves that v is tightly enclosed by two affine functions, as visualised in Figure 3. We call such a structure a *decomposition into (approximately) additive pieces (DAP)*. As proved in this section, the enclosures of v provided by DAPs allow us to bound IIs and certify SFEs.

Definition 3.1 (Decomposition into Additive Pieces)**.** Let $\mathsf{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_J\}$ be a *partition* of $\mathbb{B}^d$ into subcubes, called *pieces*, $\mathcal{A}_j = [\underline{\mathbf{a}}_j, \overline{\mathbf{a}}_j]$ with $\underline{\mathbf{a}}_j, \overline{\mathbf{a}}_j \in \mathbb{B}^d$. Given affine bounds $\underline{v}_j \leq v \leq \overline{v}_j$ on each piece, as in Equation (3), we call A an *ε-approximate decomposition into additive pieces (DAP)*, $\varepsilon \geq 0$, if

$$\overline{v}_j(\mathbf{s}) - \underline{v}_j(\mathbf{s}) \leq \varepsilon \qquad \text{for all } \mathbf{s} \in \mathcal{A}_j \text{ and } j \in [J].$$

That is, on every piece, v is additive up to an error of ε. For $\varepsilon = 0$, call A *exact*.

We now introduce our core algorithm: VERIDAP (Algorithm 1). VERIDAP computes DAPs by branch and bound (Land & Doig, 2010; Morrison et al., 2016): starting from a coarse decomposition, it grows a binary tree whose nodes are the pieces, uses ABP to compute value bounds for each node, and refines the nodes that loosen the target bounds most. Since ABP bounds become tighter for smaller decomposition pieces, VERIDAP computes ε-approximate DAPs for ever smaller ε.

Initialisation. VERIDAP starts with the whole coalition space as a single piece, $\mathsf{A}^{(1)} := \{[\mathbf{0}_d, \mathbf{1}_d]\}$, whose ABP bounds are typically loose and need to be refined by *splitting*.

Splitting Pieces (SPLIT). SPLIT partitions a piece $\mathcal{A}$ into two children $\mathcal{A}'$ and $\mathcal{A}''$ by splitting on one of its *free* features i with $\underline{\mathbf{a}}_i < \overline{\mathbf{a}}_i$. In $\mathcal{A}'$, i is fixed to be present ($\mathbf{s}_i = 1$), while in $\mathcal{A}''$, i is fixed

to be absent ($\mathbf{s}_i = 0$). Starting from $\mathsf{A}^{(1)}$, splitting grows a binary tree, in which each path fixes the features split along it. After splitting, the effect of feature i and its interactions with the remaining features are captured by the partition rather than by a single pair of affine bounds, which is why the bounds tighten. For choosing i, we use *smears* (Kearfott, 1996; Boetius et al., 2026), as introduced in Appendix B.

Selecting Pieces (SELECT). Since expanding every leaf would double the number of pieces in each iteration, VERIDAP grows the tree best-first, expanding $b \in \mathbb{N}$ leaves in every iteration. To tighten the target bounds (e.g. II bounds) as fast as possible, we select the pieces that contribute the most looseness to the target, as measured by the *priority* p_j defined in Appendix B.

Bounding IIs and Certifying SFEs. During each iteration of VERIDAP, we aggregate the value bounds into target bounds, such as guaranteed bounds on IIs. Similarly, given an SFE, we can bound the faithfulness R^2, allowing us to *certify* the SFE if the R^2 lower bound reaches a target level. Due to space constraints, we only provide short proof sketches, with full proofs in Appendix C.

Theorem 3.2 (II Bounds)**.** *Let $\mathcal{I}$ be an II as in Equation* (1)*, $\mathbf{s} \in \mathbb{B}^d$, and A a DAP. For each piece $\mathcal{A} \in \mathsf{A}$, let $\mathcal{I}_\mathcal{A}(\mathbf{s})$ be the part of the linear form in Equation* (1) *contributed by the coalitions in $\mathcal{A}$, so that $\mathcal{I}(\mathbf{s}) = \sum_{\mathcal{A}\in\mathsf{A}} \mathcal{I}_\mathcal{A}(\mathbf{s})$. Then each $\mathcal{I}_\mathcal{A}(\mathbf{s})$ admits efficiently computable bounds $\underline{\mathcal{I}}_\mathcal{A}(\mathbf{s}) \le \mathcal{I}_\mathcal{A}(\mathbf{s}) \le \overline{\mathcal{I}}_\mathcal{A}(\mathbf{s})$ computed from the affine value function bounds, and*

$$\underline{\mathcal{I}}(\mathbf{s}) := \sum_{\mathcal{A}\in\mathsf{A}} \underline{\mathcal{I}}_\mathcal{A}(\mathbf{s}) \le \mathcal{I}(\mathbf{s}) \le \sum_{\mathcal{A}\in\mathsf{A}} \overline{\mathcal{I}}_\mathcal{A}(\mathbf{s}) =: \overline{\mathcal{I}}(\mathbf{s}).$$

Proof Sketch. Since the pieces partition $\mathbb{B}^d$, $\mathcal{I}(\mathbf{s})$ splits into piece contributions. Rearranging terms, we obtain a sum of kernel weights and discrete derivatives that are enclosed using the value bounds.

While stated for a single coalition, Theorem 3.2 bounds $\mathcal{I}(\mathbf{s})$ for any number of coalitions from one shared DAP. This includes the Fourier coefficients of Equation (2). For certifying SFEs, we additionally bound the *SFE residual* $\|h - v\|_2^2$, and $\|v\|_2^2$ as the residual of the auxiliary $h \equiv 0$.

Theorem 3.3 (SFE Residual Bounds)**.** *Let h be the surrogate of an SFE and A a DAP. For each piece $\mathcal{A} \in \mathsf{A}$, the residual is enclosed by $\underline{r}_\mathcal{A} \le h(\mathbf{t}) - v(\mathbf{t}) \le \overline{r}_\mathcal{A}$ for all $\mathbf{t} \in \mathcal{A}$, with closed-form bounds computed from the affine bounds on $\mathcal{A}$. Then*

$$\underline{E} := \sum_{\mathcal{A}\in\mathsf{A}} 2^{-d}|\mathcal{A}| \min_{r\in[\underline{r}_\mathcal{A},\overline{r}_\mathcal{A}]} r^2 \le \|h - v\|_2^2 \le \sum_{\mathcal{A}\in\mathsf{A}} 2^{-d}|\mathcal{A}| \max_{r\in[\underline{r}_\mathcal{A},\overline{r}_\mathcal{A}]} r^2 =: \overline{E}.$$

Proof Sketch. The mean square $\|h - v\|_2^2$ splits into the pieces, each weighted by its share $2^{-d}|\mathcal{A}|$ of the coalitions. On each piece, surrogate terms whose coalition is fixed are constant, the others are bounded by the magnitude of their coefficient, and the affine bounds enclose v.

Corollary 3.4 (Faithfulness Bounds)**.** *Let $\underline{E} \le \|h - v\|_2^2 \le \overline{E}$, $\underline{\|v\|_2^2} \le \|v\|_2^2 \le \overline{\|v\|_2^2}$, and $\underline{\widehat{v}}(\mathbf{s}) \le \widehat{v}(\mathbf{s}) \le \overline{\widehat{v}}(\mathbf{s})$ for $\mathbf{s} \in \{\mathbf{0}_d, \mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)}\}$. Together, these provide closed-form bounds $\underline{V} \le \|v - \widehat{v}(\mathbf{0}_d)\|_2^2 \le \overline{V}$, such that $1 - \overline{E}/\underline{V} \le R^2 \le 1 - \underline{E}/\overline{V}$, whenever $\underline{V} > 0$.*

Batching and Pruning. Following Xu et al. (2021), we process pieces in batches on massively parallel hardware. For managing the memory requirements of VERIDAP, we drop pieces from memory once their contribution to the target bounds is exact. More details are provided in Appendix B.

4 CONVERGENCE AND COMPLEXITY ANALYSIS

We show that VERIDAP computes arbitrarily precise DAPs, converging to an exact DAP in finite time. While, in the worst case, VERIDAP only achieves this with an exponential number of iterations, we prove that this is an irreducible property of computing IIs. Appendix C contains our full proofs.

Proposition 4.1 (Arbitrary Precision)**.** *For every $\varepsilon \ge 0$, there exists $n \in \mathbb{N}$, such that $\mathsf{A}^{(n)}$, as computed by* VERIDAP*, is an ε-approximate DAP.*

While this shows the convergence of VERIDAP, it relies on enumerating all coalitions in $\mathcal{O}(2^d)$ iterations. Some blow-up is expected, given the #P-hardness of computing Shapley values of neural

networks (Van den Broeck et al., 2022; Arenas et al., 2023; Marzouk et al., 2025), but it is not obvious where that hardness sits. It could be carried by the input dimension d, which determines the size of the coalition space, or the network size, which determines the complexity of the value function. Surprisingly, we find that reducing the network size does *not* make the problem more tractable.

Theorem 4.2 (Hardness for Minimal Models)**.** *Computing the Shapley value φ_i of the baseline value function is #P-hard, already for Boolean $\mathbf{x}, \mathbf{y}$ and already for networks f that are (i) a single hidden layer of* two *ReLU neurons, or (ii) a* single *softmax attention layer over two tokens.*

Proof Sketch. Both cases reduce to a weighted voting game, for which computing Shapley values is #P-hard (Deng & Papadimitriou, 1994; Elkind et al., 2009).

Appendix C.4 proves complementary hardness results for bounding φ_i, the SII, k-SII, STII, and FSII, and for computing R^2. In contrast, if the input dimension is small, one can simply enumerate the coalition space, yielding tractability for every architecture. Overall, the two parameters play strongly asymmetric roles, and restricting the model size does not help scalability.

5 Training for Efficient Explanation Computation

To scale VeriDAP to larger input spaces despite Theorem 4.2, we introduce a training approach for shaping the value function geometry at little cost in accuracy. Concretely, we apply *certified training* techniques, originally developed for making neural networks more robust to adversarial attacks (Gowal et al., 2018; Müller et al., 2023; De Palma et al., 2024; Kaulen et al., 2026). As a side effect, these approaches make bound propagation, as leveraged by VeriDAP, yield tighter bounds for the trained networks. Below, we briefly introduce certified training and describe how we leverage it for obtaining more efficiently interpretable networks. Appendix E provides additional details.

Certified Training. The core idea behind certified training is to compute an upper bound on the training loss under adversarial perturbations of the training samples (Gowal et al., 2018). Adversarial perturbations either lie in ℓ_∞ balls of a certain radius or arbitrarily change a bounded number of features (Chiang et al., 2020; Shapira & Drachsler-Cohen, 2026). The loss upper bound is computed using interval bound propagation (Gowal et al., 2018; Jovanovic et al., 2022), following the approach of Section 2.2, but using intervals instead of affine bounds. There are two ways to decrease the loss upper bound: choose more robust network parameters, or choose parameters that make bound propagation produce a tighter upper bound. Practically, both occur during certified training, which enables VeriDAP to find tighter DAPs faster for the trained networks. Current state-of-the-art certified-training methods interpolate between certified and adversarial training (Madry et al., 2018).

Tradeoff Tuning. Certified training trades *clean* performance, for example, test set accuracy, for increased adversarial robustness. Hyperparameter optimisation allows for tuning this tradeoff by navigating a Pareto frontier (Kaulen et al., 2026; Daulton et al., 2020). While we found it impractical to directly tune VeriDAP runtime, the relative width of the value function bounds over the initial piece $[\mathbf{0}_d, \mathbf{1}_d]$ (Appendix E) provides a good proxy, such that tuning it against clean performance yields a rich frontier of networks.

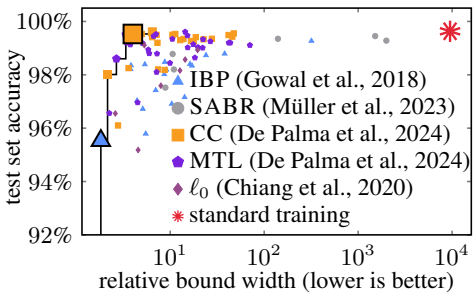


Figure 4: **MNIST Certified-Training Pareto Front.** Each point in this plot is a network trained during hyperparameter tuning. Networks on the Pareto front ⌜ provide the best tradeoffs between bound tightness and clean performance. The highlighted networks ✳, ■, ▲ are analysed in Section 6.

Figure 4 contains the results of tuning a convolutional neural network on `MNIST` (LeCun et al., 1998). The network ■ obtains comparable performance (99.53 % test set accuracy) to standard training ✳ (99.63 %), while providing orders of magnitude tighter value function bounds. The following section demonstrates the impact of certified training using ■, ✳, and ▲ (95.53 % accuracy).

Table 1: **VERIDAP Tightness.** We report the *tightness* of order-2 SII bounds after 600 s of computation, measured as normalised half-widths $\max_{\mathbf{s}}(\overline{\mathcal{I}}(\mathbf{s}) - \underline{\mathcal{I}}(\mathbf{s}))/(2|v(\mathbf{1}_d) - v(\mathbf{0}_d)|)$, where $\mathbf{s} \in \mathbb{B}^d$ with $\|\mathbf{s}\|_1 \leq 2$. Values below 0.1 % are *exact* (✓) up to a practically indiscernible tolerance. Values of at least 100 % are marked ✗.

$\lvert\mathbb{B}^d\rvert$	Tabular Dataset	Network ✳	Network ■	Network ▲
$2^{16}{>}10^{4}$	Obesity	✓	✓	0.1 %
$2^{20}{>}10^{6}$	German	✓	0.6 %	✓
$2^{22}{>}10^{6}$	Mushroom	✓	✓	*
$2^{23}{>}10^{6}$	Default	✓	✓	✓
$2^{25}{>}10^{7}$	Auto	✓	✓	✓
$2^{27}{>}10^{8}$	Steel	0.2 %	✓	✓
$2^{30}{>}10^{9}$	BreastC	6.4 %	4.4 %	3.0 %
$2^{36}{>}10^{10}$	Dropout	2.2 %	1.9 %	1.8 %
$2^{38}{>}10^{11}$	Anneal	✓	✓	✓
$2^{60}{>}10^{18}$	Sonar	✗	✗	39 %
$2^{64}{>}10^{19}$	Polish	✗	✗	53 %
$2^{79}{>}10^{23}$	NHANES I	✗	9.4 %	5.7 %
$2^{85}{>}10^{25}$	COIL2000	✗	✗	4.6 %

MNIST $\lvert\mathbb{B}^d\rvert$	Network ✳	Network ■	Network ▲
$2^{25}{>}10^{7}$	21 %	0.1 %	✓
$2^{36}{>}10^{10}$	31 %	0.2 %	✓
$2^{49}{>}10^{14}$	45 %	0.2 %	✓
$2^{64}{>}10^{19}$	49 %	0.3 %	✓
$2^{81}{>}10^{24}$	✗	✗	✓
$2^{100}{>}10^{30}$	✗	✗	✓
$2^{121}{>}10^{36}$	‡	✗	✓
$2^{144}{>}10^{43}$	‡	✗	✓
$2^{169}{>}10^{50}$	‡	‡	0.2 %
$2^{196}{>}10^{59}$	‡	‡	0.1 %
$2^{225}{>}10^{67}$	‡	‡	2.4 %
$2^{256}{>}10^{77}$	‡	‡	2.7 %
$2^{289}{>}10^{86}$	‡	‡	1.7 %

‡ Not attempted. * Pareto front only contains a single network.

6 EXPERIMENTS

In this section, we investigate *(i)* the practical capabilities of VERIDAP for computing IIs and certifying SFEs, *(ii)* the impact of certified training on the scalability of VERIDAP, and *(iii)* the performance of explanation estimators on neural networks where obtaining ground-truth exact values was previously infeasible. Questions *(i)* and *(ii)* are studied in Section 6.1, while Section 6.2 is concerned with question *(iii)*.

Datasets & Networks. We evaluate on standard tabular and vision datasets summarised in Appendix D. To manage the dimensionality of vision datasets, we group pixels into *superpixels*, using evenly spaced grids and image segmentation. On each dataset, we study networks from standard training ✳ and two networks from certified training ■▲, where ■ offers higher performance, while ▲ offers tighter bounds, as in Figure 4. Unless noted otherwise, we report median results from five repetitions at different input samples for each network and dataset. Appendix G contains more details.

6.1 SCALABILITY OF VERIDAP

To study the practical capabilities of VERIDAP and the impact of certified training, we bound the order-2 SII for a wide range of networks using the baseline value function, and certify SFEs. Appendix G provides results for additional networks, the marginal value function, further IIs, and higher-order interactions.

For space reasons, we defer a detailed discussion of certifying SFEs to Appendix G.2. We highlight that VERIDAP certifies that a PROXYSPEX explanation with only 34 of over 10^6 coalitions achieves over 95 % *guaranteed* faithfulness for the vision transformer in Figure 1, formally proving high sparsity. VERIDAP also proves that an SFE misses its faithfulness target for a space of over 10^{43} coalitions, which is relevant for discounting unreliable explanations in sensitive applications.

Tightness and Scalability. Table 1 contains our tightness results. It reveals that VERIDAP can provide tight SII bounds for standard neural networks ✳ on a tabular dataset with a search space of over 10^{11} coalitions. While this already establishes the practical value of VERIDAP, its scalability improves drastically when combined with certified training. On MNIST, sacrificing only 0.1 percentage points of accuracy ■, we compute tight SII bounds for over 10^{19} coalitions. For ▲, we are even able to tightly bound SIIs over search spaces with more than 10^{86} coalitions.

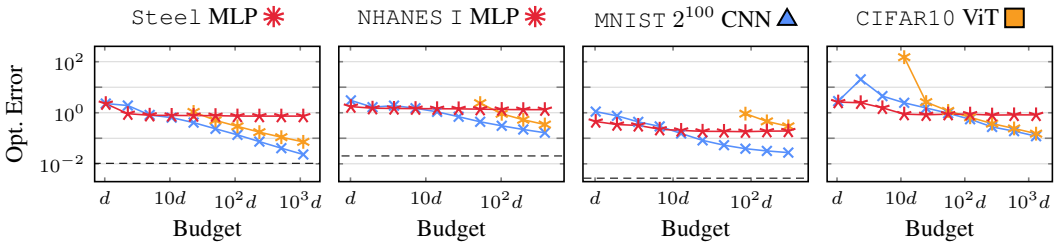


Figure 5: **Benchmarking Estimators.** Median optimistic ℓ_2 error ('Opt. Error') of KERNELSHAPIQ, PROXYSHAP, and PROXYSPEX for order-2 SII against the number of sampled coalitions ('Budget'). Below the line --- the optimistic error becomes uninformative.

Comparison to Enumeration. Table 2 compares the runtime of VERIDAP for computing exact SII to the baseline approach enumerating all coalitions, as implemented in the state-of-the-art library SHAPIQ (Muschalik et al., 2024a). VERIDAP outpaces the baseline for all search spaces over 10^6 coalitions, especially for ■. For 2^{23} coalitions, the baseline cannot provide any output within 600 s, while VERIDAP computes exact SIIs for vastly larger search spaces in Table 1.

Table 2: **Runtime Comparison.** Median runtime of the baseline enumeration approach (EXACT) and VERIDAP for computing the exact order-2 SIIs. '–' denotes not computing exact SIIs within the timeout of 600 s. Bold marks the fastest method and all runtimes faster than EXACT.

			Ours	
$\lvert\mathbb{B}^d\rvert$	**Dataset**	**EXACT**	✳	■
$2^{16}{}_{>10^4}$	`Obesity`	**5s**	7s	7s
$2^{20}{}_{>10^6}$	`German`	118s	**26s**	**7s**
$2^{22}{}_{>10^6}$	`Mushroom`	579s	**104s**	**102s**
$2^{23}{}_{>10^6}$	`Default`	–	**223s**	**9s**

Towards Transformers. Transformers (Vaswani et al., 2017) power today's most capable deep learning models, but remain challenging to deploy in sensitive applications (Liu et al., 2026). As visualised in Figure 1, VERIDAP computes the exact order-2 SII of a certifiably trained vision transformer (ViT) (Dosovitskiy et al., 2021; Brix et al., 2023) over a search space of 2^{23} coalitions within 1 440 s. Appendix G confirms this success for a second input sample. This provides an important step towards explanations with provable guarantees for safely deploying the most capable models in sensitive applications.

6.2 BENCHMARKING ESTIMATORS

Evaluating estimators of Shapley interactions on neural networks has so far been infeasible, since no ground-truth values were available for networks with more than a handful of features. We introduce the first SII benchmark for neural networks with non-trivial input dimensions, containing the 153 games (network, input, and value function) from our VERIDAP evaluation for which our bounds are tight to within 1 %. As the benchmark metric, we report the *optimistic ℓ_2 error*, a lower bound on the true ℓ_2 error computed from our bounds. Due to the small remaining gap of our bounds, this metric becomes uninformative at certain levels of error, as discussed alongside our full benchmark results in Appendix G.3. Figure 5 contains a selection of challenging instances from our benchmark. Our results provide the novel insight that tree-based estimators (Thies et al., 2026) perform well for neural networks.

7 RELATED WORK

Feature Interactions. Our framework covers every game-theoretic index of Equation (1), including cardinal-probabilistic IIs (Fujimoto et al., 2006) and weighted Banzhaf interactions (Baniecki et al., 2025), but not joint Shapley values (Harris et al., 2022) or Archipelago (Tsang et al., 2020). A complementary line of work reads interactions off network internals, through contextual decomposition (Murdoch et al., 2018; Singh et al., 2019a), Integrated Hessians (Janizek et al., 2021), and related order-2 methods (Eberle et al., 2022; Sikdar et al., 2021; Lundstrom & Razaviyayn, 2023; Pramanik et al., 2026). These measure the sensitivity of the network instead of its response to feature removal, and Hessians vanish almost everywhere in ReLU networks (Janizek et al., 2021). The metagame (Baniecki et al., 2026) bridges both by taking Shapley values of a first-order attribution.

Computing Shapley Interactions. Shapley interactions are computed exactly where the model structure limits the coalitions that matter, as for tree ensembles (Lundberg et al., 2020; Muschalik et al., 2024b) and for graph neural networks under additional assumptions (Muschalik et al., 2025). For other models, estimators use Monte Carlo sampling (Fumagalli et al., 2023), stratification (Kolpaczki et al., 2024), weighted least squares (Fumagalli et al., 2024), sparse Möbius recovery (Kang et al., 2024), or tree-based proxies (Butler et al., 2025), which become consistent with a residual correction (Thies et al., 2026), and are collected in SHAPIQ (Muschalik et al., 2024a). VERIDAP complements them with exact values for general neural networks, which we release as a benchmark for these estimators.

Neural Network Verification and Formal Explainability. Our approach builds on the rapid recent progress in neural network verification, in particular ABP and branch and bound (Zhang et al., 2018; Singh et al., 2019b; Wang et al., 2021; Zhou et al., 2024; Brix et al., 2024). Recently, formal explainability has begun to use such verifiers to compute sufficient, contrastive, and counterfactual-based explanations with provable guarantees (Ignatiev et al., 2019a;b; La Malfa et al., 2021; Wu et al., 2023; Izza et al., 2024; Bassan et al., 2025; Labbaf et al., 2025; Boumazouza et al., 2026; De Palma et al., 2026; Soria et al., 2026). These explanations are tied to adversarial robustness and therefore reduce to queries that verifiers are built to answer, whereas game-theoretic explanations aggregate the network's output over exponentially many discrete coalitions. Boetius et al. (2026) recently bridged this gap for SHAP, but only for first-order attributions of small MLPs and CNNs, mostly over search spaces of 2^{20} to 2^{30} coalitions and only rarely 2^{50} to 2^{60}. VERIDAP obtains provable guarantees for a wide family of interactions of any order and sparse Fourier explanations, all certified from a single decomposition. Combined with certified training (Gowal et al., 2018; Müller et al., 2023; De Palma et al., 2024; Kaulen et al., 2026), which we use to make networks easier to bound, it scales to 2^{289} coalitions and to vision transformers.

8 LIMITATIONS AND FUTURE WORK

Computational Hardness. Since computing Shapley values is #P-hard (Van den Broeck et al., 2022; Arenas et al., 2023; Marzouk et al., 2025), no exact method can guarantee feasible runtimes for every network. Importantly, compared with methods solving the same task, VERIDAP scales to substantially larger models and search spaces while covering a much broader family of explanations, including any-order IIs and SFEs. Moreover, when exact recovery is infeasible, VERIDAP still provides provable bounds and our training approach further improves scalability.

Neural Network Verification. Like other certification methods, VERIDAP relies on neural network verification and inherits its scalability limits. Verification has progressed rapidly in recent years (Brix et al., 2024; Wang et al., 2021; Zhou et al., 2025; 2024; Chiu et al., 2025), including for transformers (Liu et al., 2026), but remains limited for state-of-the-art models. As verification continues to advance, improved scalability and bound tightness will also expand the reach of VERIDAP.

White-Box Access. VERIDAP requires model parameters, whereas typical II and SFE estimators require only model evaluations. However, white-box access enables substantially more efficient computation and, crucially, certification without enumerating all coalitions. In many practical settings, users seek to explain models they own or control and therefore have such access; extending certification to more limited-access settings is an interesting direction for future work.

General Critiques of SHAP. The literature raises broader critiques of the axiomatic foundations and semantics of SHAP (Kumar et al., 2020; Huang & Marques-Silva, 2024; Fryer et al., 2021). We do not aim to defend these foundations, but to provide provable computation for explanations widely used in practice. Moreover, our framework covers a broad family of attribution and interaction methods with different axiomatic foundations, as well as sparse Fourier explanations.

9 CONCLUSION

We provide exact, certified game-theoretic explanations, covering a broad class of any-order interactions, including the Fourier spectrum, together with a theoretical analysis revealing what governs their hardness. Guided by this analysis, our training approach scales provable explanations to vision transformers and search spaces of up to 2^{289} coalitions. From our results, we construct the first substantial neural network benchmark for evaluating Shapley explanation estimators.

AI USE STATEMENT

In this work, we used generative AI tools for assisting in the presentation of proofs and implementing methods. We have not used generative AI tools for developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, or interpreting results. Generating synthetic datasets, assisting with translation, cleaning and reformatting datasets and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for editing our paper to improve readability. We have carefully reviewed, edited, and tested all generated code, and have carefully reviewed and edited all generated text. We take full responsibility for all content of this work, including text, claims or artefacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We only used standard, publicly available datasets in this paper. The source code of our VERIDAP implementation, as well as our estimator benchmarking code, is available in the supplementary material. The data underlying our estimator benchmark is available in anonymised form at `https://drive.google.com/file/d/1KA-fD_jPRSFR0iuwAHLZuRdYG1o_q7V9/view`. The supplementary material contains instructions on setting up and using our source code. Our code is written in Python and based on publicly obtainable open-source packages, including `JAX` (Bradbury et al., 2025), `Optuna` (Akiba et al., 2019), and `shapiq` (Muschalik et al., 2024a). We conducted our experiments on a machine with one NVIDIA L40S GPU with 48 GB of memory, an AMD EPYC 9375F CPU with 32 cores and 188 GiB of main memory. All runtime measurements reported in this paper were obtained on this machine. In total, the experiments discussed in Section 6.1 required 166 h of GPU time. We used additional hardware for training neural networks and for the estimator study of Section 6.2, but do not measure runtime on this additional hardware.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *KDD*, pp. 2623–2631. ACM, 2019. doi: 10.1145/3292500.3330701.

Marcelo Arenas, Pablo Barceló, Leopoldo Bertossi, and Mikaël Monet. On the complexity of SHAP-score-based explanations: Tractability via knowledge compilation and non-approximability results. *J. Mach. Learn. Res.*, 24:63:1–63:58, 2023. URL `https://jmlr.org/papers/v24/21-0389.html`.

Hubert Baniecki, Maximilian Muschalik, Fabian Fumagalli, Barbara Hammer, Eyke Hüllermeier, and Przemyslaw Biecek. Explaining similarity in vision-language encoders with weighted Banzhaf interactions. In *NeurIPS*, 2025.

Hubert Baniecki, Przemyslaw Biecek, and Fabian Fumagalli. Attributions all the way down? The metagame of interpretability. *CoRR*, abs/2605.06295, 2026. doi: 10.48550/arXiv.2605.06295.

John F Banzhaf, III. Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev.*, 19: 317, 1964. URL `https://access.heinonline.com/HOL/P?h=hein.journals/rutlr19&i=157`.

Shahaf Bassan, Yizhak Yisrael Elboher, Tobias Ladner, Matthias Althoff, and Guy Katz. Explaining, fast and slow: Abstraction and refinement of provable explanations. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*, pp. 3202–3225. PMLR, 2025.

David Boetius, Stefan Leue, and Tobias Sutter. Solving probabilistic verification problems of neural networks using branch and bound. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4660–4699. PMLR, 2025. URL `https://openreview.net/forum?id=suZ1pNdKrV`.

David Boetius, Shahaf Bassan, Guy Katz, Stefan Leue, and Tobias Sutter. Verified SHAP: Provable bounds for exact Shapley values of neural networks. In *ICML*, 2026. URL `https://openreview.net/forum?id=48c42LVtLx`.

Sebastian Bordt and Ulrike von Luxburg. From Shapley values to generalized additive models and back. In *AISTATS*, volume 206 of *Proceedings of Machine Learning Research*, pp. 709–745. PMLR, 2023. URL https://proceedings.mlr.press/v206/bordt23a.html.

Ryma Boumazouza, Raya Elsaleh, Melanie Ducoffe, Shahaf Bassan, and Guy Katz. FAME: Formal abstract minimal explanation for neural networks. In *ICLR*. OpenReview.net, 2026.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2025. URL http://github.com/jax-ml/jax.

Christopher Brix, Stanley Bak, Changliu Liu, and Taylor T. Johnson. The fourth international verification of neural networks competition (VNN-COMP 2023): Summary and results. *CoRR*, abs/2312.16760, 2023. doi: 10.48550/arXiv.2312.16760.

Christopher Brix, Stanley Bak, Taylor T. Johnson, and Haoze Wu. The fifth international verification of neural networks competition (VNN-COMP 2024): Summary and results. *CoRR*, abs/2412.19985, 2024. doi: 10.48550/arXiv.2412.19985.

Landon Butler, Abhineet Agarwal, Justin Singh Kang, Yigit Efe Erginbas, Bin Yu, and Kannan Ramchandran. ProxySPEX: Inference-efficient interpretability via sparse feature interactions in LLMs. In *NeurIPS*, 2025.

Yan-Hao Chen, Fei Hua, Yuwei Jin, and Eddy Z. Zhang. BGPQ: A heap-based priority queue design for GPUs. In *ICPP*, pp. 9:1–9:10. ACM, 2021. doi: 10.1145/3472456.3472463.

Ping-yeh Chiang, Renkun Ni, Ahmed Abdelkader, Chen Zhu, Christoph Studer, and Tom Goldstein. Certified defenses for adversarial patches. In *ICLR*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HyeaSkrYPH.

Hong-Ming Chiu, Hao Chen, Huan Zhang, and Richard Y. Zhang. SDP-CROWN: Efficient bound propagation for neural network verification with tightness of semidefinite programming. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10449–10468. PMLR, 2025. URL https://openreview.net/forum?id=5liHhkgvAn.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. In *NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6fec24eac8f18ed793f5eaad3dd7977c-Abstract.html.

Alessandro De Palma, Rudy Bunel, Krishnamurthy (Dj) Dvijotham, M. Pawan Kumar, Robert Stanforth, and Alessio Lomuscio. Expressive losses for verified robustness via convex combinations. In *ICLR*. OpenReview.net, 2024. URL https://openreview.net/forum?id=mzyZ4wzKlM.

Alessandro De Palma, Greta Dolcetti, and Caterina Urban. Faster verified explanations for neural networks. In *ECOOP*, volume 372 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 3:1–3:32. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026. doi: 10.4230/LIPIcs.ECOOP.2026.3.

Xiaotie Deng and Christos H. Papadimitriou. On the complexity of cooperative solution concepts. *Math. Oper. Res.*, 19(2):257–266, 1994. doi: 10.1287/MOOR.19.2.257.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Martin E. Dyer. Approximate counting by dynamic programming. In *STOC*, pp. 693–699. ACM, 2003. doi: 10.1145/780542.780643.

Oliver Eberle, Jochen Büttner, Florian Kräutli, Klaus-Robert Müller, Matteo Valleriani, and Grégoire Montavon. Building and interpreting deep similarity models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(3):1149–1161, 2022.

Edith Elkind, Leslie Ann Goldberg, Paul W. Goldberg, and Michael J. Wooldridge. On the computational complexity of weighted voting games. *Ann. Math. Artif. Intell.*, 56(2):109–131, 2009. doi: 10.1007/S10472-009-9162-5.

Daniel Vidali Fryer, Inga Strümke, and Hien D. Nguyen. Shapley values for feature selection: The good, the bad, and the axioms. *IEEE Access*, 9:144352–144360, 2021. doi: 10.1109/ACCESS.2021.3119110.

Katsushige Fujimoto, Ivan Kojadinovic, and Jean-Luc Marichal. Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices. *Games Econ. Behav.*, 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. In *NeurIPS*, pp. 11515–11551, 2023.

Fabian Fumagalli, Maximilian Muschalik, Patrick Kolpaczki, Eyke Hüllermeier, and Barbara Hammer. KernelSHAP-IQ: Weighted least square optimization for Shapley interactions. In *ICML*, volume 235 of *Proceedings of Machine Learning Research*, pp. 14308–14342. PMLR, 2024.

Ali Gorji, Andisheh Amrollahi, and Andreas Krause. SHAP values via sparse Fourier representation. In *NeurIPS*, 2025. URL https://openreview.net/forum?id=wab4BEAUt6.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy A. Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *CoRR*, abs/1810.12715, 2018. doi: 10.48550/arXiv.1810.12715.

Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *Int. J. Game Theory*, 28(4):547–565, 1999.

Michel Grabisch, Jean-Luc Marichal, and Marc Roubens. Equivalent representations of set functions. *Math. Oper. Res.*, 25(2):157–178, 2000.

Chris Harris, Richard Pymar, and Colin Rowat. Joint Shapley values: a measure of joint feature importance. In *ICLR*. OpenReview.net, 2022. URL https://openreview.net/forum?id=vcUmUvQCloe.

John C. Harsanyi. A simplified bargaining model for the n-person cooperative game. In *Papers in Game Theory*, pp. 44–70. Springer Netherlands, 1982. doi: 10.1007/978-94-017-2527-9_3.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90.

Xuanxiang Huang and João Marques-Silva. On the failings of Shapley values for explainability. *Int. J. Approx. Reason.*, 171:109112, 2024. doi: 10.1016/j.ijar.2023.109112.

Alexey Ignatiev, Nina Narodytska, and João Marques-Silva. Abduction-based explanations for machine learning models. In *AAAI*, pp. 1511–1519. AAAI Press, 2019a. doi: 10.1609/AAAI.V33I01.33011511.

Alexey Ignatiev, Nina Narodytska, and João Marques-Silva. On relating explanations and adversarial examples. In *NeurIPS*, pp. 15857–15867, 2019b.

Yacine Izza, Xuanxiang Huang, António Morgado, Jordi Planes, Alexey Ignatiev, and João Marques-Silva. Distance-restricted explanations: Theoretical underpinnings & efficient implementation. In *KR*, pp. 475–486, 2024. doi: 10.24963/KR.2024/45.

Joseph D Janizek, Pascal Sturmfels, and Su-In Lee. Explaining explanations: Axiomatic feature interactions for deep networks. *J. Mach. Learn. Res.*, 22:104:1–104:54, 2021.

Nikola Jovanovic, Mislav Balunovic, Maximilian Baader, and Martin T. Vechev. On the paradox of certified training. *Trans. Mach. Learn. Res.*, 2022. URL https://openreview.net/forum?id=atJHLVyBi8.

Justin Singh Kang, Yigit Efe Erginbas, Landon Butler, Ramtin Pedarsani, and Kannan Ramchandran. Learning to understand: Identifying interactions via the Möbius transform. In *NeurIPS*, 2024.

Justin Singh Kang, Landon Butler, Abhineet Agarwal, Yigit Efe Erginbas, Ramtin Pedarsani, Bin Yu, and Kannan Ramchandran. SPEX: scaling feature interaction explanations for LLMs. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*, pp. 28878–28903. PMLR, 2025.

Konstantin Kaulen, Hadar Shavit, and Holger H. Hoos. Rethinking evaluation paradigms in IBP-based certified training. In *ICML*, 2026. URL https://openreview.net/forum?id=XoaS2r1ezP.

R. Baker Kearfott. *Rigorous Global Search: Continuous Problems*. Springer, 1996. doi: 10.1007/978-1-4757-2495-0.

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The UCI Machine Learning repository, 2023. URL https://archive.ics.uci.edu. Accessed 2026.

Patrick Kolpaczki, Maximilian Muschalik, Fabian Fumagalli, Barbara Hammer, and Eyke Hüllermeier. SVARM-IQ: Efficient approximation of any-order Shapley interactions through stratification. In *AISTATS*, volume 238 of *Proceedings of Machine Learning Research*, pp. 3520–3528. PMLR, 2024.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

I. Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle A. Friedler. Problems with Shapley-value-based explanations as feature importance measures. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5491–5500. PMLR, 2020. URL http://proceedings.mlr.press/v119/kumar20e.html.

Emanuele La Malfa, Rhiannon Michelmore, Agnieszka M. Zbrzezny, Nicola Paoletti, and Marta Kwiatkowska. On guaranteed optimal robust explanations for NLP models. In *IJCAI*, pp. 2658–2665, 2021. doi: 10.24963/ijcai.2021/366.

Faezeh Labbaf, Tomáš Kolárik, Martin Blicha, Grigory Fedyukovich, Michael Wand, and Natasha Sharygina. Space explanations of neural network classification. In *CAV*, volume 15933 of *Lecture Notes in Computer Science*, pp. 287–303. Springer, 2025. doi: 10.1007/978-3-031-98682-6_15.

Ailsa H. Land and Alison G. Doig. An automatic method for solving discrete programming problems. In *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*, pp. 105–132. Springer, 2010. doi: 10.1007/978-3-540-68279-0_5. First published in Econometrica 28 (1960).

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Hengjie Liu, Zhenya Zhang, and Jianjun Zhao. Precise verification of transformers through ReLU-catalyzed abstraction refinement. In *CAV*, volume 16683 of *Lecture Notes in Computer Science*, pp. 527–550. Springer, 2026. doi: 10.1007/978-3-032-32526-6_25.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NeurIPS*, pp. 4765–4774, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.*, 2(1):56–67, 2020. doi: 10.1038/s42256-019-0138-9.

Daniel Lundstrom and Meisam Razaviyayn. A unifying framework to the analysis of interaction methods using synergy functions. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23005–23032. PMLR, 2023.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Reda Marzouk, Shahaf Bassan, Guy Katz, and Colin de la Higuera. On the computational tractability of the (many) Shapley values. In *AISTATS*, volume 258 of *Proceedings of Machine Learning Research*, pp. 3691–3699. PMLR, 2025. URL https://proceedings.mlr.press/v258/marzouk25a.html.

Ramon E. Moore, R. Baker Kearfott, and Michael J. Cloud. *Introduction to Interval Analysis*. SIAM, 2009. ISBN 978-0-89871-669-6. URL https://doi.org/10.1137/1.9780898717716.

David R. Morrison, Sheldon H. Jacobson, Jason J. Sauppe, and Edward C. Sewell. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discret. Optim.*, 19: 79–102, 2016. doi: 10.1016/j.disopt.2016.01.005.

Mark Niklas Müller, Franziska Eckert, Marc Fischer, and Martin T. Vechev. Certified training: Small boxes are all you need. In *ICLR*. OpenReview.net, 2023. URL https://openreview.net/forum?id=7oFuxtJtUMH.

W. James Murdoch, Peter J. Liu, and Bin Yu. Beyond word importance: Contextual decomposition to extract interactions from LSTMs. In *ICLR*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rkRwGg-0Z.

Maximilian Muschalik, Hubert Baniecki, Fabian Fumagalli, Patrick Kolpaczki, Barbara Hammer, and Eyke Hüllermeier. shapiq: Shapley interactions for machine learning. In *NeurIPS*, 2024a.

Maximilian Muschalik, Fabian Fumagalli, Barbara Hammer, and Eyke Hüllermeier. Beyond TreeSHAP: Efficient computation of any-order Shapley interactions for tree ensembles. In *AAAI*, pp. 14388–14396. AAAI Press, 2024b. doi: 10.1609/AAAI.V38I13.29352.

Maximilian Muschalik, Fabian Fumagalli, Paolo Frazzetto, Janine Strotherm, Luca Hermes, Alessandro Sperduti, Eyke Hüllermeier, and Barbara Hammer. Exact computation of any-order Shapley interactions for graph neural networks. In *ICLR*. OpenReview.net, 2025. URL https://openreview.net/forum?id=9tKC0YM8sX.

Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

Lionel S Penrose. The elementary statistics of majority voting. *J. R. Stat. Soc.*, 109(1):53–57, 1946. doi: 10.2307/2981392.

Vishal Pramanik, Maisha Maliha, Nathaniel D. Bastian, and Sumit Kumar Jha. Hessian-enhanced token attribution (HETA): Interpreting autoregressive LLMs. In *ICLR*. OpenReview.net, 2026. URL https://openreview.net/forum?id=XsEZcigEjq.

Kislaya Prasad and Jerry S. Kelly. NP-completeness of some problems concerning voting games. *Int. J. Game Theory*, 19(1):1–9, 1990. doi: 10.1007/bf01753703.

Yuval Shapira and Dana Drachsler-Cohen. Tight robustness certification through the convex hull of ℓ_0 attacks. In *AAAI*. AAAI Press, 2026. doi: 10.1609/aaai.v40i44.41128.

Lloyd S. Shapley. *A Value for N-Person Games*. RAND Corporation, Santa Monica, CA, 1952. doi: 10.7249/P0295.

Zhouxing Shi, Yihan Wang, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Fast certified robust training with short warmup. In *NeurIPS*, pp. 18335–18349, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/988f9153ac4fd966ea302dd9ab9bae15-Abstract.html.

Sandipan Sikdar, Parantapa Bhattacharya, and Kieran Heese. Integrated directional gradients: Feature interaction attribution for neural NLP models. In *ACL/IJCNLP*, pp. 865–878, 2021.

Chandan Singh, W. James Murdoch, and Bin Yu. Hierarchical interpretations for neural network predictions. In *ICLR*. OpenReview.net, 2019a. URL https://openreview.net/forum?id=SkEqro0ctQ.

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin T. Vechev. An abstract domain for certifying neural networks. *Proc. ACM Program. Lang.*, 3(POPL):41:1–41:30, 2019b. doi: 10.1145/3290354.

Jules Soria, Zakaria Chihani, Julien Girard-Satabin, Alban Grastien, Romain Xu-Darme, and Daniela Cancila. Formal abductive latent explanations for prototype-based networks. In *AAAI*, pp. 25590–25598. AAAI Press, 2026. doi: 10.1609/AAAI.V40I30.39755.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German traffic sign recognition benchmark: A multi-class classification competition. In *IJCNN*, pp. 1453–1460. IEEE, 2011. doi: 10.1109/IJCNN.2011.6033395.

Mukund Sundararajan and Amir Najmi. The many Shapley values for model explanation. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9269–9278. PMLR, 2020. URL http://proceedings.mlr.press/v119/sundararajan20b.html.

Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The Shapley Taylor interaction index. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9259–9268. PMLR, 2020.

Santo M. A. R. Thies, Hubert Baniecki, R. Teal Witter, Eyke Hüllermeier, Maximilian Muschalik, and Fabian Fumagalli. Proxy-based approximation of Shapley and Banzhaf interactions. *CoRR*, abs/2605.22738, 2026. doi: 10.48550/arXiv.2605.22738.

Che-Ping Tsai, Chih-Kuan Yeh, and Pradeep Ravikumar. Faith-Shap: The faithful Shapley interaction index. *J. Mach. Learn. Res.*, 24:94:1–94:42, 2023.

Michael Tsang, Sirisha Rambhatla, and Yan Liu. How does this interaction affect me? Interpretable attribution for feature interactions. In *NeurIPS*, pp. 6147–6159, 2020.

Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3): 410–421, 1979. doi: 10.1137/0208032.

Guy Van den Broeck, Anton Lykov, Maximilian Schleich, and Dan Suciu. On the tractability of SHAP explanations. *J. Artif. Intell. Res.*, 74:851–886, 2022. doi: 10.1613/jair.1.13283.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J. Zico Kolter. Beta-CROWN: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. In *NeurIPS*, pp. 29909–29921, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/fac7fead96dafceaf80c1daffeae82a4-Abstract.html.

Min Wu, Haoze Wu, and Clark Barrett. VeriX: Towards verified explainability of deep neural networks. In *NeurIPS*, pp. 22247–22268, 2023.

Kaidi Xu, Huan Zhang, Shiqi Wang, Yihan Wang, Suman Jana, Xue Lin, and Cho-Jui Hsieh. Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. In *ICLR*. OpenReview.net, 2021. URL https://openreview.net/forum?id=nVZtXBI6LNn.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *NeurIPS*, pp. 4944–4953, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/d04863f100d59b3eb688a11f95b0ae60-Abstract.html.

Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane S. Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. In *ICLR*. OpenReview.net, 2020. URL https://openreview.net/forum?id=Skxuk1rFwB.

Duo Zhou, Christopher Brix, Grani A. Hanasusanto, and Huan Zhang. Scalable neural network verification with branch-and-bound inferred cutting planes. In *NeurIPS*, pp. 29324–29353, 2024.

Duo Zhou, Jorge Chavez, Hesun Chen, Grani A. Hanasusanto, and Huan Zhang. Clip-and-Verify: Linear constraint-driven domain clipping for accelerating neural network verification. In *NeurIPS*, 2025. URL https://openreview.net/forum?id=HuSSR12Yot.

Appendix

Page No.

A Additional Background Material **17**
A.1 Additional Notation 17
A.2 Marginal Value Function 18
A.3 Interaction Indices 18

B Additional Details on VERIDAP **20**
B.1 Pruning and Accumulation 21
B.2 Select and Split 21

C Proofs **22**
C.1 Interaction Index Bounds 22
C.2 Sparse Fourier Explanation Bounds 27
C.3 Arbitrary Precision 31
C.4 Non-Reducible Hardness 32

D Networks and Datasets **37**

E Additional Details on Certified Training **39**

F Additional Details on Our Experimental Setup **41**

G Additional Experimental Results **41**
G.1 Interaction Indices 41
G.2 Certifying Sparse Fourier Explanations 42
G.3 Benchmarking Estimators 43
G.4 Additional Visualisations 44

A ADDITIONAL BACKGROUND MATERIAL

This appendix collects the background material on which the results of this paper and the proofs of Appendix C build. Appendix A.1 introduces remaining notation and provides a symbol reference for the entire paper. Appendix A.2 defines the marginal value function and bounds it through the affine bounds of Section 2.2. Appendix A.3 provides more details on IIs according to Equation (1).

A.1 ADDITIONAL NOTATION

For $\mathbf{s} \in \mathbb{B}^d$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, define $\mathrm{mix}_{\mathbf{x},\mathbf{y}}(\mathbf{s}) := \mathbf{s} \odot \mathbf{x} + (\mathbf{1}_d - \mathbf{s}) \odot \mathbf{y}$. For $\mathbf{x} \in \mathbb{R}^d$, we define its positive and negative parts element-wise by $[\mathbf{x}]_i^+ := \max(0, \mathbf{x}_i)$ and $[\mathbf{x}]_i^- := \min(0, \mathbf{x}_i)$, respectively, for every $i \in [d]$. In particular, $\mathbf{x} = [\mathbf{x}]^+ + [\mathbf{x}]^-$, with $[\mathbf{x}]^+ \geq \mathbf{0}_d$ and $[\mathbf{x}]^- \leq \mathbf{0}_d$. The

Bernoulli numbers B_r, $r \in \mathbb{N}_0$, are defined recursively by

$$B_0 := 1, \qquad B_r := -\frac{1}{r+1}\sum_{j=0}^{r-1}\binom{r+1}{j}B_j \quad \text{for } r \geq 1.$$

This convention gives $B_1 = -1/2$. For $x, y \in \mathbb{R}_{>0}$, we define the *Beta function* by

$$\mathrm{Beta}(x, y) := \int_0^1 t^{x-1}(1-t)^{y-1}\,\mathrm{d}t.$$

For positive integers x, y, the Beta function satisfies $\mathrm{Beta}(x, y) = (x-1)!(y-1)!/(x+y-1)!$.

Symbol Reference. Table 3 collects the notation of this paper. An underline denotes a guaranteed lower bound and an overline a guaranteed upper bound on the symbol carrying it, so that $\underline{z} \leq z \leq \overline{z}$. These variants are omitted from Table 3 for legibility.

A.2 Marginal Value Function

The baseline value function of Section 2.1 removes features by replacing them with a single baseline $\mathbf{y}$ (Sundararajan & Najmi, 2020). The *marginal value function* (Lundberg & Lee, 2017) instead removes features by averaging this replacement over several baselines. We define, for a background distribution $\mathcal{D}$ on $\mathbb{R}^d$,

$$v_{\mathcal{D}}(\mathbf{s}) := \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}\big[f\big(\mathbf{s}\odot\mathbf{x} + (\mathbf{1}_d - \mathbf{s})\odot\mathbf{y}\big)\big] = \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}\big[v_{\mathbf{y}}(\mathbf{s})\big], \qquad \mathbf{s}\in\mathbb{B}^d, \tag{4}$$

where $\mathbb{E}_{\mathbf{y}\sim\mathcal{D}}$ denotes the expectation over a baseline $\mathbf{y}$ drawn from $\mathcal{D}$, and $v_{\mathbf{y}}(\mathbf{s}) := f(\mathbf{s}\odot\mathbf{x} + (\mathbf{1}_d - \mathbf{s})\odot\mathbf{y})$ denotes the baseline value function with baseline $\mathbf{y}$. In practice, $\mathcal{D}$ is the empirical distribution of a background dataset of $|\mathcal{D}| \in \mathbb{N}$ baselines $\mathbf{y}_1, \ldots, \mathbf{y}_{|\mathcal{D}|}$, so that the expectation is the average $\frac{1}{|\mathcal{D}|}\sum_{k\in[|\mathcal{D}|]} v_{\mathbf{y}_k}(\mathbf{s})$.

We now bound the marginal value function of Equation (4) through the affine bounds of Section 2.2. The key idea is that the marginal value function is the baseline value function with an additional averaging over the baselines, and that this averaging is a linear operation. Specifically, Equation (3) applies with every baseline $\mathbf{y}$ and provides $\underline{\mathbf{w}}_{\mathbf{y}}, \overline{\mathbf{w}}_{\mathbf{y}} \in \mathbb{R}^d$ and $\underline{b}_{\mathbf{y}}, \overline{b}_{\mathbf{y}} \in \mathbb{R}$. Writing $\underline{v_{\mathbf{y}}}(\mathbf{s}) := \underline{\mathbf{w}}_{\mathbf{y}}^{\mathsf{T}}\mathbf{s} + \underline{b}_{\mathbf{y}}$ and $\overline{v_{\mathbf{y}}}(\mathbf{s}) := \overline{\mathbf{w}}_{\mathbf{y}}^{\mathsf{T}}\mathbf{s} + \overline{b}_{\mathbf{y}}$, we obtain $\underline{v_{\mathbf{y}}}(\mathbf{s}) \leq v_{\mathbf{y}}(\mathbf{s}) \leq \overline{v_{\mathbf{y}}}(\mathbf{s})$ for every $\mathbf{s} \in [\underline{\mathbf{s}}, \overline{\mathbf{s}}]$. Since each of these bounds holds pointwise for every coalition in the box, and averaging preserves both pointwise inequalities and the affine form, we obtain

$$\underline{v_{\mathcal{D}}}(\mathbf{s}) := \underline{\mathbf{w}}_{\mathcal{D}}^{\mathsf{T}}\mathbf{s} + \underline{b}_{\mathcal{D}} \leq v_{\mathcal{D}}(\mathbf{s}) \leq \overline{\mathbf{w}}_{\mathcal{D}}^{\mathsf{T}}\mathbf{s} + \overline{b}_{\mathcal{D}} =: \overline{v_{\mathcal{D}}}(\mathbf{s}), \qquad \mathbf{s}\in[\underline{\mathbf{s}}, \overline{\mathbf{s}}], \tag{5}$$

where $\underline{\mathbf{w}}_{\mathcal{D}} := \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}[\underline{\mathbf{w}}_{\mathbf{y}}]$, $\underline{b}_{\mathcal{D}} := \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}[\underline{b}_{\mathbf{y}}]$, $\overline{\mathbf{w}}_{\mathcal{D}} := \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}[\overline{\mathbf{w}}_{\mathbf{y}}]$, and $\overline{b}_{\mathcal{D}} := \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}[\overline{b}_{\mathbf{y}}]$ are the averages of the per-baseline coefficients. Hence, the averaged bounds are the averages of the per-baseline bounds, $\underline{v_{\mathcal{D}}}(\mathbf{s}) = \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}[\underline{v_{\mathbf{y}}}(\mathbf{s})]$ and $\overline{v_{\mathcal{D}}}(\mathbf{s}) = \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}[\overline{v_{\mathbf{y}}}(\mathbf{s})]$.

Equation (5) has exactly the affine form that Equation (3) prescribes. Therefore, every construction that we build on Equation (3) transfers from the baseline to the marginal value function without modification. In particular, VERIDAP computes DAPs (Definition 3.1) for the marginal value function, and Theorems 3.2 and 3.3 bound IIs and certify SFEs for the marginal value function. Furthermore, the averaged bounds inherit the exactness of the per-baseline bounds, since averaging preserves equality as well, so that they are exact for degenerate boxes and affine networks (Section 2.2).

A.3 Interaction Indices

Table 4 lists the kernels of the interaction indices that we support, and Lemma C.1 gives their positive and negative kernel masses, with closed forms for most IIs in Proposition C.3. Additionally, we define the *Möbius transform* (Harsanyi, 1982; Grabisch et al., 2000), which keeps only the empty context, $\lambda^{\mathrm{MT}}_{\mathbf{s}}(t) = \mathbb{1}[t=0]$ in Equation (1), so that $v(\mathbf{s}) = \sum_{\mathbf{t}\leq\mathbf{s}}\mu(\mathbf{t})$, where $\mu(\mathbf{s}) := \Delta v(\mathbf{s};\,\mathbf{0}_d)$.

Proposition A.1 (Kernel of k-SII)**.** *Let $\mathbf{s}\in\mathbb{B}^d$ satisfy $\|\mathbf{s}\|_1 \leq k$, and let B_r and $\mathrm{Beta}(x,y)$ be the Bernoulli numbers and the Beta function defined in Appendix A.1. The kernel of k-SII is*

$$\lambda^{k\text{-SII}}(t) := \sum_{\substack{r_1,r_2\geq 0\\ r_1+r_2\leq k-\|\mathbf{s}\|_1}} (-1)^{r_2} B_{r_1+r_2}\tbinom{t}{r_1}\tbinom{d-\|\mathbf{s}\|_1-t}{r_2}\,\mathrm{Beta}(t-r_1+1,\, d-\|\mathbf{s}\|_1-t-r_2+1). \tag{6}$$

Table 3: **Notation Table.**

Sets and vectors	
$\mathbb{B}^d$	Boolean cube $\{0,1\}^d$
$[d]$	$\{1,\dots,d\}$
$\mathbf{0}_d$, $\mathbf{1}_d$	All-zero, all-one vector
$\odot$	Element-wise product
$\|\mathbf{s}\|_1$	Coalition size
$\|g\|_2^2$	Mean square of g
$[\mathbf{x}]^+$, $[\mathbf{x}]^-$	Element-wise positive, negative part
$\mathbb{1}[P]$	Indicator, 1 if the proposition P holds and 0 otherwise
Indices and counters	
d, i	Features (players)
i^*	Feature selected for splitting
J, j	Pieces of a decomposition
n	Iteration of VERIDAP
b	Batch size
k	Order of an index
t	Kernel argument
K, k	Number and index of SFE coalitions
r	Free players inside $\mathbf{s}$
Coalitions	
$\mathbf{s}$, $\mathbf{t}$, $\mathbf{u}$	Coalitions in $\mathbb{B}^d$
S	Feature subset of $\mathbf{s}$, $\mathbf{s} \cong S$
$\mathbf{t} \leq \mathbf{s}$	$S_\mathbf{t} \subseteq S_\mathbf{s}$
$\mathbf{s} \vee \mathbf{t}$	$S_\mathbf{s} \cup S_\mathbf{t}$
$\mathbf{s} \setminus \mathbf{t}$	$S_\mathbf{s} \setminus S_\mathbf{t}$
Network and value function	
f	Network $\mathbb{R}^d \to \mathbb{R}$
$\mathbf{x}$	Network input
$\mathbf{y}$	Baseline
$\mathrm{mix}_{\mathbf{x},\mathbf{y}}(\mathbf{s})$	Element-wise interpolation between $\mathbf{y}$ and $\mathbf{x}$ by $\mathbf{s}$
v	Value function $\mathbb{B}^d \to \mathbb{R}$
$\mathcal{D}$, $v_\mathcal{D}$	Background distribution, marginal value function
g	Pseudo-Boolean function
$\mathbf{w}$, b	Coefficients of an additive g, and of the affine bounds on v
$\Delta g(\mathbf{s};\, \mathbf{t})$	Discrete derivative
$\nabla_\mathbf{s} v$	Gradient of v, extended to $[\mathbf{0}_d, \mathbf{1}_d]$, w.r.t. $\mathbf{s}$

Decompositions into additive pieces	
$\mathsf{A}^{(n)}$	Decomposition (DAP)
$\mathcal{A}_j$	Piece $[\underline{\mathbf{a}}, \overline{\mathbf{a}}]$ of A
$\mathbf{a}$	Corners of a piece
$\mathcal{A}'$, $\mathcal{A}''$	Pieces from a split
$\mathbf{f}$	Free players $\overline{\mathbf{a}} - \underline{\mathbf{a}}$
$\mathbf{i}$	Included players outside $\mathbf{s}$
$\mathbf{e}$	Excluded players outside $\mathbf{s}$
σ	Sign fixed by $\mathcal{A}$ for $\mathbf{s}$
ε	DAP tolerance
p_j	Piece priority
B, C	Selected and resulting pieces of a split
$\mathsf{O}^{(n)}$	Open pieces, those still available for splitting
$\mathsf{D}^{(n)}$	Pieces pruned up to iteration n
$\underline{\eta}_\mathcal{A}$, $\overline{\eta}_\mathcal{A}$	Piece terms of the target
$\underline{\nu}$, $\overline{\nu}$	Accumulated piece terms over the pruned pieces
$\underline{v}$, $\overline{v}$	Affine bounds on v on a piece
$\ell_\mathcal{A}$, $u_\mathcal{A}$	Extremal values of the affine bounds on $\mathcal{A}$
Interaction indices	
$\mathcal{I}$	Interaction index
λ	II kernel
φ_i	Shapley value of i
μ	Möbius transform of v
$\Lambda^+_{\mathcal{A},\mathbf{s}}$, $\Lambda^-_{\mathcal{A},\mathbf{s}}$	Kernel masses of $\mathbf{s}$ on $\mathcal{A}$
$\underline{\Delta}_{\mathcal{A},\mathbf{s}}$, $\overline{\Delta}_{\mathcal{A},\mathbf{s}}$	Discrete derivative bounds of $\mathbf{s}$ on $\mathcal{A}$
$\omega_v(\mathbf{t})$	Width of the affine bounds at $\mathbf{t}$
$\overline{\omega}_v$	Worst-case width on $\mathcal{A}$
ϱ	Radius of the derivative bounds
Fourier and sparse Fourier explanations	
$\widehat{v}$	Fourier transform of v
$\|v\|_2^2$	Mean square of v, $\sum_\mathbf{s} \widehat{v}(\mathbf{s})^2$
$\chi_\mathbf{t}$	Fourier character
m_v	Mean of v, $\widehat{v}(\mathbf{0}_d)$
$\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(K)}$	Coalitions of an SFE
$\widetilde{v}_k$	Estimated Fourier coefficient
h	Surrogate of v
$\mathcal{Q}_1$, $\mathcal{Q}_2$	Indices with $\chi_{\mathbf{s}^{(k)}}$ fixed, varying on $\mathcal{A}$
$r_\mathcal{A}$, $\underline{r}_\mathcal{A}$, $\overline{r}_\mathcal{A}$	Residual $h - v$ on $\mathcal{A}$, residual bounds
E	SFE residual $\|h - v\|_2^2$
V	Variance $\|v - \widehat{v}(\mathbf{0}_d)\|_2^2$ of the game
R^2	Faithfulness $1 - E/V$

Table 4: **Interaction Index Kernels.** For each II, the II kernel λ is as in Equation (1), where we drop the subscript $\mathbf{s}$ of $\lambda_{\mathbf{s}}$ when $\mathbf{s}$ is clear from context. The Bernoulli numbers B_r and the Beta function $\mathrm{Beta}(x, y)$ are defined in Appendix A.1. The abbreviations STII and FSII denote the Shapley–Taylor and Faithful Shapley interaction indices, respectively.

II	$\lambda(t)$	**II**	$\lambda(t)$
Möbius transform	$\mathbb{1}[t = 0]$	k-SII, $\|\mathbf{s}\|_1 \leq k$	Proposition A.1
Fourier transform	$(-1)^{\|\mathbf{s}\|_1} 2^{-d}$	STII	$\begin{cases} \frac{k}{d}\binom{d-1}{t}^{-1} & \|\mathbf{s}\|_1 = k, \\ \mathbb{1}[t=0] & \|\mathbf{s}\|_1 < k \end{cases}$
Banzhaf II	$2^{\|\mathbf{s}\|_1 - d}$		
SII	$\mathrm{Beta}(t+1, d - \|\mathbf{s}\|_1 - t + 1)$	FSII, $\|\mathbf{s}\|_1 = k$	$\frac{(2k-1)!}{((k-1)!)^2}\,\mathrm{Beta}(k+t, d-t)$

Proof. By Appendix A.2 of Bordt & von Luxburg (2023), we have

$$\mathcal{I}^{k\text{-SII}}(\mathbf{s}) = \sum_{\substack{\mathbf{t}\in\mathbb{B}^d,\ \mathbf{t}\geq\mathbf{s} \\ \|\mathbf{t}\|_1 \leq k}} B_{\|\mathbf{t}\setminus\mathbf{s}\|_1}\, \mathcal{I}^{\mathrm{SII}}(\mathbf{t}). \tag{7}$$

We next insert the linear form of each SII into Equation (7). By Equation (1), every coalition $\mathbf{t}$ satisfies

$$\mathcal{I}^{\mathrm{SII}}(\mathbf{t}) = \sum_{\mathbf{u}\in\mathbb{B}^d} \lambda^{\mathrm{SII}}(\|\mathbf{u}\setminus\mathbf{t}\|_1)(-1)^{\|\mathbf{t}\setminus\mathbf{u}\|_1} v(\mathbf{u}),$$

where, by Table 4, the SII kernel of $\mathbf{t}$ is $\lambda^{\mathrm{SII}}(t) = \mathrm{Beta}(t+1, d - \|\mathbf{t}\|_1 - t + 1)$. After exchanging the sums, the coefficient of $v(\mathbf{u})$ in $\mathcal{I}^{k\text{-SII}}(\mathbf{s})$ is

$$\sum_{\substack{\mathbf{t}\geq\mathbf{s} \\ \|\mathbf{t}\|_1 \leq k}} B_{\|\mathbf{t}\setminus\mathbf{s}\|_1} (-1)^{\|\mathbf{t}\setminus\mathbf{u}\|_1}\, \mathrm{Beta}(\|\mathbf{u}\setminus\mathbf{t}\|_1 + 1, d - \|\mathbf{t}\|_1 - \|\mathbf{u}\setminus\mathbf{t}\|_1 + 1). \tag{8}$$

Fix $\mathbf{u} \in \mathbb{B}^d$. Every coalition $\mathbf{t} \geq \mathbf{s}$ decomposes uniquely as $\mathbf{t} = \mathbf{s}\vee\mathbf{a}\vee\mathbf{b}$ with $\mathbf{a} := \mathbf{t}\odot(\mathbf{u}\setminus\mathbf{s}) \leq \mathbf{u}\setminus\mathbf{s}$ and $\mathbf{b} := \mathbf{t}\odot(\mathbf{1}_d - (\mathbf{s}\vee\mathbf{u})) \leq \mathbf{1}_d - (\mathbf{s}\vee\mathbf{u})$, where $\mathbf{a}\odot\mathbf{b} = \mathbf{0}_d$. For this decomposition,

$$\begin{aligned} \|\mathbf{t}\setminus\mathbf{s}\|_1 &= \|\mathbf{a}\|_1 + \|\mathbf{b}\|_1, & \|\mathbf{t}\|_1 &= \|\mathbf{s}\|_1 + \|\mathbf{a}\|_1 + \|\mathbf{b}\|_1, \\ \|\mathbf{u}\setminus\mathbf{t}\|_1 &= \|\mathbf{u}\setminus\mathbf{s}\|_1 - \|\mathbf{a}\|_1, & \|\mathbf{t}\setminus\mathbf{u}\|_1 &= \|\mathbf{s}\setminus\mathbf{u}\|_1 + \|\mathbf{b}\|_1, \end{aligned} \tag{9}$$

since $\mathbf{t}\setminus\mathbf{s} = \mathbf{a}\vee\mathbf{b}$ and $\mathbf{t}\setminus\mathbf{u} = (\mathbf{s}\setminus\mathbf{u})\vee\mathbf{b}$ are disjoint unions, and $\mathbf{u}\setminus\mathbf{t} = (\mathbf{u}\setminus\mathbf{s})\setminus\mathbf{a}$ with $\mathbf{a} \leq \mathbf{u}\setminus\mathbf{s}$.

We now group the summands of Equation (8) by $\|\mathbf{a}\|_1$ and $\|\mathbf{b}\|_1$, adapting the counting argument in Appendix E of Bordt & von Luxburg (2023). Writing $t := \|\mathbf{u}\setminus\mathbf{s}\|_1$, exactly $\binom{t}{r_1}$ choices of $\mathbf{a}$ satisfy $\|\mathbf{a}\|_1 = r_1$, and exactly $\binom{d-\|\mathbf{s}\|_1 - t}{r_2}$ choices of $\mathbf{b}$ satisfy $\|\mathbf{b}\|_1 = r_2$, since $\mathbf{1}_d - (\mathbf{s}\vee\mathbf{u})$ contains $d - \|\mathbf{s}\|_1 - t$ players. Moreover, the constraint $\|\mathbf{t}\|_1 \leq k$ becomes $r_1 + r_2 \leq k - \|\mathbf{s}\|_1$ by Equation (9). Applying Equation (9) to Equation (8) therefore gives the coefficient of $v(\mathbf{u})$ as

$$(-1)^{\|\mathbf{s}\setminus\mathbf{u}\|_1} \sum_{\substack{r_1, r_2 \geq 0 \\ r_1 + r_2 \leq k - \|\mathbf{s}\|_1}} (-1)^{r_2} B_{r_1+r_2} \binom{t}{r_1} \binom{d-\|\mathbf{s}\|_1 - t}{r_2}\, \mathrm{Beta}(t - r_1 + 1, d - \|\mathbf{s}\|_1 - t - r_2 + 1), \tag{10}$$

where choices with $r_1 > t$ or $r_2 > d - \|\mathbf{s}\|_1 - t$ contribute zero by the vanishing binomial coefficients. Since Equation (10) depends on $\mathbf{u}$ only through $t = \|\mathbf{u}\setminus\mathbf{s}\|_1$ and the sign $(-1)^{\|\mathbf{s}\setminus\mathbf{u}\|_1}$, it has the form of the coefficient in Equation (1) with the kernel of Equation (6). This proves Proposition A.1. □

B Additional Details on VeriDAP

This section provides additional details on VeriDAP. It presents a less abstract version of Algorithm 1 in Algorithm 2 directly computing the target bounds of Theorems 3.2 and 3.3 in a memory-efficient fashion. Furthermore, this section details the strategies for selecting features and pieces.

Algorithm 2 VERIDAP for computing the target bounds of Theorem 3.2 or Theorem 3.3.

Input: Value function $v : \mathbb{B}^d \to \mathbb{R}$, batch size $b \in \mathbb{N}$, target with per-piece contributions $\underline{\eta}_{\mathcal{A}}, \overline{\eta}_{\mathcal{A}}$.

$\mathcal{A}_1^{(1)} \leftarrow [\mathbf{0}_d, \mathbf{1}_d]$
$\mathsf{O}^{(1)} \leftarrow \{\mathcal{A}_1^{(1)}\}; \mathsf{D}^{(1)} \leftarrow \emptyset$
$(\underline{v}_1^{(1)}, \overline{v}_1^{(1)}) \leftarrow \mathrm{ABP}(v, \mathcal{A}_1^{(1)})$
$(\underline{\nu}, \overline{\nu}) \leftarrow (0, 0)$
for $n \in \{2, \ldots, 2^d\}$ **do**
  $\mathsf{D}^{(n)} \leftarrow \mathsf{D}^{(n-1)} \cup \{\mathcal{A} \in \mathsf{O}^{(n-1)} : \mathcal{A} \text{ is resolved}\}$
  $\mathsf{O}^{(n)} \leftarrow \mathsf{O}^{(n-1)} \setminus \mathsf{D}^{(n)}$
  $\underline{\nu} \leftarrow \underline{\nu} + \sum_{\mathcal{A} \in \mathsf{D}^{(n)} \setminus \mathsf{D}^{(n-1)}} \underline{\eta}_{\mathcal{A}}$
  $\overline{\nu} \leftarrow \overline{\nu} + \sum_{\mathcal{A} \in \mathsf{D}^{(n)} \setminus \mathsf{D}^{(n-1)}} \overline{\eta}_{\mathcal{A}}$
  $\mathsf{B}^{(n)} \leftarrow \textsc{Select}(\mathsf{O}^{(n)}, b)$
  $\mathsf{C}^{(n)} \leftarrow \textsc{Split}(\mathsf{B}^{(n)})$
  $\{(\underline{v}_j^{(n)}, \overline{v}_j^{(n)})\}_{j \in [|\mathsf{C}^{(n)}|]} \leftarrow \mathrm{ABP}(v, \mathsf{C}^{(n)})$
  $\mathsf{O}^{(n)} \leftarrow (\mathsf{O}^{(n)} \setminus \mathsf{B}^{(n)}) \cup \mathsf{C}^{(n)}$
  yield $(\underline{\nu} + \sum_{\mathcal{A} \in \mathsf{O}^{(n)}} \underline{\eta}_{\mathcal{A}}, \overline{\nu} + \sum_{\mathcal{A} \in \mathsf{O}^{(n)}} \overline{\eta}_{\mathcal{A}})$
end for

B.1 PRUNING AND ACCUMULATION

Both Theorems 3.2 and 3.3 bound their target as a sum of closed-form *piece terms* over the pieces, for example, as $\underline{\mathcal{I}}(\mathbf{s}) = \sum_{\mathcal{A} \in \mathsf{A}} \underline{\mathcal{I}}_{\mathcal{A}}(\mathbf{s})$ for the II of Theorem 3.2. We write $\underline{\eta}_{\mathcal{A}}, \overline{\eta}_{\mathcal{A}}$ for the lower and upper piece terms of a piece $\mathcal{A}$ under either theorem. For an II with target coalition $\mathbf{s}$, they are $\underline{\mathcal{I}}_{\mathcal{A}}(\mathbf{s}), \overline{\mathcal{I}}_{\mathcal{A}}(\mathbf{s})$. For the SFE residual, they are $2^{\|\mathbf{f}\|_1 - d} \min(-\underline{r}_{\mathcal{A}}, 0, \overline{r}_{\mathcal{A}})^2$ and $2^{\|\mathbf{f}\|_1 - d} \max(-\underline{r}_{\mathcal{A}}, \overline{r}_{\mathcal{A}})^2$.

Practically, VERIDAP does not maintain the decomposition $\mathsf{A}^{(n)}$ of Algorithm 1 in memory, but only its *open* pieces $\mathsf{O}^{(n)}$, the pieces that are still available for splitting. It tracks the contributions of the pruned pieces in *accumulators* $\underline{\nu}$, $\overline{\nu}$, initialised to 0. The pieces pruned up to iteration n are $\mathsf{D}^{(n)}$, and the decomposition of Algorithm 1 relates the three sets as $\mathsf{A}^{(n)} = \mathsf{O}^{(n)} \cup \mathsf{D}^{(n)}$, since every piece there is either still open or already pruned.

We call a piece $\mathcal{A}$ *resolved* if $\min_{\mathbf{t} \in \mathcal{A}} \underline{v}(\mathbf{t}) = \max_{\mathbf{t} \in \mathcal{A}} \overline{v}(\mathbf{t})$, so that v is constant on $\mathcal{A}$. When bounding the SFE residual, we additionally require $\underline{h}_{\mathcal{A}} = \overline{h}_{\mathcal{A}}$, so that h is constant on $\mathcal{A}$ as well. Every piece containing a single coalition is resolved, since ABP is exact for degenerate boxes (Section 2.2). The piece terms of a resolved piece are exact, that is, $\underline{\eta}_{\mathcal{A}} = \overline{\eta}_{\mathcal{A}}$, so that no split can tighten them. In contrast, exact affine bounds $\underline{v} = \overline{v}$ do not suffice, since Theorems 3.2 and 3.3 enclose some terms by the range of v over the piece.

As described in Section 3, we prune resolved pieces to manage memory. In each iteration, VERIDAP first prunes the resolved pieces. It computes their piece terms once, adds them to the accumulators, and removes the pieces from memory. SELECT then selects the batch $\mathsf{B}^{(n)}$ from the open pieces, SPLIT branches each selected piece, and ABP, which we instantiate with CROWN-IBP (Zhang et al., 2020), computes the affine bounds of the children, as in Algorithm 1.

After every iteration, the target bounds are computed from the accumulators and the piece terms over the open pieces. In our implementation, the sum over the open pieces is updated incrementally, adding the piece terms of the children and subtracting those of the selected pieces (see Appendix F for the resulting rounding error). For maintaining full batches, it can be helpful to retain a small number of resolved pieces longer than otherwise necessary.

B.2 SELECT AND SPLIT

Selecting Features to Split (SPLIT). SPLIT branches a selected piece $\mathcal{A} = [\underline{\mathbf{a}}, \overline{\mathbf{a}}]$ on one of its free features, that is, a feature i with $\underline{\mathbf{a}}_i = 0$ and $\overline{\mathbf{a}}_i = 1$. Generally, we apply the SMEARS (Kearfott, 1996; Boetius et al., 2026) strategy that selects the free feature with the largest gradient

magnitude, $i^* := \arg\max_{i \text{ free}} \overline{|\nabla_{\mathbf{s}} v|}_i$, where $\nabla_{\mathbf{s}} v := \nabla_{\mathbf{s}} v$ and $\overline{|\nabla_{\mathbf{s}} v|} \geq |\nabla_{\mathbf{s}} v|$ is computed by propagating intervals (Moore et al., 2009; Gowal et al., 2018) through the gradient of v over the piece. A feature with a large gradient magnitude affects the value function most strongly across the piece, so splitting on it promises the largest refinement. We refer to Boetius et al. (2026) for a comparison of additional SPLIT strategies.

Selecting Pieces (SELECT). Generally, VERIDAP selects the up to b open pieces for splitting in iteration $n \geq 2$ that contribute the most looseness to the target bounds we are computing, where $b \in \mathbb{N}$ is the batch size. Since SPLIT requires a free feature, SELECT only selects pieces with at least one free feature, and it selects at least one piece whenever such a piece exists. We store pieces in a batched priority queue (Chen et al., 2021) for efficiently retrieving the pieces to refine. For selecting pieces when bounding IIs, the priority is the width a piece adds to the target bounds, $p_j := \sum_{\mathbf{s} \in \mathcal{T}} \big(\overline{\mathcal{I}}_{\mathcal{A}_j}(\mathbf{s}) - \underline{\mathcal{I}}_{\mathcal{A}_j}(\mathbf{s})\big)$, where $\mathcal{T}$ is the set of target coalitions. When bounding the SFE residual, the priority of a piece $\mathcal{A}$ is $2^{-d}|\mathcal{A}|\big(\max_r r^2 - \min_r r^2\big)$ with $r \in [\underline{r}_{\mathcal{A}}, \overline{r}_{\mathcal{A}}]$.

C PROOFS

We restate all theoretical results for easier reference in this section. Table 3 summarises our notation.

C.1 INTERACTION INDEX BOUNDS

For space reasons, Section 3 states a condensed form of Theorem 3.2. We first state a more explicit form below.

Theorem 3.2 (Extended). *Let $\mathcal{I}$ be as in Equation* (1) *with kernel λ and let $\mathbf{s} \in \mathbb{B}^d$. Let $\mathsf{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_J\}$ be a DAP with affine value bounds $\underline{v}(\mathbf{t}) = \underline{\mathbf{w}}^\mathsf{T}\mathbf{t} + \underline{b}$ and $\overline{v}(\mathbf{t}) = \overline{\mathbf{w}}^\mathsf{T}\mathbf{t} + \overline{b}$ for each $\mathcal{A} \in \mathsf{A}$. For $\mathcal{A} = [\underline{\mathbf{a}}, \overline{\mathbf{a}}] \in \mathsf{A}$, define $\mathbf{f} := \overline{\mathbf{a}} - \underline{\mathbf{a}}$, $\mathbf{i} := \underline{\mathbf{a}} \setminus \mathbf{s}$, and*

$$\Lambda^+_{\mathcal{A},\mathbf{s}} := \sum_{\mathbf{t} \leq \mathbf{f} \setminus \mathbf{s}} \Big[(-1)^{\|\mathbf{s} \setminus \overline{\mathbf{a}}\|_1} \lambda(\|\mathbf{i} \vee \mathbf{t}\|_1)\Big]^+, \quad \Lambda^-_{\mathcal{A},\mathbf{s}} := \sum_{\mathbf{t} \leq \mathbf{f} \setminus \mathbf{s}} \Big[(-1)^{\|\mathbf{s} \setminus \overline{\mathbf{a}}\|_1} \lambda(\|\mathbf{i} \vee \mathbf{t}\|_1)\Big]^-$$

$$[\underline{\Delta}_{\mathcal{A},\mathbf{s}}, \overline{\Delta}_{\mathcal{A},\mathbf{s}}] := \begin{cases} [\underline{v}(\underline{\mathbf{a}}) + \mathbf{f}^\mathsf{T}[\underline{\mathbf{w}}]^-,\ \overline{v}(\underline{\mathbf{a}}) + \mathbf{f}^\mathsf{T}[\overline{\mathbf{w}}]^+] & \text{if } \mathbf{s} \odot \mathbf{f} = \mathbf{0}_d, \\ [\underline{\mathbf{w}}_i - \overline{\omega}_v, \overline{\mathbf{w}}_i + \overline{\omega}_v] & \text{if } \mathbf{s}^\mathsf{T}\mathbf{f} = 1 \text{ with } \mathbf{s}_i = \mathbf{f}_i = 1, \\ [-\varrho, \varrho] & \text{otherwise}, \end{cases}$$

where $\omega_v(\mathbf{t}) := \overline{v}(\mathbf{t}) - \underline{v}(\mathbf{t})$ is the width of the affine bounds at $\mathbf{t}$, $\overline{\omega}_v := \max_{\mathbf{u} \leq \mathbf{f} \setminus \mathbf{s}} \omega_v(\underline{\mathbf{a}} \vee \mathbf{u}) = \overline{b} - \underline{b} + (\overline{\mathbf{w}} - \underline{\mathbf{w}})^\mathsf{T}\underline{\mathbf{a}} + (\mathbf{f} \setminus \mathbf{s})^\mathsf{T}[\overline{\mathbf{w}} - \underline{\mathbf{w}}]^+$, and $\varrho := 2^{\mathbf{s}^\mathsf{T}\mathbf{f}-1}\overline{\omega}_v + 2^{\mathbf{s}^\mathsf{T}\mathbf{f}-2}(\mathbf{f} \odot \mathbf{s})^\mathsf{T}(\overline{\mathbf{w}} - \underline{\mathbf{w}})$. Then

$$\underline{\mathcal{I}}(\mathbf{s}) := \sum_{\mathcal{A} \in \mathsf{A}} \big(\Lambda^+_{\mathcal{A},\mathbf{s}}\underline{\Delta}_{\mathcal{A},\mathbf{s}} + \Lambda^-_{\mathcal{A},\mathbf{s}}\overline{\Delta}_{\mathcal{A},\mathbf{s}}\big) \leq \mathcal{I}(\mathbf{s}) \leq \sum_{\mathcal{A} \in \mathsf{A}} \big(\Lambda^+_{\mathcal{A},\mathbf{s}}\overline{\Delta}_{\mathcal{A},\mathbf{s}} + \Lambda^-_{\mathcal{A},\mathbf{s}}\underline{\Delta}_{\mathcal{A},\mathbf{s}}\big) =: \overline{\mathcal{I}}(\mathbf{s}).$$

The proof of Theorem 3.2 relies on two ingredients: the kernel masses $\Lambda^+_{\mathcal{A},\mathbf{s}}, \Lambda^-_{\mathcal{A},\mathbf{s}}$, which we derive in Lemma C.1 with closed forms for most IIs in Proposition C.3, and the discrete-derivative bounds $\underline{\Delta}_{\mathcal{A},\mathbf{s}}, \overline{\Delta}_{\mathcal{A},\mathbf{s}}$, which we prove in Lemma C.2.

Lemma C.1 (Kernel Masses). *Let $\mathbf{s} \in \mathbb{B}^d$, and let $\mathcal{A} = [\underline{\mathbf{a}}, \overline{\mathbf{a}}]$ be a piece of a DAP. We write $\mathbf{f} := \overline{\mathbf{a}} - \underline{\mathbf{a}}$ for the free players of $\mathcal{A}$, and $\mathbf{i} := \underline{\mathbf{a}} \setminus \mathbf{s}$ and $\mathbf{e} := (\mathbf{1}_d - \overline{\mathbf{a}}) \setminus \mathbf{s}$ for the players outside $\mathbf{s}$ that $\mathcal{A}$ includes and excludes, respectively. Moreover, $\sigma := (-1)^{\|\mathbf{s} \setminus \overline{\mathbf{a}}\|_1}$ is the sign fixed by $\mathcal{A}$. For an II of Table 4 with kernel λ, the kernel masses collect the positive and negative parts of the kernel factors over the contexts of $\mathcal{A}$ as*

$$\begin{aligned} \Lambda^+_{\mathcal{A},\mathbf{s}} &:= \sum_{\mathbf{t} \leq \mathbf{f} \setminus \mathbf{s}} [\sigma\, \lambda(\|\mathbf{i} \vee \mathbf{t}\|_1)]^+, \\ \Lambda^-_{\mathcal{A},\mathbf{s}} &:= \sum_{\mathbf{t} \leq \mathbf{f} \setminus \mathbf{s}} [\sigma\, \lambda(\|\mathbf{i} \vee \mathbf{t}\|_1)]^-. \end{aligned} \tag{11}$$

Then the contribution $\mathcal{I}_{\mathcal{A}}(\mathbf{s})$ of the coalitions in $\mathcal{A}$ to Equation (1) *is*

$$\mathcal{I}_{\mathcal{A}}(\mathbf{s}) = \sigma \sum_{\mathbf{t} \leq \mathbf{f} \setminus \mathbf{s}} \lambda(\|\mathbf{i} \vee \mathbf{t}\|_1)\, \Delta v(\mathbf{s} \odot \mathbf{f};\ \underline{\mathbf{a}} \vee \mathbf{t}). \tag{12}$$

Proof. We prove Equation (12). Each coalition in $\mathcal{A}$ contains the players fixed as included by $\underline{\mathbf{a}}$, together with free players outside and inside $\mathbf{s}$. Thus,

$$\mathcal{A} = \{\underline{\mathbf{a}} \vee \mathbf{t} \vee \mathbf{u} : \mathbf{t}, \mathbf{u} \in \mathbb{B}^d, \quad \mathbf{t} \le \mathbf{f} \setminus \mathbf{s}, \quad \mathbf{u} \le \mathbf{s} \odot \mathbf{f}\}.$$

Each coalition in $\mathcal{A}$ has exactly one representation, since the players in the three components belong to disjoint sets,

$$\underline{\mathbf{a}} \odot \mathbf{f} = \mathbf{0}_d, \qquad (\mathbf{f} \setminus \mathbf{s}) \odot (\mathbf{s} \odot \mathbf{f}) = \mathbf{0}_d.$$

In particular,

$$\underline{\mathbf{a}} \odot \mathbf{t} = \underline{\mathbf{a}} \odot \mathbf{u} = \mathbf{t} \odot \mathbf{u} = \mathbf{0}_d.$$

The players outside $\mathbf{s}$ and the players in $\mathbf{s}$ absent from $\underline{\mathbf{a}} \vee \mathbf{t} \vee \mathbf{u}$ satisfy

$$\begin{aligned}(\underline{\mathbf{a}} \vee \mathbf{t} \vee \mathbf{u}) \setminus \mathbf{s} &= (\underline{\mathbf{a}} \setminus \mathbf{s}) \vee \mathbf{t} = \mathbf{i} \vee \mathbf{t},\\ \mathbf{s} \setminus (\underline{\mathbf{a}} \vee \mathbf{t} \vee \mathbf{u}) &= (\mathbf{s} \setminus \overline{\mathbf{a}}) \vee ((\mathbf{s} \odot \mathbf{f}) \setminus \mathbf{u}),\\ (\mathbf{s} \setminus \overline{\mathbf{a}}) \odot ((\mathbf{s} \odot \mathbf{f}) \setminus \mathbf{u}) &= \mathbf{0}_d.\end{aligned}$$

By the disjointness in the last line,

$$\begin{aligned}(-1)^{\|\mathbf{s} \setminus (\underline{\mathbf{a}} \vee \mathbf{t} \vee \mathbf{u})\|_1} &= (-1)^{\|\mathbf{s} \setminus \overline{\mathbf{a}}\|_1 + \|(\mathbf{s} \odot \mathbf{f}) \setminus \mathbf{u}\|_1}\\ &= \sigma\,(-1)^{\|(\mathbf{s} \odot \mathbf{f}) \setminus \mathbf{u}\|_1}.\end{aligned}$$

We restrict the sum in Equation (1) to value-function evaluations at coalitions in $\mathcal{A}$. Substituting the preceding equalities gives

$$\begin{aligned}&\sum_{\mathbf{a} \in \mathcal{A}} \lambda(\|\mathbf{a} \setminus \mathbf{s}\|_1)(-1)^{\|\mathbf{s} \setminus \mathbf{a}\|_1} v(\mathbf{a})\\ &\quad= \sigma \sum_{\mathbf{t} \le \mathbf{f} \setminus \mathbf{s}} \lambda(\|\mathbf{i} \vee \mathbf{t}\|_1) \sum_{\mathbf{u} \le \mathbf{s} \odot \mathbf{f}} (-1)^{\|(\mathbf{s} \odot \mathbf{f}) \setminus \mathbf{u}\|_1} v(\underline{\mathbf{a}} \vee \mathbf{t} \vee \mathbf{u})\\ &\quad= \sigma \sum_{\mathbf{t} \le \mathbf{f} \setminus \mathbf{s}} \lambda(\|\mathbf{i} \vee \mathbf{t}\|_1) \Delta v(\mathbf{s} \odot \mathbf{f};\, \underline{\mathbf{a}} \vee \mathbf{t}).\end{aligned}$$

The last equality applies the definition of the discrete derivative, which is applicable since $(\underline{\mathbf{a}} \vee \mathbf{t}) \odot (\mathbf{s} \odot \mathbf{f}) = \mathbf{0}_d$. This is Equation (12), and the kernel masses of Equation (11) collect the positive and negative parts of its coefficients $\sigma\,\lambda(\|\mathbf{i} \vee \mathbf{t}\|_1)$. □

Lemma C.2 (Discrete Derivative Bounds)**.** *Let $\mathcal{A} = [\underline{\mathbf{a}}, \overline{\mathbf{a}}]$ be a piece of a DAP with affine value bounds $\underline{v}(\mathbf{t}) = \underline{\mathbf{w}}^{\mathsf{T}}\mathbf{t} + \underline{b}$ and $\overline{v}(\mathbf{t}) = \overline{\mathbf{w}}^{\mathsf{T}}\mathbf{t} + \overline{b}$. Define $\mathbf{f} := \overline{\mathbf{a}} - \underline{\mathbf{a}}$ and let $\mathbf{s} \in \mathbb{B}^d$. For every $\mathbf{t} \in \mathbb{B}^d$ with $\mathbf{t} \le \mathbf{f} \setminus \mathbf{s}$, it holds that*

$$\Delta v(\mathbf{s} \odot \mathbf{f};\, \underline{\mathbf{a}} \vee \mathbf{t}) \in [\underline{\Delta}_{\mathcal{A},\mathbf{s}}, \overline{\Delta}_{\mathcal{A},\mathbf{s}}] := \begin{cases}[\underline{v}(\underline{\mathbf{a}}) + \mathbf{f}^{\mathsf{T}}[\underline{\mathbf{w}}]^-,\ \overline{v}(\underline{\mathbf{a}}) + \mathbf{f}^{\mathsf{T}}[\overline{\mathbf{w}}]^+] & \text{if } \mathbf{s} \odot \mathbf{f} = \mathbf{0}_d,\\ [\underline{\mathbf{w}}_i - \overline{\omega}_v,\ \overline{\mathbf{w}}_i + \overline{\omega}_v] & \text{if } \mathbf{s}^{\mathsf{T}}\mathbf{f} = 1 \text{ with } \mathbf{s}_i = \mathbf{f}_i = 1,\\ [-\varrho,\ \varrho] & \text{otherwise},\end{cases}$$

where $\omega_v(\mathbf{t}) := \overline{v}(\mathbf{t}) - \underline{v}(\mathbf{t})$, $\overline{\omega}_v := \max_{\mathbf{u} \le \mathbf{f} \setminus \mathbf{s}} \omega_v(\underline{\mathbf{a}} \vee \mathbf{u}) = \overline{b} - \underline{b} + (\overline{\mathbf{w}} - \underline{\mathbf{w}})^{\mathsf{T}}\underline{\mathbf{a}} + (\mathbf{f} \setminus \mathbf{s})^{\mathsf{T}}[\overline{\mathbf{w}} - \underline{\mathbf{w}}]^+$, and $\varrho := 2^{\mathbf{s}^{\mathsf{T}}\mathbf{f} - 1}\overline{\omega}_v + 2^{\mathbf{s}^{\mathsf{T}}\mathbf{f} - 2}(\mathbf{f} \odot \mathbf{s})^{\mathsf{T}}(\overline{\mathbf{w}} - \underline{\mathbf{w}})$.

Proof of Lemma C.2. We recall that $\mathbf{f} := \overline{\mathbf{a}} - \underline{\mathbf{a}}$. Let $\Delta\mathbf{w} := \overline{\mathbf{w}} - \underline{\mathbf{w}}$ and $\Delta b := \overline{b} - \underline{b}$, so that $\omega_v(\mathbf{u}) = \Delta b + \mathbf{u}^{\mathsf{T}}\Delta\mathbf{w}$. Furthermore, let $\mathbf{t} \in \mathbb{B}^d$ with $\mathbf{t} \le \mathbf{f} \setminus \mathbf{s}$. We define $\mathbf{b} := \underline{\mathbf{a}} \vee \mathbf{t}$ and $\mathbf{c} := \mathbf{s} \odot \mathbf{f}$. By definition $\Delta v(\mathbf{s} \odot \mathbf{f};\, \underline{\mathbf{a}} \vee \mathbf{t}) = \Delta v(\mathbf{c};\, \mathbf{b})$ and $\mathbf{b} \odot \mathbf{c} = \mathbf{0}_d$. Since $\underline{\mathbf{a}} \odot \mathbf{t} = \mathbf{0}_d$ and $\mathbf{0}_d \le \mathbf{t} \le \mathbf{f} \setminus \mathbf{s}$,

$$\begin{aligned}\omega_v(\mathbf{b}) &= \Delta b + \underline{\mathbf{a}}^{\mathsf{T}}\Delta\mathbf{w} + \mathbf{t}^{\mathsf{T}}\Delta\mathbf{w}\\ &\le \Delta b + \underline{\mathbf{a}}^{\mathsf{T}}\Delta\mathbf{w} + (\mathbf{f} \setminus \mathbf{s})^{\mathsf{T}}[\Delta\mathbf{w}]^+ = \overline{\omega}_v.\end{aligned} \tag{13}$$

We distinguish three cases based on $r := \|\mathbf{c}\|_1 = \mathbf{s}^{\mathsf{T}}\mathbf{f}$.

- Suppose that $r = 0$. This implies $\mathbf{c} = \mathbf{0}_d$. By the definition of the discrete derivative, $\Delta v(\mathbf{c};\, \mathbf{b}) = \Delta v(\mathbf{0}_d;\, \mathbf{b}) = v(\mathbf{b})$. Since $\mathbf{t} \le \mathbf{f} \setminus \mathbf{s} \le \mathbf{f}$, we have $\mathbf{t} \le \mathbf{f}$. Therefore,

$$\begin{aligned}\Delta v(\mathbf{c};\, \mathbf{b}) = v(\mathbf{b}) &\ge \underline{v}(\underline{\mathbf{a}}) + \underline{\mathbf{w}}^{\mathsf{T}}\mathbf{t} \ge \underline{v}(\underline{\mathbf{a}}) + \mathbf{f}^{\mathsf{T}}[\underline{\mathbf{w}}]^-,\\ \Delta v(\mathbf{c};\, \mathbf{b}) = v(\mathbf{b}) &\le \overline{v}(\underline{\mathbf{a}}) + \overline{\mathbf{w}}^{\mathsf{T}}\mathbf{t} \le \overline{v}(\underline{\mathbf{a}}) + \mathbf{f}^{\mathsf{T}}[\overline{\mathbf{w}}]^+.\end{aligned}$$

- Suppose that $r = 1$. This implies that there exists exactly one $i \in [d]$ with $\mathbf{s}_i = \mathbf{f}_i = 1$. Let $i \in [d]$ be that index. The definition of the discrete derivative and the affine bounds yield

$$\begin{aligned}\Delta v(\mathbf{c};\, \mathbf{b}) &= v(\mathbf{b} \vee \mathbf{c}) - v(\mathbf{b}) \\ &\geq \underline{v}(\mathbf{b} \vee \mathbf{c}) - \overline{v}(\mathbf{b}) = \mathbf{c}^\mathsf{T}\underline{\mathbf{w}} + \underline{v}(\mathbf{b}) - \overline{v}(\mathbf{b}) = \underline{\mathbf{w}}_i - \omega_v(\mathbf{b}) \geq \underline{\mathbf{w}}_i - \overline{\omega}_v, \\ \Delta v(\mathbf{c};\, \mathbf{b}) &\leq \overline{v}(\mathbf{b} \vee \mathbf{c}) - \underline{v}(\mathbf{b}) = \mathbf{c}^\mathsf{T}\overline{\mathbf{w}} + \overline{v}(\mathbf{b}) - \underline{v}(\mathbf{b}) = \overline{\mathbf{w}}_i + \omega_v(\mathbf{b}) \leq \overline{\mathbf{w}}_i + \overline{\omega}_v,\end{aligned}$$

  where the final inequalities apply Equation (13).

- Finally, suppose that $r \geq 2$. For every coalition vector $\mathbf{u} \leq \mathbf{c}$, define the lower-bound slack $\zeta(\mathbf{u}) := v(\mathbf{b} \vee \mathbf{u}) - \underline{v}(\mathbf{b} \vee \mathbf{u})$. This gives us

$$\begin{aligned}\Delta v(\mathbf{c};\, \mathbf{b}) &= \sum_{\mathbf{u} \leq \mathbf{c}} (-1)^{\|\mathbf{c} \setminus \mathbf{u}\|_1} (\underline{v}(\mathbf{u} \vee \mathbf{b}) + \zeta(\mathbf{u})) \\ &= \Delta \underline{v}(\mathbf{c};\, \mathbf{b}) + \Delta\zeta(\mathbf{c};\, \mathbf{0}_d).\end{aligned}$$

  Since $\underline{v}$ is affine, the discrete derivative of $\underline{v}$ vanishes:

$$\begin{aligned}\Delta \underline{v}(\mathbf{c};\, \mathbf{b}) &= \sum_{\mathbf{u} \leq \mathbf{c}} (-1)^{\|\mathbf{c} \setminus \mathbf{u}\|_1} (\underline{b} + \underline{\mathbf{w}}^\mathsf{T}\mathbf{b} + \underline{\mathbf{w}}^\mathsf{T}\mathbf{u}) \\ &= \underline{v}(\mathbf{b}) \sum_{\mathbf{u} \leq \mathbf{c}} (-1)^{\|\mathbf{c} \setminus \mathbf{u}\|_1} + \sum_{\substack{i \in [d] \\ \mathbf{c}_i = 1}} \underline{\mathbf{w}}_i \sum_{\substack{\mathbf{u} \leq \mathbf{c} \\ \mathbf{u}_i = 1}} (-1)^{\|\mathbf{c} \setminus \mathbf{u}\|_1} \\ &= \underline{v}(\mathbf{b})(1-1)^r + \sum_{\substack{i \in [d] \\ \mathbf{c}_i = 1}} \underline{\mathbf{w}}_i (1-1)^{r-1} = 0,\end{aligned} \tag{14}$$

  where the last line follows from the binomial theorem. For the slack derivative, define

$$\begin{aligned}\mathcal{U}_+ &:= \{\mathbf{u} \in \mathbb{B}^d : \mathbf{u} \leq \mathbf{c}, \|\mathbf{c} \setminus \mathbf{u}\|_1 \text{ is even}\} = \{\mathbf{u} \in \mathbb{B}^d : \mathbf{u} \leq \mathbf{c}, (-1)^{\|\mathbf{c} \setminus \mathbf{u}\|_1} = 1\} \\ \mathcal{U}_- &:= \{\mathbf{u} \in \mathbb{B}^d : \mathbf{u} \leq \mathbf{c}, \|\mathbf{c} \setminus \mathbf{u}\|_1 \text{ is odd}\} = \{\mathbf{u} \in \mathbb{B}^d : \mathbf{u} \leq \mathbf{c}, (-1)^{\|\mathbf{c} \setminus \mathbf{u}\|_1} = -1\}.\end{aligned}$$

  Using Equation (14) and that $0 \leq \zeta(\mathbf{u}) \leq \omega_v(\mathbf{b} \vee \mathbf{u})$, since $\underline{v}(\mathbf{b} \vee \mathbf{u}) \leq v(\mathbf{b} \vee \mathbf{u}) \leq \overline{v}(\mathbf{b} \vee \mathbf{u})$, and grouping $\mathbf{u} \leq \mathbf{c}$ by $\mathcal{U}_+$ and $\mathcal{U}_-$ in $\Delta\zeta(\mathbf{c};\, \mathbf{0}_d)$, we obtain

$$\begin{aligned}\Delta v(\mathbf{c};\, \mathbf{b}) &= \underbrace{\Delta \underline{v}(\mathbf{c};\, \mathbf{b})}_{=0} + \Delta\zeta(\mathbf{c};\, \mathbf{0}_d) \\ &= \sum_{\mathbf{u} \in \mathcal{U}_+} \zeta(\mathbf{u}) - \sum_{\mathbf{u} \in \mathcal{U}_-} \zeta(\mathbf{u}) \leq \sum_{\mathbf{u} \in \mathcal{U}_+} \omega_v(\mathbf{b} \vee \mathbf{u}) \\ \Delta v(\mathbf{c};\, \mathbf{b}) &\geq -\sum_{\mathbf{u} \in \mathcal{U}_-} \omega_v(\mathbf{b} \vee \mathbf{u}).\end{aligned} \tag{15}$$

  Since there is an equal number of $\mathbf{u} \leq \mathbf{c}$ with even and odd $\|\mathbf{c} \setminus \mathbf{u}\|_1$, each of $\mathcal{U}_+$ and $\mathcal{U}_-$ contains 2^{r-1} coalitions. Furthermore, for each $i \in [d]$ with $\mathbf{c}_i = 1$, there are 2^{r-2} coalitions $\mathbf{u}$ in both $\mathcal{U}_+$ and $\mathcal{U}_-$ with $\mathbf{u}_i = 1$. Consequently,

$$\begin{aligned}\sum_{\mathbf{u} \in \mathcal{U}_+} \omega_v(\mathbf{u} \vee \mathbf{b}) &= \sum_{\mathbf{u} \in \mathcal{U}_+} (\Delta b + \mathbf{b}^\mathsf{T}\Delta\mathbf{w} + \mathbf{u}^\mathsf{T}\Delta\mathbf{w}) \\ &= \sum_{\mathbf{u} \in \mathcal{U}_+} \omega_v(\mathbf{b}) + \sum_{\substack{i \in [d] \\ \mathbf{c}_i = 1}} \sum_{\substack{\mathbf{u} \in \mathcal{U}_+ \\ \mathbf{u}_i = 1}} \Delta\mathbf{w}_i \\ &= 2^{r-1}\omega_v(\mathbf{b}) + 2^{r-2}\mathbf{c}^\mathsf{T}\Delta\mathbf{w} \\ &\leq 2^{r-1}\overline{\omega}_v + 2^{r-2}(\mathbf{f} \odot \mathbf{s})^\mathsf{T}\Delta\mathbf{w} = \varrho,\end{aligned} \tag{16}$$

  where the inequality follows from Equation (13). Following equivalent steps, we also obtain $-\sum_{\mathbf{u} \in \mathcal{U}_-} \omega_v(\mathbf{u} \vee \mathbf{b}) \geq -\varrho$. Together, Equations (15) and (16) prove

$$\underline{\Delta}_{\mathcal{A},\mathbf{s}} = -\varrho \leq \Delta v(\mathbf{c};\, \mathbf{b}) \leq \varrho = \overline{\Delta}_{\mathcal{A},\mathbf{s}}.$$

Since the three cases exhaust all values of r, this proves Lemma C.2. □

Proof of Theorem 3.2. We first rewrite the interaction index as one sum over Boolean coalition vectors. Expanding the discrete derivative in Equation (1) gives

$$\begin{aligned}\mathcal{I}(\mathbf{s}) &= \sum_{\mathbf{t}\leq \mathbf{1}_d-\mathbf{s}} \lambda(\|\mathbf{t}\|_1) \sum_{\mathbf{u}\leq \mathbf{s}} (-1)^{\|\mathbf{s}\setminus\mathbf{u}\|_1} v(\mathbf{t}\vee\mathbf{u}) \\ &= \sum_{\mathbf{a}\in\mathbb{B}^d} \lambda(\|\mathbf{a}\setminus\mathbf{s}\|_1)(-1)^{\|\mathbf{s}\setminus\mathbf{a}\|_1} v(\mathbf{a}) \\ &= \sum_{\mathcal{A}\in\mathsf{A}} \sum_{\mathbf{a}\in\mathcal{A}} \lambda(\|\mathbf{a}\setminus\mathbf{s}\|_1)(-1)^{\|\mathbf{s}\setminus\mathbf{a}\|_1} v(\mathbf{a}),\end{aligned} \tag{17}$$

where the second equality uses the unique decomposition of every $\mathbf{a}\in\mathbb{B}^d$ into $\mathbf{t}=\mathbf{a}\setminus\mathbf{s}$ and $\mathbf{u}=\mathbf{a}\odot\mathbf{s}$, and the third equality uses that A partitions $\mathbb{B}^d$.

Fix $\mathcal{A}=[\underline{\mathbf{a}},\overline{\mathbf{a}}]\in\mathsf{A}$, and let $\mathbf{f}:=\overline{\mathbf{a}}-\underline{\mathbf{a}}$, $\mathbf{i}:=\underline{\mathbf{a}}\setminus\mathbf{s}$, and $\mathbf{c}:=\mathbf{f}\odot\mathbf{s}$. Every coalition $\mathbf{a}\in\mathcal{A}$ has a unique representation $\mathbf{a}=\underline{\mathbf{a}}\vee\mathbf{t}\vee\mathbf{u}$, where $\mathbf{t}=\mathbf{a}\odot(\mathbf{f}\setminus\mathbf{s})$ and $\mathbf{u}=\mathbf{a}\odot\mathbf{c}$. Using that $\mathbf{t}\odot\mathbf{s}=\mathbf{0}_d$ for this representation, we obtain

$$\begin{aligned}&\|(\underline{\mathbf{a}}\vee\mathbf{t}\vee\mathbf{u})\setminus\mathbf{s}\|_1 = \|\mathbf{i}\vee\mathbf{t}\|_1, \\ &(-1)^{\|\mathbf{s}\setminus(\underline{\mathbf{a}}\vee\mathbf{t}\vee\mathbf{u})\|_1} = (-1)^{\|(\mathbf{s}\setminus\overline{\mathbf{a}})\vee(\mathbf{c}\setminus\mathbf{u})\|_1} = (-1)^{\|\mathbf{s}\setminus\overline{\mathbf{a}}\|_1}(-1)^{\|\mathbf{c}\setminus\mathbf{u}\|_1}.\end{aligned} \tag{18}$$

Applying Equation (18) to Equation (17) yields

$$\begin{aligned}\mathcal{I}(\mathbf{s}) &= \sum_{\mathcal{A}\in\mathsf{A}} \sum_{\mathbf{t}\leq\mathbf{f}\setminus\mathbf{s}} (-1)^{\|\mathbf{s}\setminus\overline{\mathbf{a}}\|_1} \lambda(\|\mathbf{i}\vee\mathbf{t}\|_1) \cdot \sum_{\mathbf{u}\leq\mathbf{c}} (-1)^{\|\mathbf{c}\setminus\mathbf{u}\|_1} v(\underline{\mathbf{a}}\vee\mathbf{t}\vee\mathbf{u}) \\ &= \sum_{\mathcal{A}\in\mathsf{A}} \sum_{\mathbf{t}\leq\mathbf{f}\setminus\mathbf{s}} (-1)^{\|\mathbf{s}\setminus\overline{\mathbf{a}}\|_1} \lambda(\|\mathbf{i}\vee\mathbf{t}\|_1) \Delta v(\mathbf{s}\odot\mathbf{f};\, \underline{\mathbf{a}}\vee\mathbf{t}).\end{aligned} \tag{19}$$

For brevity, we define

$$\alpha_{\mathcal{A},\mathbf{s}}(\mathbf{t}) := (-1)^{\|\mathbf{s}\setminus\overline{\mathbf{a}}\|_1} \lambda(\|\mathbf{i}\vee\mathbf{t}\|_1).$$

By Lemma C.2,

$$\begin{aligned}\alpha_{\mathcal{A},\mathbf{s}}(\mathbf{t})\Delta v(\mathbf{s}\odot\mathbf{f};\, \underline{\mathbf{a}}\vee\mathbf{t}) &\geq [\alpha_{\mathcal{A},\mathbf{s}}(\mathbf{t})]^+ \underline{\Delta}_{\mathcal{A},\mathbf{s}} + [\alpha_{\mathcal{A},\mathbf{s}}(\mathbf{t})]^- \overline{\Delta}_{\mathcal{A},\mathbf{s}} \\ \alpha_{\mathcal{A},\mathbf{s}}(\mathbf{t})\Delta v(\mathbf{s}\odot\mathbf{f};\, \underline{\mathbf{a}}\vee\mathbf{t}) &\leq [\alpha_{\mathcal{A},\mathbf{s}}(\mathbf{t})]^+ \overline{\Delta}_{\mathcal{A},\mathbf{s}} + [\alpha_{\mathcal{A},\mathbf{s}}(\mathbf{t})]^- \underline{\Delta}_{\mathcal{A},\mathbf{s}}.\end{aligned} \tag{20}$$

By Lemma C.1, the outer sums $\sum_{\mathbf{t}\leq\mathbf{f}\setminus\mathbf{s}} [\alpha_{\mathcal{A},\mathbf{s}}(\mathbf{t})]^+$ and $\sum_{\mathbf{t}\leq\mathbf{f}\setminus\mathbf{s}} [\alpha_{\mathcal{A},\mathbf{s}}(\mathbf{t})]^-$ are the kernel masses $\Lambda^+_{\mathcal{A},\mathbf{s}}$ and $\Lambda^-_{\mathcal{A},\mathbf{s}}$. By inserting Equation (20) into Equation (19), we obtain

$$\begin{aligned}\mathcal{I}(\mathbf{s}) &= \sum_{\mathcal{A}\in\mathsf{A}} \sum_{\mathbf{t}\leq\mathbf{f}\setminus\mathbf{s}} \alpha_{\mathcal{A},\mathbf{s}}(\mathbf{t})\Delta v(\mathbf{s}\odot\mathbf{f};\, \underline{\mathbf{a}}\vee\mathbf{t}) \\ &\geq \sum_{\mathcal{A}\in\mathsf{A}} \sum_{\mathbf{t}\leq\mathbf{f}\setminus\mathbf{s}} ([\alpha_{\mathcal{A},\mathbf{s}}(\mathbf{t})]^+ \underline{\Delta}_{\mathcal{A},\mathbf{s}} + [\alpha_{\mathcal{A},\mathbf{s}}(\mathbf{t})]^- \overline{\Delta}_{\mathcal{A},\mathbf{s}}) \\ &= \sum_{\mathcal{A}\in\mathsf{A}} \left(\left(\sum_{\mathbf{t}\leq\mathbf{f}\setminus\mathbf{s}} [\alpha_{\mathcal{A},\mathbf{s}}(\mathbf{t})]^+ \right) \underline{\Delta}_{\mathcal{A},\mathbf{s}} + \left(\sum_{\mathbf{t}\leq\mathbf{f}\setminus\mathbf{s}} [\alpha_{\mathcal{A},\mathbf{s}}(\mathbf{t})]^- \right) \overline{\Delta}_{\mathcal{A},\mathbf{s}} \right) \\ &= \sum_{\mathcal{A}\in\mathsf{A}} (\Lambda^+_{\mathcal{A},\mathbf{s}} \underline{\Delta}_{\mathcal{A},\mathbf{s}} + \Lambda^-_{\mathcal{A},\mathbf{s}} \overline{\Delta}_{\mathcal{A},\mathbf{s}}) = \underline{\mathcal{I}}(\mathbf{s}).\end{aligned}$$

Following equivalent steps, we also obtain

$$\mathcal{I}(\mathbf{s}) \leq \sum_{\mathcal{A}\in\mathsf{A}} (\Lambda^+_{\mathcal{A},\mathbf{s}} \overline{\Delta}_{\mathcal{A},\mathbf{s}} + \Lambda^-_{\mathcal{A},\mathbf{s}} \underline{\Delta}_{\mathcal{A},\mathbf{s}}) = \overline{\mathcal{I}}(\mathbf{s}).$$

This proves Theorem 3.2. □

Proposition C.3 gives closed forms for the kernel masses of most IIs. The only exception is k-SII: there, we compute the sums in Equation (11) with the kernel of Equation (6) (Proposition A.1).

Proposition C.3 (Closed Forms of Kernel Masses)**.** *Let $\mathbf{s} \in \mathbb{B}^d$, and let $\mathcal{A} = [\underline{\mathbf{a}}, \overline{\mathbf{a}}]$ be a piece of a DAP. We write $\mathbf{f} := \overline{\mathbf{a}} - \underline{\mathbf{a}}$ for the free players of $\mathcal{A}$, and $\mathbf{i} := \underline{\mathbf{a}} \setminus \mathbf{s}$ and $\mathbf{e} := (\mathbf{1}_d - \overline{\mathbf{a}}) \setminus \mathbf{s}$ for the players outside $\mathbf{s}$ that $\mathcal{A}$ includes and excludes, respectively. Moreover, $\sigma := (-1)^{\|\mathbf{s} \setminus \overline{\mathbf{a}}\|_1}$ is the sign fixed by $\mathcal{A}$. For an II of Table 4 with kernel λ, the kernel masses of Lemma C.1 evaluate as follows for the indices other than k-SII:*

(i) Möbius transform: $\Lambda^+_{\mathcal{A},\mathbf{s}} = [\sigma]^+ \mathbb{1}[\|\mathbf{i}\|_1 = 0]$ *and* $\Lambda^-_{\mathcal{A},\mathbf{s}} = [\sigma]^- \mathbb{1}[\|\mathbf{i}\|_1 = 0]$.

(ii) Fourier transform:

$$\Lambda^+_{\mathcal{A},\mathbf{s}} = \left[\sigma(-1)^{\|\mathbf{s}\|_1}\right]^+ 2^{-\|\mathbf{s}\|_1 - \|\mathbf{i}\|_1 - \|\mathbf{e}\|_1},$$
$$\Lambda^-_{\mathcal{A},\mathbf{s}} = \left[\sigma(-1)^{\|\mathbf{s}\|_1}\right]^- 2^{-\|\mathbf{s}\|_1 - \|\mathbf{i}\|_1 - \|\mathbf{e}\|_1}.$$

(iii) Banzhaf II: $\Lambda^+_{\mathcal{A},\mathbf{s}} = [\sigma]^+ 2^{-\|\mathbf{i}\|_1 - \|\mathbf{e}\|_1}$ *and* $\Lambda^-_{\mathcal{A},\mathbf{s}} = [\sigma]^- 2^{-\|\mathbf{i}\|_1 - \|\mathbf{e}\|_1}$.

(iv) SII: $\Lambda^+_{\mathcal{A},\mathbf{s}} = [\sigma]^+ \operatorname{Beta}(\|\mathbf{i}\|_1 + 1, \|\mathbf{e}\|_1 + 1)$ *and* $\Lambda^-_{\mathcal{A},\mathbf{s}} = [\sigma]^- \operatorname{Beta}(\|\mathbf{i}\|_1 + 1, \|\mathbf{e}\|_1 + 1)$.

(v) STII, $\|\mathbf{s}\|_1 = k$*:*

$$\Lambda^+_{\mathcal{A},\mathbf{s}} = [\sigma]^+ k \operatorname{Beta}(\|\mathbf{i}\|_1 + 1, k + \|\mathbf{e}\|_1),$$
$$\Lambda^-_{\mathcal{A},\mathbf{s}} = [\sigma]^- k \operatorname{Beta}(\|\mathbf{i}\|_1 + 1, k + \|\mathbf{e}\|_1).$$

(vi) STII, $\|\mathbf{s}\|_1 < k$*:* $\Lambda^+_{\mathcal{A},\mathbf{s}} = [\sigma]^+ \mathbb{1}[\|\mathbf{i}\|_1 = 0]$ *and* $\Lambda^-_{\mathcal{A},\mathbf{s}} = [\sigma]^- \mathbb{1}[\|\mathbf{i}\|_1 = 0]$.

(vii) FSII, $\|\mathbf{s}\|_1 = k$*:*

$$\Lambda^+_{\mathcal{A},\mathbf{s}} = [\sigma]^+ \frac{(2k-1)!}{((k-1)!)^2} \operatorname{Beta}(k + \|\mathbf{i}\|_1, k + \|\mathbf{e}\|_1),$$
$$\Lambda^-_{\mathcal{A},\mathbf{s}} = [\sigma]^- \frac{(2k-1)!}{((k-1)!)^2} \operatorname{Beta}(k + \|\mathbf{i}\|_1, k + \|\mathbf{e}\|_1).$$

Proof. We evaluate these masses for the listed indices. In the evaluations below, we write $a := \|\mathbf{t}\|_1$. Since $\mathbf{i}$ and $\mathbf{t}$ are disjoint, $\|\mathbf{i} \vee \mathbf{t}\|_1 = \|\mathbf{i}\|_1 + a$. For each size a, the sum contains $\binom{\|\mathbf{f} \setminus \mathbf{s}\|_1}{a}$ coalitions $\mathbf{t} \leq \mathbf{f} \setminus \mathbf{s}$. For the Möbius transform, the Banzhaf II, SII, STII, and FSII, the kernels in Table 4 are non-negative. For these kernels, we factor out the positive and negative parts of σ using

$$[\sigma\, \lambda(\|\mathbf{i} \vee \mathbf{t}\|_1)]^+ = [\sigma]^+ \lambda(\|\mathbf{i} \vee \mathbf{t}\|_1),$$
$$[\sigma\, \lambda(\|\mathbf{i} \vee \mathbf{t}\|_1)]^- = [\sigma]^- \lambda(\|\mathbf{i} \vee \mathbf{t}\|_1).$$

The items with Beta-function kernels follow from the identity

$$\sum_{a=0}^{n} \binom{n}{a} \operatorname{Beta}(a + p, n - a + q) = \operatorname{Beta}(p, q)$$

for $n \in \mathbb{N}_0$ and $p, q > 0$. The identity follows from the integral representation of the Beta function in Appendix A.1 and the binomial theorem,

$$\begin{aligned}\sum_{a=0}^{n} \binom{n}{a} \operatorname{Beta}(a + p, n - a + q) &= \int_0^1 \sum_{a=0}^{n} \binom{n}{a} t^{a+p-1}(1-t)^{n-a+q-1}\, \mathrm{d}t \\ &= \int_0^1 t^{p-1}(1-t)^{q-1}\, \mathrm{d}t = \operatorname{Beta}(p, q),\end{aligned}$$

where the second equality applies the binomial theorem. We now prove the items in order.

(i) The Möbius kernel is $\lambda^{\mathrm{MT}}(\|\mathbf{i}\|_1 + a) = \mathbb{1}[\|\mathbf{i}\|_1 + a = 0]$ (Harsanyi, 1982; Grabisch et al., 2000), so only $\mathbf{i} = \mathbf{t} = \mathbf{0}_d$ can contribute to Equation (11), and $\sum_{a=0}^{\|\mathbf{f}\setminus\mathbf{s}\|_1} \binom{\|\mathbf{f}\setminus\mathbf{s}\|_1}{a} \mathbb{1}[\|\mathbf{i}\|_1 + a = 0] = \mathbb{1}[\|\mathbf{i}\|_1 = 0]$.

(ii) The Fourier kernel is $\lambda^{\mathrm{FT}}(\|\mathbf{i}\|_1 + a) = (-1)^{\|\mathbf{s}\|_1} 2^{-d}$ (O'Donnell, 2014; Kang et al., 2025), so the sums collect $2^{\|\mathbf{f}\setminus\mathbf{s}\|_1}$ equal summands, and $2^{-d} \cdot 2^{\|\mathbf{f}\setminus\mathbf{s}\|_1} = 2^{-\|\mathbf{s}\|_1 - \|\mathbf{i}\|_1 - \|\mathbf{e}\|_1}$.

(iii) The Banzhaf kernel is $\lambda^{\mathrm{BII}}(\|\mathbf{i}\|_1 + a) = 2^{\|\mathbf{s}\|_1 - d}$ (Grabisch & Roubens, 1999), so the sums collect $2^{\|\mathbf{f}\setminus\mathbf{s}\|_1}$ equal summands, and $2^{\|\mathbf{s}\|_1 - d} \cdot 2^{\|\mathbf{f}\setminus\mathbf{s}\|_1} = 2^{-\|\mathbf{i}\|_1 - \|\mathbf{e}\|_1}$.

(iv) For SII, substituting the kernel (Grabisch & Roubens, 1999) and grouping coalitions by size gives

$$\begin{aligned}\Lambda^+_{\mathcal{A},\mathbf{s}} &= [\sigma]^+ \sum_{a=0}^{\|\mathbf{f}\setminus\mathbf{s}\|_1} \binom{\|\mathbf{f}\setminus\mathbf{s}\|_1}{a} \mathrm{Beta}(a + \|\mathbf{i}\|_1 + 1, \|\mathbf{f}\setminus\mathbf{s}\|_1 - a + \|\mathbf{e}\|_1 + 1) \\ &= [\sigma]^+ \,\mathrm{Beta}(\|\mathbf{i}\|_1 + 1, \|\mathbf{e}\|_1 + 1).\end{aligned}$$

The last equality applies the Beta-binomial identity with $n = \|\mathbf{f}\setminus\mathbf{s}\|_1$, $p = \|\mathbf{i}\|_1 + 1$, and $q = \|\mathbf{e}\|_1 + 1$. Replacing $[\sigma]^+$ by $[\sigma]^-$ gives the stated negative mass.

(v) For STII with $\|\mathbf{s}\|_1 = k$, we rewrite the kernel (Sundararajan et al., 2020) as

$$\begin{aligned}\lambda^{\mathrm{STII}}(\|\mathbf{i}\|_1 + a) &= \frac{k}{d}\binom{d-1}{\|\mathbf{i}\|_1 + a}^{-1} \\ &= k\,\mathrm{Beta}(a + \|\mathbf{i}\|_1 + 1, \|\mathbf{f}\setminus\mathbf{s}\|_1 - a + k + \|\mathbf{e}\|_1).\end{aligned}$$

The last equality uses the factorial form of the Beta function and $d = k + \|\mathbf{i}\|_1 + \|\mathbf{e}\|_1 + \|\mathbf{f}\setminus\mathbf{s}\|_1$. Substituting this kernel gives

$$\begin{aligned}\Lambda^+_{\mathcal{A},\mathbf{s}} &= [\sigma]^+ k \sum_{a=0}^{\|\mathbf{f}\setminus\mathbf{s}\|_1} \binom{\|\mathbf{f}\setminus\mathbf{s}\|_1}{a} \mathrm{Beta}(a + \|\mathbf{i}\|_1 + 1, \|\mathbf{f}\setminus\mathbf{s}\|_1 - a + k + \|\mathbf{e}\|_1) \\ &= [\sigma]^+ k\,\mathrm{Beta}(\|\mathbf{i}\|_1 + 1, k + \|\mathbf{e}\|_1).\end{aligned}$$

The last equality applies the Beta-binomial identity with $n = \|\mathbf{f}\setminus\mathbf{s}\|_1$, $p = \|\mathbf{i}\|_1 + 1$, and $q = k + \|\mathbf{e}\|_1$. Replacing $[\sigma]^+$ by $[\sigma]^-$ gives the stated negative mass.

(vi) STII with $\|\mathbf{s}\|_1 < k$ has $\lambda^{\mathrm{STII}}(\|\mathbf{i}\|_1 + a) = \mathbb{1}[\|\mathbf{i}\|_1 + a = 0]$, as in item *(i)* (Sundararajan et al., 2020).

(vii) For FSII with $\|\mathbf{s}\|_1 = k$, substituting the kernel (Tsai et al., 2023) gives

$$\begin{aligned}\Lambda^+_{\mathcal{A},\mathbf{s}} &= [\sigma]^+ \frac{(2k-1)!}{((k-1)!)^2} \sum_{a=0}^{\|\mathbf{f}\setminus\mathbf{s}\|_1} \binom{\|\mathbf{f}\setminus\mathbf{s}\|_1}{a} \\ &\qquad \cdot \mathrm{Beta}(a + k + \|\mathbf{i}\|_1, \|\mathbf{f}\setminus\mathbf{s}\|_1 - a + k + \|\mathbf{e}\|_1) \\ &= [\sigma]^+ \frac{(2k-1)!}{((k-1)!)^2} \mathrm{Beta}(k + \|\mathbf{i}\|_1, k + \|\mathbf{e}\|_1).\end{aligned}$$

The last equality applies the Beta-binomial identity with $n = \|\mathbf{f}\setminus\mathbf{s}\|_1$, $p = k + \|\mathbf{i}\|_1$, and $q = k + \|\mathbf{e}\|_1$. Replacing $[\sigma]^+$ by $[\sigma]^-$ gives the stated negative mass.

This proves Proposition C.3. □

C.2 Sparse Fourier Explanation Bounds

The proof of Theorem 3.3 relies on three ingredients: the per-piece bounds on the surrogate (Lemma C.4), the per-piece residual bounds combining them with the affine value bounds (Lemma C.5), and the square bounds over an interval (Lemma C.6). We first state Theorem 3.3 in a more explicit form than provided in Section 3.

Theorem 3.3 (Extended)**.** *Let $\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(K)} \in \mathbb{B}^d$ with the Fourier estimates $\widetilde{v}_1, \dots, \widetilde{v}_K$ and the surrogate $h(\mathbf{t}) := \sum_{k\in[K]} \widetilde{v}_k \chi_{\mathbf{s}^{(k)}}(\mathbf{t})$. Let $\mathsf{A} = \{\mathcal{A}_1, \dots, \mathcal{A}_J\}$ be a DAP with affine value bounds $\underline{v}(\mathbf{t}) = \underline{\mathbf{w}}^\mathsf{T}\mathbf{t} + \underline{b}$ and $\overline{v}(\mathbf{t}) = \overline{\mathbf{w}}^\mathsf{T}\mathbf{t} + \overline{b}$ for each $\mathcal{A} \in \mathsf{A}$. For $\mathcal{A} = [\underline{\mathbf{a}}, \overline{\mathbf{a}}] \in \mathsf{A}$, define $\mathbf{f} := \overline{\mathbf{a}} - \underline{\mathbf{a}}$, $\mathcal{Q}_1 := \{k \in [K] : \mathbf{s}^{(k)} \odot \mathbf{f} = \mathbf{0}_d\}$, $\mathcal{Q}_2 := \{k \in [K] : \mathbf{s}^{(k)} \odot \mathbf{f} \neq \mathbf{0}_d\}$, and*

$$\underline{r}_\mathcal{A} := \underline{h}_\mathcal{A} - \overline{v}(\underline{\mathbf{a}}) - \mathbf{f}^\mathsf{T}[\overline{\mathbf{w}}]^+, \qquad \overline{r}_\mathcal{A} := \overline{h}_\mathcal{A} - \underline{v}(\underline{\mathbf{a}}) - \mathbf{f}^\mathsf{T}[\underline{\mathbf{w}}]^-,$$

where $\underline{h}_\mathcal{A} := \sum_{k\in\mathcal{Q}_1} \widetilde{v}_k \chi_{\mathbf{s}^{(k)}}(\underline{\mathbf{a}}) - \sum_{k\in\mathcal{Q}_2} |\widetilde{v}_k|$, $\overline{h}_\mathcal{A} := \sum_{k\in\mathcal{Q}_1} \widetilde{v}_k \chi_{\mathbf{s}^{(k)}}(\underline{\mathbf{a}}) + \sum_{k\in\mathcal{Q}_2} |\widetilde{v}_k|$. Then

$$\underline{E} := \sum_{\mathcal{A}\in\mathsf{A}} 2^{\|\mathbf{f}\|_1 - d} \min(-\underline{r}_\mathcal{A}, 0, \overline{r}_\mathcal{A})^2 \le \|h - v\|_2^2 \le \sum_{\mathcal{A}\in\mathsf{A}} 2^{\|\mathbf{f}\|_1 - d} \max(-\underline{r}_\mathcal{A}, \overline{r}_\mathcal{A})^2 =: \overline{E}.$$

We first prove the bounds on the surrogate and the residual over each piece, and then prove the square bounds over an interval.

Lemma C.4 (Surrogate Bounds)**.** *Let $\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(K)} \in \mathbb{B}^d$ with the Fourier estimates $\widetilde{v}_1, \dots, \widetilde{v}_K$ and the* surrogate *$h(\mathbf{u}) := \sum_{k\in[K]} \widetilde{v}_k \chi_{\mathbf{s}^{(k)}}(\mathbf{u})$. Let $\mathcal{A} = [\underline{\mathbf{a}}, \overline{\mathbf{a}}]$ be a piece of a DAP, and define $\mathbf{f} := \overline{\mathbf{a}} - \underline{\mathbf{a}}$, $\mathcal{Q}_1 := \{k \in [K] : \mathbf{s}^{(k)} \odot \mathbf{f} = \mathbf{0}_d\}$, and $\mathcal{Q}_2 := \{k \in [K] : \mathbf{s}^{(k)} \odot \mathbf{f} \neq \mathbf{0}_d\}$. Furthermore, define*

$$\underline{h}_\mathcal{A} := \sum_{k\in\mathcal{Q}_1} \widetilde{v}_k \chi_{\mathbf{s}^{(k)}}(\underline{\mathbf{a}}) - \sum_{k\in\mathcal{Q}_2} |\widetilde{v}_k|,$$
$$\overline{h}_\mathcal{A} := \sum_{k\in\mathcal{Q}_1} \widetilde{v}_k \chi_{\mathbf{s}^{(k)}}(\underline{\mathbf{a}}) + \sum_{k\in\mathcal{Q}_2} |\widetilde{v}_k|.$$

For every $\mathbf{u} \in \mathcal{A}$, the surrogate satisfies

$$\underline{h}_\mathcal{A} \le h(\mathbf{u}) \le \overline{h}_\mathcal{A}. \tag{21}$$

Proof. For every $\mathbf{u} \in \mathcal{A}$, we define $\mathbf{t} := \mathbf{u} - \underline{\mathbf{a}}$, so that $\mathbf{u} = \underline{\mathbf{a}} \vee \mathbf{t}$. The sets $\mathcal{Q}_1$ and $\mathcal{Q}_2$ separate the coalitions $\mathbf{s}^{(k)}$ based on whether $\chi_{\mathbf{s}^{(k)}}(\mathbf{u})$ has a fixed value for every $\mathbf{u} \in \mathcal{A}$. In particular, for every $\mathbf{u} \in \mathcal{A}$ and every $k \in \mathcal{Q}_1$, together $\mathbf{t} \le \mathbf{f}$ and $\mathbf{s}^{(k)} \odot \mathbf{f} = \mathbf{0}_d$ give

$$\chi_{\mathbf{s}^{(k)}}(\mathbf{u}) = (-1)^{\mathbf{s}^{(k)\mathsf{T}}(\underline{\mathbf{a}}\vee\mathbf{t})} = (-1)^{\mathbf{s}^{(k)\mathsf{T}}\underline{\mathbf{a}}} = \chi_{\mathbf{s}^{(k)}}(\underline{\mathbf{a}}).$$

In contrast, $\chi_{\mathbf{s}^{(k)}}(\mathbf{u})$ takes both positive and negative values for different $\mathbf{u} \in \mathcal{A}$ for $k \in \mathcal{Q}_2$. Additionally, the index sets $\mathcal{Q}_1$ and $\mathcal{Q}_2$ satisfy $\mathcal{Q}_1 \cap \mathcal{Q}_2 = \emptyset$ and $\mathcal{Q}_1 \cup \mathcal{Q}_2 = [K]$. Therefore, for every $\mathbf{u} \in \mathcal{A}$,

$$\begin{aligned} h(\mathbf{u}) &= \sum_{k\in\mathcal{Q}_1} \widetilde{v}_k \chi_{\mathbf{s}^{(k)}}(\underline{\mathbf{a}}) + \sum_{k\in\mathcal{Q}_2} \widetilde{v}_k \chi_{\mathbf{s}^{(k)}}(\mathbf{u}) \\ &\ge \sum_{k\in\mathcal{Q}_1} \widetilde{v}_k \chi_{\mathbf{s}^{(k)}}(\underline{\mathbf{a}}) - \sum_{k\in\mathcal{Q}_2} |\widetilde{v}_k| = \underline{h}_\mathcal{A}, \end{aligned}$$

where the inequality follows from $-1 \le \chi_{\mathbf{s}^{(k)}}(\mathbf{u}) \le 1$. Following equivalent steps, we obtain

$$h(\mathbf{u}) \le \sum_{k\in\mathcal{Q}_1} \widetilde{v}_k \chi_{\mathbf{s}^{(k)}}(\underline{\mathbf{a}}) + \sum_{k\in\mathcal{Q}_2} |\widetilde{v}_k| = \overline{h}_\mathcal{A}.$$

This proves Lemma C.4. □

Lemma C.5 (Piecewise Residual Bounds)**.** *Let $\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(K)} \in \mathbb{B}^d$ with the Fourier estimates $\widetilde{v}_1, \dots, \widetilde{v}_K$ and the surrogate $h(\mathbf{u}) := \sum_{k\in[K]} \widetilde{v}_k \chi_{\mathbf{s}^{(k)}}(\mathbf{u})$. Let $\mathcal{A} = [\underline{\mathbf{a}}, \overline{\mathbf{a}}]$ be a piece of a DAP with affine value bounds $\underline{v}(\mathbf{u}) = \underline{\mathbf{w}}^\mathsf{T}\mathbf{u} + \underline{b}$ and $\overline{v}(\mathbf{u}) = \overline{\mathbf{w}}^\mathsf{T}\mathbf{u} + \overline{b}$, such that $\underline{v}(\mathbf{u}) \le v(\mathbf{u}) \le \overline{v}(\mathbf{u})$ for every $\mathbf{u} \in \mathcal{A}$. Let $\underline{h}_\mathcal{A}$ and $\overline{h}_\mathcal{A}$ be as in Lemma C.4 and let $\mathbf{f} := \overline{\mathbf{a}} - \underline{\mathbf{a}}$, as well as*

$$\underline{r}_\mathcal{A} := \underline{h}_\mathcal{A} - \overline{v}(\underline{\mathbf{a}}) - \mathbf{f}^\mathsf{T}[\overline{\mathbf{w}}]^+ \qquad \overline{r}_\mathcal{A} := \overline{h}_\mathcal{A} - \underline{v}(\underline{\mathbf{a}}) - \mathbf{f}^\mathsf{T}[\underline{\mathbf{w}}]^-.$$

For every $\mathbf{u} \in \mathcal{A}$, it holds that

$$\underline{r}_\mathcal{A} \le h(\mathbf{u}) - v(\mathbf{u}) \le \overline{r}_\mathcal{A}. \tag{22}$$

Proof. Let $\mathbf{u} \in \mathcal{A}$ and define $\mathbf{t} := \mathbf{u} - \underline{\mathbf{a}}$, such that $\mathbf{u} = \underline{\mathbf{a}} \vee \mathbf{t}$. We obtain

$$\begin{aligned} v(\mathbf{u}) \geq \underline{v}(\mathbf{u}) &= \underline{v}(\underline{\mathbf{a}}) + \mathbf{t}^{\mathsf{T}}\underline{\mathbf{w}} \\ &\geq \underline{v}(\underline{\mathbf{a}}) + \mathbf{f}^{\mathsf{T}}[\underline{\mathbf{w}}]^{-} =: \ell_{\mathcal{A}} \\ v(\mathbf{u}) \leq \overline{v}(\mathbf{u}) &= \overline{v}(\underline{\mathbf{a}}) + \mathbf{t}^{\mathsf{T}}\overline{\mathbf{w}} \\ &\leq \overline{v}(\underline{\mathbf{a}}) + \mathbf{f}^{\mathsf{T}}[\overline{\mathbf{w}}]^{+} =: u_{\mathcal{A}}. \end{aligned}$$

Combining this with Lemma C.4, we obtain

$$\begin{aligned} h(\mathbf{u}) - v(\mathbf{u}) &\geq \underline{h}_{\mathcal{A}} - u_{\mathcal{A}} = \underline{r}_{\mathcal{A}} \\ h(\mathbf{u}) - v(\mathbf{u}) &\leq \overline{h}_{\mathcal{A}} - \ell_{\mathcal{A}} = \overline{r}_{\mathcal{A}}. \end{aligned}$$

This proves Lemma C.5. □

Finally, we prove a simple bound on the quadratic z^2 over an interval.

Lemma C.6 (Square Bounds over an Interval)**.** *Let $a, b \in \mathbb{R}$ with $a \leq b$. Then*

$$\min_{z\in[a,b]} z^2 = \min(-a, 0, b)^2, \qquad \max_{z\in[a,b]} z^2 = \max(-a, b)^2.$$

Proof. We prove both identities by distinguishing the position of $[a, b]$ relative to zero.

- Suppose that $a \leq b \leq 0$. For every $z \in [a, b]$, we have $0 \leq -b \leq -z \leq -a$, and hence $b^2 \leq z^2 \leq a^2$. Moreover, $\min(-a, 0, b) = b$ and $\max(-a, b) = -a$, which proves both identities in this case.
- Suppose that $a \leq 0 \leq b$. Since $0 \in [a, b]$, the minimum equals 0. Furthermore, every $z \in [a, b]$ satisfies $|z| \leq \max(-a, b)$. Therefore, $\max_{z\in[a,b]} z^2 = \max(-a, b)^2$, while $\min(-a, 0, b) = 0$.
- Suppose that $0 \leq a \leq b$. For every $z \in [a, b]$, we have $0 \leq a \leq z \leq b$, and hence $a^2 \leq z^2 \leq b^2$. Moreover, $\min(-a, 0, b) = -a$ and $\max(-a, b) = b$, which proves both identities in this case.

The three cases cover all ordered endpoints $a \leq b$, including intervals with zero endpoints. This proves Lemma C.6. □

Proof of Theorem 3.3. Let $\mathcal{A} \in \mathsf{A}$ and $\mathbf{u} \in \mathcal{A}$. Combining Equation (22) and Lemma C.6 gives

$$\min(-\underline{r}_{\mathcal{A}}, 0, \overline{r}_{\mathcal{A}})^2 \leq \big(h(\mathbf{u}) - v(\mathbf{u})\big)^2 \leq \max(-\underline{r}_{\mathcal{A}}, \overline{r}_{\mathcal{A}})^2.$$

The piece $\mathcal{A}$ contains $2^{\|\mathbf{f}\|_1}$ coalitions and therefore accounts for the fraction $2^{\|\mathbf{f}\|_1 - d}$ of the coalition cube. By applying the definition of $\|\cdot\|_2^2$, and using that A partitions $\mathbb{B}^d$, we obtain

$$\begin{aligned} \|h - v\|_2^2 &= 2^{-d} \sum_{\mathbf{u}\in\mathbb{B}^d} \big(h(\mathbf{u}) - v(\mathbf{u})\big)^2 \\ &= 2^{-d} \sum_{\mathcal{A}\in\mathsf{A}} \sum_{\mathbf{u}\in\mathcal{A}} \big(h(\mathbf{u}) - v(\mathbf{u})\big)^2 \\ &\geq \sum_{\mathcal{A}\in\mathsf{A}} 2^{\|\mathbf{f}\|_1 - d} \min(-\underline{r}_{\mathcal{A}}, 0, \overline{r}_{\mathcal{A}})^2 = \underline{E}. \end{aligned}$$

Following equivalent steps, we similarly obtain

$$\|h - v\|_2^2 \leq \sum_{\mathcal{A}\in\mathsf{A}} 2^{\|\mathbf{f}\|_1 - d} \max(-\underline{r}_{\mathcal{A}}, \overline{r}_{\mathcal{A}})^2 = \overline{E}.$$

This proves Theorem 3.3. □

The faithfulness bound in Corollary 3.4 requires bounds on both the residual of the surrogate and the variance of the game. We obtain the residual bounds from Theorem 3.3 and derive the variance bounds in two ways. The first uses the identity that the variance equals the energy minus the squared mean. The second expresses the variance using the residual, the estimated Fourier coefficients, and the corresponding coefficients of the game. The extended corollary below gives closed-form bounds for both expressions, so that we can select the pair with the smaller width.

Corollary 3.4 (Extended)**.** *Let $\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(K)} \in \mathbb{B}^d$ be distinct, with estimates $\widetilde{v}_1, \dots, \widetilde{v}_K \in \mathbb{R}$ and surrogate $h := \sum_{k\in[K]} \widetilde{v}_k \chi_{\mathbf{s}^{(k)}}$. Assume*

$$\underline{E} \le \|h - v\|_2^2 \le \overline{E}, \qquad \underline{\|v\|_2^2} \le \|v\|_2^2 \le \overline{\|v\|_2^2}, \qquad \underline{\widehat{v}}(\mathbf{s}) \le \widehat{v}(\mathbf{s}) \le \overline{\widehat{v}}(\mathbf{s})$$

for every $\mathbf{s} \in \{\mathbf{0}_d, \mathbf{s}^{(1)}, \dots, \mathbf{s}^{(K)}\}$. Let $\widetilde{v}_0 := \widetilde{v}_k$ if $\mathbf{s}^{(k)} = \mathbf{0}_d$ for some $k \in [K]$, and let $\widetilde{v}_0 := 0$ otherwise. For every $k \in [K]$, define

$$\underline{\gamma}_k := \min\big\{\widetilde{v}_k(2\underline{\widehat{v}}(\mathbf{s}^{(k)}) - \widetilde{v}_k),\ \widetilde{v}_k(2\overline{\widehat{v}}(\mathbf{s}^{(k)}) - \widetilde{v}_k)\big\},$$
$$\overline{\gamma}_k := \max\big\{\widetilde{v}_k(2\underline{\widehat{v}}(\mathbf{s}^{(k)}) - \widetilde{v}_k),\ \widetilde{v}_k(2\overline{\widehat{v}}(\mathbf{s}^{(k)}) - \widetilde{v}_k)\big\}.$$

For every $z \in \mathbb{R}$, define

$$\underline{\kappa}(z) := \min\big(z - \underline{\widehat{v}}(\mathbf{0}_d), 0, \overline{\widehat{v}}(\mathbf{0}_d) - z\big)^2, \qquad \overline{\kappa}(z) := \max\big(z - \underline{\widehat{v}}(\mathbf{0}_d), \overline{\widehat{v}}(\mathbf{0}_d) - z\big)^2.$$

Both pairs

$$\begin{aligned} \underline{V}^{(0)} &:= \underline{\|v\|_2^2} - \overline{\kappa}(0), & \overline{V}^{(0)} &:= \overline{\|v\|_2^2} - \underline{\kappa}(0), \\ \underline{V}^{(K)} &:= \underline{E} + \sum_{\substack{k\in[K]\\ \mathbf{s}^{(k)}\neq\mathbf{0}_d}} \underline{\gamma}_k - \overline{\kappa}(\widetilde{v}_0), & \overline{V}^{(K)} &:= \overline{E} + \sum_{\substack{k\in[K]\\ \mathbf{s}^{(k)}\neq\mathbf{0}_d}} \overline{\gamma}_k - \underline{\kappa}(\widetilde{v}_0) \end{aligned} \tag{23}$$

bound $V := \|v - \widehat{v}(\mathbf{0}_d)\|_2^2$. Let $(\underline{V}, \overline{V})$ be either pair in Equation (23)*. If $\underline{V} > 0$, then*

$$1 - \overline{E}/\underline{V} \le R^2 \le 1 - \underline{E}/\overline{V}.$$

Proof of Corollary 3.4. We first express the variance using either the energy or the residual, then bound both expressions and apply the definition of faithfulness. By Equation (2), the mean of v is $\widehat{v}(\mathbf{0}_d)$, so

$$\begin{aligned} V &= 2^{-d} \sum_{\mathbf{s}\in\mathbb{B}^d} \big(v(\mathbf{s}) - \widehat{v}(\mathbf{0}_d)\big)^2 \\ &= \|v\|_2^2 - 2\widehat{v}(\mathbf{0}_d)\, 2^{-d} \sum_{\mathbf{s}\in\mathbb{B}^d} v(\mathbf{s}) + \widehat{v}(\mathbf{0}_d)^2 = \|v\|_2^2 - \widehat{v}(\mathbf{0}_d)^2. \end{aligned} \tag{24}$$

For the residual, Parseval's theorem and the distinctness of the reported frequencies give

$$\begin{aligned} \|h - v\|_2^2 &= \sum_{\mathbf{s}\in\mathbb{B}^d\setminus\{\mathbf{s}^{(1)},\dots,\mathbf{s}^{(K)}\}} \widehat{v}(\mathbf{s})^2 + \sum_{k\in[K]} \big(\widetilde{v}_k - \widehat{v}(\mathbf{s}^{(k)})\big)^2 \\ &= \|v\|_2^2 - \sum_{k\in[K]} \widetilde{v}_k\big(2\widehat{v}(\mathbf{s}^{(k)}) - \widetilde{v}_k\big). \end{aligned} \tag{25}$$

Rearranging Equation (25) gives the energy as

$$\|v\|_2^2 = \|h - v\|_2^2 + \sum_{k\in[K]} \widetilde{v}_k\big(2\widehat{v}(\mathbf{s}^{(k)}) - \widetilde{v}_k\big).$$

Substituting this expression for the energy into Equation (24) and separating the constant term gives

$$\begin{aligned} V &= \|v\|_2^2 - \widehat{v}(\mathbf{0}_d)^2 \\ &= \|h - v\|_2^2 + \sum_{k\in[K]} \widetilde{v}_k\big(2\widehat{v}(\mathbf{s}^{(k)}) - \widetilde{v}_k\big) - \widehat{v}(\mathbf{0}_d)^2 \\ &= \|h - v\|_2^2 + \sum_{\substack{k\in[K]\\ \mathbf{s}^{(k)}\neq\mathbf{0}_d}} \widetilde{v}_k\big(2\widehat{v}(\mathbf{s}^{(k)}) - \widetilde{v}_k\big) + \widetilde{v}_0\big(2\widehat{v}(\mathbf{0}_d) - \widetilde{v}_0\big) - \widehat{v}(\mathbf{0}_d)^2 \\ &= \|h - v\|_2^2 + \sum_{\substack{k\in[K]\\ \mathbf{s}^{(k)}\neq\mathbf{0}_d}} \widetilde{v}_k\big(2\widehat{v}(\mathbf{s}^{(k)}) - \widetilde{v}_k\big) - \big(\widehat{v}(\mathbf{0}_d) - \widetilde{v}_0\big)^2. \end{aligned} \tag{26}$$

The third equality uses the distinctness of the frequencies and the definition of $\widetilde{v}_0$, which is zero when the constant term is absent. The last equality completes the square.

We now bound the two expressions for the variance. For fixed $\widetilde{v}_k$, the function $z \mapsto \widetilde{v}_k(2z - \widetilde{v}_k)$ is affine on $[\underline{\widehat{v}}(\mathbf{s}^{(k)}), \overline{\widehat{v}}(\mathbf{s}^{(k)})]$. Its extrema therefore occur at the endpoints of this interval. Consequently, for every $k \in [K]$,

$$\underline{\gamma}_k \leq \widetilde{v}_k\big(2\widehat{v}(\mathbf{s}^{(k)}) - \widetilde{v}_k\big) \leq \overline{\gamma}_k.$$

By Lemma C.6 applied to $[\underline{\widehat{v}}(\mathbf{0}_d) - z, \overline{\widehat{v}}(\mathbf{0}_d) - z]$,

$$\underline{\kappa}(z) \leq \big(\widehat{v}(\mathbf{0}_d) - z\big)^2 \leq \overline{\kappa}(z)$$

for every $z \in \mathbb{R}$. Using these inequalities in Equations (24) and (26), we obtain

$$V \geq \underline{\|v\|_2^2} - \overline{\kappa}(0) = \underline{V}^{(0)},$$
$$V \geq \underline{E} + \sum_{\substack{k \in [K] \\ \mathbf{s}^{(k)} \neq \mathbf{0}_d}} \underline{\gamma}_k - \overline{\kappa}(\widetilde{v}_0) = \underline{V}^{(K)}.$$

Replacing lower bounds on the added terms by upper bounds, and upper bounds on the subtracted squares by lower bounds, gives the two upper bounds in Equation (23). Thus, either pair satisfies $\underline{V} \leq V \leq \overline{V}$.

Finally, suppose that $\underline{V} > 0$. Since $\|h - v\|_2^2 \geq 0$ and $0 < \underline{V} \leq V \leq \overline{V}$,

$$\frac{\underline{E}}{\overline{V}} \leq \frac{\|h - v\|_2^2}{\overline{V}} \leq \frac{\|h - v\|_2^2}{V} \leq \frac{\|h - v\|_2^2}{\underline{V}} \leq \frac{\overline{E}}{\underline{V}}.$$

Subtracting this chain from one and applying the definition of R^2 in Section 2.1 proves Corollary 3.4. □

Since both pairs in Equation (23) bound the same variance, in practice, we select the pair with the smaller width.

C.3 Arbitrary Precision

Proposition 4.1 (Restated)**.** *For every $\varepsilon \geq 0$, there exists $n \in \mathbb{N}$, such that $\mathsf{A}^{(n)}$, as computed by* VeriDAP*, is an ε-approximate DAP.*

Proof. We prove Proposition 4.1 for the special case $\varepsilon = 0$, which directly implies the result for every larger ε. Since Split is only defined for pieces with a free feature, Select only selects such pieces, and it selects at least one of them whenever $\mathsf{A}^{(n-1)}$ contains one (Appendix B). Splitting a piece on a free feature replaces it by two non-empty, disjoint subcubes whose union is the piece. Hence, every $\mathsf{A}^{(n)}$ is a partition of $\mathbb{B}^d$ into non-empty subcubes, and $|\mathsf{A}^{(n)}| = |\mathsf{A}^{(n-1)}| + |\mathsf{B}^{(n)}|$.

Assume that $\mathsf{A}^{(2^d)}$ contains a piece with a free feature. Every earlier $\mathsf{A}^{(n)}$ contains an ancestor of this piece, which has at least as many free features, so that $|\mathsf{B}^{(n)}| \geq 1$ for all $n \in \{2, \ldots, 2^d\}$. Together with $|\mathsf{A}^{(1)}| = 1$, this gives $|\mathsf{A}^{(2^d)}| \geq 2^d$. However, a partition of the 2^d coalitions into non-empty pieces has at most 2^d pieces, with equality only if every piece contains exactly one coalition, contradicting the assumption. Therefore, each $\mathcal{A}_j \in \mathsf{A}^{(2^d)}$ contains exactly one coalition, regardless of the batch size b. This means that $\underline{\mathbf{a}}_j = \overline{\mathbf{a}}_j$ for every $\mathcal{A}_j = [\underline{\mathbf{a}}_j, \overline{\mathbf{a}}_j] \in \mathsf{A}^{(2^d)}$. By the properties of ABP stated in Section 2.2, this implies that the lower and upper bound on the value function are equal, $\underline{v}_j(\mathbf{s}) = \overline{v}_j(\mathbf{s})$, for every $\mathcal{A}_j = [\underline{\mathbf{a}}_j, \overline{\mathbf{a}}_j] \in \mathsf{A}^{(2^d)}$. Therefore, the approximation error $\max_{\mathbf{s} \in \mathcal{A}_j} \overline{v}_j(\mathbf{s}) - \underline{v}_j(\mathbf{s})$ is zero for every piece, establishing that $\mathsf{A}^{(2^d)}$ is an *exact* (0-approximate) DAP. Since $\mathsf{A}^{(2^d)}$ is simultaneously also an ε-approximate DAP for every $\varepsilon \geq 0$, this establishes Proposition 4.1. □

C.4 NON-REDUCIBLE HARDNESS

This section proves Theorem 4.2 and Corollaries C.22 and C.23. Appendix C.4.1 fixes the computational problems and the source problem we reduce from, Appendix C.4.2 proves the tractability result, Appendix C.4.3 the two-neuron hardness result, and Appendix C.4.4 the single-attention-layer hardness result. Appendix C.4.5 collects the consequences.

C.4.1 PROBLEMS AND CONVENTIONS

Throughout this section, $v(\mathbf{s}) = f(\mathrm{mix}_{\mathbf{x},\mathbf{y}}(\mathbf{s}))$ is the baseline value function of Section 2.1. As in Section 2, we identify coalitions with the feature subsets that they indicate, writing $\mathbf{s} \cong S$ for $S = \{i : \mathbf{s}_i = 1\}$, so that we may evaluate v on subsets, as Equation (27) does. We write φ_i for the Shapley value of feature i, that is, the II of Equation (1) with the SII kernel evaluated at the singleton coalition $\{i\}$

$$\varphi_i = \sum_{S \subseteq [d] \setminus \{i\}} \frac{|S|!\,(d - |S| - 1)!}{d!} \big(v(S \cup \{i\}) - v(S)\big). \tag{27}$$

At order $k = 1$, the k-SII, STII, and FSII all coincide with φ_i, so every hardness statement below directly applies to each of them, and to any algorithm that computes interactions of orders $1, \ldots, k$. Since the networks we construct have transcendental outputs in Appendix C.4.4, we state every result for the approximation problem.

Definition C.7 (SHAPLEY$(\mathcal{F}, \delta)$)**.** Let $\mathcal{F}$ be a class of neural networks and let $\delta : \mathbb{N} \to \mathbb{R}_{\geq 0}$. An instance consists of a network $f \in \mathcal{F}$ with d inputs and one output, given by its rational parameters in binary, $\mathbf{x}, \mathbf{y} \in \mathbb{Q}^d$, and a feature index $i \in [d]$. A solution is any $\hat{\varphi}_i \in \mathbb{Q}$ with $|\hat{\varphi}_i - \varphi_i| \leq \delta(d)$. We call the problem *exact* if $\delta \equiv 0$. The exact problem is well-posed only for classes whose Shapley values at rational inputs are rational, such as the piecewise-affine networks of Appendix C.4.3; for the attention networks of Appendix C.4.4, φ_i may be irrational, and only $\delta > 0$ is meaningful.

Hardness is understood with respect to polynomial-time Turing reductions, the standard notion for #P-hardness of functions with non-integer values. We reduce from the raw Shapley–Shubik index of weighted voting games.

Definition C.8 (Weighted Voting Game)**.** A *weighted voting game* is a tuple $G = (\mathbf{w}; q)$ with $\mathbf{w} \in \mathbb{N}^d$ and $q \in \mathbb{N}$, given in binary. Its value function is $v_G(S) := \mathbb{1}[\sum_{i \in S} \mathbf{w}_i \geq q]$ for $S \subseteq [d]$.

Definition C.9 (Raw Shapley–Shubik Index)**.** For a weighted voting game G and a player $i \in [d]$,

$$SS^*(G, i) := \sum_{S \subseteq [d] \setminus \{i\}} |S|!\,(d - |S| - 1)!\big(v_G(S \cup \{i\}) - v_G(S)\big).$$

Since v_G is monotone and Boolean, every summand is non-negative, so $SS^*(G, i) \in \mathbb{N}_0$ and $SS^*(G, i) \leq d!$. We use the following known fact.

Lemma C.10 (Deng & Papadimitriou, 1994; Elkind et al., 2009)**.** *Computing $SS^*(G, i)$ is #P-hard.*

Finally, we record how sensitive an interaction index is to a perturbation of the value function. This is what lets us replace an exact threshold gate by a smooth approximation of it in Appendix C.4.4.

Lemma C.11 (Stability of Interaction Indices)**.** *Let $v, v' : \mathbb{B}^d \to \mathbb{R}$ with $\max_{\mathbf{s} \in \mathbb{B}^d} |v(\mathbf{s}) - v'(\mathbf{s})| \leq \eta$, let $\mathbf{s} \in \mathbb{B}^d$ with $\|\mathbf{s}\|_1 = k$, and let $\mathcal{I}, \mathcal{I}'$ denote the II of Equation (1) with kernel λ applied to v and v', respectively. Then*

$$|\mathcal{I}(\mathbf{s}) - \mathcal{I}'(\mathbf{s})| \leq 2^k \eta \sum_{\mathbf{t} \leq \mathbf{1}_d - \mathbf{s}} |\lambda(\|\mathbf{t}\|_1)|.$$

For the SII kernel, the sum equals 1, so $|\mathcal{I}(\mathbf{s}) - \mathcal{I}'(\mathbf{s})| \leq 2^k \eta$; in particular $|\varphi_i - \varphi_i'| \leq 2\eta$.

Proof. The discrete derivative $\Delta v(\mathbf{s}; \mathbf{t}) = \sum_{\mathbf{u} \leq \mathbf{s}} (-1)^{\|\mathbf{s} \setminus \mathbf{u}\|_1} v(\mathbf{u} \vee \mathbf{t})$ is a signed sum of 2^k evaluations of v, whence $|\Delta v(\mathbf{s}; \mathbf{t}) - \Delta v'(\mathbf{s}; \mathbf{t})| \leq 2^k \eta$ for every $\mathbf{t}$. Inserting this into Equation (1)

and applying the triangle inequality gives the claim. For the SII kernel with $\|\mathbf{s}\|_1 = k$, grouping the $\mathbf{t} \leq \mathbf{1}_d - \mathbf{s}$ by $t = \|\mathbf{t}\|_1$ gives

$$\sum_{\mathbf{t} \leq \mathbf{1}_d - \mathbf{s}} \lambda^{\mathrm{SII}}(\|\mathbf{t}\|_1) = \sum_{t=0}^{d-k} \binom{d-k}{t} \frac{(d-k-t)!\, t!}{(d-k+1)!} = \sum_{t=0}^{d-k} \frac{(d-k)!}{(d-k+1)!} = 1,$$

and the kernel is non-negative, so the sum of absolute values is 1 as well. □

C.4.2 Tractability in the Input Dimension

In this section, we provide a formal proof of tractability when restricting the input dimension.

Proposition C.12. *Let $\mathcal{I}$ be an interaction index as in Equation* (1) *with a kernel computable in time polynomial in d, and let $\mathcal{T} \subseteq \mathbb{B}^d$ be a set of target coalitions. Then $\mathcal{I}(\mathbf{s})$ for all $\mathbf{s} \in \mathcal{T}$ can be computed from 2^d evaluations of f and $\mathcal{O}(|\mathcal{T}|\, 2^d)$ further arithmetic operations. The result is exact if f maps rationals to rationals (e.g. piecewise-affine networks), and otherwise accurate to within any $\delta > 0$ at an overhead polynomial in $\log(1/\delta)$. In particular, for $d = \mathcal{O}(\log |f|)$, all IIs of all orders are computable in time polynomial in the encoding size $|f|$ of f (and $\log(1/\delta)$), for every architecture.*

Proof. Evaluate $v(\mathbf{t}) = f(\mathrm{mix}_{\mathbf{x},\mathbf{y}}(\mathbf{t}))$ for every $\mathbf{t} \in \mathbb{B}^d$ and store the 2^d values in a table; this is 2^d forward passes. By the right-hand form of Equation (1),

$$\mathcal{I}(\mathbf{s}) = \sum_{\mathbf{t} \in \mathbb{B}^d} \lambda(\|\mathbf{t} \setminus \mathbf{s}\|_1)\, (-1)^{\|\mathbf{s} \setminus \mathbf{t}\|_1}\, v(\mathbf{t}),$$

Since λ takes only $d+1$ distinct arguments, we precompute its values once in time polynomial in d, so each target costs 2^d sign computations and multiply-accumulates. Summing over $\mathcal{T}$ gives the stated count. Since $|\mathcal{T}| \leq 2^d$, the total work is $\mathcal{O}(4^d)$ arithmetic operations on top of 2^d forward passes. With $d \leq c \log_2 |f|$ we get $4^d \leq |f|^{2c}$ and $2^d \leq |f|^c$, which is polynomial in $|f|$. All kernels of Table 4 are ratios of factorials of numbers at most d and of Bernoulli numbers of index at most k, hence computable exactly with $\mathrm{poly}(d)$ bits. □

Remark C.13 (Arithmetic Model). ReLU, maxpool, and piecewise-affine networks map rationals to rationals, so Proposition C.12 is exact for them. For networks with transcendental activations, such as softmax attention, the δ-accurate case holds because the kernel weights of Equation (1) sum in absolute value to at most $2^d \max_t |\lambda(t)|$, so evaluating each of the 2^d forward passes to within δ/W, with $W := 2^d \max_t |\lambda(t)|$, that is, with $\mathcal{O}(\log W + \log(1/\delta))$ bits, suffices. The Fourier coefficients of Section 2.1 can be obtained for all 2^d coalitions at once in $\mathcal{O}(d2^d)$ operations by the fast Walsh–Hadamard transform.

C.4.3 Hardness for Two ReLU Neurons

The reduction rests on the fact that a difference of two ReLU units represents a weighted threshold function *exactly on the Boolean cube*, even though it does not represent it on $\mathbb{R}^d$.

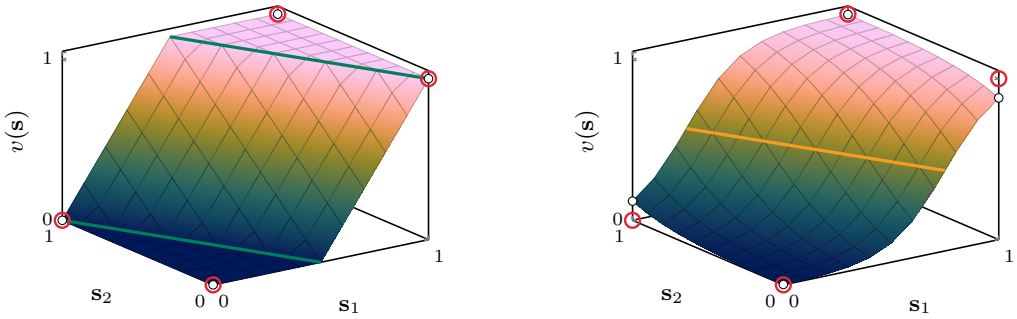
Lemma C.14 (Two-ReLU Threshold Representation)**.** *Let $\mathbf{w} \in \mathbb{Z}^d$, $q \in \mathbb{Z}$ and define $f_{\mathbf{w},q} : \mathbb{R}^d \to \mathbb{R}$ by*

$$f_{\mathbf{w},q}(\mathbf{z}) := \mathrm{ReLU}\big(\mathbf{w}^\mathsf{T}\mathbf{z} - q + 1\big) - \mathrm{ReLU}\big(\mathbf{w}^\mathsf{T}\mathbf{z} - q\big).$$

Then $f_{\mathbf{w},q}$ is a network with one hidden layer of exactly two ReLU neurons, and $f_{\mathbf{w},q}(\mathbf{z}) = \mathbb{1}[\mathbf{w}^\mathsf{T}\mathbf{z} \geq q]$ for every $\mathbf{z} \in \mathbb{B}^d$.

Proof. The first layer $\mathbf{z} \mapsto (\mathbf{w}^\mathsf{T}\mathbf{z} - q + 1, \mathbf{w}^\mathsf{T}\mathbf{z} - q)$ is affine, the second layer applies ReLU element-wise, and the read-out $(u_1, u_2) \mapsto u_1 - u_2$ is affine, so $f_{\mathbf{w},q}$ has the claimed shape. Fix $\mathbf{z} \in \mathbb{B}^d$ and set $a := \mathbf{w}^\mathsf{T}\mathbf{z} - q$. Since $\mathbf{z}$ is Boolean and $\mathbf{w}, q$ are integral, $a \in \mathbb{Z}$. If $a \leq -1$, then $a + 1 \leq 0$ and both ReLUs vanish, so $f_{\mathbf{w},q}(\mathbf{z}) = 0 = \mathbb{1}[a \geq 0]$. If $a \geq 0$, then both ReLUs are active and $f_{\mathbf{w},q}(\mathbf{z}) = (a+1) - a = 1 = \mathbb{1}[a \geq 0]$. As a is an integer, these cases are exhaustive. □

Proposition C.15 (Hardness for Two ReLU Neurons)**.** *Computing the Shapley value φ_i of the baseline value function is #P-hard, already for networks with a single hidden layer of two ReLU neurons and Boolean $\mathbf{x}, \mathbf{y}$, and already when φ_i is only required up to an additive error of $1/(4 \cdot d!)$.*



(a) Two ReLU neurons, $f_{\mathbf{w},q}$

(b) One attention head, $f_{\mathbf{w},q,\beta}$

Figure 6: **Value Functions of the Hardness Reductions.** We plot the value functions v of the networks that the reductions of Propositions C.15 and C.20 construct for the weighted voting game $G = ((2,1);2)$ over the coalition space $[0,1]^2$. In (a), the two-ReLU network $f_{\mathbf{w},q}$ of Lemma C.14 is affine on three regions, which the lines $\mathbf{w}^\mathsf{T}\mathbf{s} = q-1$ and $\mathbf{w}^\mathsf{T}\mathbf{s} = q$ — separate. Its value at every coalition $\circ$ equals v_G ○. In (b), the attention network $f_{\mathbf{w},q,\beta}$ of Lemma C.18 with $\beta = 4$ is a logistic step across the line $\mathbf{w}^\mathsf{T}\mathbf{s} = q - \frac{1}{2}$ —. Its value at every coalition $\circ$ differs from v_G ○ by at most $e^{-\beta/2}$. The reduction of Proposition C.20 chooses β large enough so that this difference does not change the rounded Shapley–Shubik index.

Proof. Let $\mathcal{F}_{2\mathrm{ReLU}}$ be the class of networks with one hidden layer of two ReLU neurons. We give a polynomial-time Turing reduction from SS^* to $\textsc{Shapley}(\mathcal{F}_{2\mathrm{ReLU}}, \delta)$ for every δ with $\delta(d) \le 1/(4 \cdot d!)$; Lemma C.10 then gives the claim, and the exact problem is the case $\delta \equiv 0$.

Let $G = (\mathbf{w}; q)$ be a weighted voting game on d players and let $i \in [d]$. Construct the instance

$$f := f_{\mathbf{w},q}, \qquad \mathbf{x} := \mathbf{1}_d, \qquad \mathbf{y} := \mathbf{0}_d,$$

with $f_{\mathbf{w},q}$ as in Lemma C.14. The construction copies $\mathbf{w}$ and q into the first layer's weights and bias, so it runs in time linear in the encoding of G, and all network parameters have the bit-length of the input.

For every $\mathbf{s} \in \mathbb{B}^d$ with $\mathbf{s} \cong S$ we have $\mathrm{mix}_{\mathbf{1}_d,\mathbf{0}_d}(\mathbf{s}) = \mathbf{s} \odot \mathbf{1}_d + (\mathbf{1}_d - \mathbf{s}) \odot \mathbf{0}_d = \mathbf{s}$, which is Boolean. Lemma C.14 therefore applies and gives

$$v(\mathbf{s}) = f_{\mathbf{w},q}(\mathbf{s}) = \mathbb{1}[\textstyle\sum_{i' \in S} \mathbf{w}_{i'} \ge q] = v_G(S).$$

Comparing Equation (27) with Definition C.9 yields

$$d!\, \varphi_i = SS^*(G, i). \tag{28}$$

Query the oracle on this instance to obtain $\hat{\varphi}_i$ with $|\hat{\varphi}_i - \varphi_i| \le \delta(d) \le 1/(4 \cdot d!)$, and output $\lfloor d!\, \hat{\varphi}_i \rceil$, the nearest integer. By Equation (28), $|d!\, \hat{\varphi}_i - SS^*(G,i)| \le 1/4 < 1/2$, and $SS^*(G,i)$ is an integer, so the output equals $SS^*(G,i)$. The number $d!$ has $\mathcal{O}(d \log d)$ bits and is computed with d multiplications, so the post-processing is polynomial. Finally, $\mathbf{x}$ and $\mathbf{y}$ are Boolean, as claimed. □

Remark C.16 (Scope of Proposition C.15). Three features of the construction are worth isolating.

- *(i)* The network has three affine regions over $\mathbb{R}^d$, so neither the neuron count nor the number of linear regions of f controls the cost of exact attribution; what is exponential is the number of *boxes* of the coalition cube on which v is affine, which is a property of how the region boundaries $\mathbf{w}^\mathsf{T}\mathbf{z} \in \{q-1, q\}$ cut $\mathbb{B}^d$.
- *(ii)* Because $\mathbf{y} = \mathbf{0}_d$ is a single point, the reduction applies unchanged to the marginal value function of Appendix A.2 with a background distribution that is a point mass at $\mathbf{0}_d$.
- *(iii)* Replacing the weights of Equation (27) by $2^{-(d-1)}$ turns Equation (28) into the statement that 2^{d-1} times the Banzhaf value of i equals the raw Banzhaf index of G, that is, the number of swings for i. Since computing the raw Banzhaf index of a weighted voting game is #P-hard (Prasad & Kelly, 1990), the same instance shows #P-hardness of the Banzhaf II of Equation (1).

C.4.4 HARDNESS FOR A SINGLE ATTENTION LAYER

We fix the architecture first. Write $\varsigma(a) := 1/(1+e^{-a})$ for the logistic function.

Definition C.17 (Attention-Only Networks)**.** A single-head softmax attention layer with parameters $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{m\times m}$ maps a token sequence $H = (\mathbf{h}_1, \dots, \mathbf{h}_T) \in \mathbb{R}^{T\times m}$ to

$$\mathrm{Attn}(H)_t := \sum_{t'=1}^{T} \frac{\exp\big(\langle \mathbf{W}_Q\mathbf{h}_t, \mathbf{W}_K\mathbf{h}_{t'}\rangle\big)}{\sum_{t''=1}^{T}\exp\big(\langle \mathbf{W}_Q\mathbf{h}_t, \mathbf{W}_K\mathbf{h}_{t''}\rangle\big)}\mathbf{W}_V\mathbf{h}_{t'}.$$

The class $\mathcal{F}_{1\mathrm{Attn}}$ consists of the networks $f = \pi \circ \mathrm{Attn} \circ \iota$, where $\iota : \mathbb{R}^d \to \mathbb{R}^{T\times m}$ is an affine embedding (token projection plus positional encoding) and $\pi : \mathbb{R}^{T\times m} \to \mathbb{R}$ is an affine read-out, with rational parameters. The softmax is the only non-linearity of f.

Lemma C.18 (A Single Head is a Logistic Gate)**.** *Let $\mathbf{w} \in \mathbb{Z}^d$, $q \in \mathbb{Z}$ and $\beta \in \mathbb{N}$. There is a network $f_{\mathbf{w},q,\beta} \in \mathcal{F}_{1\mathrm{Attn}}$ with $T = 2$ tokens and $m = 3$ that can be written down in time linear in the encoding of $(\mathbf{w}, q, \beta)$ and satisfies*

$$f_{\mathbf{w},q,\beta}(\mathbf{z}) = \varsigma\big(\beta(\mathbf{w}^\mathsf{T}\mathbf{z} - q + \tfrac{1}{2})\big) \qquad \textit{for all } \mathbf{z} \in \mathbb{R}^d.$$

Proof. Let $E_{ab} \in \mathbb{R}^{3\times 3}$ be the matrix with a one in entry (a, b) and zeros elsewhere. Define the affine embedding

$$\iota(\mathbf{z}) := (\mathbf{h}_1, \mathbf{h}_2), \qquad \mathbf{h}_1 := (1, 0, 0)^\mathsf{T}, \qquad \mathbf{h}_2 := \big(0,\ \mathbf{w}^\mathsf{T}\mathbf{z} - q + \tfrac{1}{2},\ 1\big)^\mathsf{T},$$

which is affine in $\mathbf{z}$, and the attention parameters $\mathbf{W}_Q := \beta E_{11}$, $\mathbf{W}_K := E_{12}$, $\mathbf{W}_V := E_{13}$. Then $\mathbf{W}_Q\mathbf{h}_1 = (\beta, 0, 0)^\mathsf{T}$, $\mathbf{W}_K\mathbf{h}_1 = \mathbf{0}_3$, $\mathbf{W}_K\mathbf{h}_2 = (\mathbf{w}^\mathsf{T}\mathbf{z} - q + \frac{1}{2}, 0, 0)^\mathsf{T}$, $\mathbf{W}_V\mathbf{h}_1 = \mathbf{0}_3$, and $\mathbf{W}_V\mathbf{h}_2 = (1, 0, 0)^\mathsf{T}$. The two attention logits at the query position $t = 1$ are therefore

$$\langle \mathbf{W}_Q\mathbf{h}_1, \mathbf{W}_K\mathbf{h}_1\rangle = 0, \qquad \langle \mathbf{W}_Q\mathbf{h}_1, \mathbf{W}_K\mathbf{h}_2\rangle = \beta\big(\mathbf{w}^\mathsf{T}\mathbf{z} - q + \tfrac{1}{2}\big) =: a,$$

so that $\mathrm{Attn}(\iota(\mathbf{z}))_1 = \frac{e^0}{e^0+e^a}\mathbf{0}_3 + \frac{e^a}{e^0+e^a}(1, 0, 0)^\mathsf{T}$. Choosing π to read the first coordinate of the output at position 1 gives $e^a/(1+e^a) = \varsigma(a)$. □

Lemma C.19 (Logistic Gates Approximate Threshold Gates on the Cube)**.** *Let $\mathbf{w} \in \mathbb{Z}^d$, $q \in \mathbb{Z}$, $\beta \in \mathbb{N}$. Then for every $\mathbf{z} \in \mathbb{B}^d$,*

$$\big|\varsigma\big(\beta(\mathbf{w}^\mathsf{T}\mathbf{z} - q + \tfrac{1}{2})\big) - \mathbb{1}[\mathbf{w}^\mathsf{T}\mathbf{z} \geq q]\big| \leq e^{-\beta/2}.$$

Proof. Set $a := \mathbf{w}^\mathsf{T}\mathbf{z} - q \in \mathbb{Z}$ and $a' := a + \frac{1}{2}$. If $a \geq 0$, then $a' \geq \frac{1}{2}$ and $1 - \varsigma(\beta a') = \varsigma(-\beta a') \leq e^{-\beta a'} \leq e^{-\beta/2}$. If $a < 0$, then $a \leq -1$ by integrality, so $a' \leq -\frac{1}{2}$ and $\varsigma(\beta a') \leq e^{\beta a'} \leq e^{-\beta/2}$. In both cases the indicator takes the value the logistic gate approaches, which gives the claim. □

Proposition C.20 (Hardness for a Single Attention Layer)**.** *Computing φ_i is #P-hard, already for networks whose only non-linearity is a single softmax attention layer over two tokens, with affine embedding and affine read-out, and Boolean $\mathbf{x}, \mathbf{y}$, and already when φ_i is only required up to an additive error of $1/(8 \cdot d!)$, that is, $2^{-\mathcal{O}(d\log d)}$.*

Proof. As in Appendix C.4.3, we reduce from *SS**, this time to $\textsc{Shapley}(\mathcal{F}_{1\mathrm{Attn}}, \delta)$ for every δ with $\delta(d) \leq 1/(8 \cdot d!)$.

Let $G = (\mathbf{w}; q)$ be a weighted voting game on d players and let $i \in [d]$. Set

$$\beta := 2d\lceil \log_2 d\rceil + 6, \qquad f := f_{\mathbf{w},q,\beta}, \qquad \mathbf{x} := \mathbf{1}_d, \qquad \mathbf{y} := \mathbf{0}_d,$$

with $f_{\mathbf{w},q,\beta}$ as in Lemma C.18. The integer β has $\mathcal{O}(\log d)$ bits, so the instance is constructed in polynomial time and all parameters are rational with the bit-length of the input.

As before, $\mathrm{mix}_{\mathbf{1}_d,\mathbf{0}_d}(\mathbf{s}) = \mathbf{s}$ is Boolean for every coalition, so Lemmas C.18 and C.19 give

$$\max_{\mathbf{s}\in\mathbb{B}^d} |v(\mathbf{s}) - v_G(\mathbf{s})| \leq e^{-\beta/2} =: \eta,$$

where v is the value function of f and v_G that of the weighted voting game. Let φ_i and φ_i^G be the corresponding Shapley values. Lemma C.11 with $k = 1$ gives $|\varphi_i - \varphi_i^G| \le 2\eta$, and, as in Equation (28), $d!\,\varphi_i^G = SS^*(G, i)$.

Query the oracle to obtain $\hat{\varphi}_i$ with $|\hat{\varphi}_i - \varphi_i| \le 1/(8 \cdot d!)$ and output $\lfloor d!\,\hat{\varphi}_i \rceil$. Then

$$\left|d!\,\hat{\varphi}_i - SS^*(G, i)\right| \le d!\,|\hat{\varphi}_i - \varphi_i| + d!\,|\varphi_i - \varphi_i^G| \le \tfrac{1}{8} + 2 \cdot d!\,e^{-\beta/2}.$$

It remains to check that $2 \cdot d!\,e^{-\beta/2} \le \frac{1}{8}$, that is, $\beta \ge 2\ln(16 \cdot d!)$. Using $\ln d! \le d \ln d$ and $\ln d = (\ln 2)\log_2 d \le 0.7\lceil \log_2 d \rceil$,

$$2\ln(16 \cdot d!) \le 2\ln 16 + 2d \ln d \le 5.6 + 1.4\, d\lceil \log_2 d \rceil \le \beta.$$

Hence $|d!\,\hat{\varphi}_i - SS^*(G, i)| \le 1/4 < 1/2$, and rounding recovers the integer $SS^*(G, i)$. With Lemma C.10, this proves Proposition C.20. □

Remark C.21 (On the Embedding). The only non-linearity of the construction is the softmax; the aggregation $\mathbf{w}^\mathsf{T}\mathbf{z}$ is carried out by the affine embedding ι, exactly as a patch embedding aggregates the pixels of a patch before any attention is applied. Two tokens suffice, and hardness for $\mathcal{F}_{1\mathrm{Attn}}$ implies hardness for every class containing it, in particular for single-layer transformer blocks with more tokens, more heads, residual connections, or a subsequent MLP.

C.4.5 Consequences

Theorem 4.2 (Restated). *Computing the Shapley value φ_i of the baseline value function is #P-hard, already for Boolean $\mathbf{x}, \mathbf{y}$ and already for networks f that are (i) a single hidden layer of* two *ReLU neurons, or (ii) a* single *softmax attention layer over two tokens.*

Proof. Case *(i)* is Proposition C.15, which is stronger than the statement above, since it holds already when φ_i is required only up to an additive error of $2^{-\mathcal{O}(d \log d)}$, using $d! \le d^d$. Case *(ii)* is Proposition C.20. Since φ_i may be irrational for these networks, 'computing' φ_i is understood here as computing it up to an additive error of $2^{-\mathcal{O}(d \log d)}$, which is exactly what Proposition C.20 shows to be #P-hard. □

Corollary C.22 (No Polynomial Refinement Schedule). *Assume* $\mathrm{FP} \ne \#\mathrm{P}$*, and let p be a polynomial with $2^{-p(d)} \le 1/(8 \cdot d!)$. Then no algorithm that spends polynomial time per iteration can output bounds $\underline{\varphi_i} \le \varphi_i \le \overline{\varphi_i}$ with $\overline{\varphi_i} - \underline{\varphi_i} \le 2^{-p(d)}$ within a number of iterations polynomial in d and $|f|$, not even when restricted to the two-neuron networks of Proposition C.15 or the single-head networks of Proposition C.20.*

Proof. Suppose such an algorithm existed for one of the two classes, and let p be a polynomial with $2^{-p(d)} \le 1/(8 \cdot d!)$, for instance $p(d) := d^2 + 3$. Running it to completion takes polynomial time, and the midpoint $(\underline{\varphi_i} + \overline{\varphi_i})/2$ of the returned interval approximates φ_i to within $2^{-p(d)-1}$. This solves $\textsc{Shapley}(\mathcal{F}, \delta)$ with $\delta(d) \le 1/(8 \cdot d!)$ in polynomial time, which by Propositions C.15 and C.20 places a #P-hard function in FP, contradicting $\mathrm{FP} \ne \#\mathrm{P}$. □

Corollary C.23 (Hardness of the Spectral Quantities). *For the networks of Proposition C.15, computing the Fourier coefficient $\widehat{v}(\mathbf{0}_d) = m_v$, the spectral energy $\|v\|_2^2$, or the faithfulness R^2 of a sparse Fourier explanation is #P-hard.*

Proof. Let $G = (\mathbf{w}; q)$ be a weighted voting game and write

$$N_G := \left|\{S \subseteq [d] : \textstyle\sum_{i \in S} \mathbf{w}_i \ge q\}\right|, \qquad p_G := 2^{-d} N_G.$$

Computing N_G is #P-hard, as it counts the solutions of a knapsack instance (Valiant, 1979; Dyer, 2003). We may assume $0 < N_G < 2^d$: the case $N_G = 2^d$ holds exactly when the empty coalition wins and the case $N_G = 0$ exactly when the grand coalition loses, both decidable by one evaluation of v_G.

Take $f := f_{\mathbf{w},q}$, $\mathbf{x} := \mathbf{1}_d$, $\mathbf{y} := \mathbf{0}_d$ as in the proof of Proposition C.15, so that $v = v_G$ is $\{0,1\}$-valued on $\mathbb{B}^d$. Then $v(\mathbf{s})^2 = v(\mathbf{s})$ for every coalition, and therefore

$$m_v = \widehat{v}(\mathbf{0}_d) = 2^{-d}\sum_{\mathbf{s}\in\mathbb{B}^d} v(\mathbf{s}) = p_G, \qquad \|v\|_2^2 = 2^{-d}\sum_{\mathbf{s}\in\mathbb{B}^d} v(\mathbf{s})^2 = p_G.$$

Either quantity yields $N_G = 2^d p_G$ by one multiplication, which is polynomial since p_G is a rational with denominator 2^d.

For the faithfulness, take the empty surrogate, $K = 0$, so that $h \equiv 0$ and $\|h - v\|_2^2 = \|v\|_2^2 = p_G$. Its denominator is $\|v - m_v\|_2^2 = \|v\|_2^2 - m_v^2 = p_G(1-p_G)$, which is strictly positive by the assumption above, so the definition of R^2 gives

$$R^2 = 1 - \frac{p_G}{p_G(1-p_G)} = 1 - \frac{1}{1-p_G}, \qquad \text{hence} \qquad p_G = 1 - \frac{1}{1-R^2},$$

and N_G is again recovered in polynomial time. Each of the three quantities is therefore #P-hard to compute. □

Remark C.24 (The Variance Alone Is Ambiguous). The reduction above does not go through for the variance $V = \|v\|_2^2 - m_v^2$ on its own, since $V = p_G(1-p_G)$ determines p_G only up to the substitution $p_G \mapsto 1 - p_G$. Computing V together with either m_v or $\|v\|_2^2$ resolves the ambiguity. Hence, the inputs of Corollary 3.4, which requires bounds on $\|v\|_2^2$ and $\widehat{v}(\mathbf{0}_d)$, are #P-hard to compute exactly by Corollary C.23.

Remark C.25 (Bounded Non-Linearity Helps Robustness Verification). For completeness, we record the folklore argument behind the following contrast: a network with few ReLU neurons has few activation regions, so a verifier searching the space of activation regions has little to search, while a decomposition into additive pieces may still need to partition the 2^d coalitions into exponentially many pieces, since VERIDAP searches the space of coalitions rather than the space of activation regions. Let f be a ReLU network with k ReLU neurons and let $[\underline{\mathbf{x}}, \overline{\mathbf{x}}]$ be a box. Fix an activation pattern $\pi \in \mathbb{B}^k$. Processing the neurons in topological order, substituting $\mathrm{ReLU}(a)$ by a where π says active and by 0 where it says inactive makes every pre-activation an affine function of the input, and f itself affine. The inputs consistent with π form the polyhedron cut out by the k resulting linear inequalities intersected with $[\underline{\mathbf{x}}, \overline{\mathbf{x}}]$, so maximising f over it is a linear program. Enumerating all 2^k patterns therefore decides local robustness in time $\mathcal{O}(2^k \mathrm{poly}(|f|))$, which is polynomial whenever $k = \mathcal{O}(\log |f|)$, for every input dimension d. Proposition C.15 shows that the same restriction, at $k = 2$, leaves exact attribution #P-hard.

D NETWORKS AND DATASETS

This section describes the datasets and networks used in the experiments of Section 6. Table 5 summarises the datasets and neural networks used in our experiments. All tabular datasets are taken from the UCI ML Repository (Kelly et al., 2023), which provides concise descriptions of each dataset; `MNIST` (LeCun et al., 1998), `GTSRB` (Stallkamp et al., 2011), and `CIFAR10` (Krizhevsky, 2009) are the standard image classification benchmarks.

We provide additional details on Table 5. The performance column reports the test metric used during training: accuracy for most classification tasks, balanced accuracy for the class-imbalanced ones (`German`, `Default`, `BreastC`, `Dropout`, `Anneal`, `Sonar`, `NHANES I`, `Polish`, and `COIL2000`), and the root mean square error (RMSE) for the regression task.

The ✳, ■, ●, and ▲ columns of Table 5 report the performance of a standard (naturally trained) network ✳ and three certified-training networks ■, ●, and ▲ that were manually selected from the tightness–accuracy Pareto fronts visualised in Figure 7. Concretely, ▲ provides the tightest value function bounds over $\mathbb{B}^d$ while maintaining clean performance above that of a linear model. Conversely, ■ provides the least tight value function bounds while providing the highest clean performance on the Pareto front. The additional network ●, which we only use in the appendix, is situated on the Pareto front between ▲ and ■. The `CIFAR10` ViT, which we take from a public benchmark, is IBP-trained (Gowal et al., 2018) and provides competitive test set accuracy for its architecture. Therefore, we denote it as ■. All ✳ networks were obtained from separate, standard hyperparameter tuning with `Optuna` (Akiba et al., 2019), optimising only clean performance.

Table 5: **Datasets and Networks Used in Our Experiments.** 'Architecture' lists each network as its sequence of layers, omitting input normalisation and output denormalisation. FC_k denotes a ReLU-activated fully-connected layer of k neurons, and Linear_k denotes a fully-connected layer without activation. Out_m denotes the linear output layer of dimension m. $\text{Conv}_c(k_1 \times k_2)^s$ denotes a convolution layer with c channels, a window size of $k_1 \times k_2$, a stride of s (one if omitted), and one-pixel zero padding on each side. BN, ReLU, and Clip denote batch normalisation, a ReLU activation layer, and a clipping layer with a data-dependent threshold. $\text{PatchEmbed}_m(p \times p)$, VitEmbed, Attention_h, GlobalAvgPool, and SkipConnection denote a patch embedding onto m-dimensional $p \times p$ tokens, a vision-transformer embedding that adds the class token and learned positional embeddings, multi-head attention with h heads and batch normalisation before it, token averaging, and a residual connection. On `Mushroom`, the Pareto front of certified training only contains a single network, so that no ● or ▲ network is available, as denoted by '–' in this table. Our `CIFAR10` vision transformer (ViT) (Dosovitskiy et al., 2021) is a certified-training network from the literature (Brix et al., 2023), which does not provide additional networks.

				Performance			
Input dim.	**Dataset**	**Architecture**	**Task**	**Stand. ✳**	**Cert. ■**	**Cert. ●**	**Cert. ▲**
16	`Obesity`	FC_{32}, FC_{32}, FC_{32}, Out_7	Classification	94.1 %	93.8 %	92.0 %	78.7 %
20	`German`	FC_{32}, FC_{32}, Out_2	Classification	75.0 %	75.7 %	73.7 %	72.1 %
22	`Mushroom`	FC_{16}, FC_{16}, FC_{16}, Out_2	Classification	100 %	100 %	–	–
23	`Default`	FC_{32}, FC_{32}, Out_2	Classification	71.4 %	71.6 %	71.0 %	69.8 %
25	`Auto`	FC_{32}, FC_{32}, Out_1	Regression	0.83	0.79	0.82	1.08
27	`Steel`	FC_{32}, FC_{32}, Out_7	Classification	75.2 %	77.2 %	75.9 %	70.7 %
30	`BreastC`	FC_{32}, FC_{32}, Out_2	Classification	98.6 %	98.6 %	95.7 %	94.8 %
36	`Dropout`	FC_{32}, FC_{32}, Out_2	Classification	85.2 %	86.8 %	86.3 %	85.8 %
38	`Anneal`	FC_{32}, FC_{32}, Out_2	Classification	95.2 %	97.6 %	97.2 %	94.7 %
60	`Sonar`	FC_{32}, FC_{32}, Out_2	Classification	94.1 %	97.1 %	88.2 %	82.4 %
64	`Polish`	Clip, FC_{32}, FC_{32}, Out_2	Classification	87.3 %	85.9 %	82.5 %	81.0 %
79	`NHANES I`	FC_{32}, FC_{32}, Out_2	Classification	78.5 %	79.3 %	79.1 %	79.0 %
85	`COIL2000`	FC_{256}, FC_{256}, Out_2	Classification	67.9 %	72.4 %	71.9 %	70.8 %
784	`MNIST`	CNN[†]	Classification	99.63 %	99.53 %	98.57 %	95.53 %
3072	`GTSRB`	CNN[‡]	Classification	97.73 %	99.66 %	99.47 %	96.96 %
3072	`CIFAR10`	ViT[*]	Classification	–	62.21 %	–	–

[†]CNN (Shi et al., 2021): $\text{Conv}_{64}(3 \times 3)$, BN, ReLU, $\text{Conv}_{64}(3 \times 3)$, BN, ReLU, $\text{Conv}_{128}(3 \times 3)^2$, BN, ReLU, $\text{Conv}_{128}(3 \times 3)$, BN, ReLU, $\text{Conv}_{128}(3 \times 3)$, BN, ReLU, Flatten, Linear_{512}, BN, ReLU, Out_{10}.
[‡]CNN (Shi et al., 2021): $\text{Conv}_{64}(3 \times 3)$, BN, ReLU, $\text{Conv}_{64}(3 \times 3)$, BN, ReLU, $\text{Conv}_{128}(3 \times 3)^2$, BN, ReLU, $\text{Conv}_{128}(3 \times 3)$, BN, ReLU, $\text{Conv}_{128}(3 \times 3)$, BN, ReLU, Flatten, Linear_{512}, BN, ReLU, Out_{43}.
[*]ViT (Brix et al., 2023): $\text{PatchEmbed}_{48}(8 \times 8)$, VitEmbed, Attention_3, SkipConnection, BN, FC_{96}, Linear_{48}, SkipConnection, Attention_3, SkipConnection, BN, FC_{96}, Linear_{48}, SkipConnection, Attention_3, SkipConnection, BN, FC_{96}, Linear_{48}, SkipConnection, GlobalAvgPool, BN, Out_{10}.

E Additional Details on Certified Training

We describe the hyperparameter tuning approach and the certified-training methods used in Section 5 in more detail.

Hyperparameter Tuning. In their evaluation, Kaulen et al. (2026) compare certified-training methods by Pareto fronts of clean and certified accuracy. For our explanation setting, we instead tune clean validation performance[1] against the *relative bound width* as a proxy for explanation runtime. For a sampled training input $\mathbf{x}_k$ and a fixed baseline input $\mathbf{y}$, we use the baseline value function $\mathbf{s} \mapsto f(\mathbf{s} \odot \mathbf{x}_k + (\mathbf{1}_d - \mathbf{s}) \odot \mathbf{y})$. We apply CROWN-IBP (Zhang et al., 2020) to this value function over the initial coalition piece $[\mathbf{0}_d, \mathbf{1}_d]$ and compute the relative bound width as

$$\frac{\sum_{k \in \mathcal{J}} w_k}{\sum_{k \in \mathcal{J}} |f(\mathbf{x}_k) - f(\mathbf{y})|},$$

where $\mathcal{J}$ indexes the sampled training inputs and $w_k := \max_{\mathbf{s} \in [\mathbf{0}_d, \mathbf{1}_d]} \overline{v}_k(\mathbf{s}) - \min_{\mathbf{s} \in [\mathbf{0}_d, \mathbf{1}_d]} \underline{v}_k(\mathbf{s})$ is the width of the concrete bounds that the affine bounds $\underline{v}_k, \overline{v}_k$ on the value function of sample k imply on the initial piece. While certified training can tighten these bounds, relative bound width does not directly measure the runtime of VeriDAP. The tabular searches in Figure 7 instead use the marginal value function with the same normalisation. We vary the training method and its hyperparameters to obtain networks on the Pareto front of validation performance and relative bound width.

Robust Optimisation. As one certified-training method, interval bound propagation (IBP) trains classifiers by minimising an upper bound on adversarial loss (Gowal et al., 2018). Adversarial training starts from the robust optimisation formulation of Madry et al. (2018):

$$\min_{\theta} \; \mathbb{E}_{(x,y) \sim \mathbb{P}_{\text{train}}} \left[\max_{x' \in \mathcal{B}(x)} \ell(f_\theta(x'), y) \right],$$

where θ denotes the network parameters, $\mathbb{P}_{\text{train}}$ the distribution of labelled training samples (x, y), $\mathcal{B}(x)$ the permitted perturbations of x, ℓ the cross-entropy loss, and $f_\theta(x')$ the network logits at x'. Under an ℓ_∞ threat model, the perturbations lie within a chosen radius of x and inside the valid input domain. Under an ℓ_0 threat model, the perturbations instead change at most k feature locations within their valid ranges (Chiang et al., 2020; Shapira & Drachsler-Cohen, 2026).

Interval Bound Propagation. To bound the adversarial loss, IBP propagates intervals through each network layer over an input box (Gowal et al., 2018). Let $z_\theta(x', y) := (f_\theta(x')_y - f_\theta(x')_j)_j$ be the vector of *true-class margins* indexed by class j, with a zero true-class component. IBP computes $\underline{z}_\theta(x, y)$, a componentwise lower bound on these margins over $\mathcal{B}(x)$. Applying cross-entropy to the negated lower margins gives the *verified loss* $\ell(-\underline{z}_\theta(x, y), y)$, which provides an upper bound on the worst-case classification loss. The base IBP method trains on the verified loss, so minimising it can reward both robustness and tighter interval bounds.

Convex-Combination Loss. To interpolate between adversarial and verified losses, De Palma et al. (2024) use convex combinations (CC) of adversarial margins and their IBP lower bounds in CC-IBP:

$$\mathcal{L}_{\text{CC}}(\theta; x, y) := \ell\left(-\left[(1-\alpha) z_\theta(x_{\text{adv}}, y) + \alpha\, \underline{z}_\theta(x, y)\right], y\right),$$

where $\alpha \in [0, 1]$ is the mixing coefficient and $x_{\text{adv}} \in \mathcal{B}(x)$ is an adversarial example. The objective equals the adversarial-example loss when $\alpha = 0$ and the IBP verified loss when $\alpha = 1$.

Other Certified-Training Methods. The multi-task learning (MTL) loss MTL-IBP interpolates between the two scalar losses rather than their margins (De Palma et al., 2024). Small adversarial bounding regions (SABR) apply IBP to a smaller ℓ_∞ box selected near an adversarial example (Müller et al., 2023). Because SABR bounds only this smaller box during training, its loss need not upper-bound the worst-case loss over the full perturbation set. For ℓ_0 perturbations, ℓ_0-IBP instead bounds sparse changes exactly through the first affine layer before propagating intervals through the remaining layers (Chiang et al., 2020; Shapira & Drachsler-Cohen, 2026). In our experiments, IBP, CC-IBP, MTL-IBP, and SABR use ℓ_∞ perturbations, while ℓ_0-IBP uses ℓ_0 perturbations. For regression, our implementations replace classification margins by prediction errors and cross-entropy by squared loss. Figure 7 shows that the methods yield different tradeoffs between validation performance and relative bound width for vision and tabular networks.

[1] Performance on a held-out subset of the training set, called *validation* set.

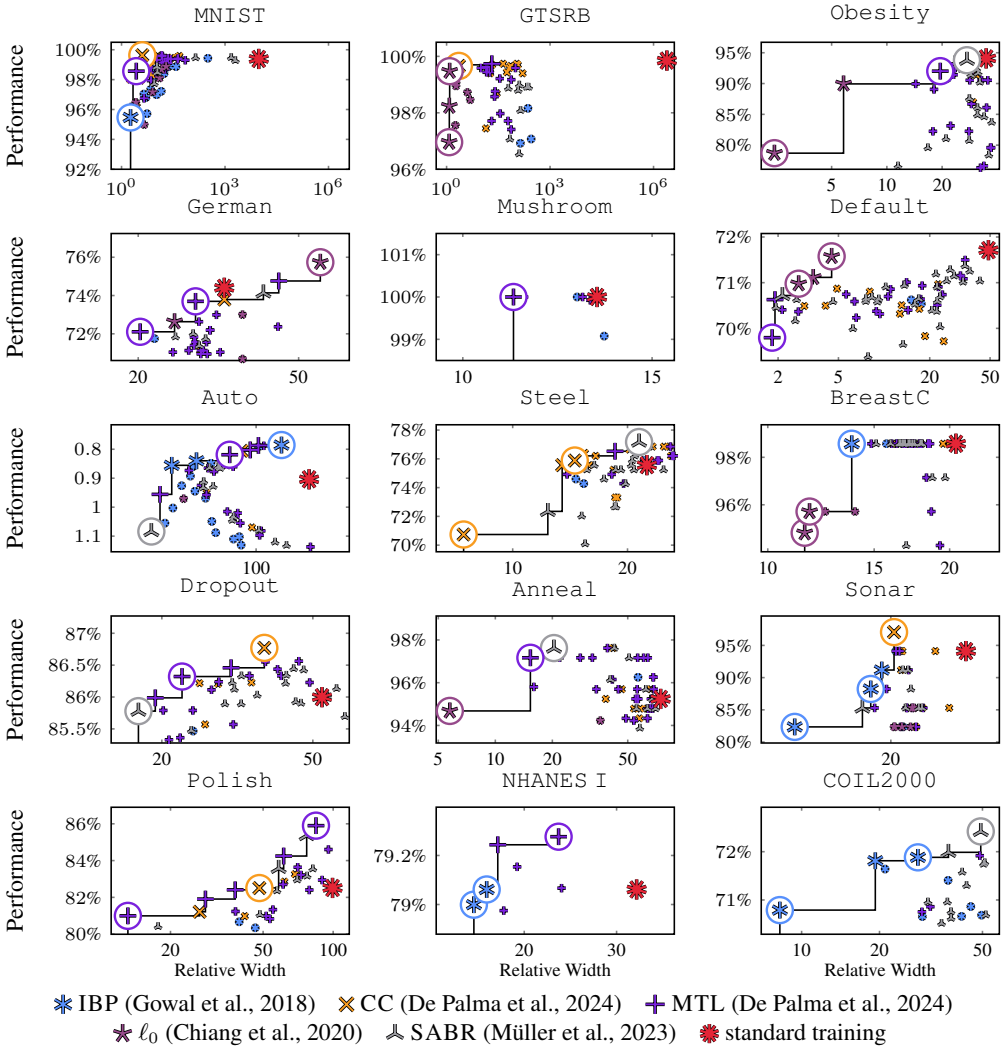


Figure 7: **Certified-Training Pareto Fronts.** Each plot shows performance against relative bound width (lower is better), computed with CROWN-IBP. We normalise the summed bound widths by the summed absolute differences between full- and empty-coalition values. The black step liner traces the certified-training front. The selected networks ■, ●, and ▲ are circled in white. `MNIST` uses the zero baseline, `GTSRB` the training-mean baseline, and the tabular datasets a marginal background of 100 examples; all report *validation* performance. Classification axes are percentages (balanced accuracy on imbalanced tabular datasets); `Auto` shows RMSE (lower is better). Some tabular panels focus on selected networks and clip other points outside their axis limits. The `MNIST` front is that of Figure 4, on validation rather than test accuracy.

Table 6: **SII Tightness for the Marginal Value Function.** Median tightness of the order-2 SII bounds VERIDAP reaches within 600 s for the marginal value function, as in Table 1.

$\lvert\mathbb{B}^d\rvert$	Dataset	✳	■	▲
$2^{16}{}_{>10^4}$	Obesity	✓	✓	✓
$2^{20}{}_{>10^6}$	German	✓	0.7 %	✓
$2^{22}{}_{>10^6}$	Mushroom	✓	✓	*
$2^{23}{}_{>10^6}$	Default	✓	✓	✓
$2^{25}{}_{>10^7}$	Auto	2.4 %	2.0 %	0.9 %
$2^{27}{}_{>10^8}$	Steel	48 %	34 %	0.5 %
$2^{30}{}_{>10^9}$	BreastC	84 %	29 %	16 %
$2^{38}{}_{>10^{11}}$	Anneal	66 %	10 %	6.1 %

* Pareto front only contains a single network.

Table 7: **Marginal SII Runtime.** Median runtime of the baseline enumeration approach (EXACT) and VERIDAP for computing the exact order-2 SIIs of the marginal value function. '–' denotes not computing exact SIIs within the timeout of 600 s. Bold marks the fastest method and all runtimes faster than EXACT.

$\lvert\mathbb{B}^d\rvert$	Dataset	EXACT	✳	■	▲
$2^{16}{}_{>10^4}$	Obesity	**5s**	10s	10s	10s
$2^{20}{}_{>10^6}$	German	118s	**30s**	–	**30s**
$2^{22}{}_{>10^6}$	Mushroom	573s	**60s**	**56s**	*
$2^{23}{}_{>10^6}$	Default	–	**233s**	**221s**	**10s**

F ADDITIONAL DETAILS ON OUR EXPERIMENTAL SETUP

This section fixes the concrete settings with which we run VERIDAP in the experiments of Section 6 and Appendix G.

For the baseline value function, we use a zero baseline for the tabular datasets and MNIST, and the dataset mean as baseline for CIFAR10 and GTSRB. The marginal value function averages over 100 samples of background data. In every run, we instantiate ABP with CROWN-IBP (Zhang et al., 2020) and process pieces in batches of $b = 1\,024$ on the GPU, following Xu et al. (2021). For the II experiments, we report the bounds reached within 600 s per run, except for our ViT experiment, which we run without a timeout. An II run converges when every target bound has width at most 1×10^{-4} or 1×10^{-4} times the attributed output $|v(\mathbf{1}_d) - v(\mathbf{0}_d)|$, whichever threshold is larger. For the SFE certifications of Appendix G.2, we bound the Fourier coefficients and the residual in separate stages, each with a timeout of 600 s. The SFE runs use an absolute tolerance of 1×10^{-4} and a relative tolerance of 1×10^{-6}. We explain five input samples per network and dataset and report median runtimes and tightness. The samples of each dataset are drawn once with a fixed per-dataset seed, so that a dataset's networks explain the same rows.

Exactness. Unlike in theoretical analysis, all computation in floating-point arithmetic is only exact up to a numerical precision. In our experiments, we consider results to be exact whose bounds would be indiscernible for the human eye when the plotted in a figure. This is the right level of precision for interpretations, as figures are the primary means in which interpretations are consumed. While numerical precision and floating-point error is a valid concern for neural network verification in safety-critical settings, it is not a practical concern when computing exact explanations. Notably, also estimators and the exact enumeration baseline are subject to floating-point error. Nonetheless, our implementation offers options for increasing numerical precision. If further precision is required, our approach is equally applicable for use with floating-point sound bound propagation frameworks.

G ADDITIONAL EXPERIMENTAL RESULTS

This section reports additional experimental results on VERIDAP and on benchmarking estimators.

G.1 INTERACTION INDICES

We complement the results of Section 6.1 with results on the marginal value function in Tables 6 and 7, the FSII (Tsai et al., 2023) in Tables 8 and 9, third-order interactions in Tables 10 and 11, and complementary results for GTSRB CNNs for superpixels from image segmentation in Table 12.

Table 8: **FSII Tightness.** Median tightness of the order-2 FSII bounds VERIDAP reaches within 600 s for the baseline value function, as in Table 1, where $\mathbf{s} \in \mathbb{B}^d$ with $\|\mathbf{s}\|_1 = 2$.

$\lvert\mathbb{B}^d\rvert$	Tabular Dataset	Network ✳	Network ■	Network ▲
$2^{16}{}_{>10^4}$	Obesity	✓	✓	0.1 %
$2^{20}{}_{>10^6}$	German	✓	0.6 %	✓
$2^{22}{}_{>10^6}$	Mushroom	✓	✓	*
$2^{23}{}_{>10^6}$	Default	✓	✓	✓
$2^{25}{}_{>10^7}$	Auto	✓	✓	✓
$2^{27}{}_{>10^8}$	Steel	0.2 %	✓	✓
$2^{30}{}_{>10^9}$	BreastC	8.6 %	5.9 %	4.0 %
$2^{36}{}_{>10^{10}}$	Dropout	2.9 %	2.5 %	2.5 %
$2^{38}{}_{>10^{11}}$	Anneal	✓	✓	✓

* Pareto front only contains a single network.

Table 9: **FSII Runtime.** Median runtime of VERIDAP for computing the exact order-2 FSIIs of the baseline value function. '–' denotes not computing exact FSIIs within the timeout of 600 s.

$\lvert\mathbb{B}^d\rvert$	Tabular Dataset	Network ✳	Network ■	Network ▲
$2^{16}{}_{>10^4}$	Obesity	6s	7s	–
$2^{20}{}_{>10^6}$	German	8s	–	8s
$2^{22}{}_{>10^6}$	Mushroom	8s	7s	*
$2^{23}{}_{>10^6}$	Default	10s	9s	3s
$2^{25}{}_{>10^7}$	Auto	82s	87s	52s
$2^{27}{}_{>10^8}$	Steel	–	501s	48s
$2^{38}{}_{>10^{11}}$	Anneal	66s	82s	70s

Table 10: **Order-3 SII Tightness.** Median tightness of the SII bounds of orders 1 to 3 that VERIDAP reaches within 600 s for the baseline value function, as in Table 1. 'Targets' is the number of coalitions of size 1 to 3 for which the SII is bounded.

$\lvert\mathbb{B}^d\rvert$	Targets	Dataset	✳
$2^{12}{}_{>10^3}$	298	Adult	✓
$2^{16}{}_{>10^4}$	696	Obesity	✓
$2^{20}{}_{>10^6}$	1 350	German	✓
$2^{22}{}_{>10^6}$	1 793	Mushroom	✓
$2^{23}{}_{>10^6}$	2 047	Default	0.2 %
$2^{25}{}_{>10^7}$	2 625	Auto	1.9 %
$2^{27}{}_{>10^8}$	3 303	Steel	3.7 %
$2^{30}{}_{>10^9}$	4 525	BreastC	9.3 %
$2^{36}{}_{>10^{10}}$	7 806	Dropout	✗
$2^{38}{}_{>10^{11}}$	9 177	Anneal	28 %

Table 11: **Order-3 SII Runtime.** Median runtime of the baseline enumeration approach (EXACT) and VERIDAP for computing the exact SIIs of orders 1 to 3 of the baseline value function with the dataset mean as baseline. 'Targets' is the number of coalitions of size 1 to 3 for which they are computed. '–' denotes not computing exact SIIs within the timeout of 600 s.

$\lvert\mathbb{B}^d\rvert$	Targets	Dataset	EXACT	Ours ✳
$2^{12}{}_{>10^3}$	298	Adult	**1s**	5s
$2^{16}{}_{>10^4}$	696	Obesity	25s	**6s**
$2^{20}{}_{>10^6}$	1 350	German	–	**25s**
$2^{22}{}_{>10^6}$	1 793	Mushroom	–	**53s**

G.2 CERTIFYING SPARSE FOURIER EXPLANATIONS

This section reports the certified faithfulness of the SFEs discussed briefly in Section 6.1. We explain the tabular MLPs and the MNIST networks of Section 6 with PROXYSPEX (Butler et al., 2025). PROXYSPEX fits an explainable proxy model to samples of $v(\mathbf{s})$ and reports the coalitions that carry 95 % of the energy of the proxy's Fourier transform. PROXYSPEX thereby targets a faithfulness of 0.95. VERIDAP *certifies* this target when its lower bound on R^2 reaches 0.95, and *disproves* the target when its upper bound falls below 0.95. We use the marginal value function of Appendix A.2 over 100 background rows for the tabular datasets and the baseline value function for MNIST. As in Section 6, we explain five samples per dataset and network and report the median lower and upper faithfulness bounds.

Given PROXYSPEX coalitions, VERIDAP bounds the faithfulness of the explanation as in Corollary 3.4. In two separate runs, it first bounds the reported Fourier coefficients and then bounds the SFE residual, each with a timeout of 600 s.

Tabular Datasets. Table 13 shows that VERIDAP decides the faithfulness target for over 75% of the explanations. Concretely, it certifies the target for 34 of the 66 explanations and disproves it for 16. Where VERIDAP disproves the target, PROXYSPEX misses it by at most 0.03: the frequencies that carry 95 % of the proxy's energy still explain at least 92 % of the variance of the network's value

Table 12: **VERIDAP SII Tightness for `GTSRB`.** We report the tightness of order-2 SII bounds for the certified `GTSRB` CNNs over 23 to 25 superpixels and the mean-baseline value function, as in Table 1. We consider values below 0.1 % to be *exact* (✓) and values above 100 % to be *vacuous* (✗). Each cell is the median over five images, where an image is not attempted for more superpixels once its tightness exceeds 50 % for fewer superpixels. The timeout for each run in this table is 600 s.

`GTSRB`	**Network**	
$\lvert\mathbb{B}^d\rvert$	■	▲
$2^{23}{}_{>10^6}$	43 %	4.9 %
$2^{24}{}_{>10^7}$	76 %	7.6 %
$2^{25}{}_{>10^7}$	16 %	2.0 %

Table 13: **SFE Faithfulness on Tabular Datasets.** Median certified bounds on the faithfulness R^2 of PROXYSPEX explanations that VERIDAP reaches within 600 s per stage, for the marginal value function. Every network has a column for the median lower bound $\underline{R^2}$ and one for the median upper bound $\overline{R^2}$, each over the samples of its row. A value in bold green with a check mark spans both columns where the two bounds agree to three decimals. A lower bound of at least 0.95 certifies the faithfulness target of PROXYSPEX, and an upper bound below 0.95 disproves it.

		Network					
		✳		■		▲	
$\lvert\mathbb{B}^d\rvert$	**Dataset**	$\underline{R^2}$	$\overline{R^2}$	$\underline{R^2}$	$\overline{R^2}$	$\underline{R^2}$	$\overline{R^2}$
$2^{16}{}_{>10^4}$	`Obesity`	**0.981 ✓**		**0.980 ✓**		**0.965 ✓**	
$2^{20}{}_{>10^6}$	`German`	0.952	0.953	0.914	0.963	0.953	0.954
$2^{22}{}_{>10^6}$	`Mushroom`	**0.945 ✓**		**0.948 ✓**			*
$2^{23}{}_{>10^6}$	`Default`	**0.938 ✓**		0.938	0.946	0.967	0.968
$2^{25}{}_{>10^7}$	`Auto`	0.926	0.958	0.931	0.964	0.943	0.960

✓ Exact. * Pareto front only contains a single network.

function. This means that even if the target is not reached, VERIDAP proves a high degree of sparsity in the Fourier transform. Regarding runtime, certified training accelerates the certification as it does for the SII in Table 7: on `Default`, both stages converge in a median of 13 s for ▲, compared to 491 s for ✳.

`MNIST`. Table 14 shows that certified training extends the reach of VERIDAP to SFEs as it does to SIIs in Table 1. For the two networks from certified training, across meshes up to 12×12 (a search space of 2^{144} coalitions), VERIDAP disproves the target for 21 of the 36 explanations and certifies it for one.

Vision Transformer. For the `CIFAR10` vision transformer of Figure 1, we bound the faithfulness of the explanations of two images, segmented by hand into 23 superpixels, for the baseline value function with the dataset mean as baseline, computing in double precision. The bounds converge within 48 min per image to a faithfulness of 0.9506 and 0.9399, so that VERIDAP certifies the target for the first image and disproves it for the second.

G.3 BENCHMARKING ESTIMATORS

This section describes the estimator benchmark of Section 6.2 and reports its results for all games.

Games. The benchmark contains the games of our experiments for which the order-2 SII bounds of VERIDAP have half-widths of at most 1 % of $|v(\mathbf{1}_d) - v(\mathbf{0}_d)|$. These are 153 games: 118 games of tabular MLPs, 33 games of `MNIST` CNNs, and the 2 games of the `CIFAR10` ViT, one for each of the two hand-segmented images. We select games individually, so a network can contribute games

Table 14: **SFE Faithfulness on `MNIST`.** Median certified bounds on the faithfulness R^2 of PROXYSPEX explanations that VERIDAP reaches within 600 s per stage, for the baseline value function and the two networks from certified training, as in Table 13.

	Network			
	■		▲	
$\lvert\mathbb{B}^d\rvert$	$\underline{R^2}$	$\overline{R^2}$	$\underline{R^2}$	$\overline{R^2}$
$2^{49}{>}10^{14}$	0.908	0.932	0.938	0.952
$2^{64}{>}10^{19}$	0.905	0.922	0.928	0.932
$2^{81}{>}10^{24}$		‡	0.914	0.948
$2^{100}{>}10^{30}$		‡	0.885	0.956
$2^{121}{>}10^{36}$		‡	0.876	0.985
$2^{144}{>}10^{43}$		‡	0.923	0.935

‡ Not attempted.

even if its median tightness in Table 1 exceeds 1 %. The targets of a game are all coalitions of one and two players.

Estimators. We run PROXYSHAP, KERNELSHAPIQ, and PROXYSPEX from SHAPIQ 1.5.1 at 10 budgets per game, spaced logarithmically from $d + 1$ to 30 000 coalitions, with 10 seeds each. PROXYSPEX returns a sparse set of interactions, with a median of 2 % of the targets, and returns none in 8 % of the runs. Following the convention of SHAPIQ, we count an interaction that PROXYSPEX does not return as zero. KERNELSHAPIQ solves a weighted least-squares problem over the sampled coalitions, and we omit the runs in which this problem is ill-posed. The problem is exactly singular in 10 % of the runs and numerically singular in 32 %, because fewer distinct coalitions than pairs of players receive a non-zero weight. This leaves 57 % of the runs.

Optimistic Error. Let $\hat{\mathbf{e}}$ be an estimate and let $\underline{\mathbf{e}}$ and $\overline{\mathbf{e}}$ be the lower and upper VERIDAP bounds, where every vector has one entry per target. The optimistic error of $\hat{\mathbf{e}}$ is $\|\mathbf{r}\|_2/|v(\mathbf{1}_d) - v(\mathbf{0}_d)|$, where $\mathbf{r} = \max(\mathbf{0}, \underline{\mathbf{e}} - \hat{\mathbf{e}}, \hat{\mathbf{e}} - \overline{\mathbf{e}})$ is taken element-wise. By the triangle inequality, the error of $\hat{\mathbf{e}}$ with respect to the exact values lies between $\|\mathbf{r}\|_2$ and $\|\mathbf{r}\|_2 + \|\overline{\mathbf{e}} - \underline{\mathbf{e}}\|_2$. Therefore, the bounds determine an error up to a factor of two when its optimistic error exceeds their relative ℓ_2 width. At 30 000 coalitions, this is the case in 65 % of the games for PROXYSHAP, 83 % for KERNELSHAPIQ, and 94 % for PROXYSPEX.

Results. Figure 8 summarises the optimistic errors of each domain, and Table 15 reports them per network at 30 000 coalitions. At this budget, PROXYSHAP reaches the lowest median optimistic error in every domain, with 0.9 % for the tabular MLPs, 1.5 % for the `MNIST` CNNs, and 9.4 % for the ViT. Beyond $10d$ coalitions, the median optimistic error of PROXYSPEX decreases by less than 16 % in every domain. This suggests that the interactions PROXYSPEX omits dominate its error. On the `MNIST` ▲, the optimistic error of KERNELSHAPIQ grows with the number of players from 6.5 % on the 7×7 mesh to 248 % on the 15×15 mesh, compared to 0.44 % and 5.4 % for PROXYSHAP. Across the seeds, the median range of the optimistic error is 24 % of the median error for PROXYSHAP, 16 % for KERNELSHAPIQ, and 6 % for PROXYSPEX.

G.4 ADDITIONAL VISUALISATIONS

We provide additional visualisations of the VERIDAP results computed in our experiments. Figure 9 visualises the tight SII bounds computed by VERIDAP for the standard network and two certified-training networks. Figure 10 visualises the certified PROXYSPEX results of the same networks. Figure 11 provides an enlarged version of Figure 1, and Figure 12 provides the same visualisation for a second `CIFAR10` sample.

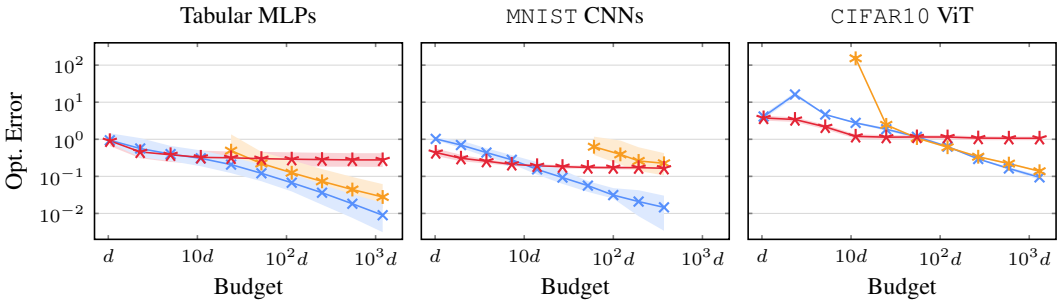


Figure 8: **Estimator Error per Domain.** We plot the optimistic error of order-2 SII estimates against the number of coalitions an estimator evaluates, divided by the number of players d, for KERNELSHAPIQ, PROXYSHAP, and PROXYSPEX. Each game contributes its median over the seeds. We omit budgets at which fewer than half the games of a domain have well-posed KERNELSHAPIQ runs.

Table 15: **Estimator Error per Network.** We report the optimistic errors in percent at 30 000 coalitions. For each network, we take the median over seeds per game and then the median over its games. The networks are standard ✳ or certifiably trained □○△. An entry of 0 means that the estimates lie within the bounds, and '–' marks a network without games.

		PROXYSHAP				KERNELSHAPIQ				PROXYSPEX			
Dataset	d	✳	□	○	△	✳	□	○	△	✳	□	○	△
Tabular, baseline value function													
Default	23	1.8	0.53	0.82	0.17	6	3.1	5.7	2.4	36	18	27	0.32
Auto	25	0.83	0.99	0.61	0.26	2.4	2.7	1.7	0.71	34	38	23	16
Steel	27	2.7	0.91	1.2	2.6	9.4	2.2	2.8	8.4	105	41	39	94
Dropout	36	–	–	0.82	–	–	–	2.5	–	–	–	36	–
Anneal	38	1.4	1.8	2.1	4.2	7.3	3.2	3.5	10	28	21	32	26
NHANES I	79	16	3.6	2	2.1	35	10	5.3	6.3	132	35	23	26
Tabular, marginal value function													
Default	23	1.7	0.5	0.13	0.01	3	1.4	1.9	0.59	33	23	20	15
Auto	25	–	0	0	0	–	0	0	0.13	–	12	7.9	8.8
Steel	27	–	–	–	0.043	–	–	–	0.99	–	–	–	27
$\text{MNIST}_{7\times7}$	49	–	0	–	0.44	–	5.7	–	6.5	–	18	–	18
$\text{MNIST}_{8\times8}$	64	–	0.16	–	1.4	–	11	–	10	–	16	–	15
$\text{MNIST}_{9\times9}$	81	–	0.027	–	1.3	–	14	–	16	–	17	–	19
$\text{MNIST}_{10\times10}$	100	–	–	–	2.7	–	–	–	29	–	–	–	15
$\text{MNIST}_{11\times11}$	121	–	–	–	2.1	–	–	–	33	–	–	–	17
$\text{MNIST}_{12\times12}$	144	–	–	–	2.2	–	–	–	39	–	–	–	16
$\text{MNIST}_{13\times13}$	169	–	–	–	2.5	–	–	–	61	–	–	–	17
$\text{MNIST}_{14\times14}$	196	–	–	–	4	–	–	–	126	–	–	–	14
$\text{MNIST}_{15\times15}$	225	–	–	–	5.4	–	–	–	248	–	–	–	15
CIFAR10 ViT	23	–	9.4	–	–	–	14	–	–	–	106	–	–

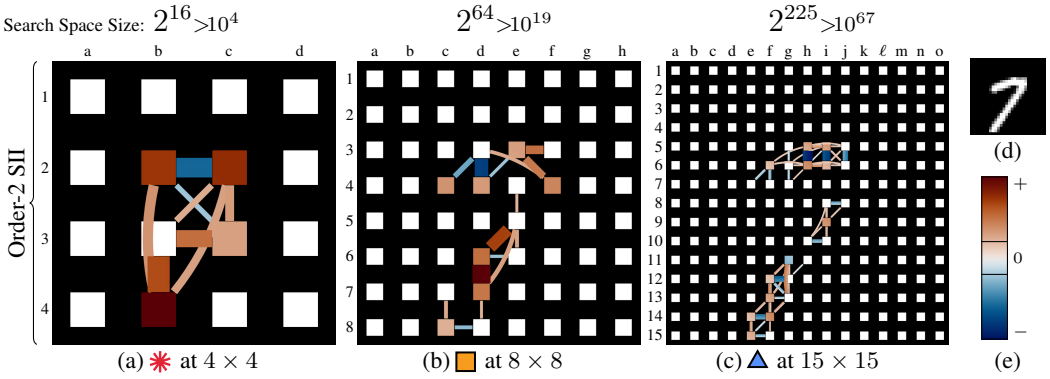


Figure 9: **Shapley Interaction Index Bounds on `MNIST`.** We plot the bounds on the order-2 SII that VERIDAP computes for a standard `MNIST` network ✳ in (a) and for two certified-training networks ■ and ▲ in (b) and (c), respectively. The SII attributes the prediction for the image of a seven in (d) to a mesh of superpixels. Each mesh cell shows the bounds on the SII of its superpixel, and each arc between two cells shows the bounds on the SII of that pair. Both contain a colour gradient from the lower bound to the upper bound, read by the colour bar in (e). We normalise all bounds by the largest absolute bound of each network and hide the coalitions whose bounds stay within 20 % of zero in magnitude, the band marked in (e).

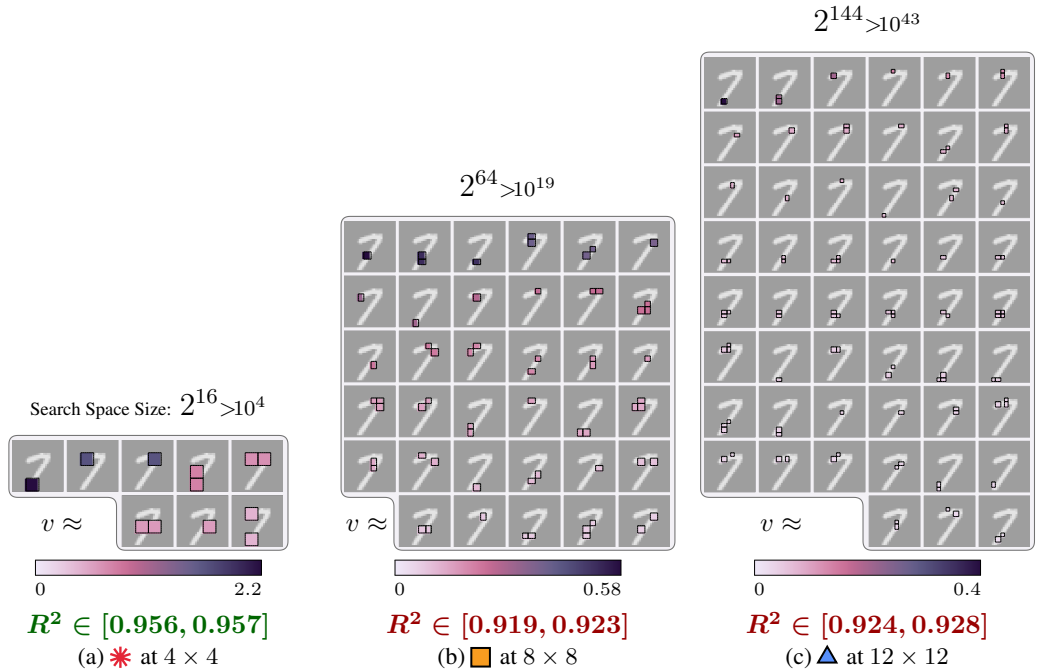
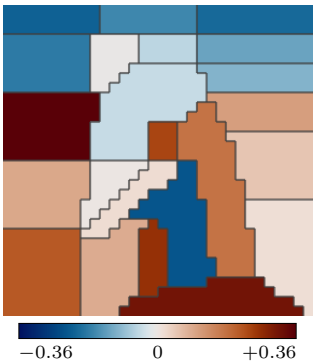


Figure 10: **Sparse Fourier Explanations on `MNIST`.** We show the SFEs that PROXYSPEX (Butler et al., 2025) reports for the image of a seven in Figure 9(d). The explanations are for the standard network ✳ on a 4 × 4 mesh in (a) and for the certified-training networks ■ and ▲ on an 8 × 8 and a 12 × 12 mesh in (b) and (c), respectively.

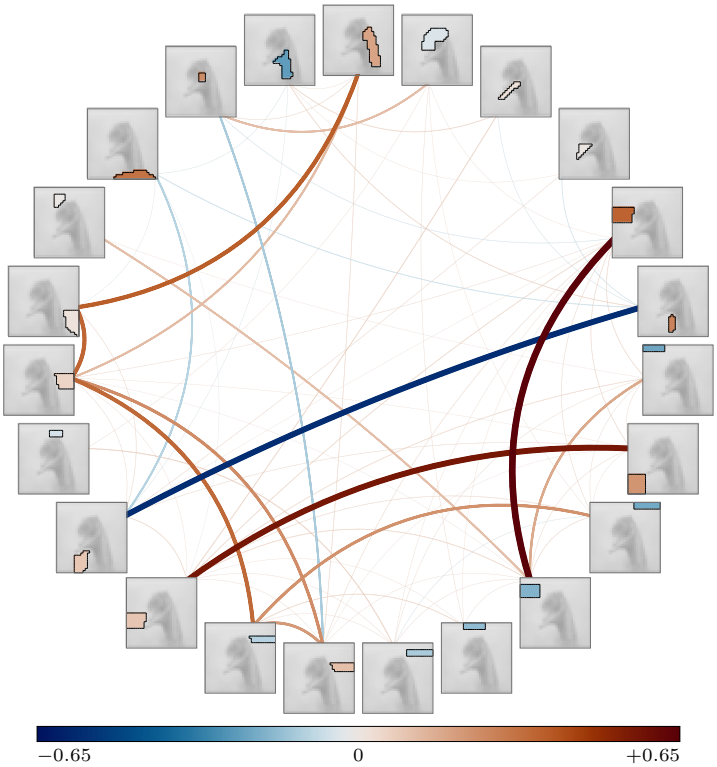


Class: **Bird**

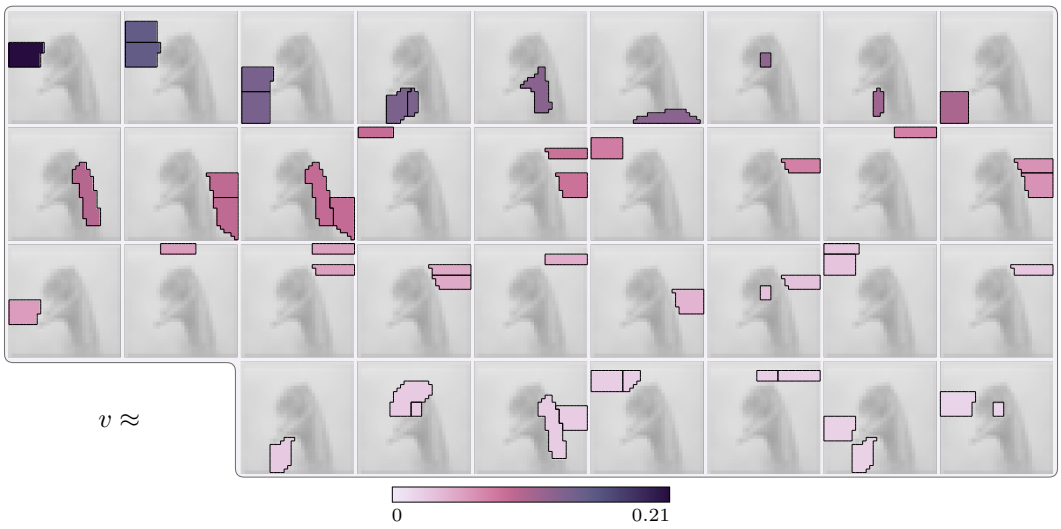
(a) Image



(b) Exact SHAP



(c) Order-2 SII



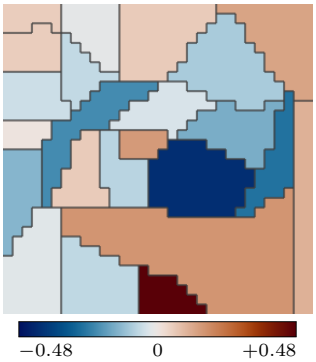
(d) PROXYSPEX, $R^2 \geq 0.950$

Figure 11: **Explaining a Vision Transformer on `CIFAR10`.** Enlarged version of Figure 1.

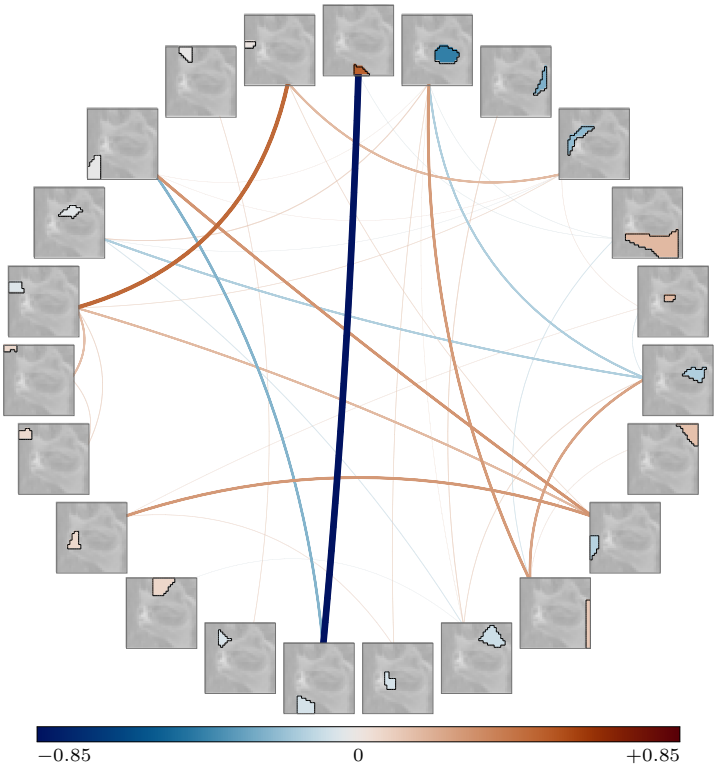


Class: **Bird**

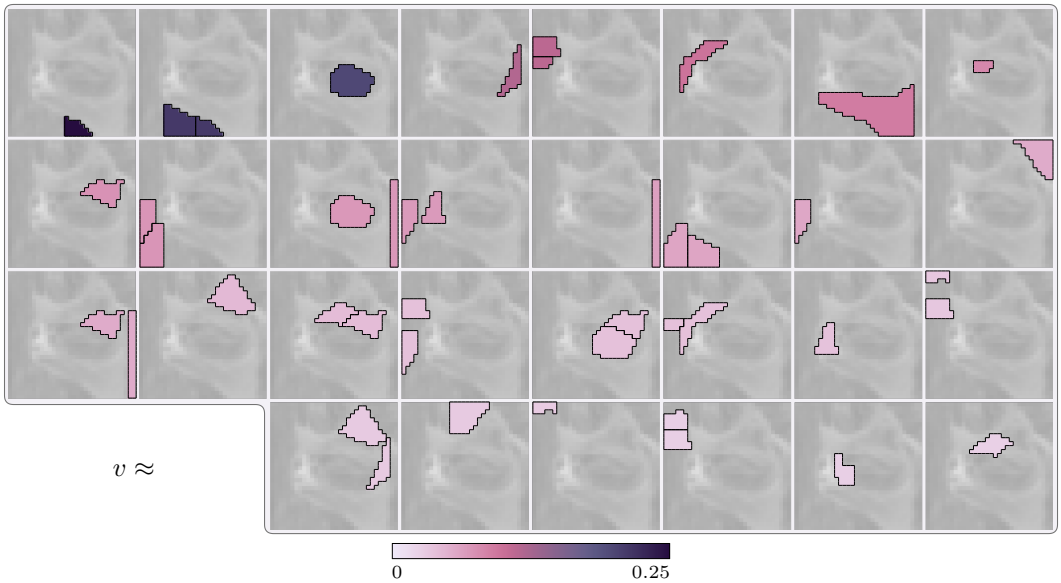
(a) Image



(b) Exact SHAP



(c) Order-2 SII



(d) PROXYSPEX, $R^2 \geq 0.939$

Figure 12: **Explaining a Vision Transformer on `CIFAR10`.** This figure provides the explanations as in Figures 1 and 11 for a second `CIFAR10` test set image.