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

March 25, 2025

Abstract

Stability guarantees are an emerging tool for 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 present a model-agnostic, sample-efficient certification algorithm that provides non-trivial guarantees that are practically interpretable. 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 impact of smoothing on classifier behavior. We demonstrate the improvement of soft stability over hard stability through experiments on vision and language tasks with various feature attribution methods.

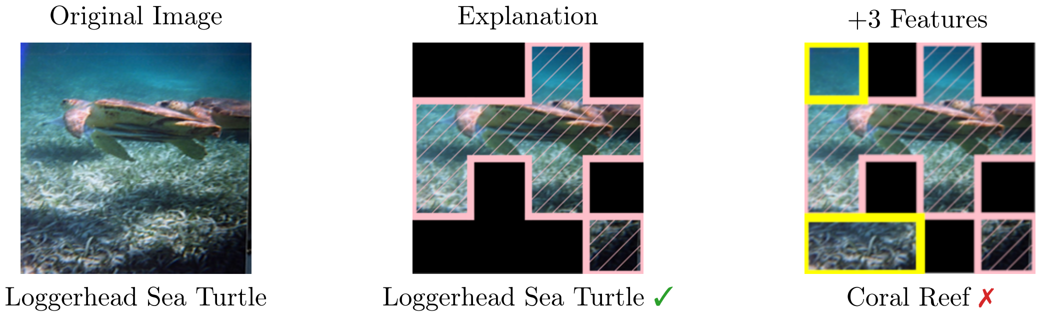

Figure 1: **An unstable explanation.** Given an input image (left), the features selected by LIME [45] (middle) are enough to preserve Vision Transformer's [16] prediction. However, adding just three more features (right, in yellow) flips the prediction, suggesting that the explanation is not robust.

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 [46].

There is much interest in explaining the behavior of complex models. One popular class of explanation methods are *feature attributions* [34, 45], 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. This kind of non-robustness is often undesirable because

[*] Equal contribution.

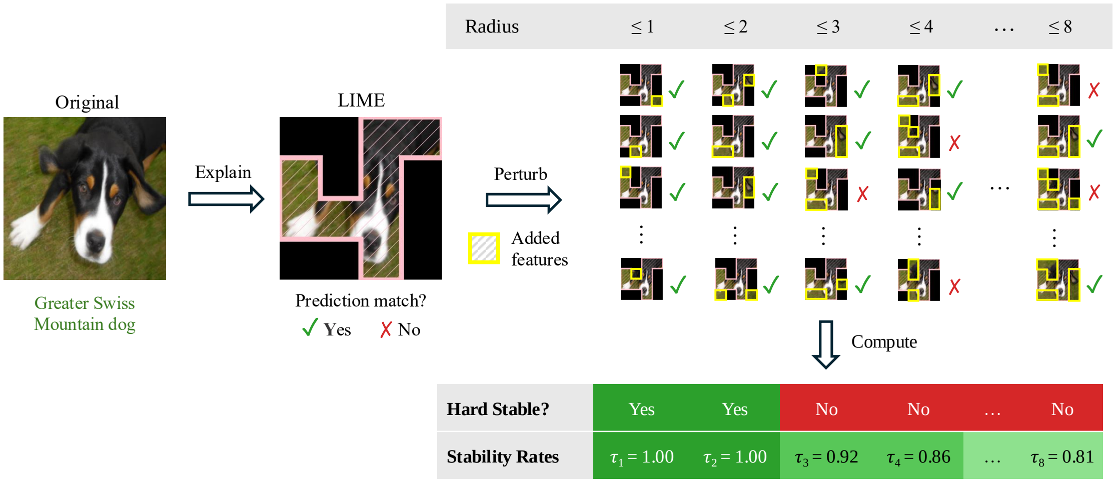

Figure 2: **Soft stability provides a fine-grained measure of robustness.** For the Vision Transformer classifier [16], LIME's explanation [45] is only hard stable at radius $r \leq 2$. In contrast, the stability rate — the key metric of soft stability — offers a more nuanced measure of sensitivity to added features.

it suggests that the given explanation is unreliable [41, 54, 59]. Such phenomena has motivated efforts to quantify how model predictions vary with explanations, including the effects of adding or removing features [48, 55] and the influence of the selection's shape [20, 47]. 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 prediction, whereas soft stability quantifies *how often* the prediction remains unchanged. Soft stability may thus be interpreted as a probabilistic relaxation of hard stability enables a more fine-grained analysis of explanation robustness. Crucially, this shift in perspective allows for model-agnostic applicability and admits efficient certification algorithm 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 Attains Useful Guarantees We introduce soft stability to address the limitations of hard stability. Soft stability certification is model-agnostic and provides non-trivial guarantees that are practically interpretable. In contrast, hard stability certification relies on heavily smoothed classifiers, which not only suffer from accuracy degradation but also yield overly conservative guarantees. In Section 3, we examine the drawbacks of hard stability and introduce an efficient soft stability certification algorithm.

Mild Smoothing Improves Soft Stability Although soft stability certification does not require smoothing, we found that mild smoothing can improve guarantees without incurring the severe accuracy degradation typically associated with the smoothing-based certification of hard stability. To explain this phenomenon, we developed new techniques for analyzing the effects of smoothing on stability using Boolean function analysis. We give an overview of our main theoretical results in Section 4.

Soft Stability Outperforms Hard Stability We empirically validated the improvement of soft stability over hard stability on a combination of vision and language tasks and various feature attribution methods. In particular, the attainable soft stability certificates are often two orders of magnitude stronger than the hard stability certificates. Moreover, we show that mildly smoothed classifiers attain a good trade-off between accuracy and stability. We present our experimental results in Section 5.

2 Background and Overview

First, we will 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

Feature attributions are widely used in explainability due to their simplicity and generality. To formalize our discussion, we consider classifiers of the form $f : \mathbb{R}^n \to \mathbb{R}^m$, which map n-dimensional inputs to m logits representing class probabilities. A feature attribution method assigns a score α_i to each input feature x_i to indicate its importance to the model's prediction $f(x)$. The definition of importance depends on the method. In gradient-based methods [50, 52], each α_i might be dependent on $\nabla_{x_i} f(x)$, whereas in Shapley value-based methods [34, 51], the α_i might to measure the Shapley value at x_i. Although attribution scores are typically real-valued, it is common to binarize them to $\alpha \in \{0, 1\}^n$ by selecting only the top-k highest-scoring features [45], thereby aligning with the general preference for concise explanations [39].

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 study 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. Next, we write $f(x \odot \alpha) \cong f(x \odot \alpha')$ to mean that the masked inputs $x \odot \alpha$ and $x \odot \alpha'$ have the same prediction on f, which holds if:

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

This form of evaluating feature sets is related to notions of *fidelity*, *consistency*, and *preservation* in the explainability literature [41], 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 conceptualize 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 radius $r > 0$, define its r-additive perturbation set as:

$$\Delta_r(\alpha) = \{\alpha' \in \{0, 1\}^n : \alpha' \geq \alpha,\ \|\alpha' - \alpha\|_0 \leq r\},$$

where $\alpha' \geq \alpha$ iff each $\alpha'_i \geq \alpha_i$ and $\|\cdot\|_0$ denotes the ℓ^0 norm, which counts the number of non-zero entries.

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 *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.

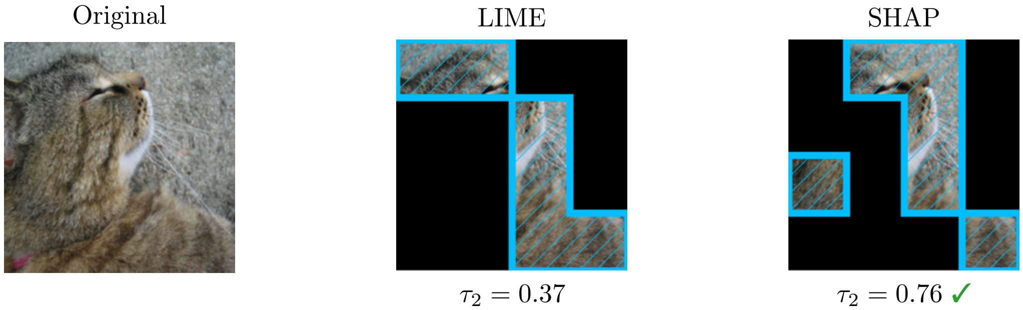

Figure 3: **Similar explanations may have different stability rates.** Despite visual similarities, the explanations generated by LIME [45] (middle) and SHAP [34] (right) have different stability rates at radius $r = 2$. In this example, SHAP's explanation is considerably more stable than LIME's.

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

However, hard stability is no straightforward to certify, and existing algorithms suffer from costly tradeoffs 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 [17], 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* of attribution α at radius r as:

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

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 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.

Alternative Formulations 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 $\tau_{=1}, \dots, \tau_{=r}$. The choice of formulation affects the implementation of the certification algorithm.

3 Certifying Soft Stability

We first discuss the limitations of existing methods for certifying hard stability. We then introduce a sampling-based algorithm to efficiently certify the soft stability of any model.

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 [18, 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.

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

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

where $x^{(1)}, \dots, x^{(N)} \sim \mathcal{D}(x)$ are sampled perturbations of x. If $\mathcal{D}$ is properly chosen, then the smoothed classifier $\tilde{f}$ has a Lipschitz constant κ that is explicitly known in expectation. [2]

Since κ measures a function's sensitivity to input perturbations, a smaller κ implies a smoother (i.e., more robust) classifier. Crucially, because $\tilde{f}$ is designed to have a known Lipschitz constant, this enables efficient computation of hard stability guarantees: in general, a smaller κ leads to larger certified radius. We describe how the certified radius is computed 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. Additionally, since smoothing relies on evaluation with perturbed inputs, it typically leads to accuracy degradation compared to f. This relation between smoothness, certified radii, and accuracy follows a well-known trade-off:

$$\text{Smoothness}(\tilde{f}) \approx \text{CertifiedRadius}(\tilde{f}) \approx \big(1 - \text{Accuracy}(\tilde{f})\big),$$

where $\approx$ indicates a general trend rather than an exact numerical relation. In other words, increased smoothness leads to larger certified radii (stronger hard stability guarantees) but at the cost of accuracy. This trade-off 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 these radii depend on a global property (the Lipschitz constant κ) to make local guarantees about feature perturbations. In general, the certified radius of $\tilde{f}$ scales as $\mathcal{O}(1/\kappa)$ for any input x and attribution α.

3.2 Estimating Soft Stability

Unlike hard stability, which requires destructively smoothing the classifier and often yields conservative guarantees, soft stability can be estimated efficiently for any classifier. Its key measure, the *stability rate* τ_r, can be efficiently estimated via the following algorithm.

Theorem 3.1 (Estimation Algorithm)**.** *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 stability rate estimator:*

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

Then, with probability $\geq 1 - \delta$, it holds that $|\tau_r - \hat{\tau}_r| \leq \varepsilon$.

Proof. Apply Hoeffding's inequality to estimate the mean of independently distributed Bernoulli random variables $X^{(1)}, \dots, X^{(N)}$, where each $X^{(i)} = \mathbf{1}[f(x \odot \alpha^{(i)}) \cong f(x \odot \alpha)]$. □

We illustrate this algorithm in Figure 4. Notably, the required sample size N depends only on ε and δ, since τ_r is a one-dimensional statistic. Because N is independent of f, the estimation algorithm scales linearly in the cost of evaluating f. Moreover, certifying soft stability does not require deriving a smoothed classifier through a destructive smoothing classifier. Unlike hard stability, which applies to the smoothed classifier $\tilde{f}$,

[2] Specifically, $\tilde{f}$ is Lipschitz with respect to the masking of features. For any $f : \mathbb{R}^n \to [0, 1]$, Xue et al. [57] yields a $\tilde{f} : \mathbb{R}^n \to [0, 1]$ where: $\|\tilde{f}(x \odot \alpha) - \tilde{f}(x \odot \alpha')\| \leq \kappa\|\alpha - \alpha'\|_0$. This is then lifted to m classes by standard robustness arguments.

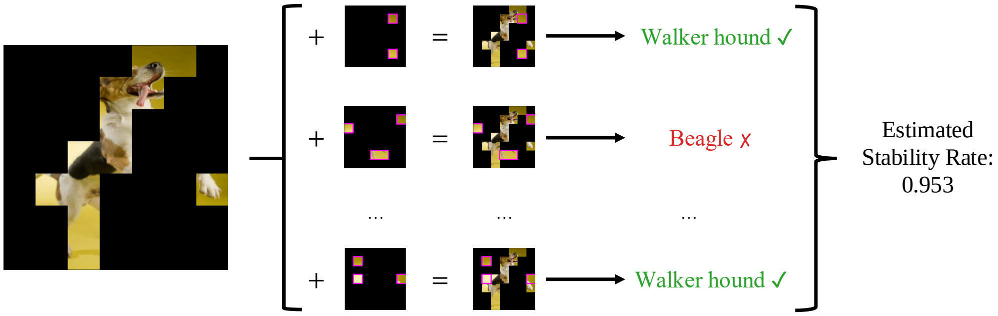

Figure 4: **Estimating the stability rate τ_r.** Given an explanation $\alpha \in \{0,1\}^n$ for a model f and input $x \in \mathbb{R}^n$: First, uniformly sample with replacement the additively perturbed explanations $\alpha' \sim \Delta_r(\alpha)$. Then, let $\hat{\tau}_r$ be the rate at which the perturbed explanation match the explanation's prediction, i.e., $\hat{\tau}_r = \frac{1}{N}\sum_{\alpha'} \mathbf{1}[f(x \odot \alpha') \cong f(x \odot \alpha)]$, 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 estimation accuracy is $|\hat{\tau}_r - \tau_r| \leq \varepsilon$. In this example, $\hat{\tau}_r = 0.953$.

soft stability provides robustness guarantees directly on the original classifier f. This eliminates the need for a destructive smoothing process that risks degrading accuracy.

For practical implementation, sampling from $\Delta_r(\alpha)$ may be done in two steps: first, suppose there are $m = n - \|\alpha\|_0$ zeros positions available in α and sample $j \sim \{0, 1, \ldots r\}$ with probability proportional to $\binom{m}{j} / \sum_{i=0}^{r} \binom{m}{i}$; then, uniformly select j zero positions among the m available in α and set them to ones. However, note that the values of $\binom{m}{i}$ can be large to the point of numerical instability. We thus recommend using a Gumbel softmax reparametrization [24] to sample in log space.

4 Mild Smoothing Improves Soft Stability

Smoothing is commonly used to certify robustness guarantees, but 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 in Theorem 3.1 does *not* require smoothing. Rather, mildly smoothing the model can empirically improve stability rates.

We now introduce the smoothing operator used to certify hard stability in Xue et al. [57], wherein the main idea is for the smoothed classifier to be more robust to the inclusion and exclusion of features. This is achieved by randomly masking features in the following process.

Definition 4.1 (Random Masking[3])**.** For any classifier f and smoothing parameter $\lambda \in [0, 1]$, define the random masking operator M_λ as:

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

The smoothing parameter λ is the probability that any given feature is kept. That is, each feature is randomly masked (zeroed, dropped) with probability $1 - \lambda$. We say that 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 give an overview of our results in Section 4.1 and a more technical presentation in Section 4.2.

[3] This is also called multiplicative smoothing because the noise scales the input, unlike standard additive noising [12]. In Xue et al. [57], the noise distribution is not restricted to coordinate-wise i.i.d. Bernoulli sampling over a 2^n-sized space. Instead, introducing structured statistical dependencies enables a deterministic, sample-efficient variant of M_λ.

4.1 Summary of Theoretical Results

Our main theoretical tooling is Boolean function analysis [42], 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 function $f_x(\alpha) = f(x \odot \alpha)$. Such $f_x : \{0,1\}^n \to \mathbb{R}^m$ is then a Boolean function, for which M_λ is well-defined because of the identity $M_\lambda f(x \odot \alpha) = M_\lambda f_x(\alpha)$.

Next, we state some simplifying assumptions to make our theoretical analysis tractable. Specifically, we consider classifiers of form $f : \mathbb{R}^n \to [0,1]$, where at a given input x, attribution α, and $\alpha' \sim \Delta_r(\alpha)$, we have the prediction match $f_x(\alpha) \cong f_x(\alpha')$ if $\lvert f_x(\alpha) - f_x(\alpha') \rvert \leq 1/2$. That is, α and α' induce the same class only when their classifier outputs are sufficiently close. These let us establish the following.

Theorem 4.2 (Main Stability Result, Informal)**.** *Smoothing improves the stability rate lower bound by a factor of λ. Let $\mathcal{Q}$ be a quantity that depends on f_x (specifically, its monotone basis weights), then:*

$$1 - \mathcal{Q} \leq \tau_r(f_x, \alpha) \implies 1 - \lambda \mathcal{Q} \leq \tau_r(M_\lambda f_x, \alpha).$$

In other words, smoothing improves the worst-case stability rate by a factor of λ. Although this result is on a lower bound, it aligns with our empirical observation 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 [42]. This motivated us to develop novel analytic tooling, the process of which we describe in the next section.

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 operator M_λ. We refer to Appendix A.1 for a more extensive exposition on Boolean function analysis and defer additional technical details to Appendix A and Appendix B.

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

$$h(\alpha) = \sum_{S \subseteq [n]} \widehat{h}(S) \chi_S(\alpha), \quad \text{where } \chi_S(\alpha) = \prod_{i \in S} (-1)^{\alpha_i}, \text{ and } \widehat{h}(S) = \langle h, \chi_S \rangle = \underset{\alpha \sim \mathsf{Bern}(1/2)^n}{\mathbb{E}} [h(\alpha) \chi_S(\alpha)].$$

Because there are 2^n possible values for S, this expansion captures all the $0, 1, \ldots, n$-degree interactions between the input bits. Each subset S is associated with a coefficient (weight) $\widehat{h}(S) \in \mathbb{R}$ and function $\chi_S : \{0,1\}^n \to \{\pm 1\}$. The set of functions $\{\chi_S : S \subseteq [n]\}$ is also known as the *standard (Fourier) basis* and its members are orthonormal in the sense that: $\langle \chi_S, \chi_T \rangle = 1$ if $S = T$, and $\langle \chi_S, \chi_T \rangle = 0$ if $S \neq T$. As an example, the 2-bit conjunction (AND) $h(\alpha_1, \alpha_2) = \alpha_1 \wedge \alpha_2$ may be uniquely expressed in this basis as:

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

where $\alpha = (\alpha_1, \alpha_2)$ and $\chi_\emptyset(\alpha) = 1$ for all α by convention.

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 basis, the random masking operator M_λ acts as follows.

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

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

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

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

This characterizes how M_λ redistributes the weight at each S to the lower-order terms $T \subseteq S$ according to a binomial distribution of $\mathsf{Bin}(|S|, \lambda)$. Qualitatively, M_λ rapidly decays the weights at the higher degree terms (at larger S, T). However, this decay is distinct from those of more commonly studied smoothing operators in that it is due to a redistribution of weights from S to $T \subseteq S$ rather than by a point-wise contraction. [4]

While it may be possible to derive meaningful stability results with respect to the standard basis, and we refer to Appendix A for interesting results, we found it fruitful to search for more user-friendly bases on which M_λ may transform more naturally. In particular, we arrived at the *monotone basis*, described as follows.

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

$$\mathbf{1}_S(\alpha) = \begin{cases} 1, & \text{if } \alpha_i = 1 \text{ for all } i \in S, \\ 0, & \text{otherwise.} \end{cases}$$

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:

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

where $\widetilde{h}(S)$ are the recursively defined monotone weights at S and let $h(S)$ denote the evaluation of h on the natural $\{0,1\}^n$-valued representation of S. Crucially, the monotone basis exhibits an algebraically convenient point-wise contraction under M_λ.

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

$$\widetilde{M_\lambda h}(S) = \begin{cases} \widetilde{h}(\emptyset), & \text{if } S = \emptyset, \\ \lambda^{|S|} \widetilde{h}(S), & \text{if } S \neq \emptyset. \end{cases}$$

In the monotone basis, smoothing exponentially decays the weights by a power of λ, which is simpler than the redistribution of weights under the standard basis. This behavior more closely aligns with the more commonly studied paradigms, such as random flipping on the standard basis, allowing us to adapt existing tooling towards analyzing M_λ via the monotone basis. In particular, we established a more direct bound on the stability rate for smoothed classifiers in the manner of Theorem 4.2, where the value of $\mathcal{Q}$ is dependent on $\{\widetilde{h}(S) : |S| \leq r\}$, 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, investigate how smoothing affects classifier stability and accuracy, and analyze how different feature attribution methods perform under these conditions. Our findings reinforce that soft stability provides significantly larger certificates than hard stability, while mild smoothing often improves soft stability without substantial accuracy degradation. We give an overview of our results here and refer to Appendix C for a more comprehensive list of experiments.

Setup We evaluated a combination of vision and language models: for vision models, we used Vision Transformer (VIT) [16] and ResNet18 [21], while for language models, we used RoBERTa[33]. Our datasets included a 1000-image subset of ImageNet and the emotion subset of TweetEval. The images were of size $3 \times 224 \times 224$, which we segmented into patches of size 16×16, for a total of $n = (224/16)^2 = 196$ features per image. We considered five feature attribution methods: LIME [45], 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-k features in their ranking, where $k = 25\%$ unless specified otherwise. Our experiments were conducted using NVIDIA GeForce RTX 3090 and NVIDIA RTX A6000 GPUs.

[4] 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.

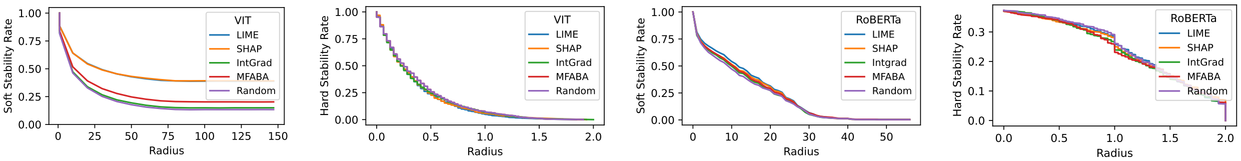

Figure 5: **Soft stability certifies more than hard stability.** We show the attainable soft (for $\varepsilon = \delta = 0.1$) and hard stability rates. Hard stability can only achieve up to a radius of 2, whereas soft stability can achieve radii of more than two orders of magnitude. This is because soft stability is a probabilistic certificate, whereas hard stability is a deterministic one. We show the soft stability rates for Vision Transformer (far left) and RoBERTa (center right), as well as their respective hard stability rates (center left; far right).

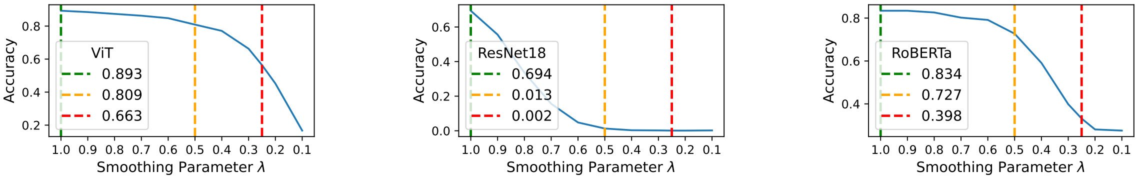

Figure 6: **Accuracy decreases as smoothing intensifies.** We report accuracy values at key thresholds: ($\lambda = 1.0$) the original, unmodified classifier; ($\lambda = 0.5$) the threshold above which hard stability certificates are not attainable; ($\lambda = 0.25$) at which hard stability can only certify additive perturbations of up to 2 features.

Question 1: How much do soft stability certificates improve over hard stability? We compare soft stability guarantees with hard stability across different inputs and radii. While soft stability is measured via the (soft) stability rate, hard stability is conventionally defined by its (hard) certifiable radius. To facilitate comparison, we define a hard stability rate as follows:

$$\text{Hard stability rate at radius } r = \frac{\#\{\text{CertifiedRadius}(M_\lambda f, x, \alpha) \geq r\}}{\text{Total number of } x\text{'s}}$$

We describe the computation of the certified radius in Theorem C.1 and present the results in Figure 5. To estimate soft stability rates we use $\delta = \varepsilon = 0.1$, which means that 150 samples per (x, α) instance are required for a confidence interval of size ε with probability $\geq 1 - \delta$ (c.f. Theorem 3.1). For hard stability, we set $\lambda = 0.25$. Since hard stability certification involves random masking (Definition 4.1), the smoothed classifier incurs an accuracy penalty compared to the classifier used for soft stability — an effect we further examine in later experiments.

We find that soft stability consistently yields larger certified radii than hard stability across all models and attribution methods. Moreover, soft stability effectively distinguishes attribution methods, with LIME and SHAP outperforming IntGrad, MFABA, and random baselines across all radii. In contrast, hard stability produces overly conservative radii, which limits its ability to differentiate stability across methods, which aligns with the empirical observations in Xue et al. [57]. While soft stability is inherently probabilistic, one can improve confidence by taking more samples. We give more detailed comparisons of the soft vs. hard stability rates in Appendix C.1 and give more details on the hard stability rates in Appendix C.2.

Question 2: How much does smoothing degrade accuracy? We analyze the impact of smoothing on classifier accuracy and observe that values of $\lambda > 0.5$ largely suffice to maintain the original accuracy. We plot our results in Figure 6, where we note 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].

For evaluation, we used $N = 64$ Bernoulli samples when computing the smoothed classifier $M_\lambda f$. We observe that accuracy generally remains okay until around $\lambda \leq 0.5$, after which performance begins to decline sharply.

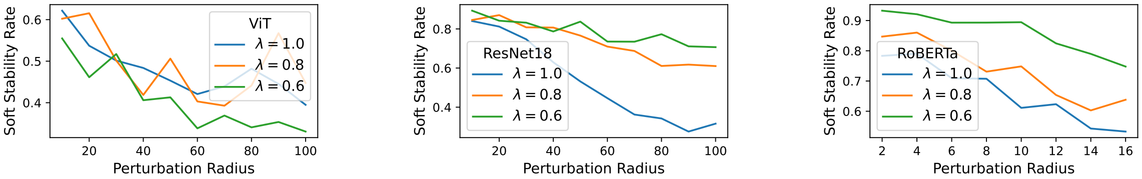

Figure 7: **Mild smoothing can improve the stability rate.** However, the improvement is not necessarily monotonic, and some models benefit more than others. The values reported are for when α is a random selection of 25% of the input features.

Attribution-masked is more prevalent for vision models, where fine-tuning on masked images is known to mitigate the accuracy loss [57], though avoiding degradation entirely remains challenging.

Question 3: How does smoothing affect soft stability? We study the effect of the smoothing parameter on the stability rate. For the vision dataset, we use a random sample of $N = 50$ images from our ImageNet subset, and for each image, we randomly select 25% of the input features to be α. For the language dataset, we use only those in the emotions dataset with input token sequences of length ≥ 40, of which there were 50 items, where we similarly choose to include 25% of them in α. We do this because the average token length is only 28, so having too great of a perturbation radius might accidentally reveal too many features.

We show our results in Figure 7, where we observe that smoothing improves stability for ResNet18 and RoBERTa, but it does not help much for Vision Transformer. This is likely because Vision Transformer can already attain good accuracy on highly masked images, so further smoothing may not have much effect. We give additional experiments in Appendix C.3 on the effect of different values of α.

6 Related Work

Feature-based Explanations Feature attributions have long been used in explainability and remain popular. Early examples include gradient saliency [50], LIME [45], SHAP [34], and Integrated Gradients [52]. More recent works include DIME [37], LAFA [60], CAFE [13], DoRaR [44], 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 modeling [36]. For general surveys on explainability, 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. [43]. For explainability in law, we refer to [4, 46]. Furthermore, "stability" is a widely used and overloaded term in the explainability literature, but many definitions relate to some notion of robustness [41].

Evaluating Feature Attributions Although feature attributions are popular, their correctness and usefulness have often been called into question [1, 15, 27]. This is because each attribution method computes importance by a different measure, which may not necessarily be indicative of the underlying model behavior [2, 61], as well as theoretical results on their limitations [9]. This has prompted a large number of evaluation metrics for feature attributions [3, 25, 41, 47], in particular for various notions of robustness [19, 26].

Certifying Feature Attributions While many empirical metrics exist, there is also growing interest in ensuring that feature attributions are well-behaved through formal, mathematical guarantees. In particular, there is interest in 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 in some sense [10]. However, the literature on explicit guarantees for feature attributions is still emerging, largely because formalizing desirable properties and algorithmically certifying them is difficult.

7 Discussion

Towards More Reliable Explanations Our primary goal is to provide reliable explanations for machine learning models, with stability guarantees serving as a key measure of reliability. While existing certification algorithms offer provable guarantees, they are often overly conservative due to their reliance on deterministic methods. As noted by Xue et al. [57], empirically attainable certified radii tend to exceed their robustly certified counterparts. To mitigate this conservativeness, we explore probabilistic guarantees, which yield larger, more practically interpretable certificates that enhance their usefulness for explainability.

Boolean Function Analysis in Explainability Boolean analytic techniques are well-suited for explainability, as many manipulations in this domain are often 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 Lipschitz-based robustness analysis and instead leverage Boolean analytic methods for a more discrete perspective on stability. Our findings suggest that similar techniques could be valuable in other machine learning tasks, particularly those involving voting, aggregation, or other discrete perturbation schemes.

Future Directions One interesting direction for future work is adaptive smoothing, where the smoothing parameter is dynamically adjusted based on feature importance or model confidence. For instance, weaker smoothing could be applied to confident predictions, while stronger smoothing stabilizes uncertain ones. Another promising avenue is stability-aware training, where models are explicitly trained to optimize for higher soft stability rates. This could involve regularizing unstable attributions or integrating stability constraints into pre-training, such that models are encouraged to induce inherently stable explanations. Additionally, exploring the connection between stability and generalization could offer deeper insights into explainability. If higher soft stability correlates with better generalization, then stability metrics could serve as a proxy for model reliability. Furthermore, stability shares similarities with adversarial robustness, as both measure sensitivity to input perturbations. Therefore, it would be interesting to study settings in which stability arises as the natural notion of robustness.

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 work was supported by a gift from Amazon AWS.

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

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

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

[17] 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.

[18] 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.

[19] 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.

[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] Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17, 2018.

[41] 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.

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

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[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 section, we analyze the random masking operator using standard tooling from Boolean function analysis. Recall that this operator is defined as follows.

Definition A.1 (Xue et al. [57])**.** 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:

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

Tooling from Boolean analysis is applicable to M_λ in the following manner. For any function $f : \mathbb{R}^n \to \mathbb{R}^m$ and input $x \in \mathbb{R}^n$, define the function $f_x(\alpha) = f(x \odot \alpha)$. Such $f_x : \{0,1\}^n \to \mathbb{R}^m$ is then a Boolean function, which then satisfies the following relation:

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

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*:

$$h(\alpha) = \sum_{S \subseteq [n]} \widehat{h}(S) \chi_S(\alpha), \quad \text{where } \chi_S(\alpha) = (-1)^{\sum_{i \in S} \alpha_i} = \prod_{i \in S} (-1)^{\alpha_i} = \prod_{i \in S} (1 - 2\alpha_i),$$

where the summation is indexed by all the 2^n subsets of $[n] = \{1, \ldots, n\}$. Each subset $S \subseteq [n]$ is associated with a *Fourier coefficient* (*weight*) $\widehat{h}(S) \in \mathbb{R}$ and *basis function* $\chi_S : \{0,1\}^n \to \{-1, +1\}$. The definition here is a special case of the p-biased basis functions, which we describe in Definition A.5. Expressing a Boolean function in this form makes all the $k = 0, 1, \ldots, n$ degree interactions between input variables into a linear combination, where the degree of $\widehat{h}(S)\chi_S(\alpha)$ is the size of S.

These χ_S are also known as *parity functions* because they count whether there are an even $(+1)$ or odd (-1) number of $\alpha_i = 1$ for $i \in S$. Crucially, they form an orthonormal basis on $\{0,1\}^n$ via:

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

This orthonormality allows each Boolean function to be uniquely expressed as a linear combination of χ_S for $S \subseteq [n]$. As an example, the function $h(\alpha_1, \alpha_2) = \alpha_1 \wedge \alpha_2$ may be uniquely expressed as:

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

where $\alpha = (\alpha_1, \alpha_2)$, and let $\chi_\emptyset(\alpha) = 1$ for all α by convention.

A.2 Basic Results in the Standard Basis

A common strategy for studying operators like M_λ is to inspect how they act on each basis function. We now present some first results with respect to the standard basis functions.

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

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

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

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

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

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

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

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

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:

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

where recall that $\widehat{h}(S) = 1/4$ for all $S \subseteq \{1,2\}$. In general, it is helpful to lexicographically sort the rows and columns of M_λ and partition them by degree. As an example with $n = 3$, one may write $M_\lambda \in \mathbb{R}^{8\times 8}$ as:

$$M_\lambda \;\cong\; \begin{array}{c||c|ccc|ccc|c} & \emptyset & \{1\} & \{2\} & \{3\} & \{1,2\} & \{1,3\} & \{2,3\} & \{1,2,3\} \\ \hline\hline \emptyset & 1 & (1-\lambda) & (1-\lambda) & (1-\lambda) & (1-\lambda)^2 & (1-\lambda)^2 & (1-\lambda)^2 & (1-\lambda)^3 \\ \hline \{1\} & & \lambda & & & \lambda(1-\lambda) & \lambda(1-\lambda) & & \lambda(1-\lambda)^2 \\ \{2\} & & & \lambda & & \lambda(1-\lambda) & & \lambda(1-\lambda) & \lambda(1-\lambda)^2 \\ \{3\} & & & & \lambda & & \lambda(1-\lambda) & \lambda(1-\lambda) & \lambda(1-\lambda)^2 \\ \hline \{1,2\} & & & & & \lambda^2 & & & \lambda^2(1-\lambda) \\ \{1,3\} & & & & & & \lambda^2 & & \lambda^2(1-\lambda) \\ \{2,3\} & & & & & & & \lambda^2 & \lambda^2(1-\lambda) \\ \hline \{1,2,3\} & & & & & & & & \lambda^3 \end{array} \qquad (1)$$

This visualization will help us reason about later proofs. Because the columns of M_λ sum to 1, we have

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

Moreover, $M_\lambda h$ is a downshift of h 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 $S \supseteq T$. This relation is given as follows.

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

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

Proof. This follows by analyzing the T-th row of M_λ as in Equation (1). More specifically, we have:

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

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

This expression for $\widehat{M_\lambda h}(T)$ is useful because it allows us to characterize the spectral mass down-shift. This helps us bound how much mass $M_\lambda h$ retains for $\geq k$-degree terms.

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

$$\sum_{T : |T| \geq k} |\widehat{M_\lambda h}(T)| \leq C(k,n,\lambda) \sum_{S : |S| \geq k} |\widehat{h}(S)|,$$

where $C(k,n,\lambda) \leq 1$ is the tail cumulative of $\mathsf{Bin}(n,\lambda)$, i.e., $C(k,n,\lambda) = \Pr[X \geq k]$ for any $X \sim \mathsf{Bin}(n,\lambda)$.

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

$$\begin{aligned} \sum_{T : |T| \geq k} |\widehat{M_\lambda h}(T)| &\leq \sum_{T : |T| \geq k} \sum_{S \supseteq T} \lambda^{|T|} (1-\lambda)^{|S-T|} |\widehat{h}(S)| \\ &= \sum_{S : |S| \geq k} |\widehat{h}(S)| \underbrace{\sum_{j=k}^{|S|} \binom{|S|}{j} \lambda^j (1-\lambda)^{|S|-j}}_{C(k,|S|,\lambda)} \end{aligned}$$

where we re-indexed the summations to track the contribution of each $|\widehat{h}(S)|$ for $|S| \geq k$. As a visual aid, we refer to Equation (1). Finally, applying $C(k,|S|,\lambda) \leq C(k,n,\lambda)$ yields the stated contraction factor. □

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 challenging to work with.

A.3 Analysis in the 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:

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

Observe that when $p = 1/2$, this is the standard basis discussed earlier. We first rederive a basic fact that the p-biased basis functions are orthonormal.

Proposition A.6 (Orthonormality of p-Biased Basis)**.** *For any p-biased basis functions χ_S^p and χ_T^p,*

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

Proof. For any coordinate $i \in [n]$, note the following identities for the first and second moments:

$$\mathop{\mathbb{E}}_{\alpha \sim \mathrm{Bern}(p)} \left[\frac{p - \alpha_i}{\sqrt{p - p^2}} \right] = 0, \qquad \mathop{\mathbb{E}}_{\alpha \sim \mathrm{Bern}(p)} \left[\left(\frac{p - \alpha_i}{\sqrt{p - p^2}} \right)^2 \right] = 1.$$

The inner product is then:

$$\begin{aligned} \mathop{\mathbb{E}}_{\alpha \sim \mathrm{Bern}(p)^n} [\chi_S^p(\alpha)\chi_T^p(\alpha)] &= \mathop{\mathbb{E}}_{\alpha} \left[\prod_{i \in S} \frac{p - \alpha_i}{\sqrt{p - p^2}} \prod_{j \in T} \frac{p - \alpha_j}{\sqrt{p - p^2}} \right] \\ &= \underbrace{\prod_{i \in S \cap T} \mathop{\mathbb{E}}_{\alpha} \left[\left(\frac{p - \alpha_i}{\sqrt{p - p^2}} \right)^2 \right]}_{= 1, \text{ for any } S \text{ and } T} \underbrace{\prod_{j \in S \triangle T} \mathop{\mathbb{E}}_{\alpha} \left[\frac{p - \alpha_j}{\sqrt{p - p^2}} \right]}_{= 0, \text{ if } S \triangle T \neq \emptyset} \end{aligned}$$

where we have used the coordinate-wise independence of $\alpha_1, \ldots, \alpha_n$ to swap the expectation and products. □

We remark that setting $p = 1/2$ recovers the proof of orthonormality for the standard basis. On the p-biased basis, smoothing with a well-chosen λ induces a change-of-basis effect.

Lemma A.7 (Change-of-Basis)**.** *For any p-biased basis function χ_S^p and smoothing parameter $\lambda \in [p, 1]$,*

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

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

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

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

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

□

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

Theorem A.8 (Variance Reduction). *For any function $h : \{0,1\}^n \to \mathbb{R}$ and smoothing parameter $\lambda \in [p, 1]$,*

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

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

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

Proof. We use the previous results to compute:

$$\begin{aligned}
\operatorname*{Var}_{\alpha\sim\mathrm{Bern}(p/\lambda)^n} [M_\lambda h(\alpha)] &= \operatorname*{Var}_{\alpha\sim\mathrm{Bern}(p/\lambda)^n} \left[M_\lambda \sum_{S\subseteq[n]} \widehat{h}(S)\chi_S^p(\alpha)\right] && \text{(by unique p-biased representation of h)} \\
&= \operatorname*{Var}_{\alpha\sim\mathrm{Bern}(p/\lambda)^n} \left[\sum_{S\subseteq[n]} \left(\frac{\lambda-p}{1-p}\right)^{|S|/2} \widehat{h}(S)\chi_S^{p/\lambda}(\alpha)\right] && \text{(by linearity and Lemma A.7)} \\
&= \sum_{S\neq\emptyset} \left(\frac{\lambda-p}{1-p}\right)^{|S|} \widehat{h}(S)^2 && \text{(Parseval's theorem by orthonormality of $\chi_S^{p/\lambda}$)} \\
&\leq \left(\frac{\lambda-p}{1-p}\right) \sum_{S\neq\emptyset} \widehat{h}(S)^2 && \text{($0 \leq \tfrac{\lambda-p}{1-p} \leq 1$ because $p \leq \lambda \leq 1$)} \\
&= \left(\frac{\lambda-p}{1-p}\right) \operatorname*{Var}_{\alpha\sim\mathrm{Bern}(p)^n} [h(\alpha)] && \text{(Parseval's by orthonormality of χ_S^p)}
\end{aligned}$$

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

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

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

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

In other words, the smoothed function is less sensitive to input perturbations, assuming it is measured with respect to the appropriate input distribution. However, this does not directly give us a bound on the stability rate. In fact, we generally found it difficult to compute the stability rate with respect to both the standard basis and the p-biased basis. While our results above give novel and interesting characterizations of M_λ, we were thus motivated to develop better-suited Boolean analytic tooling.

B Analysis of Stability and Smoothing in the Monotone Basis

The analysis of feature attribution stability naturally leads to studying Boolean functions under one-way perturbations. While Fourier analysis is the standard tool for Boolean function analysis, it has key limitations for our setting. First, it treats 0→1 and 1→0 transitions symmetrically, making it harder to analyze perturbations that only add features ($\beta \geq \alpha$) and smoothing operations that only remove features (via masking). Second, the traditional spectral analysis focuses on global properties, while our stability guarantees are inherently local (they depend on the specific attribution α). This asymmetry in our setting, combined with our focus on mild smoothing ($\lambda \approx 1$), motivates the development of new analytical tools.

B.1 Monotone Basis for Boolean Functions

To respect this one-way nature of perturbations, we introduce a monotone basis. For any set $T \subseteq [n]$:

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

Unlike the standard Fourier basis, the monotone basis is not orthonormal. However, it satisfies certain desirable properties:

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

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

where $\widetilde{h}(T)$ are the monotone basis coefficients of h, $\widetilde{h}(\emptyset)$ is a constant term, and $\mathbf{1}_\emptyset(\alpha) = 1$ for all α. The basis functions satisfy:

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

and the coefficients can be computed recursively:

$$\widetilde{h}(T) = h(T) - \sum_{S \subsetneq T} \widetilde{h}(S)$$

where $h(T)$ means evaluating h on the attribution with 1's exactly at positions in T.

Proof of Lemma B.1. First, we prove existence and uniqueness. For any attribution α, let $S_\alpha = \{i : \alpha_i = 1\}$ be its support. By definition of $\mathbf{1}_T$:

$$\begin{aligned} h(\alpha) &= \widetilde{h}(\emptyset) + \sum_{\substack{T \subseteq [n] \\ T \neq \emptyset}} \widetilde{h}(T)\mathbf{1}_T(\alpha) \\ &= \widetilde{h}(\emptyset) + \sum_{T \subseteq S_\alpha} \widetilde{h}(T) && \text{(support restriction)} \end{aligned}$$

This gives a system of 2^n linear equations (one for each α) in 2^n unknowns (the coefficients $\widetilde{h}(T)$). When we order both attributions and sets by inclusion, for each set T, all proper subsets $S \subsetneq T$ appear before T in the ordering. This creates an upper triangular matrix with 1's on the diagonal (since $\mathbf{1}_T(T) = 1$ and $\mathbf{1}_T(S) = 0$ for $|S| < |T|$), proving existence and uniqueness.

For the inner product formula:

$$\begin{aligned} \mathop{\mathbb{E}}_{\alpha}\left[\mathbf{1}_S(\alpha)\mathbf{1}_T(\alpha)\right] &= \Pr_{\alpha}\left[\alpha_i = 1 \text{ for all } i \in S \cup T\right] && \text{(product rule)} \\ &= 2^{-|S\cup T|} && \text{(uniform distribution)} \end{aligned}$$

For the recursive formula, fix a set T and consider $h(T)$. By the expansion:

$$\begin{aligned} h(T) &= \widetilde{h}(\emptyset) + \sum_{S \subseteq T} \widetilde{h}(S) && \text{(basis expansion)} \\ &= \widetilde{h}(T) + \widetilde{h}(\emptyset) + \sum_{S \subsetneq T} \widetilde{h}(S) && \text{(split largest term)} \end{aligned}$$

Rearranging gives the recursive formula:

$$\widetilde{h}(T) = h(T) - \sum_{S \subsetneq T} \widetilde{h}(S) \qquad \text{(recursion)}$$

□

To build intuition for this basis, consider the following example:

Example B.2 (Conjunction vs Fourier)**.** Consider the conjunction of two features: $h(\alpha) = \alpha_1 \wedge \alpha_2$. This function outputs 1 only when both features are present. In the standard Fourier basis with $\chi_T(\alpha) = (-1)^{|\operatorname{supp}(\alpha)\cap T|}$, we have:

$$h(\alpha) = \frac{1}{4} + \frac{1}{4}\chi_{\{1\}}(\alpha) + \frac{1}{4}\chi_{\{2\}}(\alpha) + \frac{1}{4}\chi_{\{1,2\}}(\alpha).$$

In the monotone basis, by contrast, this is simply $h(\alpha) = \mathbf{1}_{\{1,2\}}(\alpha)$. This directly captures the AND operation: the function is 1 exactly when both features are present.

B.2 Connection between Stability and Monotone Basis Expansion

We will focus on $f : \mathbb{R}^n \to \mathbb{R}$ (binary classification) and consider the following notion of model prediction equivalence.

Definition B.3 (Model Prediction Equivalence)**.** For a classifier $f : \mathcal{X} \to \mathcal{Y}$ and input x, we say two predictions are equivalent, denoted $f(x \odot \beta) \cong f(x \odot \alpha)$, if:

$$|f(x \odot \beta) - f(x \odot \alpha)| \leq 1/2$$

For binary classifiers where $\mathcal{Y} = \{0, 1\}$, this means the predictions must be identical. For probabilistic classifiers where $\mathcal{Y} = [0, 1]$, this allows for small variations in confidence while preserving the predicted class. The monotone basis let us derive tight bounds on both soft and hard stability. We begin with soft stability:

Lemma B.4 (Soft Stability)**.** *For any Boolean function $h : \{0,1\}^n \to [0,1]$ and attribution α, the stability rate τ_r satisfies:*

$$1 - \tau_r \leq \frac{2}{\sum_{i=0}^{r} \binom{n-|\alpha|}{i}} \sum_{\substack{j=1 \\ S:|S|=j \\ S\cap \operatorname{supp}(\alpha)=\emptyset}}^{r} \left| \sum_{T \subseteq S \setminus \{\emptyset\}} \widetilde{h}(T) \right|$$

where $\widetilde{h}(T)$ are the coefficients in the monotone basis.

Proof of Lemma B.4. We begin with the definition of stability rate. By Markov's inequality:

$$\begin{aligned}1 - \tau_r &= \Pr_{\beta \sim \Delta_r(\alpha)} \left[|h(\beta) - h(\alpha)| > 1/2 \right] \\ &\leq 2 \mathop{\mathbb{E}}_{\beta \sim \Delta_r(\alpha)} \left[|h(\beta) - h(\alpha)| \right] && \text{(Markov)}\end{aligned}$$

To analyze the difference $h(\beta) - h(\alpha)$, we express it using the monotone basis:

$$h(\beta) - h(\alpha) = \sum_{T \subseteq [n]} \widetilde{h}(T) \big(\mathbf{1}_T(\beta) - \mathbf{1}_T(\alpha) \big)$$

Since perturbations only add features ($\beta \geq \alpha$), the difference in indicator functions simplifies considerably. In particular, we have for any set T that:

$$\mathbf{1}_T(\beta) - \mathbf{1}_T(\alpha) = \begin{cases} 1 & \text{if } T \neq \emptyset \text{ and } \alpha_i = 0, \beta_i = 1 \text{ for all } i \in T \\ 0 & \text{otherwise} \end{cases}$$

This allows us to rewrite the difference as a sum over only the relevant sets:

$$h(\beta) - h(\alpha) = \sum_{T : \alpha_i = 0, \beta_i = 1 \text{ for all } i \in T \setminus \{\emptyset\}} \widetilde{h}(T)$$

To compute the expectation of $|h(\beta) - h(\alpha)|$, we first need to understand the structure of this difference for any fixed β. Note that β is completely determined by the set of positions S where it differs from α (where zeros become ones). By construction of $\Delta_r(\alpha)$, this set S must satisfy two properties: $|S| = j$ for some $j \leq r$, and $S \cap \operatorname{supp}(\alpha) = \emptyset$ since we can only flip zeros to ones. For such a fixed set S, we can simplify our expression for $h(\beta) - h(\alpha)$:

$$h(\beta) - h(\alpha) = \sum_{T \subseteq S \setminus \{\emptyset\}} \widetilde{h}(T)$$

This simplification follows because a set T contributes to the difference if and only if it is contained in S (the positions where β differs from α).

Now we can compute the expectation by considering how β is sampled under $\Delta_r(\alpha)$. The sampling process has two steps: first, choose the number of positions $j \sim \{0, 1, \ldots, r\}$ to flip with probability proportional to $\frac{\binom{n-|\alpha|}{j}}{\sum_{i=0}^{r} \binom{n-|\alpha|}{i}}$; then, uniformly select j positions from the zeros in α. This gives us:

$$\mathop{\mathbb{E}}_{\beta} \left[|h(\beta) - h(\alpha)| \right] = \sum_{j=1}^{r} \sum_{\substack{S : |S| = j \\ S \cap \operatorname{supp}(\alpha) = \emptyset}} \frac{\binom{n-|\alpha|}{j}}{\sum_{i=0}^{r} \binom{n-|\alpha|}{i}} \left| \sum_{T \subseteq S \setminus \{\emptyset\}} \widetilde{h}(T) \right|$$

Combining this with our initial Markov inequality bound completes the proof:

$$1 - \tau_r \leq \frac{2}{\sum_{i=0}^{r} \binom{n-|\alpha|}{i}} \sum_{\substack{j=1 \\ S : |S| = j \\ S \cap \operatorname{supp}(\alpha) = \emptyset}}^{r} \left| \sum_{T \subseteq S \setminus \{\emptyset\}} \widetilde{h}(T) \right|$$

□

Here we present a simplification of the soft-stability bound above to make it easier to parse.

Lemma B.5 (Simplified Soft Stability Bound)**.** *Under the same conditions, we also have:*

$$1 - \tau_r \leq 2 \sum_{k=1}^{r} \sum_{\substack{T:|T|=k \\ T \cap \text{supp}(\alpha)=\emptyset}} |\widetilde{h}(T)| \cdot \frac{\sum_{j=k}^{r} \binom{n-|\alpha|-k}{j-k}}{\sum_{j=0}^{r} \binom{n-|\alpha|}{i}}$$

Proof. Starting from the bound in Lemma B.4:

$$\begin{aligned}
1 - \tau_r &\leq \frac{2}{\sum_{i=0}^{r} \binom{n-|\alpha|}{i}} \sum_{j=1}^{r} \sum_{\substack{S:|S|=j \\ S \cap \text{supp}(\alpha)=\emptyset}} \left| \sum_{T \subseteq S \setminus \{\emptyset\}} \widetilde{h}(T) \right| \\
&\leq \frac{2}{\sum_{i=0}^{r} \binom{n-|\alpha|}{i}} \sum_{j=1}^{r} \sum_{\substack{S:|S|=j \\ S \cap \text{supp}(\alpha)=\emptyset}} \sum_{T \subseteq S \setminus \{\emptyset\}} |\widetilde{h}(T)| && \text{(triangle inequality)} \\
&= \frac{2}{\sum_{i=0}^{r} \binom{n-|\alpha|}{i}} \sum_{k=1}^{r} \sum_{\substack{T:|T|=k \\ T \cap \text{supp}(\alpha)=\emptyset}} |\widetilde{h}(T)| \sum_{j=k}^{r} \binom{n-|\alpha|-k}{j-k} && \text{(reorder sums)} \\
&= 2 \sum_{k=1}^{r} \sum_{\substack{T:|T|=k \\ T \cap \text{supp}(\alpha)=\emptyset}} |\widetilde{h}(T)| \cdot \frac{\sum_{j=k}^{r} \binom{n-|\alpha|-k}{j-k}}{\sum_{j=0}^{r} \binom{n-|\alpha|}{i}} && \text{(rearrange)}
\end{aligned}$$

The derivation proceeds in three steps. We begin by applying the triangle inequality to separate the coefficients. Next, we reorder the summation to group terms by coefficient size. Finally, we count the occurrences of each coefficient in the sum. The final expression weights each coefficient $\widetilde{h}(T)$ by $\frac{\sum_{j=k}^{r} \binom{n-|\alpha|-k}{j-k}}{\sum_{j=0}^{r} \binom{n-|\alpha|}{i}}$, which is the probability that a random perturbation contains set T of size k. □

Observe that the bound depends only on the monotone expansion terms of degree $\leq r$. We can use the same technique above to derive a hard stability bound in terms of the monomial expansion as well.

Lemma B.6 (Hard Stability Bound)**.** *For any Boolean function $h : \{0,1\}^n \to [0,1]$ and attribution α, let*

$$r^* = \max \left\{ r \geq 0 : \max_{\substack{S \subseteq [n] \\ 1 \leq |S| \leq r \\ S \cap \text{supp}(\alpha)=\emptyset}} \left| \sum_{T \subseteq S \setminus \{\emptyset\}} \widetilde{h}(T) \right| \leq \frac{1}{2} \right\}$$

Then h is hard-stable at radius r^*.

Proof. For any $\beta \in \Delta_r(\alpha)$:

$$|h(\beta) - h(\alpha)| = \left| \sum_{T \subseteq \text{diff}(\beta,\alpha) \setminus \{\emptyset\}} \widetilde{h}(T) \right|$$

since $\text{diff}(\beta, \alpha)$ is always a non-empty subset of size at most r disjoint from $\text{supp}(\alpha)$. By definition of r^*, $|h(\beta) - h(\alpha)| \leq 1/2$ for all $\beta \in \Delta_{r^*}(\alpha)$, proving hard stability. □

B.3 Stability Bound for Smoothed Distribution

The monotone basis allows us to capture random masking as a simple transformation.

Theorem B.7 (Smoothing in Monotone Basis)**.** *Let M_λ be the smoothing operator that randomly masks features with probability $1 - \lambda$:*

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

where z represents a random mask and $\odot$ denotes element-wise multiplication as defined in Appendix A.

For any Boolean function $h : \{0,1\}^n \to [0,1]$, the smoothed function $M_\lambda h$ in the monotone basis satisfies:

$$\widetilde{M_\lambda h}(T) = \begin{cases} \widetilde{h}(\emptyset), & \text{if } T = \emptyset \text{ (constant term preserved)}, \\ \lambda^{|T|} \widetilde{h}(T), & \text{if } T \neq \emptyset \text{ (coefficients damped)}, \end{cases}$$

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

Proof of Theorem B.7. First, note that M_λ is a linear operator since expectation is linear. For the empty set, $\widetilde{M_\lambda h}(\emptyset) = \widetilde{h}(\emptyset)$ since smoothing preserves constants.

For any non-empty set T:

$$\begin{aligned} M_\lambda \mathbf{1}_T(\alpha) &= \mathop{\mathbb{E}}_{z \sim \mathsf{Bern}(\lambda)^n} \big[\mathbf{1}_T(\alpha \odot z) \big] \\ &= \mathop{\mathbb{E}}_{z \sim \mathsf{Bern}(\lambda)^n} \Big[\prod_{i \in T} (\alpha_i z_i) \Big] \\ &= \prod_{i \in T} \big(\alpha_i \mathop{\mathbb{E}}_{z_i \sim \mathsf{Bern}(\lambda)} [z_i] \big) \\ &= \lambda^{|T|} \mathbf{1}_T(\alpha) \end{aligned}$$

The result follows by linearity of expectation. □

With the above theorem in hand, we can now compute the stability of the smoothed classifier:

Corollary B.8 (Stability of Smoothed Functions)**.** *For any Boolean function $h : \{0,1\}^n \to [0,1]$, attribution α, and smoothing parameter $\lambda \in [0,1]$, the stability rate of the smoothed function satisfies:*

$$1 - \tau_r(M_\lambda h) \leq 2 \sum_{k=1}^{r} \lambda^k \sum_{\substack{T : |T| = k \\ T \cap \mathsf{supp}(\alpha) = \emptyset}} |\widetilde{h}(T)| \cdot \frac{\sum_{j=k}^{r} \binom{n - |\alpha| - k}{j - k}}{\sum_{j=0}^{r} \binom{n - |\alpha|}{i}}$$

This shows that smoothing improves stability by exponentially dampening the influence of larger feature sets.

Proof of Corollary B.8. Apply the stability bound from Lemma B.4 to $M_\lambda h$ and use Theorem B.7 which shows that $\widetilde{M_\lambda h}(T) = \lambda^{|T|} \widetilde{h}(T)$ for non-empty T:

$$\begin{aligned} 1 - \tau_r(M_\lambda h) &\leq 2 \sum_{k=1}^{r} \sum_{\substack{T : |T| = k \\ T \cap \mathsf{supp}(\alpha) = \emptyset}} |\widetilde{M_\lambda h}(T)| \cdot \frac{\sum_{j=k}^{r} \binom{n - |\alpha| - k}{j - k}}{\sum_{j=0}^{r} \binom{n - |\alpha|}{i}} && \text{(by Lemma B.4)} \\ &= 2 \sum_{k=1}^{r} \sum_{\substack{T : |T| = k \\ T \cap \mathsf{supp}(\alpha) = \emptyset}} \lambda^k |\widetilde{h}(T)| \cdot \frac{\sum_{j=k}^{r} \binom{n - |\alpha| - k}{j - k}}{\sum_{j=0}^{r} \binom{n - |\alpha|}{i}} && \text{(by Theorem B.7)} \\ &= 2 \sum_{k=1}^{r} \lambda^k \sum_{\substack{T : |T| = k \\ T \cap \mathsf{supp}(\alpha) = \emptyset}} |\widetilde{h}(T)| \cdot \frac{\sum_{j=k}^{r} \binom{n - |\alpha| - k}{j - k}}{\sum_{j=0}^{r} \binom{n - |\alpha|}{i}} && \text{(rearrange terms)} \end{aligned}$$

The final expression shows how smoothing affects stability through three key mechanisms. First, each coefficient $\widetilde{h}(T)$ is weighted by λ^k where $k = |T|$. Second, larger sets T are dampened more strongly since λ^k decreases exponentially with k. Finally, the combinatorial terms $\frac{\sum_{j=k}^{r}\binom{n-|\alpha|-k}{j-k}}{\sum_{j=0}^{r}\binom{n-|\alpha|}{i}}$ represent the probability of including set T in a random perturbation. □

Remark B.9 (Smoothing Effect). The upper bound for the smoothed function in Corollary B.8 is at least a factor of λ smaller than the upper bound for the original function, since $\lambda^k \leq \lambda$ for all $k \geq 1$. However, these are only upper bounds - the actual improvement from smoothing could be either better or worse than suggested by comparing these bounds.

B.4 Discussion and Practical Implications

Our analysis using a monotone basis reveals some key mechanisms affecting stability. First, mild smoothing ($\lambda \approx 1$) can be effective because it exponentially dampens higher-order terms while preserving essential low-order structure — for instance, with $\lambda = 0.9$, single-feature terms are dampened by 0.9. In comparison, five-feature terms are dampened by $0.9^5 \approx 0.59$. While our bounds guarantee at least a factor of λ improvement in stability (since $\lambda^k \leq \lambda$ for all $k \geq 1$), the actual improvement could be either better or worse in practice. Second, stability becomes harder to maintain at larger radii because both the number of terms and their combinatorial weights grow with r, suggesting that λ should be chosen based on the distribution of $|\widetilde{h}(T)|$ across different set sizes.

While this work establishes the theoretical foundations, we could use these insights to design new attribution methods that explicitly control the monotone basis expansion of their output — for instance, by regularizing higher-order coefficients or by constructing explanations primarily from small low-order terms. This suggests a new shift approach to attribution stability: rather than focusing solely on Lipschitz constants of the model, we should study the distribution of monotone basis coefficients, as these more directly capture the stability properties we care about.

C Additional Experiments and Figures

We present additional experiments in this section.

Experiment Setup For vision models, we use Vision Transformer (ViT) [16] and ResNet [21]. For language models, we use RoBERTa [33]. For the vision dataset, we use a 1000-sized subset [5] of ImageNet [14] that contains one sample per each of its 1000 classes. 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 the emotion subset of TweetEval [40], which consists of four classes: "anger", "joy", "optimism", and "sadness". For feature attribution methods, we used LIME [45], SHAP [34], Integrated Gradients [52], and MFABA [62]. We convert real-valued attribution to binary-valued ones by selecting the top-k features. We used the implementations from exlib [6]. For GPUs, we had access to a combination of NVIDIA GeForce RTX 3090 and NVIDIA RTX A6000.

Computing the Hard Certified Stability Radius An important part of hard stability is in computing the (hard) certified radius. Below, we describe how Xue et al. [57] compute this for a smoothed classifier.

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

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

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 $\tilde{f}(x \odot \alpha)$.

[5] `https://github.com/EliSchwartz/imagenet-sample-images`

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

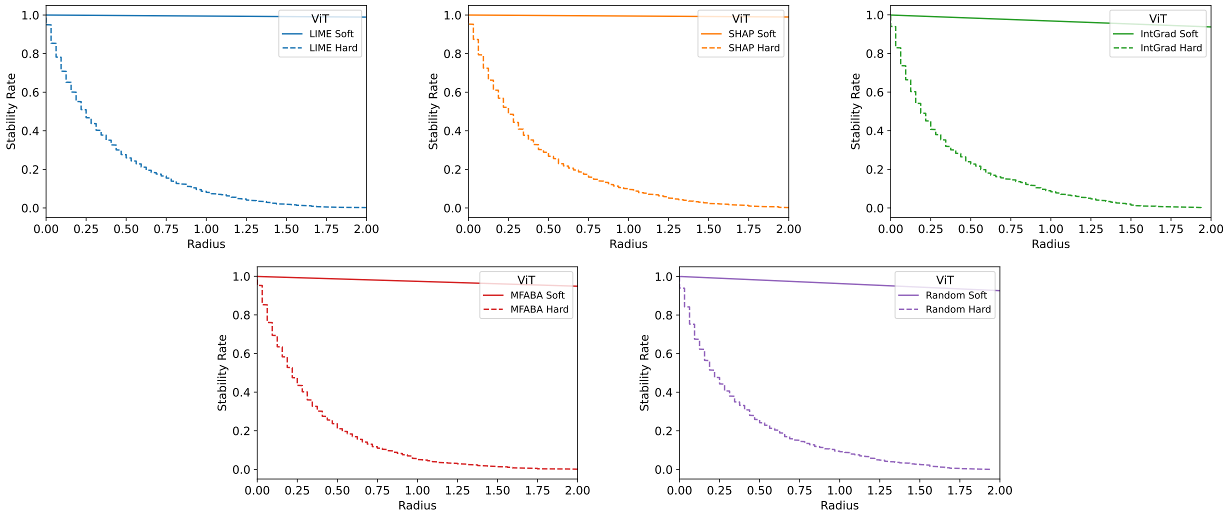

Figure 8: **(Vision Transformer) There is a large gap between the soft and hard stability rates.**

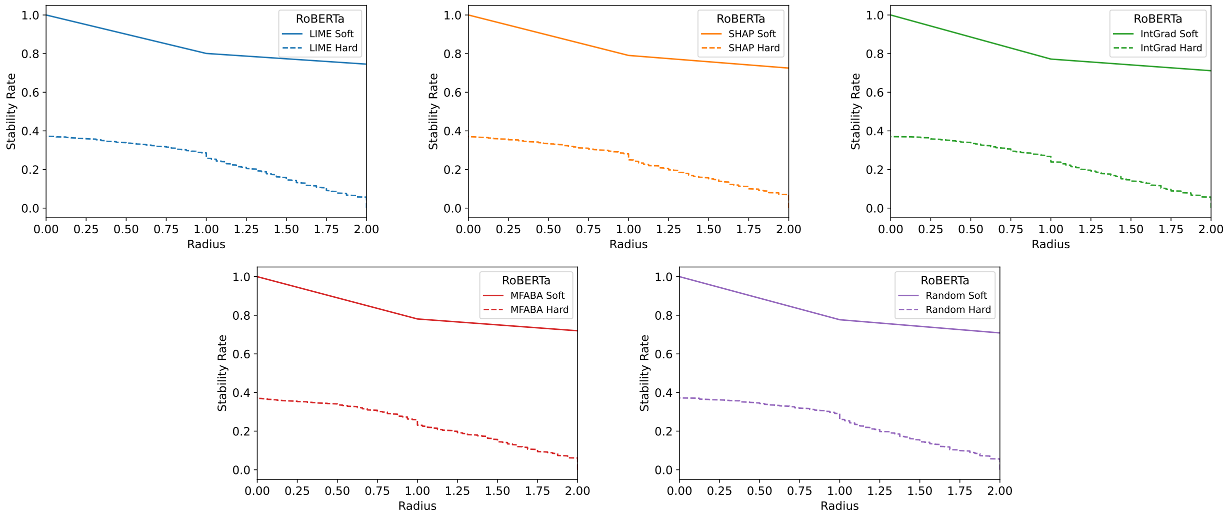

Figure 9: **(RoBERTa) There is a large gap between the soft and hard stability rates.**

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

$$|\tilde{f}_i(x \odot \alpha) - \tilde{f}_i(x \odot \alpha')| \leq \lambda \|\alpha - \alpha'\|_0, \quad \text{for all } \alpha, \alpha' \in \{0,1\}^n \text{ and } i = 1, \ldots, m.$$

That is, the keep-probability of each feature is also the Lipschitz constant (per earlier discussion: $\kappa = \lambda$). Note that deterministically evaluating $M_\lambda f$ 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] construct such a distribution 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 constructed from non-$\mathsf{Bern}(\lambda)^n$ distributions.

C.1 Soft vs. Hard Stability Rates

We compare the soft and hard stability rates across different explanation methods at a zoomed-in level. We use top-25% of features from LIME, SHAP, IntGrad, MFABA, and a random selection baseline. We show our results in Figure 8 for Vision Transformer, and in Figure 9 for RoBERTa. We truncated the curves at a

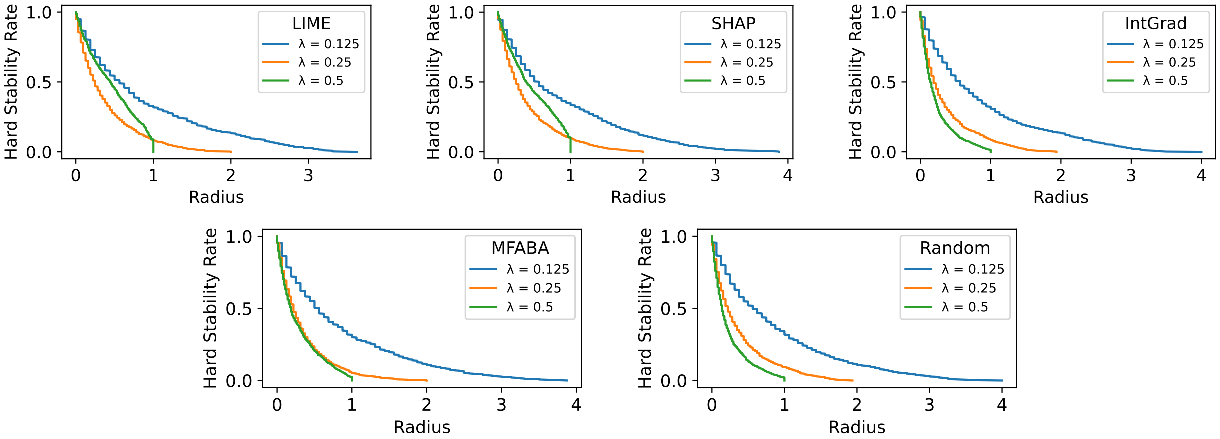

Figure 10: **(Vision Transformer) Hard stability rates for varying lambda parameters.** We show the hard stability rates for different explanation methods at different values of $\lambda = 0.125, 0.25, 0.5$.

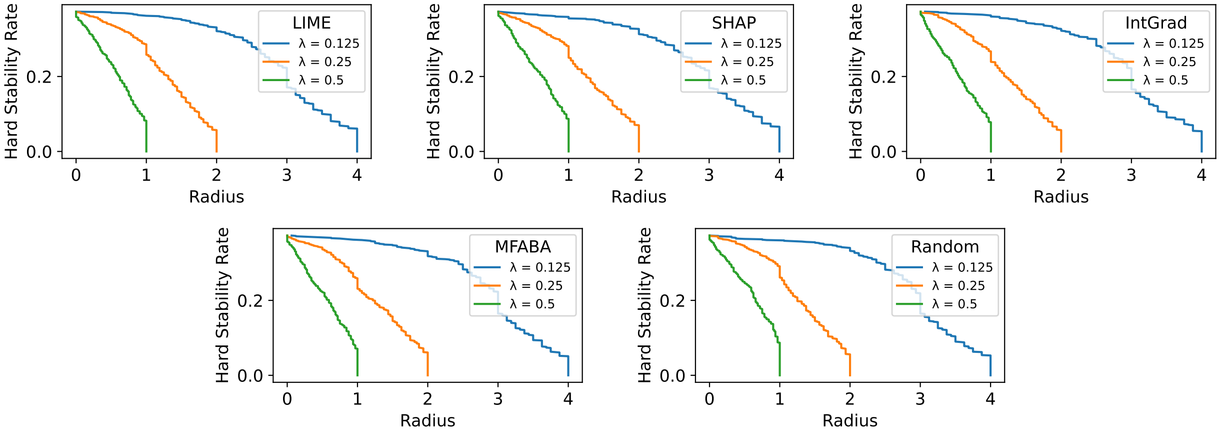

Figure 11: **(RoBERTa) Hard stability rates for varying lambda parameters.** We show the hard stability rates for different explanation methods at different values of $\lambda = 0.125, 0.25, 0.5$.

radius of 2, which is the maximum certifiable hard stability radius given our choice of $\lambda = 0.25$. Recall that the hard stability rate is computed as:

$$\text{Hard stability rate at radius } r = \frac{\#\{\text{CertifiedRadius}(M_\lambda f, x, \alpha) \geq r\}}{\text{Total number of } x\text{'s}},$$

where the certified radius is computed as in Theorem C.1. We note that the soft stability rates are consistently higher than that of hard stability, owing to the fact that soft stability is a probabilistic guarantee rather than a deterministic one.

C.2 Hard Stability Rates

We give a further zoomed-in perspective on the attainable hard stability rates for Vision Transformer in Figure 10, and for RoBERTa in Figure 11. By Theorem C.1 the maximum certifiable radius is $1/2\lambda$.

C.3 Stability vs. Smoothing

We investigate the stability rate at different λ when the initial α is allowed to vary. Most of our experiments were conducted where 25% of the entries in α were ones. We show in Figure 12 where this initial fraction

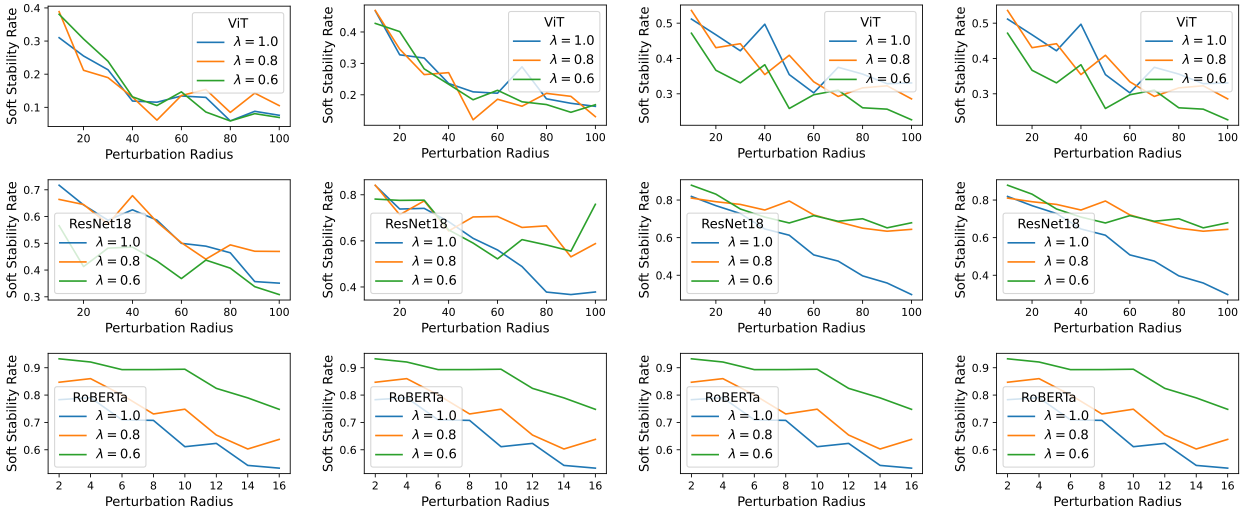

Figure 12: **Fraction of ones in α can affect stability.** We show the soft stability rates at different λ for Vision Transformer (top row), ResNet18 (middle row), and RoBERTa (bottom row). The four columns correspond to α where the fraction of ones entries are 0.10, 0.15, 0.20, and 0.25, respectively.

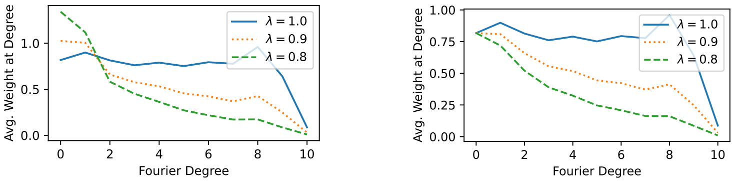

Figure 13: **Random masking (left) and random flipping (right) affect the spectrum differently.** With respect to the standard Fourier basis, random masking causes a down-shift in spectral mass, whereas the more commonly studied random flipping causes a point-wise contraction. This shows that the two smoothing operators are qualitatively different.

of ones was taken to be 10%, 15%, 20%, and 25%. In general, Vision Transformer did not benefit from smoothing. However, ResNet18 and RoBERTa saw improved stability rates when smoothed.

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 the lightly noised setting, whereas $\lambda \approx 0$ is the heavily noised setting.

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:

$$\sum_{S : |S| = k} |\widehat{M_\lambda h}(S)|, \qquad \sum_{S : |S| = k} |\widehat{T_\lambda h}(S)|, \qquad \text{for each degree } k = 0, 1, \dots, n.$$

We plot the result in Figure 13 for $n = 10$ features. Note that random masking causes the spectrum to down-shift, whereas random flipping induces a point-wise contraction at each S.

D Additional Discussion

Our primary goal is to investigate reliable explanations for machine learning models, with stability serving as a key measure of reliability. 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. 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.