Probabilistic Stability Guarantees for Feature Attributions

Helen Jin*, Anton Xue*, Weiqiu You, Surbhi Goel, and Eric Wong
Department of Computer and Information Science, University of Pennsylvania

April 16, 2025

Abstract

Stability guarantees are an emerging tool to evaluate feature attributions, but existing certification methods rely on specialized architectures and yield conservative bounds that are not useful in practice. To address this gap, we introduce soft stability and propose a model-agnostic, sample-efficient certification algorithm that offers nontrivial, practically interpretable guarantees. Interestingly, we show that mild smoothing enhances soft stability without incurring the accuracy degradation observed in existing smoothing-based stability certifications. To explain this phenomenon, we leverage techniques from Boolean function analysis to characterize and provide insights into the behavior of smoothed classifiers. Lastly, we demonstrate the improvement of soft stability over hard stability through experiments on vision and language tasks with various feature attribution methods.

1 Introduction

Powerful machine learning models are increasingly deployed in practice. However, their opacity presents a major challenge in being adopted in high-stake domains, where transparent explanations are needed in decision-making. In healthcare, for instance, doctors require insights into the diagnostic steps to trust the model and integrate them into clinical practice effectively [28]. Similarly, in the legal domain, attorneys must ensure that decisions reached with the assistance of models meet stringent judicial standards [45].

There is much interest in explaining the behavior of complex models. One popular class of explanation methods are *feature attributions* [34, 44], which aim to select the input features most important to a model's prediction. However, many explanations are *unstable*, as illustrated in Figure 1: additionally including a few features may change the model's prediction. Such instability suggests that the explanation may be unreliable [40, 54, 59]. This phenomenon has motivated efforts to quantify how model predictions vary with explanations, including the effects of adding or removing features [47, 55] and the influence of the selection's shape [20, 46]. However, most existing works focus on empirical measures [3], with limited formal, mathematical guarantees on robustness.

To address this gap, prior work in Xue et al. [57] considers *stability* as a formal certification framework for robust explanations. In particular, a *stable explanation* is one where adding any small number of features does not alter the model's prediction, up to some maximum tolerance. However, finding this tolerance is non-trivial: for an arbitrary model, one must exhaustively enumerate and check all possible perturbations in a computationally intractable manner. To overcome this, Xue et al. [57] apply adversarial robustness techniques [12, 30] to construct *smoothed classifiers*, which have mathematical properties for efficiently and non-trivially lower-bounding the maximum tolerance. While this is a first step towards certifiably robust explanations, such notions of tolerance are coarse, and smoothing-based certificates are often conservative.

In this work, we introduce *soft stability*, a new variant of stability defined in contrast to *hard stability* [57]. As illustrated in Figure 2, hard stability certifies whether *all* small perturbations to an explanation yield the same

*Equal contribution.

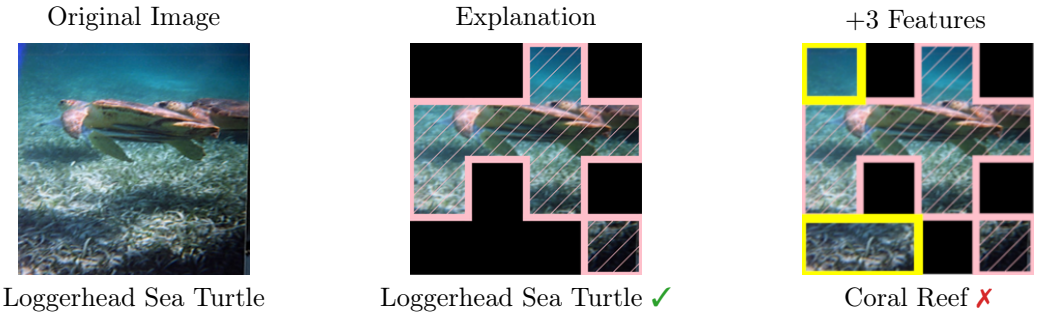

Figure 1: **An unstable explanation.** Given an input image (left), the LIME explanation method [44] identifies features (middle, in pink) that preserve the Vision Transformer model's [15] prediction. However, adding just three more features (right, in yellow) flips the prediction, suggesting an unstable explanation.

prediction, whereas soft stability quantifies *how often* the prediction remains unchanged. Soft stability may thus be interpreted as a probabilistic relaxation of hard stability, which enables a more fine-grained analysis of explanation robustness. Crucially, this shift in perspective allows for model-agnostic applicability and admits efficient certification algorithms that provide stronger guarantees. This work advances our understanding of robust feature-based explanations, and we summarize our contributions below.

Soft Stability is Practical and Certifiable To address the limitations of hard stability, we introduce *soft stability* as a more practical, model-agnostic alternative (Section 2). Unlike hard stability, which relies on heavily smoothed classifiers and yields overly conservative guarantees, soft stability can be certified without modifying the classifier and retains non-vacuous guarantees at large perturbations. We present an efficient certification algorithm and highlight key conceptual differences in Section 3.

Soft Stability Outperforms Hard Stability We empirically evaluate soft stability on vision and language tasks across several attribution methods and compare it against hard stability (Section 5). We show that soft stability provides informative guarantees for larger perturbations, whereas hard stability quickly becomes vacuous. We also show that soft stability can distinguish between explanation methods, with LIME and SHAP achieving better stability rates than gradient-based and randomized baselines.

Mild Smoothing Can Improve Soft Stability Interestingly, although soft stability certification does not require smoothing, we found that *mild* smoothing can improve certified rates while preserving classifier accuracy. We analyze this effect theoretically using Boolean function analysis and develop a novel monotone basis to explain how smoothing influences stability. We present our theoretical results in Section 4 and our empirical validations in Section 5.

2 Background and Overview

Feature attributions are widely used in explainability due to their simplicity and generality, but they are not without drawbacks. In this section, we first give an overview of feature attributions. We then discuss the existing work on hard stability and introduce the notion of soft stability.

2.1 Feature Attributions as Explanations

We consider classifiers of the form $f : \mathbb{R}^n \to \mathbb{R}^m$, which map n features to logits representing m classes. A feature attribution method assigns a score α_i to each input feature x_i that indicates its importance to the model prediction $f(x)$. The definition of importance depends on the method. In gradient-based methods [50, 52], α_i typically denotes the gradient at x_i, while in Shapley-based methods [34, 51], it represents

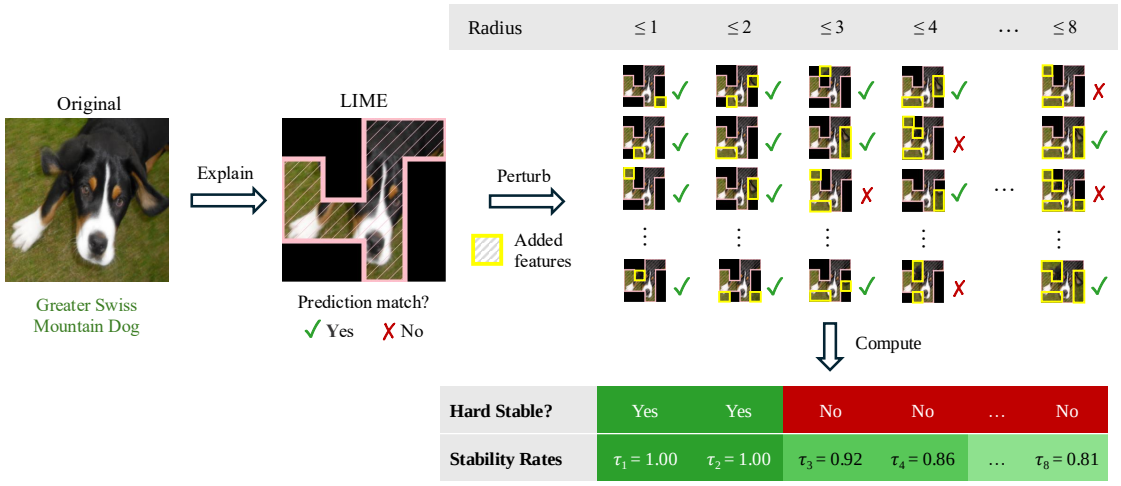

Figure 2: **Soft stability offers a fine-grained measure of robustness.** For Vision Transformer [15], LIME's explanation [44] is only hard stable up to radius $r \leq 2$. In contrast to hard stability's binary decision at each r, soft stability uses the *stability rate* τ_r to quantify the fraction of $\leq r$-sized perturbations that preserve the prediction. This yields a more nuanced view of an explanation's robustness to added features.

the Shapley value of x_i. Attribution scores are typically real-valued, but it is common to binarize them to $\alpha \in \{0,1\}^n$ by selecting the top-k highest-scoring features [39, 44].

2.2 Hard Stability and Soft Stability

Many evaluation metrics exist for binary-valued feature attributions [3]. To compare two attributions $\alpha, \alpha' \in \{0,1\}^n$, it is common to check whether they *induce* the same prediction with respect to a given classifier $f : \mathbb{R}^n \to \mathbb{R}^m$ and input $x \in \mathbb{R}^n$. Let $(x \odot \alpha) \in \mathbb{R}^n$ be the α-masked variant of x, where $\odot$ is the coordinate-wise product of two vectors. We write $f(x \odot \alpha) \cong f(x \odot \alpha')$ to mean that the masked inputs $x \odot \alpha$ and $x \odot \alpha'$ yield the same prediction under f, which holds if:

$$\arg\max_k f(x \odot \alpha)_k = \arg\max_{k'} f(x \odot \alpha')_{k'}, \tag{1}$$

where $k \in \{1, \ldots, m\}$ indexes the predicted class. This form of evaluating feature sets is related to notions of *faithfulness*, *fidelity*, and *consistency*, in the explainability literature [40], but the specific terminology and definition vary by author and source. Furthermore, attribution-masked evaluation is more commonly seen in vision tasks [23], though it is also present in language modeling [35, 58].

It is often desirable that two "similar" attributions induce the same prediction [59]. Although various measures of similarity exist, we are interested in the notion of additive perturbations. Specifically, we treat an additively perturbed attribution α' as one that contains *more information* (features) than α, where the desiderata is that adding more features to a "good quality" α should not easily alter the prediction.

Definition 2.1 (Additive Perturbations)**.** For an attribution α and integer-valued radius $r \geq 0$, define r-additive perturbation set of α as:

$$\Delta_r(\alpha) = \{\alpha' \in \{0,1\}^n : \alpha' \geq \alpha,\ |\alpha' - \alpha| \leq r\}, \tag{2}$$

where $\alpha' \geq \alpha$ iff each $\alpha'_i \geq \alpha_i$ and $|\cdot|$ counts the non-zeros in a binary vector (i.e., the ℓ^0 norm).

Intuitively, $\Delta_r(\alpha)$ represents the set of attributions that are at least as informative as α, differing by at most r features. This allows us to study the robustness of explanations by analyzing whether small modifications in feature selection affect the model's prediction. A natural way to formalize such robustness is through

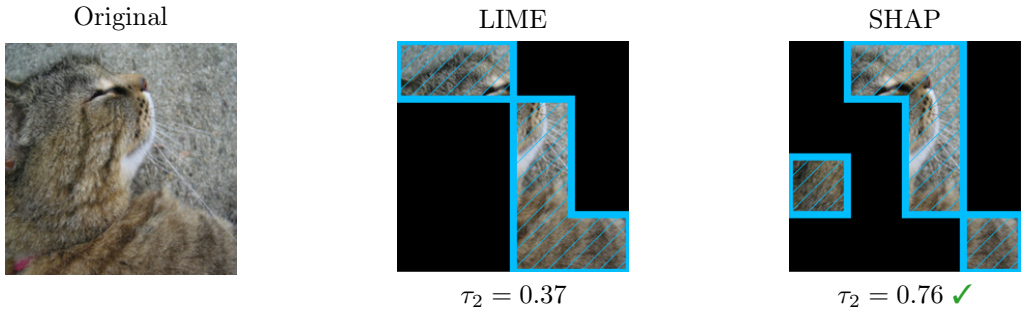

Figure 3: **Similar explanations may have different stability rates.** Despite visual similarities, the explanations generated by LIME [44] (middle) and SHAP [34] (right), both in blue, have very different stability rates at radius $r = 2$. In this example, SHAP's explanation is more soft stable than LIME's.

stability: an attribution α should be considered stable if adding a small number of features does not alter the classifier's decision. We now define *hard stability*, which reinforces this concept strictly.

Definition 2.2 (Hard Stability [1] [57])**.** For a classifier f and input x, the explanation α is *hard-stable* with radius r if: $f(x \odot \alpha') \cong f(x \odot \alpha)$ for all $\alpha' \in \Delta_r$.

However, hard stability is not straightforward to certify, and existing algorithms suffer from costly trade-offs that we later discuss in Section 3.1. This motivates us to investigate *relaxations* that admit efficient certification algorithms while remaining practically useful. In particular, we are motivated by the increasing usage of probabilistic guarantees in domains such as medical imaging [16], drug discovery [6], and autonomous driving [32], which are often formulated in terms of confidence [7, 11]. We thus present a probabilistic relaxation of hard stability, quantified by the *stability rate*, as follows.

Definition 2.3 (Soft Stability)**.** For a classifier f and input x, define the *stability rate* $\tau_r(f, x, \alpha)$ as the probability that the model's prediction remains unchanged when α is perturbed by up to r features:

$$\tau_r(f, x, \alpha) = \Pr_{\alpha' \sim \Delta_r} [f(x \odot \alpha') \cong f(x \odot \alpha)], \quad \text{where } \alpha' \sim \Delta_r \text{ is uniformly sampled.} \tag{3}$$

A higher stability rate τ_r indicates a greater likelihood that a perturbation of at most r features preserves the prediction. In fact, soft stability generalizes hard stability, as the extreme case of $\tau_r = 1$ recovers the hard stability condition. In other words, the binary hard stability certificate is a very loose but valid lower bound on the stability rate. In the context of robustness, soft stability may differ greatly between two explanations that appear similar, as shown in Figure 3, where two explanations that differ at only two features have drastically different stability rates.

3 Certifying Soft Stability

We first discuss the limitations of certifying hard stability, particularly with smoothing-based methods such as the Multiplicative Smoothing (MuS) framework from Xue et al. [57]. We then introduce the Stability Certification Algorithm (SCA) in Theorem 3.1, a model-agnostic and sample-efficient approach for certifying soft stability that yields non-vacuous guarantees across every perturbation radius.

3.1 Challenges in Certifying Hard Stability

Existing approaches to certifying hard stability rely on a classifier's *Lipschitz constant*, which is a measure of function smoothness. While useful for robustness certification [12], the Lipschitz constant is often intractable to compute [53] and challenging to approximate [17, 56]. To address this, Xue et al. [57] derive smoothed

[1] Xue et al. [57] equivalently call this property "incrementally stable" and define "stable" as a stricter property.

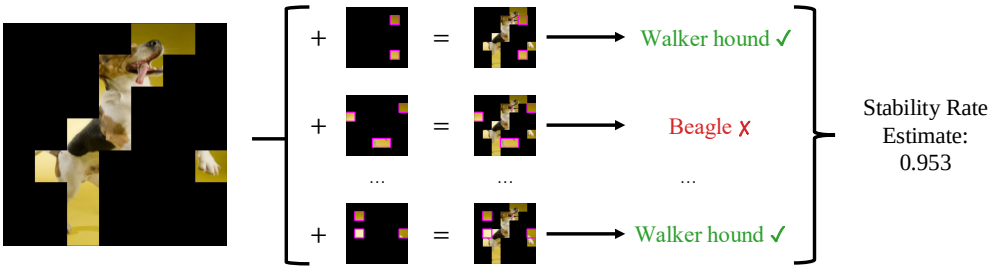

Figure 4: **The Stability Certification Algorithm (SCA).** Given an explanation $\alpha \in \{0,1\}^n$ for a model f and input $x \in \mathbb{R}^n$, we certify the stability rate τ_r as follows. First, uniformly sample $\alpha' \sim \Delta_r(\alpha)$ with replacement. Then, let $\hat{\tau}_r$ be the rate at which the perturbed explanation match the original explanation's prediction, $\hat{\tau}_r = \frac{1}{N}\sum_{\alpha'} \mathbf{1}[f(x \odot \alpha') \cong f(x \odot \alpha)]$, where N is the number of samples. If $N \geq \frac{\log(2/\delta)}{2\varepsilon^2}$, then with probability at least $1-\delta$, the estimator $\hat{\tau}_r$ is accurate to the true stability rate τ_r with error $|\hat{\tau}_r - \tau_r| \leq \varepsilon$.

classifiers with a known Lipschitz constant by construction. Starting with any classifier f, one defines the smoothed classifier $\tilde{f}$ as the expectation over randomly perturbed inputs:

$$\tilde{f}(x) = \frac{1}{N}\big[f(x^{(1)}) + \cdots + f(x^{(N)})\big], \tag{4}$$

where $x^{(1)}, \ldots, x^{(N)} \sim \mathcal{D}(x)$ are sampled perturbations of x. If the distribution and samples are appropriately chosen, then the smoothed classifier $\tilde{f}$ has a Lipschitz constant κ that is explicitly known in expectation. This κ measures sensitivity to input perturbations and, since it is known by construction, allows efficient certification of hard stability on $\tilde{f}$. We present the MuS algorithm from Xue et al. [57] in Definition 4.1 and describe how to extract hard stability guarantees from MuS-smoothed classifiers in Theorem C.1.

Smoothing Has Performance Trade-offs A key limitation of smoothing-based certificates is that the stability guarantees apply to $\tilde{f}$, not the original classifier f. Typically, the smoother the model, the stronger its guarantees (larger certified radii), but this comes at the cost of accuracy. This tension arises because excessive smoothing reduces a model's sensitivity, making it harder to distinguish between classes [5, 22].

Smoothing-based Hard Stability is Conservative Even when smoothing-based certification is feasible, the resulting certified radii are often conservative. The main reason is that this approach depends on a global property, the Lipschitz constant κ, to make guarantees about local perturbations $\alpha' \sim \Delta_r(\alpha)$. In particular, the certified hard stability radius of $\tilde{f}$ scales as $\mathcal{O}(1/\kappa)$, which we elaborate on in Theorem C.1.

3.2 Certifying Soft Stability via Sampling

Unlike hard stability, which requires destructively smoothing the classifier and often yields conservative guarantees, soft stability can be certified efficiently for any classifier. Its key measure, the stability rate τ_r, can be efficiently and reliably estimated via the following algorithm, which we also illustrate in Figure 4.

Theorem 3.1 (Stability Certification Algorithm (SCA)). *Let $N \geq \frac{\log(2/\delta)}{2\varepsilon^2}$ for any $\varepsilon > 0$ and $\delta > 0$. For a classifier f, input x, explanation α, and radius r, define the estimator:*

$$\hat{\tau}_r = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\big[f(x \odot \alpha^{(i)}) \cong f(x \odot \alpha)\big], \quad \textit{where } \alpha^{(1)}, \ldots, \alpha^{(N)} \sim \Delta_r(\alpha) \textit{ are sampled i.i.d.} \tag{5}$$

Then, with probability at least $1-\delta$, the estimator $\hat{\tau}_r$ is accurate to τ_r with error $|\hat{\tau}_r - \tau_r| \leq \varepsilon$.

Proof. The sample complexity N follows from applying Hoeffding's inequality to the mean estimation of independent Bernoulli random variables $X^{(1)}, \ldots, X^{(N)}$, where each $X^{(i)} = \mathbf{1}[f(x \odot \alpha^{(i)}) \cong f(x \odot \alpha)]$. □

Note that because τ_r is a one-dimensional statistics, the sample size N depends only on ε and δ. Moreover, certifying soft stability does not require deriving a smoothed classifier. Unlike hard stability, which applies to the smoothed classifier $\tilde{f}$, soft stability provides robustness guarantees directly on the original classifier f. This eliminates the need for a destructive smoothing process that risks degrading accuracy.

Implementation Details Suppose that $r \leq n - |\alpha|$, then sampling from $\Delta_r(\alpha)$ may be done in two steps: first, sample the perturbation size $k \sim \{0, 1, \ldots, r\}$ with probability $\binom{n-|\alpha|}{k}/|\Delta_r(\alpha)|$, where $|\Delta_r(\alpha)| = \sum_{i=0}^{r} \binom{n-|\alpha|}{i}$; then, uniformly select k among the $n - |\alpha|$ zero positions in α and set them to one. To avoid numerical instability from large binomial coefficients, we use a Gumbel softmax reparametrization [24] to sample in the log probability space.

4 Theoretical Link Between Soft Stability and Smoothing

While smoothing is used to certify hard stability, this is often at a high cost to the smoothed classifier's accuracy. Interestingly, however, we found that a milder variant of the smoothing proposed in [57] can improve soft stability while incurring only a minor accuracy trade-off. We emphasize that the soft stability certification algorithm, SCA (Theorem 3.1), does *not* require smoothing. Rather, mildly smoothing the model can empirically improve stability rates.

To formalize the discussion, we next present multiplicative smoothing (MuS), the algorithm used in hard stability certification. The main idea is to make the smoothed classifier robust to the inclusion and exclusion of features, which is achieved by taking an averaged evaluation over randomly masked (dropped) features.

Definition 4.1 (MuS[2] (Random Masking))**.** For any classifier f and smoothing parameter $\lambda \in [0, 1]$, define the random masking operator M_λ as:

$$M_\lambda f(x) = \mathop{\mathbb{E}}_{z \sim \mathrm{Bern}(\lambda)^n} f(x \odot z), \quad \text{where } z_1, \ldots, z_n \sim \mathrm{Bern}(\lambda) \text{ are i.i.d. samples.} \tag{6}$$

We show in Theorem C.1 how to certify hard stability with MuS-smoothed classifiers. The smoothing parameter λ is the probability that a feature is kept, i.e., each feature is randomly masked (dropped) with probability $1 - \lambda$. Smoothing becomes stronger as λ shrinks: at $\lambda = 1$, no smoothing occurs because $M_1 f(x) = f(x)$; at $\lambda = 1/2$, half the features of $x \odot z$ are zeroed out on average; at $\lambda = 0$, the classifier predicts on an entirely zeroed input because $M_0 f(x) = f(\mathbf{0}_n)$. In the following, we summarize our theoretical results on the soft stability of smoothed classifiers in Section 4.1 and defer technical details to Section 4.2.

4.1 Summary of Theoretical Results

Our main theoretical tooling is Boolean function analysis [41], which studies real-valued functions of Boolean-valued inputs. To connect this with attribution-masked classification: for any classifier $f : \mathbb{R}^n \to \mathbb{R}^m$ and input $x \in \mathbb{R}^n$, define the masked evaluation $f_x(\alpha) = f(x \odot \alpha)$. Such $f_x : \{0, 1\}^n \to \mathbb{R}^m$ is then a Boolean function, for which the random masking (MuS) operator M_λ is well-defined because $M_\lambda f(x \odot \alpha) = M_\lambda f_x(\alpha)$.

To simplify our analysis, we consider a specific form of prediction agreement. Namely, we consider classifiers of the form $f_x : \{0, 1\}^n \to \mathbb{R}$, where for $\alpha' \sim \Delta_r(\alpha)$ let:

$$f_x(\alpha') \cong f_x(\alpha) \quad \text{if} \quad |f_x(\alpha') - f_x(\alpha)| \leq \gamma. \tag{7}$$

This setup, including the decision boundary distance γ, can be derived from a general m-class classifier once the x and α are given. In summary, we establish the following.

Theorem 4.2 (Smoothed Stability, Informal)**.** *Smoothing improves the lower bound on the stability rate by shrinking its gap to* 1 *by a factor of* λ*. Consider any classifier* f_x *and attribution* α *that satisfy Equation* (7)*,*

[2]We use the terms MuS, random masking, and M_λ interchangeably, depending on the context.

and let Q be a quantity that depends on f_x (specifically, its monotone weights of degree $\leq r$), then:

$$1 - \frac{Q}{\gamma} \leq \tau_r(f_x, \alpha) \implies 1 - \frac{\lambda Q}{\gamma} \leq \tau_r(M_\lambda f_x, \alpha). \tag{8}$$

We present the full version in Theorem B.4. Theoretically, smoothing improves the worst-case stability rate by a factor of λ. Empirically, we observe that smoothed classifiers tend to be more stable. Interestingly, we found it challenging to bound the stability rate of M_λ-smoothed classifiers using standard Boolean analytic techniques, such as those presented in widely used references like [41]. This motivated us to develop novel analytic tooling. We describe these challenges and developments next.

4.2 Challenges with Standard Boolean Analytic Tooling and New Techniques

We now describe the challenges encountered with standard Boolean analytic tooling and introduce novel techniques for analyzing the random masking (MuS) operator M_λ. We refer to Appendix A for a more extensive exposition on Boolean function analysis and defer additional technical details to Appendix B.

It is common to study Boolean functions through their Fourier expansion. For any $h : \{0,1\}^n \to \mathbb{R}$, its Fourier expansion exists uniquely as a linear combination over the subsets S of $[n] = \{1, \ldots, n\}$, taking the form:

$$h(\alpha) = \sum_{S \subseteq [n]} \widehat{h}(S)\chi_S(\alpha), \tag{9}$$

where each $\chi_S(\alpha)$ is a Fourier basis function with weight $\widehat{h}(S)$, respectively defined as:

$$\chi_S(\alpha) = \prod_{i \in S} (-1)^{\alpha_i}, \quad \chi_\emptyset(\alpha) = 1, \quad \widehat{h}(S) = \frac{1}{2^n} \sum_{\alpha \in \{0,1\}^n} h(\alpha)\chi_S(\alpha). \tag{10}$$

A key benefit of studying a Boolean function's Fourier expansion is that all the $k = 0, 1, \ldots, n$ degree (order) interactions between input bits are made explicit. For example, the 2-bit conjunction (AND) $h(\alpha_1, \alpha_2) = \alpha_1 \wedge \alpha_2$ is uniquely expressible as:

$$h(\alpha_1, \alpha_2) = \frac{1}{4}\chi_\emptyset(\alpha) - \frac{1}{4}\chi_{\{1\}}(\alpha) - \frac{1}{4}\chi_{\{2\}}(\alpha) + \frac{1}{4}\chi_{\{1,2\}}(\alpha), \tag{11}$$

A common way to study operators on Boolean functions is to examine how they affect each basis function in an expansion. With respect to the standard Fourier basis, the operator M_λ acts as follows.

Theorem 4.3. *For any standard basis function χ_S and smoothing parameter $\lambda \in [0,1]$,*

$$M_\lambda \chi_S(\alpha) = \sum_{T \subseteq S} \lambda^{|T|}(1-\lambda)^{|S-T|}\chi_T(\alpha). \tag{12}$$

For any function $h : \{0,1\}^n \to \mathbb{R}$, its smoothed variant M_λ has expansion

$$M_\lambda h(\alpha) = \sum_{T \subseteq [n]} \widehat{M_\lambda h}(T)\chi_T(\alpha), \quad \textit{where } \widehat{M_\lambda h}(T) = \lambda^{|T|} \sum_{S \supseteq T} (1-\lambda)^{|S-T|}\widehat{h}(S). \tag{13}$$

Intuitively, this expression shows that smoothing redistributes weights from each term S down to all of its subsets $T \subseteq S$, scaled by a binomial decay $\mathsf{Bin}(|S|, \lambda)$. However, this behavior introduces significant complexity in the algebraic manipulations and is distinct from that of other operators commonly studied in literature, which makes it hard to adapt existing techniques for stability analysis. [3]

[3] The prototypical smoothing operator is random flipping: for $0 \leq \rho \leq 1$, define $T_\rho h(\alpha) = \mathbb{E}_{z \sim \mathsf{Bern}(q)^n}[h((\alpha + z) \bmod 2)]$, where $q = (1-\rho)/2$. This point-wise contracts the spectral weight at S via $T_\rho \chi_S(\alpha) = \rho^{|S|}\chi_S(\alpha)$, which is distinct from redistribution.

Although stability results could, in principle, be derived using the standard basis, we introduce the *monotone basis* that yields cleaner analytical expressions. This development is motivated by the fact that the monotone basis is better equipped to describe properties that depend on the inclusion and exclusion of features, such as the additive perturbations from $\Delta_r(\alpha)$ or the random deletions from M_λ. To our knowledge, this is the first application of such a technique for analyzing feature attribution stability.

Definition 4.4 (Monotone Basis)**.** For any subset $T \subseteq [n]$, define its respective monotone basis function as:

$$\mathbf{1}_T(\alpha) = \begin{cases} 1 & \text{if } \alpha_i = 1 \text{ for all } i \in T \text{ (all features of } T \text{ are present)}, \\ 0 & \text{otherwise (any feature of } T \text{ is absent)}. \end{cases} \tag{14}$$

The monotone basis provides a direct encoding of set inclusion, where the earlier example of $h(\alpha_1, \alpha_2) = \alpha_1 \wedge \alpha_2$ is now concisely represented as $h(\alpha) = \mathbf{1}_{\{1,2\}}(\alpha)$. Similar to the standard basis, the monotone basis also admits a unique *monotone expansion* for any $h : \{0,1\}^n \to \mathbb{R}$ of the form:

$$h(\alpha) = \sum_{T \subseteq [n]} \widetilde{h}(T)\mathbf{1}_T(\alpha), \quad \text{where } \widetilde{h}(T) = h(T) - \sum_{S \subsetneq T} \widetilde{h}(S), \quad \widetilde{h}(\emptyset) = h(\mathbf{0}_n), \tag{15}$$

such that $\widetilde{h}(T)$ are the recursively defined monotone weights at each $T \subseteq [n]$, with $h(T)$ being the evaluation of h on the natural $\{0,1\}^n$-valued representation of T. A key property of the monotone basis is that the action of M_λ is now point-wise at each T.

Theorem 4.5. *For any function $h : \{0,1\}^n \to \mathbb{R}$, subset $T \subseteq [n]$, and smoothing parameter $\lambda \in [0,1]$,*

$$\widetilde{M_\lambda h}(T) = \lambda^{|T|}\widetilde{h}(T), \tag{16}$$

where $\widetilde{M_\lambda h}(T)$ and $\widetilde{h}(T)$ are the monotone basis coefficients of $M_\lambda h$ and h at T, respectively.

In the monotone basis, smoothing exponentially decays the weight at each $T \subseteq [n]$ by a factor of $\lambda^{|T|}$. This is algebraically simpler than the redistribution of weights in Theorem 4.3 and aligns more closely with the motifs in existing techniques. As previewed in Theorem 4.2 (full version in Theorem B.4), we use the monotone basis to bound the stability rate of smoothed classifiers, where Q is a value that depends on $\{\widetilde{h}(T) : |T| \leq r\}$, i.e., the monotone weights of degree $\leq r$. We refer to Appendix B for additional details.

5 Experiments

We evaluate soft and hard stability guarantees on vision and language models, examine the effects of smoothing on classifier stability and accuracy, and assess how different feature attribution methods perform under these conditions. Our findings reinforce that soft stability provides significantly larger certificates than those attained from smoothing-based hard stability. Moreover, we show that mild smoothing, defined as $\lambda \geq 0.5$, often improves soft stability while preserving classifier accuracy. We summarize our key findings here and defer full experimental details to Appendix C.

Setup For vision models, we used Vision Transformer (ViT) [15] and ResNet50/18 [21], while for language models, we used RoBERTa[33]. For datasets, we used a 2000-image subset of ImageNet (2 images per class) and six subsets of TweetEval (emoji, emotion, hate, irony, offensive, sentiment), totaling 10653 samples. We segmented the $3 \times 224 \times 224$ images into non-overlapping 16×16 patches, resulting in $n = 196$ features per image. For text, each token was treated as one feature. We used five feature attribution methods: LIME [44], SHAP [34], Integrated Gradients (IntGrad) [52], MFABA [62], and a baseline where random features are selected. We binarized real-valued attributions by selecting the top-25% of features unless stated otherwise. When certifying soft stability with SCA (Theorem 3.1), we used parameters of $\varepsilon = \delta = 0.1$ for a sample size of $N = 150$. Where appropriate, we used 1000 iterations of bootstrap sampling to compute the 95% confidence intervals. For compute, we used NVIDIA GeForce RTX 3090 and NVIDIA RTX A6000 GPUs.

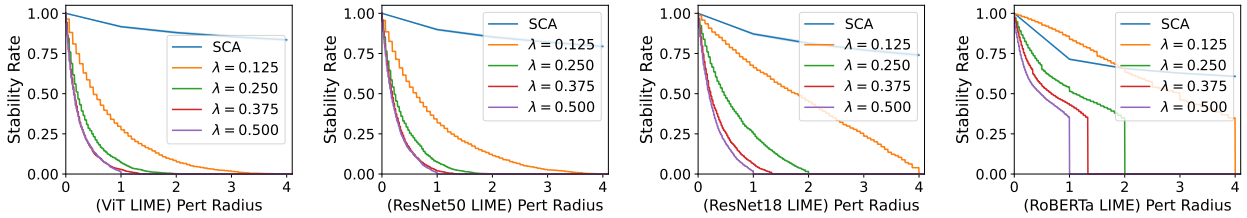

Figure 5: **Soft stability certifies more than hard stability.** Soft stability remains informative even for larger perturbations, whereas smoothing-based hard stability certificates quickly become vacuous. When using MuS with parameter λ, the maximum certifiable radius is only $1/2\lambda$. However, the smaller the λ, the worse the smoothed classifier accuracy, which we show in Figure 8.

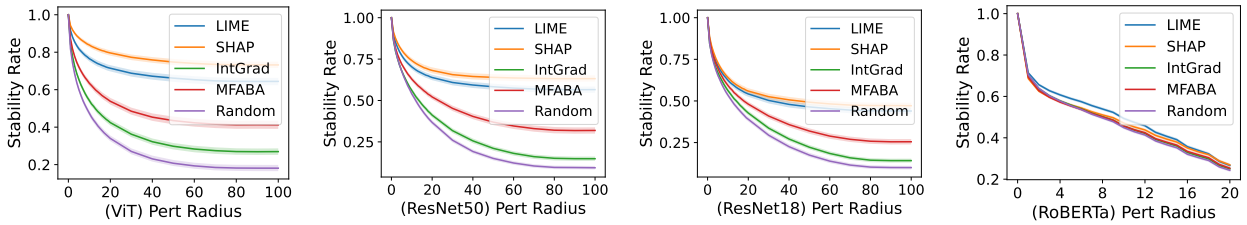

Figure 6: **Soft stability varies across explanation methods.** For vision models, LIME and SHAP yield higher stability rates than gradient-based methods, with all methods outperforming the random baseline. On RoBERTa, however, the methods are less distinguishable. In all cases, SCA-based soft stability certification yields non-trivial guarantees at radii where MuS-based hard stability becomes vacuous.

Question 1: How does soft stability compare to hard stability? To compare MuS-certified hard stability radii (Theorem C.1) with SCA-based stability rates, we plot the following:

$$\text{Stability rate at radius } r = \frac{|\{(x, \alpha) : \text{CertifiedRadius}(M_\lambda f_x, \alpha) \geq r\}|}{\text{Total number of } x\text{'s}}, \tag{17}$$

We present results for LIME in Figure 5 and defer results for other explanation methods to Appendix C.2. Soft stability yields non-trivial certificates at larger perturbations, and this is evident for the vision models at radius $r = 4$. With smoothing parameter λ, the maximum certifiable radius is $1/2\lambda$. However, smaller λ also degrades model accuracy, meaning that these guarantees with respect to a less accurate model.

Question 2: How does soft stability vary across explanation methods? We study how the stability rate varies across different explanation methods in Figure 6. For vision, soft stability can effectively distinguish between different explanation methods: LIME and SHAP yield higher stability rates than gradient-based methods, with all methods outperforming the random baseline. However, this distinction is not clear in our language setting. Because there are only 196 features in the vision setting with each explanation consisting of $0.25 \times 196 = 49$ features, a perturbation of size 100 would consist of over half the available features.

Question 3: How does (mild) smoothing affect soft stability? We study the effects of smoothing on the stability rate, where we focus on the case of mild smoothing, i.e., $\lambda \geq 0.5$. Notably, these λ values do not induce enough smoothing to allow hard stability certification. We show accuracy curves in Figure 7 for MuS-smoothed classifiers at different values of λ, where we used 32 Bernoulli samples to compute the smoothed classifier as in Definition 4.1. For data, we used 200 samples from our subset of ImageNet and 200 samples from TweetEval that had at least 40 tokens. This length-based filter was to ensure that a non-trivial number of additive perturbations was possible, and we used the random baseline to select the attribution. Smoothing generally improves stability, particularly for ResNet50 and ResNet18 compared to ViT. We give a more detailed comparison of different perturbation radii in Appendix C.3.

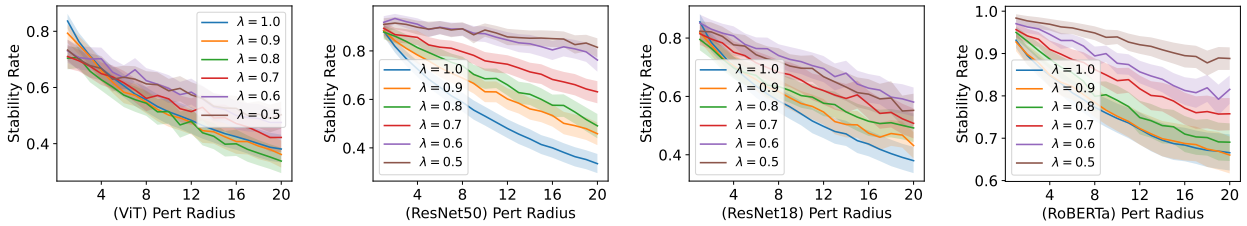

Figure 7: **Mild smoothing ($\lambda \geq 0.5$) can improve soft stability.** For vision, this is most prominent for ResNet50 and ResNet18. While transformer-based models also benefit, RoBERTa improves more than ViT.

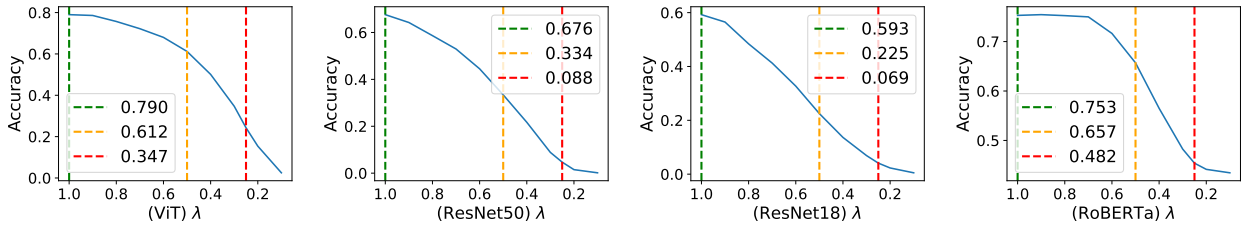

Figure 8: **Mild smoothing ($\lambda \geq 0.5$) preserves classifier accuracy.** We report classification accuracy at three key smoothing levels: ($\lambda = 1.0$, in green) the original, unsmoothed model; ($\lambda = 0.5$, in orange) a mildly smoothed model, the largest λ for which hard stability certificates can be obtained; ($\lambda = 0.25$, in red) a strongly smoothed model, where hard stability only certifies perturbations of up to 2 added features.

Question 4: How accurate are smoothed classifiers? We analyze the impact of smoothing on classifier accuracy in Figure 8 and highlight three key values: the original, unmodified classifier accuracy ($\lambda = 1.0$), the largest smoothing parameter usable in the certification of hard stability ($\lambda = 0.5$), and ($\lambda = 0.25$), the smoothing parameter used in many hard stability experiments of [57]. We used 64 Bernoulli samples to compute the smoothed classifier as in Definition 4.1. Transformer-based models (ViT, RoBERTa) exhibit a more gradual accuracy decline as smoothing intensifies, likely because their training involves random masking.

6 Related Work

Feature-based Explanations Feature attributions have long been used in explainability and remain popular. Early examples include gradient saliency [50], LIME [44], SHAP [34], and Integrated Gradients [52]. More recent works include DIME [37], LAFA [60], CAFE [13], DoRaR [43], MFABA [62], various Shapley value-based methods [51], and methods based on influence functions [8, 29]. Moreover, while feature attributions are commonly associated with vision models, they are also used in language [36] and time series modeling [48]. For surveys on explainability in general, we refer to Milani et al. [38], Schwalbe and Finzel [49]. For explainability in medicine, we refer to Klauschen et al. [28], Patrício et al. [42]. For explainability in law, we refer to [4, 45].

Evaluating Feature Attributions Although feature attributions are popular, their usefulness is often challenged [1, 14, 27]. This is because each attribution method computes importance differently and in ways that may not be faithful to the underlying model behavior [2, 61]. Moreover, theoretical results exist on their fundamental limitations [9]. There is a large number of evaluation metrics for feature attributions [3, 25, 40, 46], in particular for various notions of robustness [18, 26]. Perturbation-based robustness metrics similar to ours include incremental insertion [40] and ranking stability [19]. Masking-based evaluations are also vulnerable to missingness bias, which is more prominent for CNNs than ViTs on vision-based tasks [23].

Certifying Feature Attributions While many empirical metrics exist, there is also growing interest in ensuring that feature attributions are well-behaved through formal guarantees. There exists work on certifying the robustness properties of adding [57] and removing [31] features from an attribution. There is also work on

selecting feature sets that are provably optimal with respect to some metric [10], such as in their ranking [19]. However, the literature on explicit guarantees for feature attributions is still emergent.

7 Discussion

This work aims to make post hoc explanations more reliable by introducing soft stability, a probabilistic and model-agnostic notion of robustness. While prior certification methods tend to be overly conservative or tied to smoothed models, our approach yields stronger, practically useful guarantees. We now discuss some broader implications and promising extensions.

Boolean Function Analysis in Explainability Boolean analytic techniques are well-suited for explainability, as many manipulations are inherently discrete. This makes Boolean function analysis a natural tool for both developing new algorithms and analyzing existing ones. In our case, this approach enabled us to shift away from traditional continuous Lipschitz-based robustness analysis to provide a discrete perspective. Our findings suggest that similar techniques could be valuable in other machine learning tasks, especially those involving voting, aggregation, or other discrete perturbation schemes.

Future Directions An exciting direction for future work is adaptive smoothing, where the smoothing parameter is tuned based on feature importance or model confidence, for example, applying stronger smoothing to uncertain predictions. Another is stability-aware training, where models are explicitly regularized to produce more stable attributions. Exploring connections between stability and generalization may also yield insights, particularly if higher soft stability aligns with better model reliability. Alternative smoothing schemes based on attribution rankings [19] merit investigation as well. Finally, since stability and adversarial robustness both capture sensitivity to perturbations, future work could study when stability naturally emerges as the appropriate robustness notion.

8 Conclusion

We introduce soft stability, a probabilistic relaxation of hard stability that provides a more flexible and efficient way to certify the robustness of feature attributions. Unlike hard stability, soft stability is model-agnostic, sample-efficient, and does not require destructively modifying the classifier. Interestingly, we show that mild smoothing can improve the soft stability certificate of classifiers while incurring only a small cost to accuracy. We study this phenomenon from the perspective of Boolean function analysis and present novel characterizations and techniques that would be of interest to explainability researchers. Furthermore, we validate our theory through experiments on vision and language tasks.

Acknowledgements This research was partially supported by the ARPA-H program on Safe and Explainable AI under the grant D24AC00253-00, by NSF award CCF 2313010, by the AI2050 program at Schmidt Sciences, by an Amazon Research Award Fall 2023, by an OpenAI SuperAlignment grant, and Defense Advanced Research Projects Agency's (DARPA) SciFy program (Agreement No. HR00112520300). The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

[2] Julius Adebayo, Michael Muelly, Harold Abelson, and Been Kim. Post hoc explanations may be ineffective for detecting unknown spurious correlation. In *International Conference on Learning Representations*, 2022.

[3] Chirag Agarwal, Satyapriya Krishna, Eshika Saxena, Martin Pawelczyk, Nari Johnson, Isha Puri, Marinka Zitnik, and Himabindu Lakkaraju. Openxai: Towards a transparent evaluation of model explanations. *Advances in neural information processing systems*, 35:15784–15799, 2022.

[4] Kasun Amarasinghe, Kit T Rodolfa, Hemank Lamba, and Rayid Ghani. Explainable machine learning for public policy: Use cases, gaps, and research directions. *Data & Policy*, 5:e5, 2023.

[5] Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301. PMLR, 2019.

[6] Staffan Arvidsson McShane, Ulf Norinder, Jonathan Alvarsson, Ernst Ahlberg, Lars Carlsson, and Ola Spjuth. Cpsign: conformal prediction for cheminformatics modeling. *Journal of Cheminformatics*, 16(1): 75, 2024.

[7] Pepa Atanasova. A diagnostic study of explainability techniques for text classification. In *Accountable and Explainable Methods for Complex Reasoning over Text*, pages 155–187. Springer, 2024.

[8] Samyadeep Basu, Philip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. *arXiv preprint arXiv:2006.14651*, 2020.

[9] Blair Bilodeau, Natasha Jaques, Pang Wei Koh, and Been Kim. Impossibility theorems for feature attribution. *Proceedings of the National Academy of Sciences*, 121(2):e2304406120, 2024.

[10] Guy Blanc, Jane Lange, and Li-Yang Tan. Provably efficient, succinct, and precise explanations. *Advances in Neural Information Processing Systems*, 34:6129–6141, 2021.

[11] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.

[12] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019.

[13] Adam Dejl, Hamed Ayoobi, Matthew Williams, and Francesca Toni. Cafe: Conflict-aware feature-wise explanations. *arXiv preprint arXiv:2310.20363*, 2023.

[14] Jonathan Dinu, Jeffrey Bigham, and J Zico Kolter. Challenging common interpretability assumptions in feature attribution explanations. *arXiv preprint arXiv:2012.02748*, 2020.

[15] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[16] Jamil Fayyad, Shadi Alijani, and Homayoun Najjaran. Empirical validation of conformal prediction for trustworthy skin lesions classification. *Computer Methods and Programs in Biomedicine*, page 108231, 2024.

[17] Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. *Advances in neural information processing systems*, 32, 2019.

[18] Yuyou Gan, Yuhao Mao, Xuhong Zhang, Shouling Ji, Yuwen Pu, Meng Han, Jianwei Yin, and Ting Wang. "is your explanation stable?" a robustness evaluation framework for feature attribution. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1157–1171, 2022.

[19] Jeremy Goldwasser and Giles Hooker. Provably stable feature rankings with shap and lime. *arXiv preprint arXiv:2401.15800*, 2024.

[20] Peter Hase, Harry Xie, and Mohit Bansal. The out-of-distribution problem in explainability and search methods for feature importance explanations. *Advances in neural information processing systems*, 34: 3650–3666, 2021.

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

[22] Todd Huster, Cho-Yu Jason Chiang, and Ritu Chadha. Limitations of the lipschitz constant as a defense against adversarial examples. In *ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings 18*, pages 16–29. Springer, 2019.

[23] Saachi Jain, Hadi Salman, Eric Wong, Pengchuan Zhang, Vibhav Vineet, Sai Vemprala, and Aleksander Madry. Missingness bias in model debugging. In *International Conference on Learning Representations*, 2022.

[24] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[25] Helen Jin, Shreya Havaldar, Chaehyeon Kim, Anton Xue, Weiqiu You, Helen Qu, Marco Gatti, Daniel Hashimoto, Bhuvnesh Jain, Amin Madani, Masao Sako, Lyle Ungar, and Eric Wong. The fix benchmark: Extracting features interpretable to experts. *arXiv preprint arXiv:2409.13684*, 2024.

[26] Sandesh Kamath, Sankalp Mittal, Amit Deshpande, and Vineeth N Balasubramanian. Rethinking robustness of model attributions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 2688–2696, 2024.

[27] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer, 2019.

[28] Frederick Klauschen, Jonas Dippel, Philipp Keyl, Philipp Jurmeister, Michael Bockmayr, Andreas Mock, Oliver Buchstab, Maximilian Alber, Lukas Ruff, Grégoire Montavon, et al. Toward explainable artificial intelligence for precision pathology. *Annual Review of Pathology: Mechanisms of Disease*, 19(1):541–570, 2024.

[29] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

[30] Alexander J Levine and Soheil Feizi. Improved, deterministic smoothing for l_1 certified robustness. In *International Conference on Machine Learning*, pages 6254–6264. PMLR, 2021.

[31] Chris Lin, Ian Covert, and Su-In Lee. On the robustness of removal-based feature attributions. *Advances in Neural Information Processing Systems*, 36, 2024.

[32] Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 2023.

[33] Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364, 2019.

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

[35] Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329, 2023.

[36] Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. Towards faithful model explanation in nlp: A survey. *Computational Linguistics*, pages 1–67, 2024.

[37] Yiwei Lyu, Paul Pu Liang, Zihao Deng, Ruslan Salakhutdinov, and Louis-Philippe Morency. Dime: Fine-grained interpretations of multimodal models via disentangled local explanations. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 455–467, 2022.

[38] Stephanie Milani, Nicholay Topin, Manuela Veloso, and Fei Fang. Explainable reinforcement learning: A survey and comparative review. *ACM Computing Surveys*, 56(7):1–36, 2024.

[39] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[40] Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice Van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Computing Surveys*, 55 (13s):1–42, 2023.

[41] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[42] Cristiano Patrício, João C Neves, and Luís F Teixeira. Explainable deep learning methods in medical image classification: A survey. *ACM Computing Surveys*, 56(4):1–41, 2023.

[43] Dong Qin, George T Amariucai, Daji Qiao, Yong Guan, and Shen Fu. A comprehensive and reliable feature attribution method: Double-sided remove and reconstruct (dorar). *Neural Networks*, 173:106166, 2024.

[44] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[45] Karen McGregor Richmond, Satya M Muddamsetty, Thomas Gammeltoft-Hansen, Henrik Palmer Olsen, and Thomas B Moeslund. Explainable ai and law: an evidential survey. *Digital Society*, 3(1):1, 2024.

[46] Yao Rong, Tobias Leemann, Vadim Borisov, Gjergji Kasneci, and Enkelejda Kasneci. A consistent and efficient evaluation strategy for attribution methods. In *International Conference on Machine Learning*, pages 18770–18795. PMLR, 2022.

[47] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.

[48] Udo Schlegel, Hiba Arnout, Mennatallah El-Assady, Daniela Oelke, and Daniel A Keim. Towards a rigorous evaluation of xai methods on time series. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4197–4201. IEEE, 2019.

[49] Gesina Schwalbe and Bettina Finzel. A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*, 38(5): 3043–3101, 2024.

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

[51] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International conference on machine learning*, pages 9269–9278. PMLR, 2020.

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

[53] Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

[54] Jorg Wagner, Jan Mathias Kohler, Tobias Gindele, Leon Hetzel, Jakob Thaddaus Wiedemer, and Sven Behnke. Interpretable and fine-grained visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9097–9107, 2019.

[55] Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R Lyu, and Yu-Wing Tai. Towards global explanations of convolutional neural networks with concept attribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8652–8661, 2020.

[56] Anton Xue, Lars Lindemann, Alexander Robey, Hamed Hassani, George J Pappas, and Rajeev Alur. Chordal sparsity for lipschitz constant estimation of deep neural networks. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 3389–3396. IEEE, 2022.

[57] Anton Xue, Rajeev Alur, and Eric Wong. Stability guarantees for feature attributions with multiplicative smoothing. *Advances in Neural Information Processing Systems*, 36, 2024.

[58] Fanghua Ye, Mingming Yang, Jianhui Pang, Longyue Wang, Derek F Wong, Emine Yilmaz, Shuming Shi, and Zhaopeng Tu. Benchmarking llms via uncertainty quantification. *arXiv preprint arXiv:2401.12794*, 2024.

[59] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. *Advances in neural information processing systems*, 32, 2019.

[60] Sheng Zhang, Jin Wang, Haitao Jiang, and Rui Song. Locally aggregated feature attribution on natural language model understanding. *arXiv preprint arXiv:2204.10893*, 2022.

[61] Yilun Zhou, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. Do feature attribution methods correctly attribute features? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9623–9633, 2022.

[62] Zhiyu Zhu, Huaming Chen, Jiayu Zhang, Xinyi Wang, Zhibo Jin, Minhui Xue, Dongxiao Zhu, and Kim-Kwang Raymond Choo. Mfaba: A more faithful and accelerated boundary-based attribution method for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17228–17236, 2024.

A Analysis of Smoothing with Standard Techniques

In this appendix, we analyze the smoothing operator M_λ using classical tools from Boolean function analysis. Specifically, we study how smoothing redistributes the spectral mass of a function by examining its action on standard Fourier basis functions. This sets up the foundation for our later motivation to introduce a more natural basis in Appendix B. First, recall the definition of the random masking-based smoothing operator.

Definition A.1 (MuS [57] (Random Masking))**.** For any classifier $f : \mathbb{R}^n \to \mathbb{R}^m$ and smoothing parameter $\lambda \in [0, 1]$, define the random masking operator M_λ as:

$$M_\lambda f(x) = \mathop{\mathbb{E}}_{z \sim \mathrm{Bern}(\lambda)^n} f(x \odot z), \quad \text{where } z_1, \ldots, z_n \sim \mathrm{Bern}(\lambda) \text{ are i.i.d. samples.} \tag{18}$$

To study M_λ via Boolean function analysis, we fix the input $x \in \mathbb{R}^n$ and view the masked classifier $f_x(\alpha) = f(x \odot \alpha)$ as a Boolean function $f_x : \{0, 1\}^n \to \mathbb{R}^m$. In particular, we have the following identities:

$$M_\lambda f(x \odot \alpha) = M_\lambda f_x(\alpha) = M_\lambda f_{x \odot \alpha}(\mathbf{1}_n). \tag{19}$$

This relation is useful from an explainability perspective because it means that features not selected by α (the x_i where $\alpha_i = 0$) will not be seen by the classifier. In other words, this prevents a form of information leakage when evaluating the informativeness of a feature selection.

A.1 Background on Boolean Function Analysis

A key approach in Boolean function analysis is to study functions of the form $h : \{0, 1\}^n \to \mathbb{R}$ by their unique *Fourier expansion*. This is a linear combination indexed by the subsets $S \subseteq [n]$ of form:

$$h(\alpha) = \sum_{S \subseteq [n]} \widehat{h}(S) \chi_S(\alpha), \tag{20}$$

where each $\chi_S(\alpha)$ is a Fourier basis function, also called the standard basis function, with weight $\widehat{h}(S)$. These quantities are respectively defined as:

$$\chi_S(\alpha) = \prod_{i \in S} (-1)^{\alpha_i}, \quad \chi_\emptyset(\alpha) = 1, \quad \widehat{h}(S) = \frac{1}{2^n} \sum_{\alpha \in \{0,1\}^n} h(\alpha) \chi_S(\alpha). \tag{21}$$

The functions $\chi_S : \{0, 1\}^n \to \{\pm 1\}$ form an orthonormal basis on $\{0, 1\}^n$ in the sense that:

$$\langle \chi_S, \chi_T \rangle = \mathop{\mathbb{E}}_{\alpha \sim \mathsf{Bern}(1/2)^n} [\chi_S(\alpha) \chi_T(\alpha)] = \frac{1}{2^n} \sum_{\alpha \in \{0,1\}^n} \chi_S(\alpha) \chi_T(\alpha) = \begin{cases} 1 & \text{if } S = T, \\ 0 & \text{if } S \neq T. \end{cases} \tag{22}$$

Consequently, all of the 2^n weights $\widehat{h}(S)$ (one for each $S \subseteq [n]$) are uniquely determined by the 2^n values of $h(\alpha)$ (one for each $\alpha \in \{0, 1\}^n$) under the linear relation $\widehat{h}(S) = \langle h, \chi_S \rangle$ as in Equation (21). For example, one can check that the function $h(\alpha_1, \alpha_2) = \alpha_1 \wedge \alpha_2$ is uniquely expressible in this basis as:

$$h(\alpha_1, \alpha_2) = \frac{1}{4} \chi_\emptyset(\alpha) - \frac{1}{4} \chi_{\{1\}}(\alpha) - \frac{1}{4} \chi_{\{2\}}(\alpha) + \frac{1}{4} \chi_{\{1,2\}}(\alpha). \tag{23}$$

We defer to O'Donnell [41] for a more comprehensive introduction to Boolean function analysis.

A.2 Basic Results in the Standard Basis

We now study how smoothing affects stability by analyzing how M_λ transforms Boolean functions in the standard Fourier basis. A common approach is to examine how M_λ acts on each basis function χ_S, and we show that smoothing causes a spectral mass shift from higher-order to lower-order terms.

Lemma A.2. *For any standard basis function χ_S and smoothing parameter $\lambda \in [0,1]$,*

$$M_\lambda \chi_S(\alpha) = \sum_{T \subseteq S} \lambda^{|T|}(1-\lambda)^{|S-T|}\chi_T(\alpha). \tag{24}$$

Proof. We first expand the definition of $\chi_S(\alpha)$ to derive:

$$\begin{aligned} M_\lambda \chi_S(\alpha) &= \mathop{\mathbb{E}}_{z} \prod_{i \in S} (-1)^{\alpha_i z_i} && (25)\\ &= \prod_{i \in S} \mathop{\mathbb{E}}_{z} (-1)^{\alpha_i z_i} && \text{(by independence of } z_1, \ldots, z_n) \\ &= \prod_{i \in S} [(1-\lambda) + \lambda(-1)^{\alpha_i}], && (26) \end{aligned}$$

We then use the distributive property (i.e., expanding products over sums) to rewrite the product $\prod_{i \in S}(\cdots)$ as a summation over $T \subseteq S$ to get

$$\begin{aligned} M_\lambda \chi_S(\alpha) &= \sum_{T \subseteq S} \left(\prod_{j \in S-T} (1-\lambda) \right) \left(\prod_{i \in T} \lambda(-1)^{\alpha_i} \right) && (27)\\ &= \sum_{T \subseteq S} (1-\lambda)^{|S-T|}\lambda^{|T|}\chi_T(\alpha), && (28) \end{aligned}$$

where T acts like an enumeration over the 2^n choices of $z \in \{0,1\}^n$ and recall that $\chi_T(\alpha) = \prod_{i \in T}(\alpha)$. □

In other words, M_λ redistributes the Fourier weight at each basis χ_S over to the $2^{|S|}$ subsets $T \subseteq S$ according to a binomial distribution $\mathsf{Bin}(|S|, \lambda)$. Because this redistribution acts linearly on the input, we can visualize M_λ as a $\mathbb{R}^{2^n \times 2^n}$ upper-triangular matrix whose entries are indexed by $T, S \subseteq [n]$, such that

$$(M_\lambda)_{T,S} = \begin{cases} \lambda^{|T|}(1-\lambda)^{|S-T|} & \text{if } T \subseteq S, \\ 0 & \text{otherwise.} \end{cases} \tag{29}$$

Using the earlier example of $h(\alpha_1, \alpha_2) = \alpha_1 \wedge \alpha_2$, the Fourier coefficients of $M_\lambda h$ may be expressed as:

$$\begin{bmatrix} \widehat{M_\lambda h}(\emptyset) \\ \widehat{M_\lambda h}(\{1\}) \\ \widehat{M_\lambda h}(\{2\}) \\ \widehat{M_\lambda h}(\{1,2\}) \end{bmatrix} = \begin{bmatrix} 1 & (1-\lambda) & (1-\lambda) & (1-\lambda)^2 \\ & \lambda & & \lambda(1-\lambda) \\ & & \lambda & \lambda(1-\lambda) \\ & & & \lambda^2 \end{bmatrix} \begin{bmatrix} \widehat{h}(\emptyset) \\ \widehat{h}(\{1\}) \\ \widehat{h}(\{2\}) \\ \widehat{h}(\{1,2\}) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} (2-\lambda)^2 \\ -\lambda(2-\lambda) \\ -\lambda(2-\lambda) \\ \lambda^2 \end{bmatrix} \tag{30}$$

where recall that $\widehat{h}(S) = 1/4$ for all $S \subseteq \{1,2\}$. For visualization, it is useful to sort the rows and columns of M_λ by inclusion and partition them by degree. Below is an illustrative expansion of $M_\lambda \in \mathbb{R}^{8 \times 8}$ for $n = 3$, sorted by inclusion and partitioned by degree:

$(M_\lambda)_{T,S}$ =

	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$\emptyset$	1	$(1-\lambda)$	$(1-\lambda)$	$(1-\lambda)$	$(1-\lambda)^2$	$(1-\lambda)^2$	$(1-\lambda)^2$	$(1-\lambda)^3$
$\{1\}$		λ			$\lambda(1-\lambda)$	$\lambda(1-\lambda)$		$\lambda(1-\lambda)^2$
$\{2\}$			λ		$\lambda(1-\lambda)$		$\lambda(1-\lambda)$	$\lambda(1-\lambda)^2$
$\{3\}$				λ		$\lambda(1-\lambda)$	$\lambda(1-\lambda)$	$\lambda(1-\lambda)^2$
$\{1,2\}$					λ^2			$\lambda^2(1-\lambda)$
$\{1,3\}$						λ^2		$\lambda^2(1-\lambda)$
$\{2,3\}$							λ^2	$\lambda^2(1-\lambda)$
$\{1,2,3\}$								λ^3

(31)

Because the columns of M_λ sum to 1, we have the identity:

$$\sum_{T\subseteq[n]} \widehat{M_\lambda h}(T) = \sum_{S\subseteq[n]} \widehat{h}(S), \quad \text{for any function } h:\{0,1\}^n \to \mathbb{R}. \tag{32}$$

Moreover, M_λ may be interpreted as a downshift operator in the sense that: for each $T \subseteq [n]$, the Fourier coefficient $\widehat{M_\lambda h}(T)$ depends only on those of $\widehat{h}(S)$ for $S \supseteq T$. The following result gives a more precise characterization of each $\widehat{M_\lambda h}(T)$ in the standard basis.

Lemma A.3. *For any function $h:\{0,1\}^n \to \mathbb{R}$ and smoothing parameter $\lambda \in [0,1]$,*

$$M_\lambda h(\alpha) = \sum_{T\subseteq[n]} \widehat{M_\lambda h}(T)\chi_T(\alpha), \quad \textit{where } \widehat{M_\lambda h}(T) = \lambda^{|T|} \sum_{S\supseteq T} (1-\lambda)^{|S-T|}\widehat{h}(S). \tag{33}$$

Proof. This follows by analyzing the T-th row of M_λ as in Equation (31). More specifically, we have:

$$\begin{aligned} M_\lambda h(\alpha) &= \sum_{S\subseteq[n]} \widehat{h}(S) M_\lambda \chi_S(\alpha) && (34)\\ &= \sum_{S\subseteq[n]} \widehat{h}(S) \sum_{T\subseteq S} \lambda^{|T|}(1-\lambda)^{|S-T|}\chi_T(\alpha) && \text{(Lemma A.2)}\\ &= \sum_{T\subseteq[n]} \chi_T(\alpha) \underbrace{\sum_{S\supseteq T} \lambda^{|T|}(1-\lambda)^{|S-T|}\widehat{h}(S)}_{\widehat{M_\lambda h}(T)}, && (35) \end{aligned}$$

where the final step follows by noting that each $\widehat{M_\lambda h}(T)$ depends only on $\widehat{h}(S)$ for $S \supseteq T$. □

The expression derived in Lemma A.3 shows how spectral mass gets redistributed from higher-degree to lower-degree terms. To understand how smoothing affects model robustness, it is helpful to quantify how much of the original function's complexity (i.e., higher-degree interactions) survive after smoothing. The following result shows how smoothing suppresses higher-order interactions by bounding how much mass survives in terms of degree $\geq k$.

Theorem A.4 (Higher-order Spectral Mass After Smoothing)**.** *For any function $h:\{0,1\}^n \to \mathbb{R}$, smoothing parameter $\lambda \in [0,1]$, and $0 \leq k \leq n$,*

$$\sum_{T:|T|\geq k} |\widehat{M_\lambda h}(T)| \leq \Pr_{X\sim \mathsf{Bin}(n,\lambda)}[X \geq k] \sum_{S:|S|\geq k} |\widehat{h}(S)|. \tag{36}$$

Proof. We first apply Lemma A.3 to expand each $\widehat{M_\lambda h}(T)$ and derive

$$\begin{aligned} \sum_{T:|T|\geq k} |\widehat{M_\lambda h}(T)| &\leq \sum_{T:|T|\geq k} \sum_{S\supseteq T} \lambda^{|T|}(1-\lambda)^{|S-T|}|\widehat{h}(S)| && (37)\\ &= \sum_{S:|S|\geq k} |\widehat{h}(S)| \underbrace{\sum_{j=k}^{|S|} \binom{|S|}{j} \lambda^j (1-\lambda)^{|S|-j}}_{\Pr_{Y\sim \mathsf{Bin}(|S|,\lambda)}[Y\geq k]} && (38) \end{aligned}$$

where we re-indexed the summations to track the contribution of each $|\widehat{h}(S)|$ for $|S| \geq k$. To yield the desired result, we next apply the following inequality of binomial tail CDFs given $|S| \leq n$:

$$\Pr_{Y\sim \mathsf{Bin}(|S|,\lambda)}[Y \geq k] \leq \Pr_{X\sim \mathsf{Bin}(n,\lambda)}[X \geq k]. \tag{39}$$

□

Our analyses with respect to the standard basis provide a first step towards understanding the random masking operator M_λ. However, the weight-mixing from our initial calculations suggests that the standard basis may be algebraically challenging to work with.

A.3 Analysis in the p-Biased Basis

While analysis on the standard Fourier basis reveals interesting properties about M_λ, it suggests that this may not be the natural choice of basis in which to analyze random masking. Principally, this is because each $M_\lambda \chi_S$ is expressed as a linear combination of χ_T where $T \subseteq S$. By "natural", we instead aim to express the image of M_λ as a single term. One partial attempt is an extension of the standard basis, known as the p-biased basis, which is defined as follows.

Definition A.5 (p-Biased Basis)**.** For any subset $S \subseteq [n]$, define its corresponding p-biased function basis as:

$$\chi_S^p(\alpha) = \prod_{i \in S} \frac{p - \alpha_i}{\sqrt{p - p^2}}. \tag{40}$$

Observe that when $p = 1/2$, this is the standard basis discussed earlier. The p-biased basis is orthonormal with respect to the p-biased distribution on $\{0,1\}^n$ in that:

$$\mathop{\mathbb{E}}_{\alpha \sim \mathrm{Bern}(p)^n} [\chi_S^p(\alpha) \chi_T^p(\alpha)] = \begin{cases} 1 & \text{if } S = T, \\ 0 & \text{if } S \neq T. \end{cases} \tag{41}$$

On the p-biased basis, smoothing with a well-chosen λ induces a change-of-basis effect.

Lemma A.6 (Change-of-Basis)**.** *For any p-biased basis function χ_S^p and smoothing parameter $\lambda \in [p, 1]$,*

$$M_\lambda \chi_S^p(\alpha) = \left(\frac{\lambda - p}{1 - p}\right)^{|S|/2} \chi_S^{p/\lambda}(\alpha). \tag{42}$$

Proof. Expanding the definition of M_λ, we first derive:

$$\begin{aligned} M_\lambda \chi_S^p(\alpha) &= \mathop{\mathbb{E}}_{z \sim \mathrm{Bern}(\lambda)^n} \left[\prod_{i \in S} \frac{p - \alpha_i z_i}{\sqrt{p - p^2}} \right] && (43) \\ &= \prod_{i \in S} \mathop{\mathbb{E}}_{z} \left[\frac{p - \alpha_i z_i}{\sqrt{p - p^2}} \right] && \text{(by independence of } z_1, \ldots, z_n) \\ &= \prod_{i \in S} \frac{p - \lambda \alpha_i}{\sqrt{p - p^2}}, && (44) \end{aligned}$$

We then rewrite the above in terms of a (p/λ)-biased basis function as follows:

$$\begin{aligned} M_\lambda \chi_S^p(\alpha) &= \prod_{i \in S} \lambda \frac{(p/\lambda) - \alpha_i}{\sqrt{p - p^2}} && (45) \\ &= \prod_{i \in S} \lambda \frac{\sqrt{(p/\lambda) - (p/\lambda)^2}}{\sqrt{p - p^2}} \frac{(p/\lambda) - \alpha_i}{\sqrt{(p/\lambda) - (p/\lambda)^2}} && (\lambda \geq p) \\ &= \prod_{i \in S} \sqrt{\frac{\lambda - p}{1 - p}} \frac{(p/\lambda) - \alpha_i}{\sqrt{(p/\lambda) - (p/\lambda)^2}} && (46) \\ &= \left(\frac{\lambda - p}{1 - p}\right)^{|S|/2} \underbrace{\prod_{i \in S} \frac{(p/\lambda) - \alpha_i}{\sqrt{(p/\lambda) - (p/\lambda)^2}}}_{\chi_S^{p/\lambda}(\alpha)} && (47) \end{aligned}$$

□

When measured with respect to this changed basis, we can show that M_λ provably contracts the variance.

Theorem A.7 (Variance Reduction)**.** *For any function $h : \{0,1\}^n \to \mathbb{R}$ and smoothing parameter $\lambda \in [p, 1]$,*

$$\operatorname*{Var}_{\alpha \sim \mathrm{Bern}(p/\lambda)^n} [M_\lambda h(\alpha)] \leq \left(\frac{\lambda - p}{1 - p}\right) \operatorname*{Var}_{\alpha \sim \mathrm{Bern}(p)^n} [h(\alpha)]. \tag{48}$$

If the function is centered at $\mathbb{E}_{\alpha \sim \mathsf{Bern}(p)^n}[h(\alpha)] = 0$, then we also have:

$$\mathop{\mathbb{E}}_{\alpha \sim \mathrm{Bern}(p/\lambda)^n} \left[M_\lambda h(\alpha)^2\right] \leq \mathop{\mathbb{E}}_{\alpha \sim \mathrm{Bern}(p)} \left[h(\alpha)^2\right]. \tag{49}$$

Proof. We use the previous results to compute:

$$\begin{aligned}
\operatorname*{Var}_{\alpha \sim \mathrm{Bern}(p/\lambda)^n} [M_\lambda h(\alpha)] &= \operatorname*{Var}_{\alpha \sim \mathrm{Bern}(p/\lambda)^n} \left[M_\lambda \sum_{S \subseteq [n]} \widehat{h}(S) \chi_S^p(\alpha) \right] && \text{(by unique } p\text{-biased representation of } h) \\
&= \operatorname*{Var}_{\alpha \sim \mathrm{Bern}(p/\lambda)^n} \left[\sum_{S \subseteq [n]} \left(\frac{\lambda - p}{1 - p}\right)^{|S|/2} \widehat{h}(S) \chi_S^{p/\lambda}(\alpha) \right] && \text{(by linearity and Lemma A.6)} \\
&= \sum_{S \neq \emptyset} \left(\frac{\lambda - p}{1 - p}\right)^{|S|} \widehat{h}(S)^2 && \text{(Parseval's theorem by orthonormality of } \chi_S^{p/\lambda}) \\
&\leq \left(\frac{\lambda - p}{1 - p}\right) \sum_{S \neq \emptyset} \widehat{h}(S)^2 && (0 \leq \tfrac{\lambda - p}{1 - p} \leq 1 \text{ because } p \leq \lambda \leq 1) \\
&= \left(\frac{\lambda - p}{1 - p}\right) \operatorname*{Var}_{\alpha \sim \mathrm{Bern}(p)^n} [h(\alpha)] && \text{(Parseval's by orthonormality of } \chi_S^p)
\end{aligned}$$

which leads to the first desired inequality. For the second inequality, we have:

$$\mathop{\mathbb{E}}_{\alpha \sim \mathsf{Bern}(p)^n} [h(\alpha)^2] = \widehat{h}(\emptyset)^2 + \underbrace{\sum_{S \neq \emptyset} \widehat{h}(S)^2}_{\mathrm{Var}\,[h(\alpha)]}, \tag{50}$$

$$\mathop{\mathbb{E}}_{\alpha \sim \mathsf{Bern}(p/\lambda)^n} [M_\lambda h(\alpha)^2] = \widehat{M_\lambda h}(\emptyset)^2 + \underbrace{\sum_{S \neq \emptyset} \widehat{M_\lambda h}(S)^2}_{\mathrm{Var}\,[M_\lambda h(\alpha)]}, \tag{51}$$

where recall that $\widehat{h}(\emptyset) = \mathbb{E}_\alpha[h(\alpha)]$ is zero by assumption. □

The smoothing operator M_λ acts like a downshift on the standard basis and as a change-of-basis on a well-chosen p-biased basis. In both cases, the algebraic manipulations can be cumbersome and inconvenient, suggesting that neither is the natural choice of basis for studying M_λ. To address this, we next introduce in Appendix B a new set of basis functions, the *monotone basis* that allows for a more tractable characterization of how smoothing affects the structure and stability of Boolean functions in the context of classification.

B Analysis of Stability and Smoothing in the Monotone Basis

While the standard Fourier basis is a common starting point for studying Boolean functions, its interaction with M_λ is algebraically complex. The main reason is because this basis treats $0 \to 1$ and $1 \to 0$ perturbations symmetrically. In contrast, we wish to analyze perturbations that add features, (i.e., $\beta \geq \alpha$) and smoothing operations that remove features. This mismatch results in a complex redistribution of terms that is algebraically inconvenient to manipulate. We were thus motivated to develop a new set of analytical tooling, principally in a new set of basis functions that we call the *monotone basis*. In contrast to the Fourier basis, where smoothing spreads weights across subsets, the monotone basis aligns more directly with subset containment and better reflects the additive structure of $\Delta_r(\alpha)$.

B.1 Monotone Basis for Boolean Functions

For any subset $T \subseteq [n]$, define its corresponding *monotone basis function* $\mathbf{1}_T : \{0,1\}^n \to \{0,1\}$ as:

$$\mathbf{1}_T(\alpha) = \begin{cases} 1 & \text{if } \alpha_i = 1 \text{ for all } i \in T \text{ (all features in } S \text{ present)}, \\ 0 & \text{otherwise (any feature in } T \text{ is absent)}, \end{cases} \tag{52}$$

where let $\mathbf{1}_\emptyset(\alpha) = 1$. First, we flexibly identify subsets of $[n]$ with binary vectors in $\{0,1\}^n$, which lets us write $T \subseteq \alpha$ if $i \in T$ implies $\alpha_i = 1$. This gives us some useful ways to equivalently express $\mathbf{1}_T(\alpha)$:

$$\mathbf{1}_T(\alpha) = \prod_{i \in T} \alpha_i = \begin{cases} 1 & \text{if } T \subseteq \alpha, \\ 0 & \text{otherwise.} \end{cases} \tag{53}$$

The monotone basis lets us more compactly express properties that depend on the inclusion or exclusion of features. For instance, the earlier example of conjunction $h(\alpha) = \alpha_1 \wedge \alpha_2$ may be equivalently written as:

$$\begin{aligned} \alpha_1 \wedge \alpha_2 &= \mathbf{1}_{\{1,2\}}(\alpha) && \text{(monotone basis)} \\ &= \frac{1}{4}\chi_\emptyset(\alpha) - \frac{1}{4}\chi_{\{1\}}(\alpha) - \frac{1}{4}\chi_{\{2\}}(\alpha) + \frac{1}{4}\chi_{\{1,2\}}(\alpha) && \text{(standard basis)} \end{aligned}$$

Unlike the standard bases (both standard Fourier and p-biased Fourier), the monotone basis is not orthonormal with respect to $\{0,1\}^n$ because

$$\mathop{\mathbb{E}}_{\alpha \sim \{0,1\}^n}[\mathbf{1}_S(\alpha)\mathbf{1}_T(\alpha)] = \Pr_{\alpha \sim \{0,1\}^n}[S \cup T \subseteq \alpha] = 2^{-|S \cup T|}, \tag{54}$$

where note that $S \cup T \subseteq \alpha$ iff both $S \subseteq \alpha$ and $T \subseteq \alpha$. However, the monotone basis does satisfy some interesting properties, which we describe next.

Theorem B.1. *Any Boolean function $h : \{0,1\}^n \to \mathbb{R}^n$ can be uniquely expressed in the monotone basis as:*

$$h(\alpha) = \sum_{T \subseteq [n]} \widetilde{h}(T)\mathbf{1}_T(\alpha), \tag{55}$$

where $\widetilde{h}(T) \in \mathbb{R}$ are the monotone basis coefficients of h that can be recursively computed by the formula:

$$\widetilde{h}(T) = h(T) - \sum_{S \subsetneq T} \widetilde{h}(S), \quad \widetilde{h}(\emptyset) = h(\mathbf{0}_n), \tag{56}$$

where $h(T)$ denotes the evaluation of h on the binary vectorized representation of T.

Proof. We first prove existence and uniqueness. By definition of $\mathbf{1}_T$, we have the simplification:

$$h(\alpha) = \sum_{T \subseteq [n]} \widetilde{h}(T)\mathbf{1}_T(\alpha) = \sum_{T \subseteq \alpha} \widetilde{h}(T). \tag{57}$$

This yields a system of 2^n linear equations (one for each $h(\alpha)$) in 2^n unknowns (one for each $\widetilde{h}(T)$). We may treat this as a matrix of size $2^n \times 2^n$ with rows indexed by $h(\alpha)$ and columns indexed by $\widetilde{h}(T)$, sorted by inclusion and degree. This matrix is lower-triangular with ones on the diagonal ($\mathbf{1}_T(T) = 1$ and $\mathbf{1}_T(\alpha) = 0$ for $|T| > \alpha$; like a transposed Equation (31)), and so the 2^n values of $h(\alpha)$ uniquely determine $\widetilde{h}(T)$.

For the recursive formula, we simultaneously substitute $\alpha \mapsto T$ and $T \mapsto S$ in Equation (57) to write:

$$h(T) = \widetilde{h}(T) + \sum_{S \subsetneq T} \widetilde{h}(S), \tag{58}$$

and re-ordering terms yields the desired result. □

B.2 Smoothing and Stability in the Monotone Basis

A key advantage of the monotone basis is that it yields a convenient analytical expression for how smoothing affects the spectrum.

Theorem B.2 (Smoothing in the Monotone Basis)**.** *Let M_λ be the smoothing operator as in Definition A.1. Then, for any Boolean function $h : \{0,1\}^n \to \mathbb{R}$ and subset $T \subseteq [n]$, we have the spectral contraction:*

$$\widetilde{M_\lambda h}(T) = \lambda^{|T|}\widetilde{h}(T),$$

where $\widetilde{M_\lambda h}(T)$ and $\widetilde{h}(T)$ are the monotone basis coefficients of $M_\lambda h$ and h at T, respectively.

Proof. By linearity of expectation, it suffices to study the action of M_λ on each monotone basis function:

$$\begin{aligned}
M_\lambda \mathbf{1}_T(\alpha) &= \mathop{\mathbb{E}}_{z\sim\mathsf{Bern}(\lambda)^n}[\mathbf{1}_T(\alpha \odot z)] && \text{(by definition of } M_\lambda) \\
&= \mathop{\mathbb{E}}_{z\sim\mathsf{Bern}(\lambda)^n}\left[\prod_{i\in T}(\alpha_i z_i)\right] && \text{(by definition of } \mathbf{1}_T(\alpha)) \\
&= \prod_{i\in T}\left(\alpha_i \mathop{\mathbb{E}}_{z_i\sim\mathsf{Bern}(\lambda)}[z_i]\right) && \text{(by independence of } z_1,\ldots,z_n) \\
&= \lambda^{|T|}\mathbf{1}_T(\alpha) && (\mathbb{E}[z_i] = \lambda)
\end{aligned}$$

□

The monotone basis also gives a computationally tractable way of bounding the stability rate. Crucially, the difference between two Boolean functions is easier to characterize. As a simplified setup, we consider classifiers of form $h : \{0,1\}^n \to \mathbb{R}$, where for $\beta \sim \Delta_r(\alpha)$ let:

$$h(\beta) \cong h(\alpha) \quad \text{if} \quad |h(\beta) - h(\alpha)| \le \gamma. \tag{59}$$

Such h and its decision boundary γ may be derived from a general classifier $f : \mathbb{R}^n \to \mathbb{R}^m$ once x and α are known. This relation of the decision boundary then motivates the difference computation:

$$h(\beta) - h(\alpha) = \sum_{T\subseteq[n]} \widetilde{h}(T)(\mathbf{1}_T(\beta) - \mathbf{1}_T(\alpha) = \sum_{T\subseteq\beta\setminus\alpha, T\neq\emptyset} \widetilde{h}(T), \tag{60}$$

where recall that $\mathbf{1}_T(\beta) - \mathbf{1}_T(\alpha) = 1$ iff $T \neq \emptyset$ and $T \subseteq \beta \setminus \alpha$. This algebraic property plays a key role in tractably bounding the stability rate. More precisely, we upper-bound the *instability rate* $1 - \tau_r$, which is

$$1 - \tau_r = \Pr_{\beta\sim\Delta_r(\alpha)}[|h(\beta) - h(\alpha)| > \gamma]. \tag{61}$$

An upper bound of form $1 - \tau_r \le Q$, where Q depends on the monotone coefficients of h, then implies a lower bound on the stability rate $1 - Q \le \tau_r$. We show this next.

Lemma B.3 (Soft Stability Bound)**.** *For any Boolean function $h : \{0,1\}^n \to [0,1]$ and attribution $\alpha \in \{0,1\}^n$ that satisfy Equation* (59)*, the stability rate τ_r is bounded by:*

$$1 - \tau_r \le \frac{1}{\gamma}\sum_{k=1}^{r}\sum_{\substack{T\subseteq[n]\setminus\alpha \\ |T|=k}} |\widetilde{h}(T)| \cdot \Pr_{\beta\sim\Delta_r}[|\beta\setminus\alpha| \ge k], \tag{62}$$

where

$$\Pr_{\beta\sim\Delta_r}[|\beta\setminus\alpha| \ge k] = \frac{1}{|\Delta_r|}\sum_{j=k}^{r}\binom{n-|\alpha|-k}{j-k}, \quad |\Delta_r| = \sum_{i=0}^{r}\binom{n-|\alpha|}{i} \tag{63}$$

Proof. We can directly bound the stability rate as follows:

$$
\begin{aligned}
1-\tau_r &= \Pr_{\beta\sim\Delta_r}\left[|h(\beta)-h(\alpha)|>\gamma\right] && (64)\\
&\le \frac{1}{\gamma}\,\mathbb{E}_{\beta\sim\Delta_r}\left[|h(\beta)-h(\alpha)|\right] && \text{(Markov's inequality)}\\
&\le \frac{1}{\gamma}\,\mathbb{E}_{\beta\sim\Delta_r}\sum_{\substack{T\subseteq\beta\setminus\alpha\\ T\neq\emptyset}}|\widetilde{h}(T)| && \text{(by Equation (60), triangle inequality)}\\
&= \frac{1}{\gamma|\Delta_r|}\sum_{k=0}^{r}\sum_{|\beta\setminus\alpha|=k}\sum_{\substack{T\subseteq\beta\setminus\alpha\\ T\neq\emptyset}}|\widetilde{h}(T)| && \text{(enumerate } \beta\in\Delta_r(\alpha) \text{ by its size, } k)\\
&= \frac{1}{\gamma|\Delta_r|}\sum_{k=1}^{r}\sum_{S\subseteq[n]\setminus\alpha\atop |S|=k}\sum_{\substack{T\subseteq S\\ T\neq\emptyset}}|\widetilde{h}(T)| && \text{(the } k=0 \text{ term is zero, and let } S=\beta\setminus\alpha)\\
&= \frac{1}{\gamma|\Delta_r|}\sum_{k=1}^{r}\sum_{T\subseteq[n]\setminus\alpha\atop |T|=k}|\widetilde{h}(T)|\cdot\underbrace{\left|\{S\subseteq[n]\setminus\alpha : S\supseteq T, |S|\le r\}\right|}_{\text{Total times that }\widetilde{h}(T)\text{ appears}} && \text{(re-index by } T)\\
&= \frac{1}{\gamma}\sum_{k=1}^{r}\sum_{T\subseteq[n]\setminus\alpha\atop |T|=k}|\widetilde{h}(T)|\cdot\Pr_{\beta\sim\Delta_r}\left[|\beta\setminus\alpha|\ge k\right] && (65)
\end{aligned}
$$

□

An immediate consequence from Theorem B.2 is a stability rate bound on smoothed functions.

Theorem B.4 (Stability of Smoothed Functions)**.** *Consider any Boolean function $h:\{0,1\}^n\to[0,1]$ and attribution $\alpha\in\{0,1\}^n$ that satisfy Equation* (59)*. Then, for any smoothing parameter $\lambda\in[0,1]$,*

$$
1-\frac{Q}{\gamma}\le\tau_r(h,\alpha)\implies 1-\frac{\lambda Q}{\gamma}\le\tau_r(M_\lambda h,\alpha), \tag{66}
$$

where

$$
Q=\sum_{k=1}^{r}\sum_{T\subseteq[n]\setminus\alpha\atop |T|=k}|\widetilde{h}(T)|\cdot\Pr_{\beta\sim\Delta_r}\left[|\beta\setminus\alpha|\ge k\right]. \tag{67}
$$

Proof. This follows from applying Theorem B.2 to Lemma B.3 by noting that:

$$
1-\tau_r(M_\lambda h,\alpha)\le\frac{1}{\gamma}\sum_{k=1}^{r}\lambda^k\sum_{T\subseteq[n]\setminus\alpha\atop |T|=k}|\widetilde{h}(T)|\cdot\Pr_{\beta\sim\Delta_r}\left[|\beta\setminus\alpha|\ge k\right]. \tag{68}
$$

□

Moreover, we also present the following result on hard stability in the monotone basis.

Theorem B.5 (Hard Stability Bound)**.** *For any Boolean function $h:\{0,1\}^n\to[0,1]$ and attribution $\alpha\in\{0,1\}^n$ that satisfy Equation* (59)*, let*

$$
r^\star=\arg\max_{r\ge0}\max_{\beta:|\beta\setminus\alpha|\le r}\left[\left|\sum_{T\subseteq\beta\setminus\alpha,T\neq\emptyset}\widetilde{h}(T)\right|\le\gamma\right]. \tag{69}
$$

Then, h is hard stable at α with radius $r^\star$.

Proof. This follows from Equation (60) because it is equivalent to stating that:

$$r^\star = \arg\max_{r\geq 0} \max_{\beta:|\beta\setminus\alpha|\leq r} \underbrace{[|h(\beta) - h(\alpha)| \leq \gamma]}_{h(\beta)\cong h(\alpha)}. \tag{70}$$

□

In summary, the monotone basis provides a more natural setting in which to study the smoothing operator M_λ. While M_λ yields an algebraically complex weight redistribution under the standard basis, its effect is more compactly described in the monotone basis as a point-wise contraction at each $T \subseteq [n]$. In particular, we are able to derive a lower-bound improvement on the stability of smoothed functions in Theorem B.4.

C Additional Experiments

In this section, we include experiment details and additional experiments.

Models For vision models, we used Vision Transformer (ViT) [15], ResNet50, and ResNet18 [21]. For language models, we used RoBERTa [33].

Datasets For the vision dataset, we used a subset of ImageNet that contains two classes per sample, for a total of 2000 images. The images are of size $3 \times 224 \times 224$, which we segmented into grids with patches of size 16×16, for a total of $n = (224/16)^2 = 196$ features. For the language dataset, we used six subsets of TweetEval (emoji, emotion, hate, irony, offensive, sentiment) for a total of 10653 items; we omitted the stance subset because their corresponding fine-tuned models were not readily available.

Explanation Methods For feature attribution methods, we used LIME [44], SHAP [34], Integrated Gradients [52], and MFABA [62] using the implementation from exlib. [4] Each attribution method outputs a ranking of features by their importance score, which we binarized by selecting the top-25% of features.

Certifying Soft Stability We used SCA (Theorem 3.1) for certifying soft stability with parameters of $\varepsilon = \delta = 0.1$, for a sample size of $N = 150$. Where appropriate, we used 1000 iterations of bootstrap to compute the 95% confidence intervals.

Compute We used a shared cluster with NVIDIA GeForce RTX 3090 and NVIDIA RTX A6000 GPUs.

C.1 Certifying Hard Stability

A key part of hard stability certification lies in computing the certified radius. Below, we describe how Xue et al. [57] computes this for a MuS-smoothed classifier.

Theorem C.1 (Certifying Hard Stability via MuS [57])**.** *For any classifier $f : \mathbb{R}^n \to [0,1]^m$ and smoothing parameter $\lambda \in [0,1]$, let $\tilde{f} = M_\lambda f$ be the MuS-smoothed classifier. Then, for any input $x \in \mathbb{R}^n$ and explanation $\alpha \in \{0,1\}^n$, the certifiable hard stability radius is given by:*

$$r_{\text{cert}} = \frac{\tilde{f}_1(x \odot \alpha) - \tilde{f}_2(x \odot \alpha)}{2\lambda}, \tag{71}$$

where $\tilde{f}_1(x \odot \alpha)$ and $\tilde{f}_2(x \odot \alpha)$ denote the top-1 and top-2 class probabilities of the smoothed output $\tilde{f}(x \odot \alpha)$.

Each output coordinate $\tilde{f}_1, \ldots, \tilde{f}_m$ is also λ-Lipschitz to the masking of features:

$$|\tilde{f}_i(x \odot \alpha) - \tilde{f}_i(x \odot \alpha')| \leq \lambda |\alpha - \alpha'|, \quad \text{for all } \alpha, \alpha' \in \{0,1\}^n \text{ and } i = 1, \ldots, m. \tag{72}$$

[4] `https://github.com/BrachioLab/exlib`

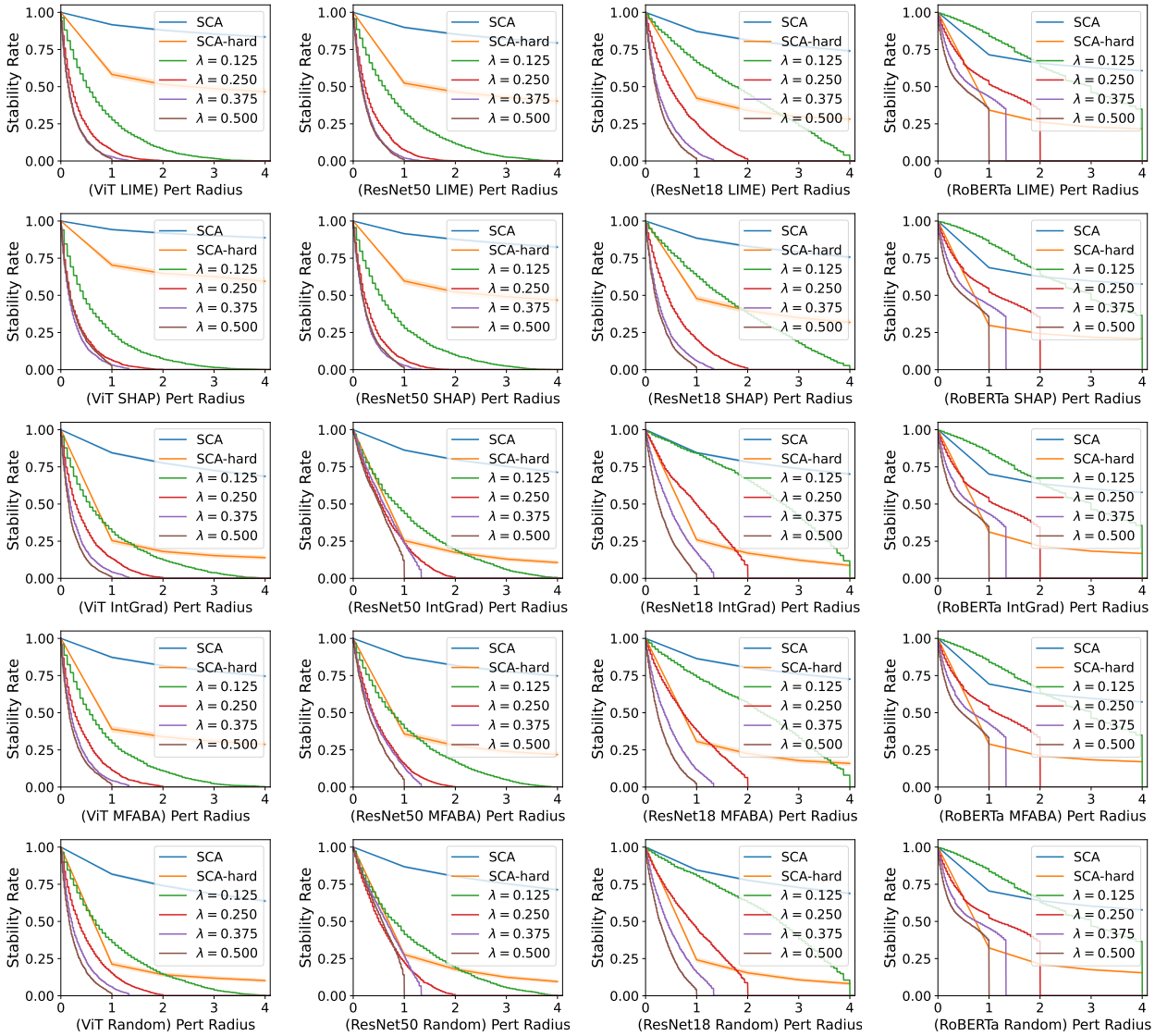

Figure 9: **Soft stability certifies more than hard stability.** An extended version of Figure 5.

That is, the keep-probability of each feature is also the Lipschitz constant (per earlier discussion: $\kappa = \lambda$). Note that deterministically evaluating $M_\lambda f_x$ would require 2^n samples in total, as there are 2^n possibilities for $\mathsf{Bern}(\lambda)^n$. Interestingly, distributions other than $\mathsf{Bern}(\lambda)^n$ also suffice to attain the desired Lipschitz, and thus hard certified radius, guarantees. In fact, Xue et al. [57] constructs such a distribution based on de-randomized sampling [30], for which a smoothed classifier can be deterministically evaluated in $\ll 2^n$ samples. However, our Boolean analytic results do not readily extend to classifiers smoothed with non-Bernoulli distributions.

C.2 Soft vs. Hard Stability of Different Explanation Methods

We show in Figure 9 an extension of Figure 5, where we also plot the estimated hard stability rate (SCA-hard). In particular, if we have $\hat{\tau}_r = 1$, then we also claim hard stability at radius r.

C.3 Stability vs. Smoothing

We show in Figure 10 an extension of Figure 7, where we plot perturbations at larger radii.

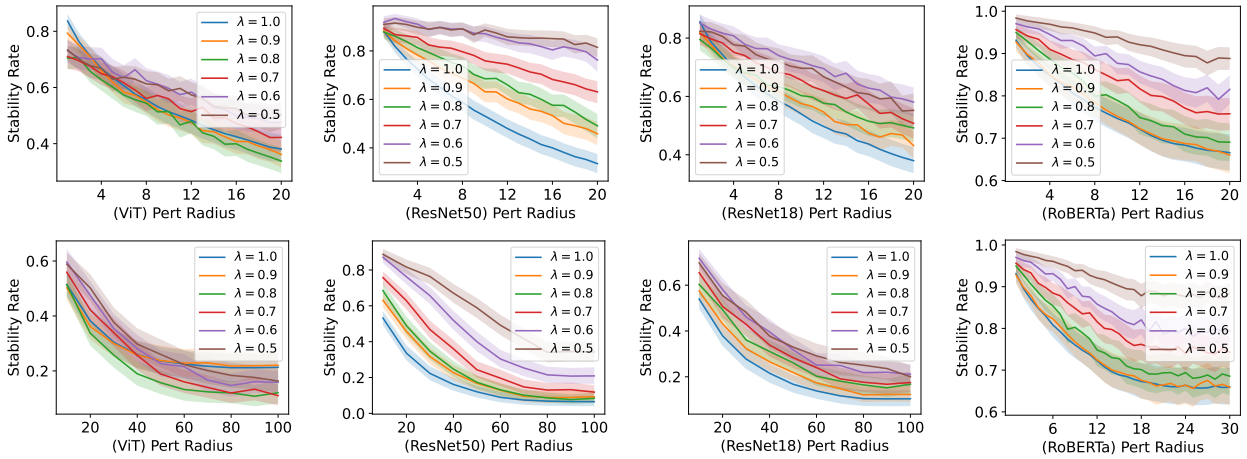

Figure 10: **Mild smoothing ($\lambda \geq 0.5$) can improve soft stability.** An extended version of Figure 7. The improvement is more pronounced at smaller radii (top row) than at larger radii (bottom row).

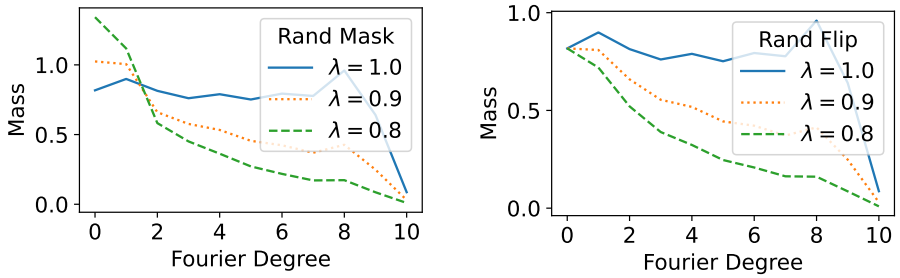

Figure 11: **Spectral effects of random masking (left) and flipping (right) are different.** With respect to the standard Fourier basis, random masking causes a down-shift in spectral mass, whereas the more commonly studied random flipping causes a point-wise contraction.

C.4 Random Masking vs. Random Flipping

We next study how the Fourier spectrum is affected by random masking and random flipping (i.e., the noise operator), which are respectively defined for Boolean functions as follows:

$$M_\lambda h(\alpha) = \mathop{\mathbb{E}}_{z \sim \mathsf{Bern}(\lambda)^n} [h(\alpha \odot z)] \qquad \text{(random masking)}$$

$$T_\lambda h(\alpha) = \mathop{\mathbb{E}}_{z \sim \mathsf{Bern}(q)^n} [h((\alpha + z) \bmod 2)], \quad q = \frac{1-\lambda}{2} \qquad \text{(random flipping)}$$

In both cases, $\lambda \approx 1$ corresponds to mild smoothing, whereas $\lambda \approx 0$ corresponds to heavy smoothing. To study this, we randomly generated a spectrum via $\widehat{h}(S) \sim N(0,1)$ for each $S \subseteq [n]$. We then average the mass of the randomly masked and randomly flipped spectrum at each degree, which are respectively:

$$\text{Average mass at degree } k \text{ from random masking } = \sum_{S:|S|=k} |\widehat{M_\lambda h}(S)|, \tag{73}$$

$$\text{Average mass at degree } k \text{ from random flipping } = \sum_{S:|S|=k} |\widehat{T_\lambda h}(S)|, \tag{74}$$

We plot the results in Figure 11, which qualitatively demonstrates the effects of random masking and random flipping on the standard Fourier basis.

D Additional Discussion

Our primary goal is to investigate reliable explanations for machine learning models, with stability serving as a key measure of reliability. Here, we provide some additional discussion.

Alternative Formulations of Stability Our definition of soft stability is one of many possible variants. For example, one might define $\tau_{=k}$ as the probability that the prediction remains unchanged under an *exactly* k-sized additive perturbation. A conservative variant could then take the minimum over $\tau_{=1}, \ldots, \tau_{=r}$. The choice of formulation affects the implementation of the certification algorithm.

Soft vs. Hard Stability While hard stability offers deterministic guarantees, it is highly conservative and limited to small certified radii, making it less practical for distinguishing between feature attribution methods. In contrast, soft stability leverages probabilistic certification to provide significantly larger guarantees while maintaining strong reliability.

Smoothing Our results also indicate that mild smoothing enhances soft stability without substantial accuracy degradation, suggesting broader applicability beyond robustness certification. These findings suggest the possibility of studying stability-aware training and adaptive smoothing techniques to improve the reliability and interpretability of feature-based explanations.

Limitations While soft stability provides a more fine-grained and model-agnostic robustness measure than hard stability, it remains sensitive to the choice of attribution thresholding and masking strategy. Our use of square patches and top-25% binarization, while standard, may not capture all forms of attribution uncertainty. Additionally, soft stability offers probabilistic rather than worst-case guarantees, which may be insufficient in some high-stakes settings without further calibration or confidence adjustment.