Transparent Trade-offs between Properties of Explanations

Hiwot Belay Tadesse[1] **Alihan Hüyük**[2] **Yaniv Yacoby**[3] **Weiwei Pan**[4] **Finale Doshi-Velez**[5]

[1, 2, 4, 5]John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA
[3]Wellesley College, Wellesley, MA, USA

Abstract

When explaining machine learning models, it is important for explanations to have certain properties like faithfulness, robustness, smoothness, low complexity, etc. However, many properties are in tension with each other, making it challenging to achieve them simultaneously. For example, reducing the complexity of an explanation can make it less expressive, compromising its faithfulness. The ideal balance of trade-offs between properties tends to vary across different tasks and users. Motivated by these varying needs, we aim to find explanations that make *optimal trade-offs* while allowing for *transparent control* over the balance between different properties. Unlike existing methods that encourage desirable properties implicitly through their design, our approach optimizes explanations explicitly for a linear mixture of multiple properties. By adjusting the mixture weights, users can control the balance between those properties and create explanations with precisely what is needed for their particular task.

1 INTRODUCTION

When explaining machine learning models, it is desirable for our explanations to satisfy various *properties*. For instance, explanations should be *faithful* and accurately reflect the actual computations carried out by the underlying model [Yeh et al., 2019]. We might also want them to be *robust* so that similar inputs result in similar explanations [Alvarez Melis and Jaakkola, 2018], or *smooth* so that similar dimensions in the input are assigned similar values in the explanation [Ajalloeian et al., 2022]. Some applications may require explanations to be simple with low *complexity* [Bhatt et al., 2020]. Accordingly, a range of metrics has been proposed to evaluate these properties [Chen et al., 2022, Wang et al., 2024].

Table 1: Different tasks require explanations with different properties. For instance, when auditing models, explanations need to be faithful no matter how complex so that users can investigate every detail and catch the smallest breaches in regulation. Meanwhile, counterfactual reasoning involves extrapolating a model's behavior to new cases, where robust explanations that capture global trends might be more useful than explanations faithful to local variations.

Task	**Property Preference**
Auditing models	Faithfulness $\succ$ Complexity [Nofshin et al., 2024]
Counterfactual reasoning	Robustness $\succ$ Faithfulness [Nofshin et al., 2024]
Detecting model biases	Smoothness $\succ$ Faithfulness [Colin et al., 2022]

While all of these properties can be useful, many of them are in tension with each other, making it difficult to achieve them simultaneously. For instance, reducing the complexity of an explanation often makes it less expressive, which can undermine its faithfulness [Bhatt et al., 2020]. Previous work has shown similar trade-offs between faithfulness vs. robustness [Tan and Tian, 2023], faithfulness vs. sensitivity [Bansal et al., 2020], and faithfulness vs. homogeneity [Balagopalan et al., 2022].

Not only these trade-offs exist, but it is also the case that different tasks or different users require different balances among these properties [Zhou et al., 2021, Liao et al., 2022, Nofshin et al., 2024]. For instance, users who work more closely with AI might prefer more faithful explanations despite their complexity, while non-AI experts might prefer explanations that are shorter and clearer instead Wang et al. [2024]. Likewise, studies show that users perform better with faithful explanations when auditing black-box models for compliance [Nofshin et al., 2024], robust explanations when performing counterfactual ("what-if") reasoning [Nofshin et al., 2024], or smooth explanations when trying to detect model biases (see Table 1) [Colin et al., 2022].

Motivated by these varying user needs, we aim to provide a method for finding explanations that make *optimal* trade-offs between different properties, where the balance between each property can be controlled *transparently*. We focus

Proceedings of the 41st Conference on Uncertainty in Artificial Intelligence (UAI 2025), PMLR 244:4092–4112.

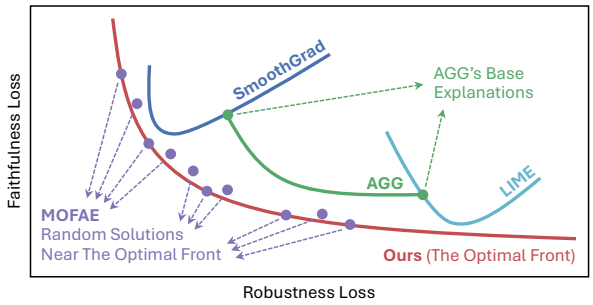


Figure 1: **Summary of Related Work.** Consider the trade-off between two properties. Methods without explicit optimization (like **SmoothGrad** and **LIME**) fail to reliably produce optimal explanations. Varying their hyperparameters leads to different explanations, but does not consistently balance one property againts the other. **AGG** can offer more direct control over the trade-off by aggregating multiple explanations, but its span and optimality are limited by the quality of its base explanations. **MOFAE** can generate a random set of near-optimal explanations, but provides little control over where exactly these explanations land along the optimality front (making it difficult to fine-tune the balance for an individual explanation). In contrast, **our approach** finds optimal explanations with transparent control across the entire optimality front.

specifically on feature attribution explanations [e.g. Ribeiro et al., 2016, Lundberg and Lee, 2017, Smilkov et al., 2017, Selvaraju et al., 2020]. This popular type of explanation assigns a contribution score to each input feature, capturing the impact of that feature on the final output. We achieve our goal by directly optimizing these contribution scores for a linear mixture of properties, where the mixture weights can be adjusted freely by the user.

We derive computationally efficient ways to perform such optimization in both inductive and transductive settings. In the inductive setting, we show that optimizing a combination of faithfulness, robustness, and smoothness can be expressed as an equivalent Gaussian process (GP) inference problem. While the inductive setting is more generally applicable, the transductive setting allows us to consider an even wider range of properties and efficient optimization strategies. In particular, in the transductive setting, we consider optimizing for complexity and alternative formulations of faithfulness and robustness from the literature.

Related Work: Limitations of Current Methods. Most feature attribution methods tend to target a specific property when generating explanations. For instance, SmoothGrad [Smilkov et al., 2017] improves robustness by averaging gradients around input points. Meanwhile, LIME [Ribeiro et al., 2016] prioritizes faithfulness by fitting local linear models at individual input points. Like SmoothGrad and LIME, methods are typically designed as forward computations, where the desired property is encouraged heuristically through the design of those computations. Explanations are almost never optimized directly for a target property or for a desired trade-off between multiple properties.

As a consequence, such methods often make implicit trade-offs against properties other than the one targeted. For instance, methods like LIME or SHAP [Lundberg and Lee, 2017], which focus on faithfulness, have been shown to generate explanations that are not always robust [Alvarez Melis and Jaakkola, 2018, Ghorbani et al., 2019, Slack et al., 2020]. Similarly, methods like SmoothGrad and GradCAM [Selvaraju et al., 2020], which focus on robustness, can yield explanations that lack faithfulness [Adebayo et al., 2018].

These implicit trade-offs make it impossible to control the balance between competing properties in a transparent manner. To give an example, Tan and Tian [2023] demonstrated for SmoothGrad that the balance between faithfulness and robustness is sensitive to a hyperparameter, δ, but how exactly δ affects this balance remains unclear (as we will see in the experiments, varying δ can lead to unintuitive changes in faithfulness and robustness). Furthermore, without explicit optimization, the resulting trade-off may not even be optimal or the method might fail to produce the intended property altogether. Frameworks for recommending explanations tailored to specific user needs [e.g. Cugny et al., 2022] often rely on tuning the hyperparameters of existing explanation methods, inheriting the same limitations.

In order to produce a desired property more reliably, Bhatt et al. [2020], Decker et al. [2024] proposed aggregating explanations generated by multiple methods. While the properties induced by individual methods can be unpredictable, with sufficient diversity, finding a desirable aggregation can be possible. The algorithm of Decker et al. [2024], AGG, searches for the best convex combination of some base explanations to optimize either for faithfulness or for robustness. AGG can be extended to target any arbitrary trade-off between these two properties as well, offering transparent control, however, the span and optimality of achievable trade-offs would remain constrained by the properties of the base explanations used (see Figure 1).

Wang et al. [2024] proposed a framework where explanations are directly optimized for multiple properties simultaneously (rather than a single mixture as in our approach). Their method, MOFAE, uses a genetic algorithm to generate a collection of explanations, each achieving a different but optimal trade-off. However, since this collection is generated through a stochastic process, MOFAE is ineffective at targeting a particular balance of properties or at fine-tuning the balance achieved by an individual solution (see Figure 1).

Contributions. We make three contributions: (i) We propose a new method called **POE (Property Optimized Explanations)**, which directly optimizes for a combination of desired properties. POE allow for fine-grained control over the trade-offs between competing objectives, enabling users

Table 2: **Summary of Property Definitions.** We consider faithfulness (gradient matching), robustness (average differences), and smoothness for both inductive and transductive settings. While the inductive setting is more generally applicable, the transductive setting allows us to consider additional definitions of faithfulness (function matching) and robustness (maximum difference) as well as complexity. $^{\dagger}s(\boldsymbol{x}, \boldsymbol{x}')$ and $S_{nn'} \doteq s(\boldsymbol{x}_n, \boldsymbol{x}_{n'})$ quantify the similarity between two inputs. $^{\ddagger}\tilde{S}_{dd'}$ quantifies the similarity between two input dimensions.

Property	Inductive Definition	Transductive Definition
Faithfulness *(Gradient Matching)*	**(a)** $\mathcal{L}_{\texttt{F-grad}}(E) = \int_\Omega \|E(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|_2^2 d\boldsymbol{x}$	**(d)** $\mathcal{L}_{\texttt{F-grad}}(W_E) = \sum_n \|\boldsymbol{w}_{E_n} - \nabla f(\boldsymbol{x}_n)\|_2^2$
Faithfulness *(Function Matching)*	–	**(e)** $\mathcal{L}_{\texttt{F-func}}(W_E) = \sum_n \|(\boldsymbol{w}_{E_n})^\top \boldsymbol{x}_n - f(\boldsymbol{x}_n)\|^2$
Robustness *(Average Difference)* †	**(b)** $\mathcal{L}_{\texttt{R-avg}}(E) = \iint_\Omega \|\mathrm{E}(\boldsymbol{x}) - \mathrm{E}(\boldsymbol{x}')\|_2^2 \; s(\boldsymbol{x}, \boldsymbol{x}') \; d\boldsymbol{x} d\boldsymbol{x}'$	**(f)** $\mathcal{L}_{\texttt{R-avg}}(W_E) = \sum_{n,n'} \|\boldsymbol{w}_{E_n} - \boldsymbol{w}_{E_{n'}}\|_2^2 \; S_{nn'}$
Robustness *(Maximum Difference)*	–	**(g)** $\mathcal{L}_{\texttt{R-max}}(W_E) = \sum_n \max_{n'} \|\boldsymbol{w}_{E_n} - \boldsymbol{w}_{E_{n'}}\|_2^2$
Smoothness ‡	**(c)** $\mathcal{L}_{\texttt{S}}(E) = \int_\Omega \sum_{d,d'} \|E_d(\boldsymbol{x}) - E_{d'}(\boldsymbol{x})\|^2 \; \tilde{S}_{dd'} \; d\boldsymbol{x}$	**(h)** $\mathcal{L}_{\texttt{S}}(W_E) = \sum_{n,d,d'} \|(\boldsymbol{w}_{E_n})_d - (\boldsymbol{w}_{E_n})_{d'}\|^2 \; \tilde{S}_{dd'}$
Complexity	–	**(i)** $\mathcal{L}_{\texttt{C}}(W_E) = \sum_n \|\boldsymbol{w}_{E_n}\|_1$

to create explanations with precisely the balance of properties needed for their task. (ii) We demonstrate that popular feature attributions methods (e.g. SmoothGrad, LIME) not only fail to reliably generate explanations with optimal properties, but also lack mechanisms for controlling the prioritization of properties. Among recent frameworks, AGG brings some control but does not ensure optimality, and MOFAE finds optimal explanations but does not allow for fine-grained control. (iii) Finally, we demonstrate through a variety of experiments that POE consistently yields explanations with optimal properties and easy-to-adjust trade-offs.

2 PROBLEM FORMULATION

Setting. We consider the problem of explaining some fixed function $f : \mathbb{R}^D \to \mathbb{R}$. We assume that we can query the function for an output $y = f(\boldsymbol{x})$ for any input $\boldsymbol{x}$ as well as a gradient $\nabla f(\boldsymbol{x})$. Depending on how explanations are requested, we consider two different settings: (i) the *inductive setting*, where the requests for explanations arrive in an online fashion (i.e. one input point $\boldsymbol{x}$ at a time), and (ii) the *transductive setting*, where we are given the full set of input points $\{\boldsymbol{x}_n\}_{n=1}^N$ that we will need to explain in advance. While the inductive setting is more generally applicable, it is also a more challenging setting (because properties like robustness that rely on simultaneously optimizing explanations at multiple inputs requires us to reason about the entire input domain); the assumption of the transductive setting will allow us to optimize for a wider range of properties.

Explanations. We restrict our focus to local explanations that are based on feature-attributions. This is so that we can give concrete definitions of properties such as faithfulness, robustness, and smoothness as an exemplar of our approach. In local feature attribution, an explanation $\boldsymbol{E} \in \mathbb{R}^D$ is a vector assigning an attribution score to each component of some input point $\boldsymbol{x}$. In the inductive settings, these explanations are given by a function $E : \mathbb{R}^D \to \mathbb{R}^D$ (from input to explanation). In the transductive setting, we do not assume an explicit explanation function, rather, for each input $\boldsymbol{x}_n$, we denote the explanation at $\boldsymbol{x}_n$ by $\boldsymbol{w}_{E_n} \in \mathbb{R}^D$ and the matrix of explanations for the N inputs as $W_E \in \mathbb{R}^{N \times D}$.

The General Problem. We are given a set of desired properties $\{\text{prop}_i\}$, each characterized by a loss function $\mathcal{L}_{\text{prop}_i}$. In the inductive setting, these losses are functions of E, and in the transductive setting, they are functions of W_E. Our objective is to find the explanation function E—or the explanation matrix W_E—that optimizes for the given properties (i.e. minimizes their characteristic loss functions):

$$E^* = \text{argmin}_E \textstyle\sum_i \lambda_{\text{prop}_i} \mathcal{L}_{\text{prop}_i}(E; f) \tag{1}$$

where λ_{prop_i} are weights that provide a transparent, intuitive way to manage trade-offs between different properties.

3 PROPERTY DEFINITIONS

In this paper, we optimize for a range of common properties that have been shown to be useful for downstream tasks in literature. In existing literature, the same conceptual property is mathematically formalized in many different ways [Chen et al., 2022, e.g. Alvarez Melis and Jaakkola [2018] vs. Yeh et al. [2019] for robustness]. We focus on the properties of faithfulness, robustness, smoothness, and complexity. We choose to work with these four properties because they are in tension, that is, it is not possible to maximize all four unless the function has constant gradients (i.e. it is linear). Thus, it is necessary for any explanation method to manage the trade-off between faithfulness, robustness, smoothness, and complexity in ways that are appropriate for specific tasks. We consider formalizations of these properties that have been well studied in literature as well as lend themselves to efficient optimization. Table 2 summarizes all the properties we consider and their corresponding losses.

Faithfulness. Faithfulness quantifies the extent to which an explanation accurately reflects the behavior of the function that is being explained. The class of explanations wherein one computes the marginal contribution of each dimension, by using small perturbations or by analytically computing input gradients, can be seen as optimizing faithfulness formalized as gradient matching [Table 2a, Baehrens et al., 2010, Simonyan, 2013]:

$$\mathcal{L}_{\texttt{F-grad}}(E) = \textstyle\int_\Omega \|E(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|_2^2 d\boldsymbol{x} \tag{2}$$

Many faithfulness metrics proposed in the literature can be reduced to gradient matching under certain settings. For example, the faithfulness metric in Tan and Tian [2023] is equivalent to gradient matching, when the perturbation is Gaussian and the similarity measure is Euclidean distance. A number of metrics define faithfulness as matching the function with linear approximations based on the explanation, e.g. local-fidelity [Yeh et al., 2019], local-accuracy [Tan and Tian, 2023], loss-based-fidelity [Balagopalan et al., 2022], see $\mathcal{L}_{\texttt{F-func}}$ in Table 2e. We will consider this alternative formalization in the transductive setting.

Robustness. Robustness quantifies the extent to which an explanation varies with respect to the input. A common way to formalize robustness as a metric is to take the weighted sum of pairwise differences between explanations for different inputs in the data. More formally, in the inductive setting, this can be written (Table 2b):

$$\mathcal{L}_{\texttt{R-avg}}(E) = \iint_{\Omega} \|\mathrm{E}(\boldsymbol{x}) - \mathrm{E}(\boldsymbol{x}')\|_2^2 \; s(\boldsymbol{x}, \boldsymbol{x}') \; d\boldsymbol{x} d\boldsymbol{x}' \quad (3)$$

where $s(\boldsymbol{x}, \boldsymbol{x}') \in \mathbb{R}$ is a weight that quantifies the similarity between $\boldsymbol{x}$ and $\boldsymbol{x}'$. Metrics like (Lipschitz) local-stability [Alvarez Melis and Jaakkola, 2018, Wang et al., 2020] are instances of the above equation, where we set the weights $s(\boldsymbol{x}, \boldsymbol{x}')$ to be a function of $\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2$. Notably, our robustness loss depends on the similarity of the explanation at input $\boldsymbol{x}$ to the explanation at input $\boldsymbol{x}'$. Thus, optimizing the robustness of an explanation for even *one* input requires assigning explanations to *many* other inputs. By explicitly optimizing explanations, our approach will ensure that all explanation assignments are consistent, a property not present in most existing explanation methods. We detail our choice of s in the appendix. Rather than averaged differences, metrics like max-sensitivity [Yeh et al., 2019] capture a notion of robustness defined by the maximum difference between the explanations at "neighboring" input points, see $\mathcal{L}_{\texttt{R-max}}$ in Table 2g. We will consider this alternative formalization in the transductive setting.

Smoothness. Smoothness or "internal robustness" captures the variability across different dimensions of an explanation that is assigned to a fixed input. Similar to robustness, suppose we are given a similarity matrix $\tilde{S}_{dd'}$ that tells how similar an input dimension $d \in [D]$ is to another input dimension $d' \in [D]$ (for instance, when inputs are images, this similarity can be related to the distance between two pixels). Then, we define the smoothness loss in a form similar to robustness (Table 2c):

$$\mathcal{L}_{\texttt{S}}(E) = \int_{\Omega} \sum_{d,d'} |E_d(\boldsymbol{x}) - E_{d'}(\boldsymbol{x})|^2 \; \tilde{S}_{dd'} \; d\boldsymbol{x} \quad (4)$$

Complexity. A common way to formalize explanation complexity is via sparsity, i.e. by counting the number of non-zero weights in the explanations. In the transductive setting, this can be captured through the ℓ_1-norm of explanations (Table 2i): $\mathcal{L}_{\texttt{C}}(W_E) = \sum_n \|\boldsymbol{w}_{E_n}\|_1$, which we prefer over ℓ_0-norm for efficient optimization.

4 INDUCTION: OPTIMIZING EXPLANATION FUNCTIONS

In this section, we tackle the inductive version of our problem: learning explanation functions that are optimized for a desired mixture of explanation properties. We instantiate the general problem in Equation 1 with three properties: faithfulness (`F-grad`), robustness (`R-avg`) and smoothness (`S`). Then, we drive a computationally efficient and mathematically consistent solution based on GP inference.

Instantiation. When linearly combined, the losses corresponding to these three properties ($\mathcal{L}_{\texttt{F-grad}}$, $\mathcal{L}_{\texttt{R-avg}}$, $\mathcal{L}_{\texttt{S}}$ in Table 2a–c) lead to the following optimization problem:

$$\begin{aligned} E^* = \mathrm{argmin}_E \; & \int_{\Omega} \lambda_{\texttt{F-avg}} \|E(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|_2^2 d\boldsymbol{x} \\ & + \lambda_{\texttt{R-avg}} \iint_{\Omega} \|E(\boldsymbol{x}) - E(\boldsymbol{x}')\|_2^2 s(\boldsymbol{x}, \boldsymbol{x}') d\boldsymbol{x} d\boldsymbol{x}' \\ & + \lambda_{\texttt{S}} \int_{\Omega} \sum_{d,d'} |E_d(\boldsymbol{x}) - E_{d'}(\boldsymbol{x})|^2 \tilde{S}_{dd'} d\boldsymbol{x} \end{aligned} \quad (5)$$

Here, the robustness loss in Equation 5 is what makes optimization challenging: When assigning an explanation even to one input, we must ensure that the explanation is similar to explanations for nearby inputs; those explanations, to be optimal, must in turn also be faithful, robust, and smooth. To get a consistent solution, we must reason over the entire space of explanation functions $E(\boldsymbol{x})$. This challenge is what sets the inductive setting apart from the transductive setting, where we only need to optimize a finite set of explanations.

Equivalence to Gaussian Process Regression. We overcome the above challenge via Gaussian process (GP) regression, which provides a theoretically grounded and computationally tractable framework for inference in function spaces. Below, we show that the solution to the optimization problem in Equation 5 can be expressed as the maximum aposteriori function of a multi-output GP, where gradients $\nabla f(\boldsymbol{x})$ can be viewed as observations of some latent explanation function $E(\boldsymbol{x})$, and the similarity measures $s, \tilde{S}$ can be written as the inverse of a multi-output kernel. Viewed as GP regression, we can drive a consistent, analytic solution to Equation 5 for any number of query points.

Formally, let us start with the following GP prior over explanation functions $E(\boldsymbol{x})$:

$$E \sim \mathrm{GP}(m, K), \quad m(\boldsymbol{x}) = \mu \mathbf{1}_D, \quad \mu \sim \mathcal{N}(0, \sigma^2) \quad (6)$$

where the mean m has a constant value $\mu \in \mathbb{R}$ for all inputs and all dimensions, and the kernel K is a matrix-valued function such that $\mathrm{cov}(E_d(\boldsymbol{x}), E_{d'}(\boldsymbol{x}')) = K_{dd'}(\boldsymbol{x}, \boldsymbol{x}')$ with its inverse denoted as K^{-1}. We interpret the gradient ∇f as a random observation of some latent E, which determines the likelihood for posterior inference:

$$\nabla f(\boldsymbol{x}) \sim \mathcal{N}(E(\boldsymbol{x}), \varsigma^2 I) \quad (7)$$

Then, we have the following equivalence:

Proposition 1. *For* $\{\boldsymbol{x}_n\}_{n=1}^N$, $\boldsymbol{x}_n \sim$ *Uniform*(Ω), *and* $\varsigma^2 = 1/N$, *the maximum aposteriori function*

$$\operatorname{argmax}_E \, p(E|\{\boldsymbol{x}_n, \nabla f(\boldsymbol{x}_n)\}) \tag{8}$$

approaches to the optimal explanation function

$$\begin{aligned}&\operatorname{argmin}_E \int_\Omega \|E(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|_2^2 d\boldsymbol{x} \\ &- \iint_\Omega \textstyle\sum_{d,d'} |E_d(\boldsymbol{x}) - E_{d'}(\boldsymbol{x}')|^2 K^{-1}_{dd'}(\boldsymbol{x}, \boldsymbol{x}') d\boldsymbol{x} d\boldsymbol{x}'\end{aligned} \tag{9}$$

as $\sigma^2 \to \infty$ *and* $N \to \infty$.

Proof. The complete proof can be found in the appendix. However, the key insight is to write the inductive optimization in Eq. 9 as the limiting case of a transductive optimization, which allows us to manipulate it algebraically to show its equivalence to the inference problem in Eq. 8. □

When the similarity measures in Equation 5 are precision function corresponding to kernels, that is $s(\boldsymbol{x}, \boldsymbol{x}') = -k^{-1}(\boldsymbol{x}, \boldsymbol{x}')$ and $\tilde{S} = -\tilde{K}^{-1}$ for some $k, \tilde{K}$, we can define the GP kernel K in Equation 6 such that

$$\begin{aligned}K^{-1}_{dd'}(\boldsymbol{x}, \boldsymbol{x}') = \tfrac{\lambda_{\texttt{R-avg}}}{\lambda_{\texttt{F-grad}}} \cdot k^{-1}(\boldsymbol{x}, \boldsymbol{x}') \cdot \mathbb{1}\{d = d'\} \\ + \tfrac{\lambda_{\texttt{S}}}{\lambda_{\texttt{F-grad}}} \cdot \mathbb{1}\{\boldsymbol{x} = \boldsymbol{x}'\} \cdot \tilde{K}^{-1}_d\end{aligned} \tag{10}$$

Then, the objective in the proposition (Equation 9) becomes equivalent to the objective in our explanation optimization problem (Equation 5). Thus, the solution to our optimization problem can be computed via GP inference (via Proposition 1). In particular, we can find efficient solutions to our optimization problem through approximate GP inference by using a set of *inducing points* $\{\boldsymbol{x}_n\}$. In our experiments, we investigate the sensitivity of our approach to selection of inducing points (specifically to their size N, and additionally in the appendix, to their distribution).

5 TRANSDUCTION: OPTIMIZING EXPLANATIONS FOR FIXED INPUTS

Optimization in the inductive setting is generally challenging because the explanation function must be consistent across all possible inputs. In this section, we describe how optimization can be made even more efficient in the transductive setting, where we have in advance, a fixed set of inputs $\boldsymbol{x}_n$ that we want to explain. In particular, we identify specific classes of properties for which the optimization problem in the transductive setting can be solved analytically or efficiently, and with guarantees, by recasting these problems as linear or quadratic programs.

Connection to the Inductive Setting. When we instantiate Equation 1 for the transductive setting using the same definitions of faithfulness (`F-grad`), robustness (`R-avg`), and smoothness (`S`) from Section 4, the optimization problem defines a quadratic program:

$$\begin{aligned}W^*_E = \operatorname{argmin}_{W_E} \lambda_{\texttt{F-grad}} \textstyle\sum_n \|\boldsymbol{w}_{E_n} - \nabla f(\mathbf{x}_n)\|_2^2 \\ + \lambda_{\texttt{R-avg}} \textstyle\sum_{n,n'} \|\boldsymbol{w}_{E_n} - \boldsymbol{w}_{E_{n'}}\|_2^2 S_{nn'} \\ + \lambda_{\texttt{S}} \textstyle\sum_{n,d,d'} |(\boldsymbol{w}_{E_n})_d - (\boldsymbol{w}_{E_n})_{d'}|^2 \tilde{S}_{dd'}\end{aligned} \tag{11}$$

Furthermore, we note that when we choose the similarity measures S and $\tilde{S}$ to be the negative precision matrices for some kernels K and $\tilde{K}$, we can still take advantage of Proposition 1 (when N is large!) and use GP inference to find a solution. Doing so, the faithfulness objective follows from the Gaussian likelihood in Equation 7, and the robustness and smoothness objectives are determined by our choice of kernel in the prior (Equation 6). This connection to GP inference provides theoretical grounding for the multi-property optimization problem in Equation 11: Even though we are optimizing these properties for a fixed set of inputs, the resulting explanations are consistent with the wider, inductive notions of faithfulness, robustness, and smoothness.

Additional Properties. In the transductive setting, we have the ability to efficiently optimize a large number of additional formalizations of properties by recasting them as linear or quadratic programs (as we no longer need the GP framework to keep track of functions). For example, we consider one additional formalization of faithfulness (*function matching*, `F-func` in Table 2e), one additional formalization of robustness (*maximum difference*, `R-max` in Table 2g) and an entirely new property: complexity (captured through *sparsity*, `C` in Table 2i).

- *Faithfulness as function matching.* This property is defined by a quadratic loss, $\sum_n \|\boldsymbol{w}_{E_n} - \nabla f(\boldsymbol{x}_n)\|_2^2$, and can be optimized with an unconstrained quadratic program.
- *Robustness as maximum difference.* The loss defining this property involves a max operation: $\sum_n \max_{n'} \|\boldsymbol{w}_{E_n} - \boldsymbol{w}_{E_{n'}}\|_2^2$. Optimizing it can be recast as a linear program with quadratic constraints:

$$\begin{aligned}&\textit{minimize}_{\; W_E, \Delta_n} \quad \textstyle\sum_n \Delta_n \\ &\quad \textit{s.t.} \; \|\boldsymbol{w}_{E_n} - \boldsymbol{w}_{E_{n'}}\|_2^2 \le \Delta_n \quad \forall n, n'\end{aligned} \tag{12}$$

- *Complexity as sparsity.* Optimizing the loss defining this property, $\sum_n \|W_{E_n}\|_1$ involves an unconstrained ℓ_1 optimization, which can be rewritten as a linear program with linear constraints:

$$\begin{aligned}&\textit{minimize}_{\; W_E, \Delta_{nd}} \quad \textstyle\sum_{n,d} \Delta_{nd} \\ &\quad \textit{s.t.} \; \Delta_{nd} \ge 0 \qquad\qquad\qquad \forall n, d \\ &\qquad\quad \Delta_{nd} \ge (W_{E_n})_d \ge -\Delta_{nd} \quad \forall n, d\end{aligned} \tag{13}$$

  When integrated with other objectives, it can be efficiently solved via sequential quadratic programs [Schmidt, 2005].

In all of these optimizations, the trade-off parameters of our method, that is $\lambda_{\texttt{F-grad}}$, $\lambda_{\texttt{F-func}}$, $\lambda_{\texttt{R-avg}}$, $\lambda_{\texttt{R-max}}$, $\lambda_{\texttt{S}}$,

and λ_{C}, allow us to explicitly and flexibly prioritize their corresponding properties. As we will show next with experiments, baselines like SmoothGrad, LIME, AGG, and MOFAE can only do so in limited ways, if at all.

6 EXPERIMENTS

First, we focus on the transductive setting (Section 6.1, and demonstrate that our method can optimize for desired properties and manage the trade-offs between competing properties, while other baselines cannot reliably. We highlight how different formalizations of a property can be used in our method while still providing similar explanations across alternative formalizations. Afterwards in Section 6.2, we show that the same optimization can also be performed efficiently in the inductive setting, by selecting a small subset of the input over which to perform GP inference. Specifically, we check that the explanations generated for a test set inductively, based on a separate set of inducing points, approach those that could have been generated transductively if the test points were available ahead of time. This results means the two settings provide solutions that are consistent with each other, and the inductive setting enjoys the same benefits we highlight for the transductive setting. Our implementation is available at: `https://github.com/dtak/POE`

6.1 TRANSDUCTIVE EXPERIMENTS

Baselines. We compare our approach to two popular explanation methods, SmoothGrad, and LIME as well as the two multi-property frameworks that we have discussed in our introduction, AGG and MOFAE:

- *SmoothGrad* is defined by averaging gradients of f over a set of points sampled from a neighborhood of $\boldsymbol{x}_n$:

$$\boldsymbol{w}_{\text{SG}_n} = \tfrac{1}{S}\sum_{s=1}^{S} \nabla f(\tilde{\boldsymbol{x}}_{n,s}),\ \tilde{\boldsymbol{x}}_{n,s} \sim \mathcal{N}(\boldsymbol{x}_n, \delta_{\text{SG}}^2 I) \quad (14)$$

  where the parameter δ_{SG} controls the size of the neighborhood, hence the degree of smoothing applied to ∇f.

- *LIME* approximates f at $\boldsymbol{x}_n$ with a linear model trained on points $\tilde{\boldsymbol{x}}_{n,s}$ sampled from a δ_{LIME}-neighborhood of $\boldsymbol{x}_n$:

$$\boldsymbol{w}_{\text{LIME}_n} = \text{argmin}_{\boldsymbol{w}} \sum_{s=1}^{S} (f(\tilde{\boldsymbol{x}}_{n,s}) - \boldsymbol{w}^\top \tilde{\boldsymbol{x}}_{n,s})^2 \quad (15)$$

- *AGG* linearly combines a base set of explanations $\{E^{(m)}\}$ with weights α_m to optimize for a desired property $\mathcal{L}$:

$$\min_{\substack{\alpha_m : \Sigma_m \alpha_m = 1 \\ \alpha_m \geq 0}} \mathcal{L}(W_{\text{AGG}} \doteq \textstyle\sum_m \alpha_m W_{E^{(m)}}) \quad (16)$$

  Like our approach, AGG can also manage trade-offs between different properties if we set $\mathcal{L} = \sum_i \lambda_{\text{prop}_i} \mathcal{L}_{\text{prop}_i}$. However, the range and the optimality of trade-offs it can achieve would naturally be limited by the initial properties of the base explanations $\{E^{(m)}\}$. We initialize this set via SmoothGrad and LIME with varying hyperparameters (picking three explanations from each method).

- *MOFAE* tackles multiple properties at once and searches for Pareto optimal explanations—those are explanations, W_{MOFAE}, which are better than any other explanation in terms of at least one property:

$$\forall W_{E'},\ \exists i,\ \mathcal{L}_{\text{prop}_i}(W_{E'}) < \mathcal{L}_{\text{prop}_i}(W_{\text{MOFAE}}) \quad (17)$$

  However, MOFAE relies on a genetic algorithm to achieve this, and as such, returns random explanations along the Pareto front without any mechanism to control which properties are prioritized. We initialize MOFAE using the same base explanations as AGG but supplemented with 100 additional SmoothGrad explanations (as the genetic algorithm requires a significantly larger initial set).

Functions. We consider a number of functions with different degrees of curvature, and periodicity/quasi-periodicity. Specifically, we consider:

- *Polynomials:* $f(\boldsymbol{x}) = \sum_d (\boldsymbol{x})_d^3$ (*Cubic*), $f(\boldsymbol{x}) = \sum_d (\boldsymbol{x})_d^3 + \sum_{d,d',d''} (\boldsymbol{x})_d (\boldsymbol{x})_{d'} (\boldsymbol{x})_{d''}$ (*Cubic with Interactions*).
- *Periodic:* $f(\boldsymbol{x}) = \sum_d \sin((\boldsymbol{x})_d)$.
- *Quasi-Periodic:* We create quasi-periodic functions by modifying the above sinusoidal with various non-periodic terms, $f(\boldsymbol{x}) = \sum_d (\boldsymbol{x})_d + \sin(3(\boldsymbol{x})_d)$ (*with Linear Term*), $f(\boldsymbol{x}) = \sum_d (\boldsymbol{x})_d^2/10 + \sin(3(\boldsymbol{x})_d)$ (*with Quadratic Term*), $f(\boldsymbol{x}) = \sum_d \sin(e^{(\boldsymbol{x})_d/10})$ (*with Exponentiated Inputs*), $f(\boldsymbol{x}) = (\boldsymbol{x})_d + \sum_d \sin(e^{(\boldsymbol{x})_d/10})$ (*with Linear Terms and Exponentiated Inputs*).
- *Exponential:* $f(\boldsymbol{x}) = \sum_d e^{(\boldsymbol{x})_d}$.

In addition to these simple functions, we also consider neural networks with one hidden layer in the appendix as well as pretrained convolutional neural networks for image classification later in Section 6.2 for the inductive setting. As query points, we choose $N = 100^D$ points from a D-dimensional grid with values in $(-5, 5)$ for each dimension. For our main experiments, we set $D = 3$ but experiments for larger D can be found in the appendix. For robustness losses, we consider a uniform similarity measure such that $S_{nn'} = 1$. Finally, for our method, we vary the trade-off parameters λ_{prop_i} so that they always sum to one.

Results. In Figure 2, we optimize for faithfulness (gradient matching) and robustness (average difference), varying λ for us and AGG. We also run SmoothGrad and LIME, varying δ. In Figure 3, we consider complexity in addition to faithfulness and robustness.

Intuitive interpretations of the hyperparameters of SmoothGrad and LIME do not always hold. A number of works show that δ, the sampling hyper-parameter of LIME and SmoothGrad, affect the robustness of these methods. In particular, when the δ is larger, LIME and SmoothGrad generate

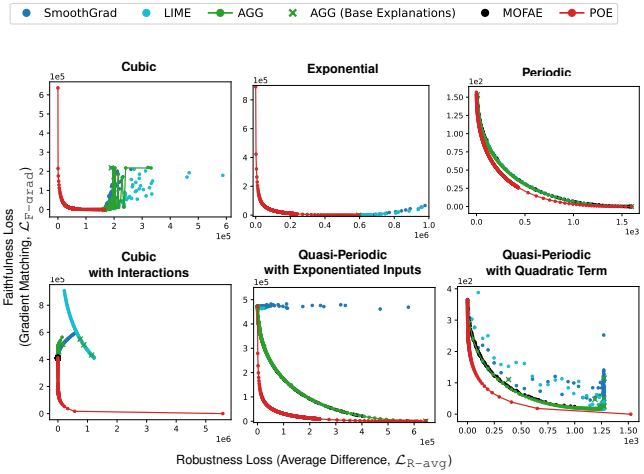


Figure 2: **Faithfulness vs. Robustness.** For most functions, POE is the only method capable of generating explanations that cover the entire optimality front. SmoothGrad and LIME do not always result in a faithfulness-robustness balance, sometimes outputting strictly worse explanations when their hyperparameters are varied (e.g. *Exponential*). AGG is constrained by the limited range of these methods. MOFAE fail to consistently cover the optimality front, often missing large portions (e.g. *Exponential*) or even outputting suboptimal explanations (e.g. *Cubic with Interactions*).

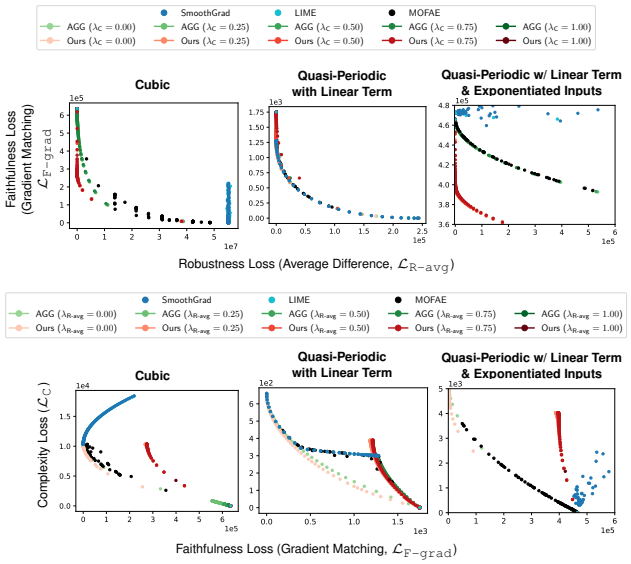


Figure 3: **Faithfulness vs. Robustness vs. Complexity.** In the top row, we report faithfulness vs. robustness for fixed values of λ_{C}. In the bottom row, we report complexity vs. faithfulness for fixed values of $\lambda_{\text{R-avg}}$.

explanations that are less sensitive to local perturbations. However, robustness in these analyses is formalized as local-Lipschitzness, and it is not clear how their insight generalize to other formalizations. For instance, in Figure 2 for *Cubic*, we see that the robustness of SmoothGrad (as defined in Equation 3) does not depend on δ at all—increasing δ does not smooth the explanations, only reduces their faithfulness.

Without explicit optimization, SmoothGrad and LIME can fail to find robust and faithful explanations. In Figure 2 for *Exponential* and *Quasi-Periodic with Exponentiated Inputs*, we see that varying δ not only fails to control the balance between faithfulness and robustness, but also might lead to strictly worse explanations in terms of both properties.

AGG is limited by the quality of its base explanations. Although AGG, similar to our approach, allows transparent control over different properties, the optimality of the trade-offs it can achieve is constrained by the properties of its base explanations. For instance, in Figure 2 for *Cubic*, we see that AGG is less effective in achieving more moderate trade-offs between faithfulness and robustness since its base explanations are either extremely faithful or extremely robust.

MOFAE does not offer any control over trade-offs and might not always cover the complete range of possible trade-offs. Since it uses a genetic algorithm to search for explanations, MOFAE essentially produces a random collection of explanations without offering any control over where these explanations land in terms of their properties. Sometimes, they do not achieve optimal trade-offs at all (e.g. Figure 2, *Cubic with Interactions*). At other times, they only offer a limited range of trade-offs, leaving portions of the Pareto front sparsely populated (e.g. Figure 2, *Cubic* and *Exponential*, where explanations with low robustness loss are missed).

Transparent Trade-offs between Properties. In contrast to all our baselines, POE can consistently find optimal trade-offs between different properties, but even more importantly, it provides fine control over these trade-offs through hyperparameters λ_{prop_i}. POE makes it significantly easier to include additional properties to the optimization as well, see results for complexity in Figure 3. The addition of this third property does not affect the explanations generated by SmoothGrad and LIME since they do not optimize for any property. Similarly, the base explanation set of AGG does not get any richer either. It increases the dimensionality of the optimality front that MOFAE needs to cover, which makes its coverage even worse. For instance, we see in Figure 3 for *Quasi-Periodic with Exponentiated Inputs*, MOFAE covers the faithfulness-robustness front well (top row) but misses a large portion of solutions that have low complexity loss (bottom row). Meanwhile, our approach with $\lambda_{\text{R-avg}} = 0$ covers the faithfulness-complexity front as well.

6.2 INDUCTIVE EXPERIMENTS

Quantitative Evaluation. Our goal is to check the consistency between the transductive and inductive settings. We optimize for faithfulness (gradient-matching) and robustness (average differences), fixing $\lambda_{\text{F-grad}} = 1$ and varying $\lambda_{\text{R-avg}}$. We report the total loss as we vary the number of inducing points N that are used for GP inference, comparing

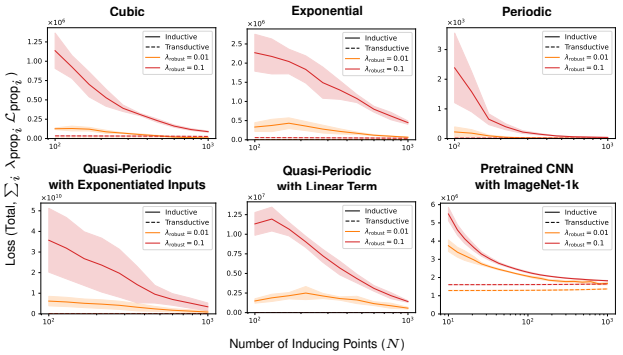


Figure 4: As the number of inducing points increases, inductive solutions obtained via GP inference (optimizing faithfulness and robustness) approach to transductive solutions on an exponential scale.

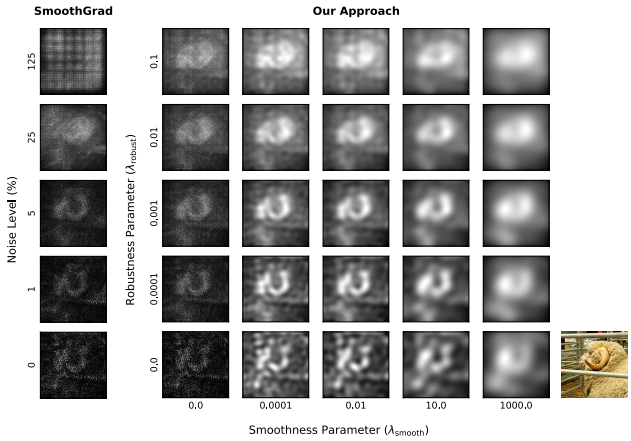


Figure 5: Our approach can achieve an arbitrary balance between faithfulness, robustness, and smoothness. Meanwhile, SmoothGrad can only manage the trade-off between faithfulness and robustness, and does not improve smoothness.

this with the total loss achievable transductively. We investigate the impact of how the inducing points are selected in the appendix. For now, just like in Proposition 1, we assume that they share the same distribution as our query points.

In addition to functions from Section 6.1, we also consider a *Convolution Neural Network (CNN)*, specifically the model architecture ResNet-50 [He et al., 2016] pretrained on the dataset ImageNet-1k [Russakovsky et al., 2015]. For quantitative results, we sample 1000 query images from ImageNet-1k and up to $N = 1000$ separate images to be used as inducing points. Note that, for these images, $D = 244 \times 244$. For the robustness loss, we consider similarity measures obtained by inverting Guassian kernels, $s = k^{-1}$ and $k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-0.5\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2/\Lambda^2)$.

In Figure 4, we show that inductive solutions approach transductive solutions exponentially fast as the number of inducing points are increased. This means the benefits we have seen so far can also be achieved in the inductive setting (although for a more restricted set of properties).

Qualitative Evaluation. Finally, we also provide a visual example to highlight the control over different properties provided by POE. We pick a single image from ImageNet-1k (one more example in the appendix) and compare the explanations generated by our approach against SmoothGrad. Using POE, we optimize for faithfulness, robustness, and smoothness via GP inference, using perturbations of the original image as inducing points (uniform perturbations as opposed to Gaussian perturbations). We fix $\lambda_{\texttt{F-grad}} = 1$ and vary $\lambda_{\texttt{R-avg}}$ and $\lambda_{\texttt{S}}$. For SmoothGrad, we can only vary δ in Equation 14 (using Gaussian perturbations to match the Gaussian kernel in our approach).

In Figure 5, we see that POE can achieve an arbitrary balance between faithfulness, robustness, and smoothness. Furthermore, the range of explanations we obtain shows that robustness and smoothness each capture distinct aspects of what makes an explanation "noisy". While the main motivation behind SmoothGrad is to de-noise explanations, it attributes noisiness solely to sharp variations in the gradients around an input point, thereby only improving robustness while missing the smoothness aspect entirely. Meanwhile, users seeking de-noised explanations might have different preferences regarding which aspect to prioritize. Without a precise definition of what constitutes a noisy explanation, and without an explicit optimization targeting that definition, it is not clear how to modify SmoothGrad to also promote smoothness alongside robustness, whereas in our approach, noisiness can be expressed as a specific mixture of robustness and smoothness based on user needs.

7 ETHICAL AND SOCIETAL IMPLICATIONS

As argued by Alpsancar et al. [2024], explanations are a means to an end, they are valuable not for their own sake, but for how they support broader goals such as fairness, safety, accountability and trust in machine learning systems. Our work is intended to help users tailor explanations to serve these goals more effectively. We believe that our approach can give end users more control over how explanations align with their specific needs. For example, in a recent work Nofshin et al. [2024] showed that when attempting to detect bias in a model, a less compact but more faithful explanation is essential. With our approach, the properties of an explanation – how faithful, how compact, etc– are transparent. This provides a means for users to know what the explanation can and cannot be used for, whether that is to help mitigating bias and supporting fairness in real-world applications, safety, or any other task.

8 CONCLUSION AND FUTURE WORK

A growing number of studies show that different downstream applications may require explanations with different properties. Yet existing feature attribution methods like

SmoothGrad and LIME neither directly optimize for arbitrary explanation properties, nor can they manage trade-offs between properties that are in tension. Multi-property frameworks like AGG and MOFAE are either constrained by the limitations of these methods or do not offer mechanisms to control trade-offs when optimizing for properties.

In our work, we introduced POE, a framework that can efficiently optimize for a set of desired explanation properties. In addition, POE can also explicitly and intuitively manage trade-offs between competing properties through its hyperparameters. We addressed both settings in which we have the entire dataset that needs explanations (the transductive setting) as well as settings in which we must generalize our property-optimized explanations to unseen query points (the inductive setting).

There are numerous potentially useful explanation properties. Some of those could use the same ideas in our work straightforwardly; others will require algorithmic innovation. Developing such methods is an interesting direction for future work.

Another limitation of our framework, shared with many feature-attribution explanation methods, is scalability. While POE is faster than existing methods for some properties, it becomes slower when solving a quadratic program is required. We see efficiently optimizing for a range of properties e.g. with clever amortization schemes as a ripe direction for future work.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. IIS-2007076. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1750358. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation

Supported by NIH award 5R01MH123804-02.

References

Solar Flare. UCI Machine Learning Repository, 1989. DOI: https://doi.org/10.24432/C5530G.

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.

Ahmad Ajalloeian, Seyed-Mohsen Moosavi-Dezfooli, Michalis Vlachos, and Pascal Frossard. On smoothed explanations: Quality and robustness. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 15–25, 2022.

Suzana Alpsancar, Heike M Buhl, Tobias Matzner, and Ingrid Scharlau. Explanation needs and ethical demands: unpacking the instrumental value of xai. *AI and Ethics*, pages 1–19, 2024.

David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

Aparna Balagopalan, Haoran Zhang, Kimia Hamidieh, Thomas Hartvigsen, Frank Rudzicz, and Marzyeh Ghassemi. The road to explainability is paved with bias: Measuring the fairness of explanations. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1194–1206, 2022.

Naman Bansal, Chirag Agarwal, and Anh Nguyen. Sam: The sensitivity of attribution methods to hyperparameters. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 8673–8683, 2020.

Umang Bhatt, Adrian Weller, and José M. F. Moura. Evaluating and aggregating feature-based model explanations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3016–3022, 2020.

Zixi Chen, Varshini Subhash, Marton Havasi, Weiwei Pan, and Finale Doshi-Velez. What makes a good explanation?: A harmonized view of properties of explanations. *arXiv preprint arXiv:2211.05667*, 2022.

Julien Colin, Thomas Fel, Rémi Cadène, and Thomas Serre. What i cannot predict, i do not understand: A human-centered evaluation framework for explainability methods. *Advances in neural information processing systems*, 35: 2832–2845, 2022.

Robin Cugny, Julien Aligon, Max Chevalier, Geoffrey Roman Jimenez, and Olivier Teste. Autoxai: A framework to automatically select the most adapted xai solution. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 315–324, 2022.

Thomas Decker, Ananta R Bhattarai, Jindong Gu, Volker Tresp, and Florian Buettner. Provably better explanations with optimized aggregation of feature attributions. *arXiv preprint arXiv:2406.05090*, 2024.

Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3681–3688, 2019.

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.

Q Vera Liao, Yunfeng Zhang, Ronny Luss, Finale Doshi-Velez, and Amit Dhurandhar. Connecting algorithmic research and usage contexts: a perspective of contextualized evaluation for explainable ai. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 10, pages 147–159, 2022.

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

Eura Nofshin, Esther Brown, Brian Lim, Weiwei Pan, and Finale Doshi-Velez. A sim2real approach for identifying task-relevant properties in interpretable machine learning. *arXiv preprint arXiv:2406.00116*, 2024.

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.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

Mark Schmidt. Least squares optimization with l1-norm regularization. *CS542B Project Report*, 504(2005):195–221, 2005.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: visual explanations from deep networks via gradient-based localization. *International journal of computer vision*, 128:336–359, 2020.

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

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

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

Zeren Tan and Yang Tian. Robust explanation for free or at the cost of faithfulness. In *International Conference on Machine Learning*, pages 33534–33562, 2023.

Zifan Wang, Haofan Wang, Shakul Ramkumar, Piotr Mardziel, Matt Fredrikson, and Anupam Datta. Smoothed geometry for robust attribution. *Advances in neural information processing systems*, 33:13623–13634, 2020.

Ziming Wang, Changwu Huang, Yun Li, and Xin Yao. Multi-objective feature attribution explanation for explainable machine learning. *ACM Transactions on Evolutionary Learning and Optimization*, 4(1):1–32, 2024.

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.

Jianlong Zhou, Amir H Gandomi, Fang Chen, and Andreas Holzinger. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10(5):593, 2021.

Transparent Trade-offs between Properties of Explanations (Supplementary Material)

Hiwot Belay Tadesse[1] **Alihan Hüyük**[2] **Yaniv Yacoby**[3] **Weiwei Pan**[4] **Finale Doshi-Velez**[5]

[1, 2, 4, 5]John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA
[3]Wellesley College, Wellesley, MA, USA

A CHOICE OF SIMILARITY FUNCTION

For robustness losses, we need to choose a similarity function $s(\boldsymbol{x}, \boldsymbol{x}')$. Certain similarity functions lend themselves to more efficient optimization. In particular, in Section 4, we showed that when $s(\boldsymbol{x}, \boldsymbol{x}') = -k^{-1}(\boldsymbol{x}, \boldsymbol{x}')$ is the precision of a kernel function k, optimization in the inductive setting becomes equivalent to Gaussian process inference. As an example, in this work, we consider similarity measures that are induced by Gaussian kernels: $k(\boldsymbol{x}, \boldsymbol{x}'; \Lambda) = \exp(-\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2 / 2\Lambda^2)$. In practice, we note that the choice of s or an associated kernel k should be informed by domain knowledge (e.g. capture a notion of similarity between inputs that is meaningful for the user's downstream task).

B PROOF OF PROPOSITION 1

For a sufficiently large N, the objective in Equation 9 can be written as:

$$\operatorname{argmin}_E \frac{1}{N} \sum\nolimits_n \|E(\boldsymbol{x}_n) - \nabla f(\boldsymbol{x}_n)\|_2^2 - \frac{1}{N^2} \sum\nolimits_{n,n,d,d'} |E_d(\boldsymbol{x}_n) - E_{d'}(\boldsymbol{x}_{n'})|^2 K^{-1}_{dd'}(\boldsymbol{x}_n, \boldsymbol{x}'_n) \tag{18}$$

We show that, as $N \to \infty$ and $\sigma^2 \to \infty$, maximizing $\log p(E|\{\boldsymbol{x}_n, \nabla f(\boldsymbol{x}_n)\})$ becomes equivalent to this objective.

For simplicity, we sitch to a vectorized notation so that we can work with a single-dimensional GP: $\boldsymbol{E} = [E(\boldsymbol{x}_1) \dots E(\boldsymbol{x}_N)] \in \mathbb{R}^{ND}$, $\boldsymbol{\nabla f} = [\nabla f(\boldsymbol{x}_1) \dots \nabla f(\boldsymbol{x}_N)] \in \mathbb{R}^{ND}$, and

$$\boldsymbol{K} = \begin{bmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}_1) & \dots & K(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_N, \boldsymbol{x}_1) & \dots & K(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix} \in \mathbb{R}^{ND \times ND} \tag{19}$$

Let us first consider only the prior term in the GP, marginalizing out the random mean explanation μ from the joint distribution:

$$p(\boldsymbol{E}, \mu) = p(\boldsymbol{E}|\mu)p(\mu) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{E} - \mu \mathbf{1}_{ND})^\top \boldsymbol{K}^{-1}(\boldsymbol{E} - \mu \mathbf{1}_{ND})\right\} \exp\left\{-\frac{1}{2} \cdot \frac{\mu^2}{\sigma^2}\right\} \tag{20}$$

where $\mathbf{1}_{ND} \in \mathbb{R}^{ND}$ is a vector of ones (written as $\mathbf{1}$ henceforth). By marginalizing out μ in Equation 20 (after completing the square, followed by some algebra), we obtain $p(E)$ as the following Gaussian:

$$p(\boldsymbol{E}) \propto \exp\left\{ -\frac{1}{2} \boldsymbol{E}^\top \Bigg(\underbrace{\boldsymbol{K}^{-1} - \frac{\boldsymbol{K}^{-1}\mathbf{1}\mathbf{1}^\top \boldsymbol{K}^{-1}}{\mathbf{1}^\top \boldsymbol{K}^{-1}\mathbf{1} + 1/\sigma^2}}_{\doteq \Sigma^{-1}} \Bigg) \boldsymbol{E} \right\} \tag{21}$$

We denote the covariance of $p(\boldsymbol{E})$ by Σ. This marginalization can be interpreted as a new Gaussian prior on the explanations $\boldsymbol{E}$; we will still have a multivariate Gaussian—and associated GP—with this new kernel.

Next, we note that the posterior mean of multivariate Gaussian $p(\boldsymbol{E}|\nabla \boldsymbol{f})$ is the solution to the least square regression problem, regularized by the precision matrix Σ^{-1} of the prior $p(E)$ that we just derived above in Equation 21:

$$\max_{\boldsymbol{E}} \log p(\boldsymbol{E}|\nabla \boldsymbol{f}) \equiv \min_{\boldsymbol{E}} \frac{1}{s^2}(\boldsymbol{E} - \nabla \boldsymbol{f})^\top (\boldsymbol{E} - \nabla \boldsymbol{f}) + \boldsymbol{E}^\top \Sigma^{-1} \boldsymbol{E} \tag{22}$$

$$\equiv \min_{\boldsymbol{E}} N(\boldsymbol{E} - \nabla \boldsymbol{f})^\top (\boldsymbol{E} - \nabla \boldsymbol{f}) + \boldsymbol{E}^\top \Sigma^{-1} \boldsymbol{E} \tag{23}$$

$$\equiv \min_{\boldsymbol{E}} \frac{1}{N}(\boldsymbol{E} - \nabla \boldsymbol{f})^\top (\boldsymbol{E} - \nabla \boldsymbol{f}) + \frac{1}{N^2} \boldsymbol{E}^\top \Sigma^{-1} \boldsymbol{E} \tag{24}$$

With a little algebraic manipulation, the regularization term $E^\top \Sigma^{-1} E$ can be further expanded as follows:

$$\boldsymbol{E}^\top \Sigma^{-1} \boldsymbol{E} = \underbrace{-\frac{1}{2} \sum_{\ell=1}^{ND} \sum_{\ell'=1}^{ND} |\boldsymbol{E}_\ell - \boldsymbol{E}_{\ell'}|^2 \Sigma^{-1}_{\ell\ell'}}_{\text{Term 1}} + \underbrace{\sum_{\ell=1}^{ND} \left(|\boldsymbol{E}_\ell|^2 \sum_{\ell'=1}^{ND} \Sigma^{-1}_{\ell\ell'} \right)}_{\text{Term 2}} \tag{25}$$

We first observe that the second term goes to zero as the variance of the explanation mean σ^2 goes to infinity. Specifically, Term 2 in Equation 25 can be expanded as:

$$\sum_\ell |\boldsymbol{E}_\ell|^2 \left(\delta_\ell^\top \boldsymbol{K}_\ell^{-1} \mathbf{1} - \delta_\ell^\top \boldsymbol{K}_\ell^{-1} \mathbf{1} \left(\frac{\mathbf{1}^\top \boldsymbol{K}^{-1} \mathbf{1}}{\mathbf{1}^\top \boldsymbol{K}^{-1} \mathbf{1} + 1/\sigma^2} \right) \right) \tag{26}$$

where δ_ℓ is a vector of zeros except for the ℓ-th element, which is one. As σ^2 goes to infinity (representing an improper, uninformative prior over the value of the mean explanation), the fraction goes to one and the first and second terms in the kernel expression cancel.

Thus, we are left with only the first term:

$$\boldsymbol{E}^\top \Sigma^{-1} \boldsymbol{E} \approx \underbrace{-\frac{1}{2} \sum_{\ell=1}^{ND} \sum_{\ell'=1}^{ND} |\boldsymbol{E}_\ell - \boldsymbol{E}_{\ell'}|^2 \Sigma^{-1}_{\ell\ell'}}_{\text{Term 1}} \tag{27}$$

We are going to finish our proof by showing that Σ^{-1} tends to $\boldsymbol{K}^{-1}$ as $\sigma^2 \to \infty$ and $N \to \infty$. Let $r = \boldsymbol{K}\mathbf{1}$ and $r' = \boldsymbol{K}^{-1}\mathbf{1}$ be the vector of row sums for $\boldsymbol{K}$ and $\boldsymbol{K}^{-1}$ respectively. Now, as $\sigma^2 \to \infty$, we can re-write Σ^{-1} as

$$\Sigma^{-1}_{\ell\ell'} = \boldsymbol{K}^{-1}_{\ell\ell'} - \frac{\delta_\ell^\top (\boldsymbol{K}^{-1}\mathbf{1})(\boldsymbol{K}^{-1}\mathbf{1})^\top \delta_{\ell'}}{\mathbf{1}^\top (\boldsymbol{K}^{-1}\mathbf{1}) + 1/\sigma^2} = \boldsymbol{K}^{-1}_{\ell,\ell'} - \frac{r'_\ell r'_{\ell'}}{\sum_s r'_s + 1/\sigma^2} \approx \boldsymbol{K}^{-1}_{\ell\ell'} - \frac{r'_\ell r'_{\ell'}}{\sum_s r'_s} \tag{28}$$

Note that, if $r_{\text{lower}} \leq r_s \leq r_{\text{upper}}$ for all s, then $1/r_{\text{upper}} \leq r'_s \leq 1/r_{\text{lower}}$ for all s. Letting

$$K_{\max} \doteq \max_{\boldsymbol{x},\boldsymbol{x}',d,d'} K_{dd'}(\boldsymbol{x}, \boldsymbol{x}') \tag{29}$$

$$K_{\min} \doteq \min_{\boldsymbol{x},d} K_{dd}(\boldsymbol{x}, \boldsymbol{x}) \tag{30}$$

we have $K_{\min} \leq r_s \leq NDK_{\max}$ hence $1/NDK_{\max} \leq r'_s \leq 1/K_{\min}$. Using these bounds, we obtain

$$\frac{K_{\min}}{N^3 D^3 K^2_{\max}} \leq \frac{r'_\ell r'_{\ell'}}{\sum_s r'_s} \leq \frac{(1/\min_s r_s)^2}{ND(1/\max_s r_s)} = \frac{\max_s r_s/N}{N^2 D \cdot (\min_s r_s/N)^2} \tag{31}$$

The left-hand side goes to zero as $N \to \infty$. For the right-hand side, let

$$n[s] = \lceil s/D \rceil \in \{1, \ldots, N\} \tag{32}$$

$$d[s] = (s \mod D) + 1 \in \{1, \ldots, D\} \tag{33}$$

be the original dimensions corresponding to the flattened dimension $s \in \{1, \ldots, ND\}$. Then, for input points $\boldsymbol{x}_n \sim \mathcal{P}$ that are i.i.d. and as $N \to \infty$, we have

$$\frac{r_s}{N} = \frac{1}{N} \sum\nolimits_{s'} \boldsymbol{K}_{ss'} = \frac{1}{N} \sum\nolimits_{n',d'} K_{d[s],d'}(\boldsymbol{x}_{n[s]}, \boldsymbol{x}_{n'}) \to \mathbb{E}_{\boldsymbol{x}' \sim \mathcal{P}} \left[\sum\nolimits_{d'} K_{d[s],d'}(\boldsymbol{x}_{n[s]}, \boldsymbol{x}') \right] \doteq \bar{K}_{d[s]}(\boldsymbol{x}_{n[s]}) \tag{34}$$

Note that $\bar{K}_d(\boldsymbol{x})$ is a function independent of $\{\boldsymbol{x}_n\}$ and N with maxima $\bar{K}_{\max} = \max_{\boldsymbol{x},d} \bar{K}_d(\boldsymbol{x})$ and minima $\bar{K}_{\min} = \min_{\boldsymbol{x},d} \bar{K}_d(\boldsymbol{x})$ independent of $\{\boldsymbol{x}_n\}$ and N as well. For sufficiently large N, we have

$$\frac{\max_s r_s/N}{N^2 D \cdot (\min_s r_s/N)^2} \approx \frac{\max_s \bar{K}_{d[s]}(\boldsymbol{x}_{n[s]})}{N^2 D \cdot (\min_s \bar{K}_{d[s]}(\boldsymbol{x}_{n[s]}))^2} \leq \frac{\bar{K}_{\max}}{N^2 D \bar{K}_{\min}^2} \approx 0 \tag{35}$$

C ADDITIONAL TRANSDUCTIVE SETTING RESULTS

C.1 SHAP

SHAP Lundberg and Lee [2017] generates explanations by sampling subsets of features to quantify their contributions. When comparing SHAP's faithfulness and robustness losses to other explanation methods—including our proposed method, POE we observe distinct behavior which for some functions the SHAP explanations have a higher robustness and faithfulness loss. One potential reason for this difference could be SHAP's lack of tunable hyperparameters, which sets it apart from alternative methods. We plan to explore this in our future work.

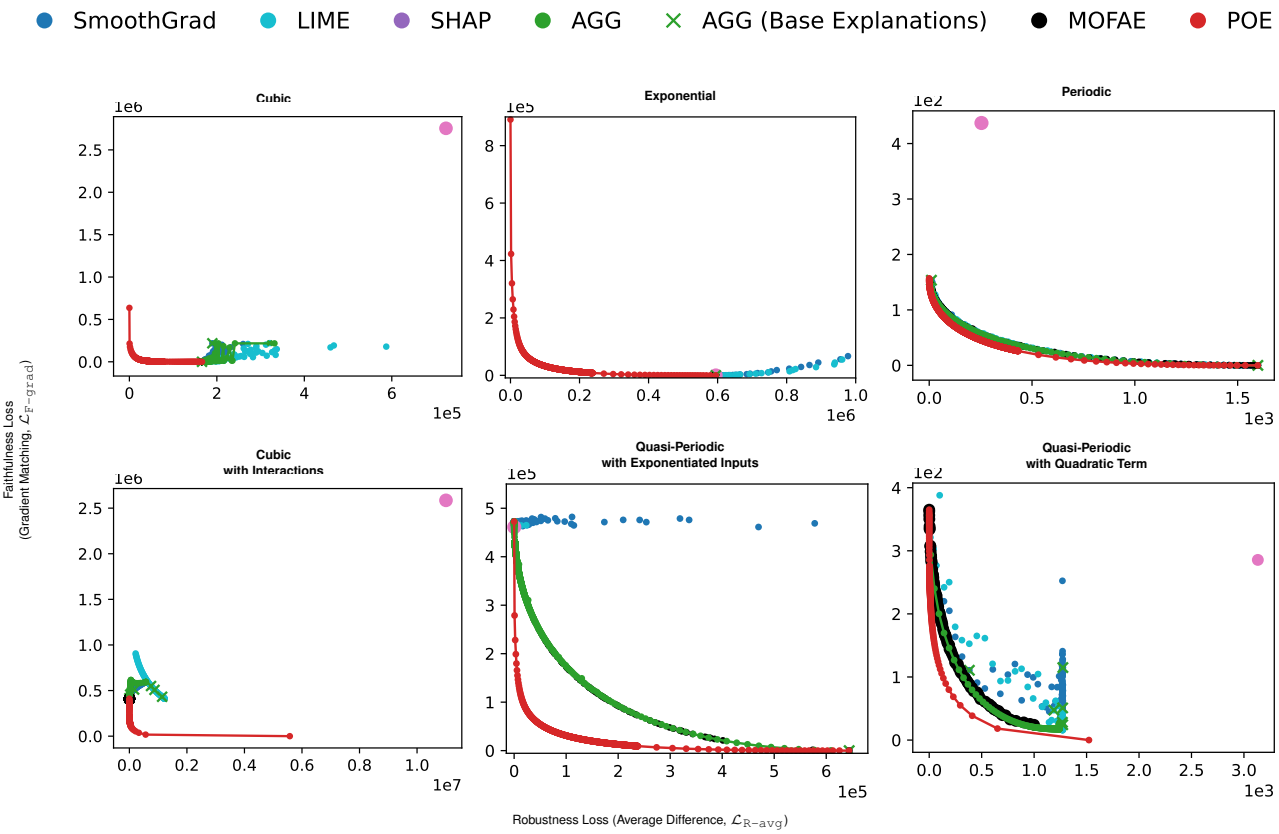
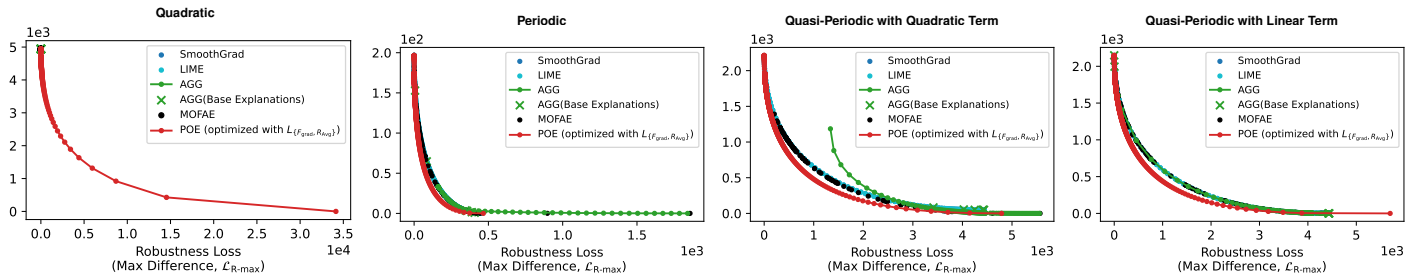


Figure 6: **Faithfulness vs. Robustness.** ss. Comparison of POE and baselines (SmoothGrad, LIME, AGG, and MOFAE) with SHAP. Due to its lack of tunable hyperparameter, SHAP provides only a single explanation, which is not optimal for all the functions above compared to our methods and other baselines.

C.2 HIGHER DIMENSIONAL EXPERIMENTS

C.2.1 POLYNOMIALS AND PERIODIC FUNCTIONS

Below we have additional results for polynomials and periodic functions for $D = 10$. We show that our method consistently provides optimal explanations with an option to control the trade-off between faithfulness and robustness. For $D = 10$ inputs, we randomly sampled 100 points from a uniform distribution $U(-5, 5)$.



(a) (b) (c) (d)

Figure 7: **Faithfulness vs. Robustness.**Trade-off between faithfulness loss and robustness loss for various λ values, the hyperparameter controlling trade-off between faithfulness and robustness in our method and AGG.

C.2.2 Experiment with Images

We compared our method to baselines in an experiment using a pretrained ResNet model He et al. [2016] which contains 25,557,032 parameters on 10 different input images from ImageNet Russakovsky et al. [2015]

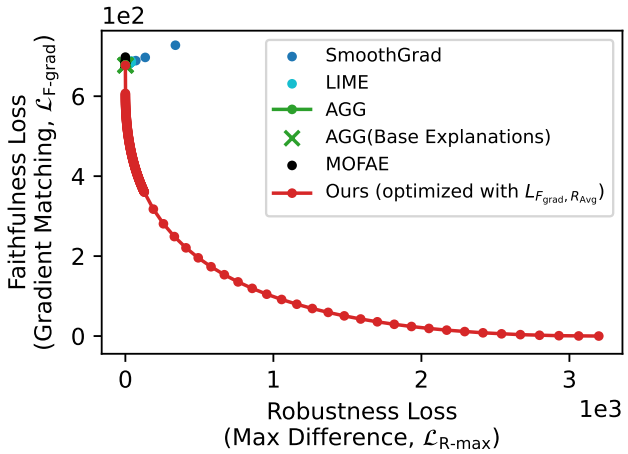
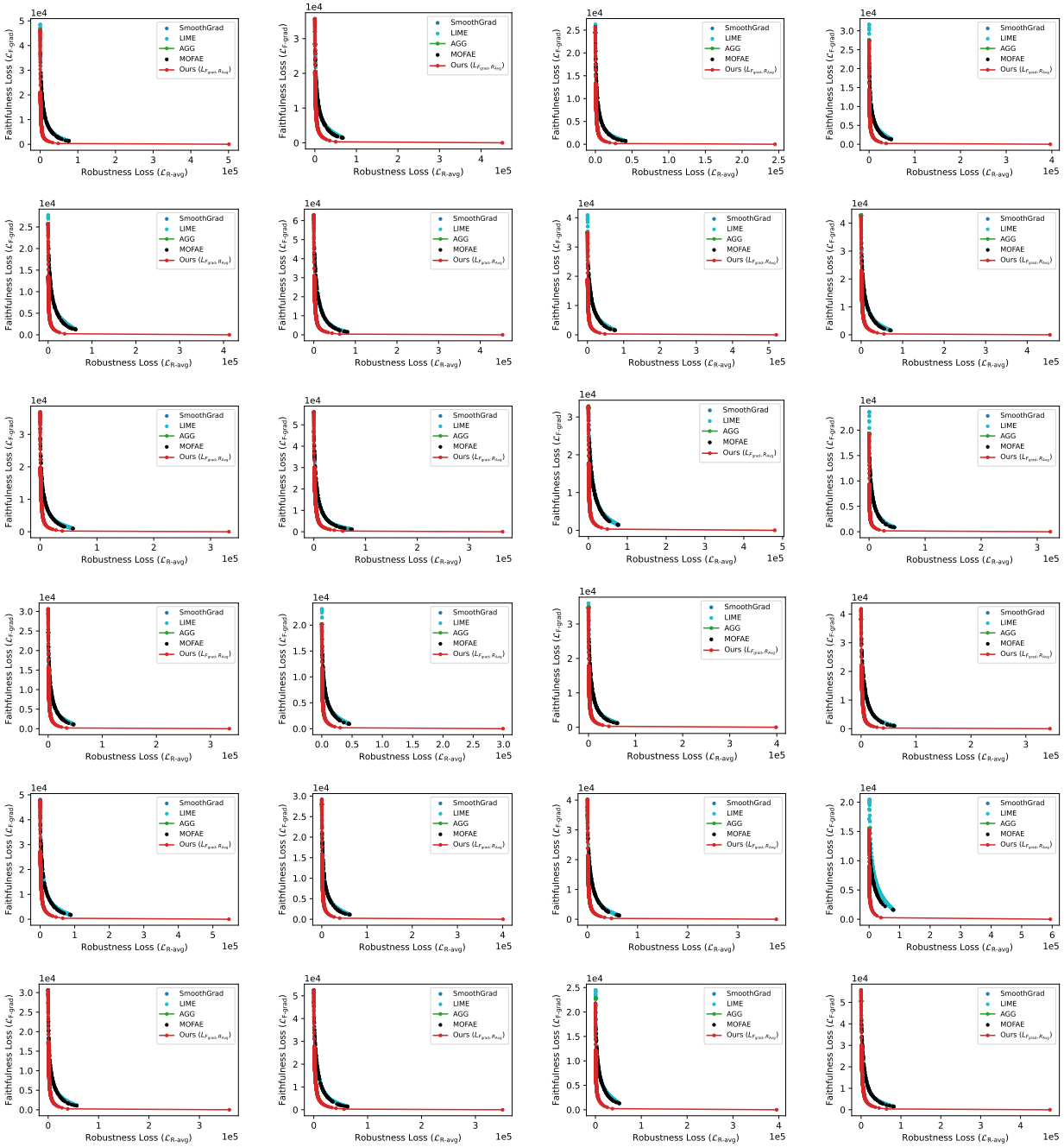
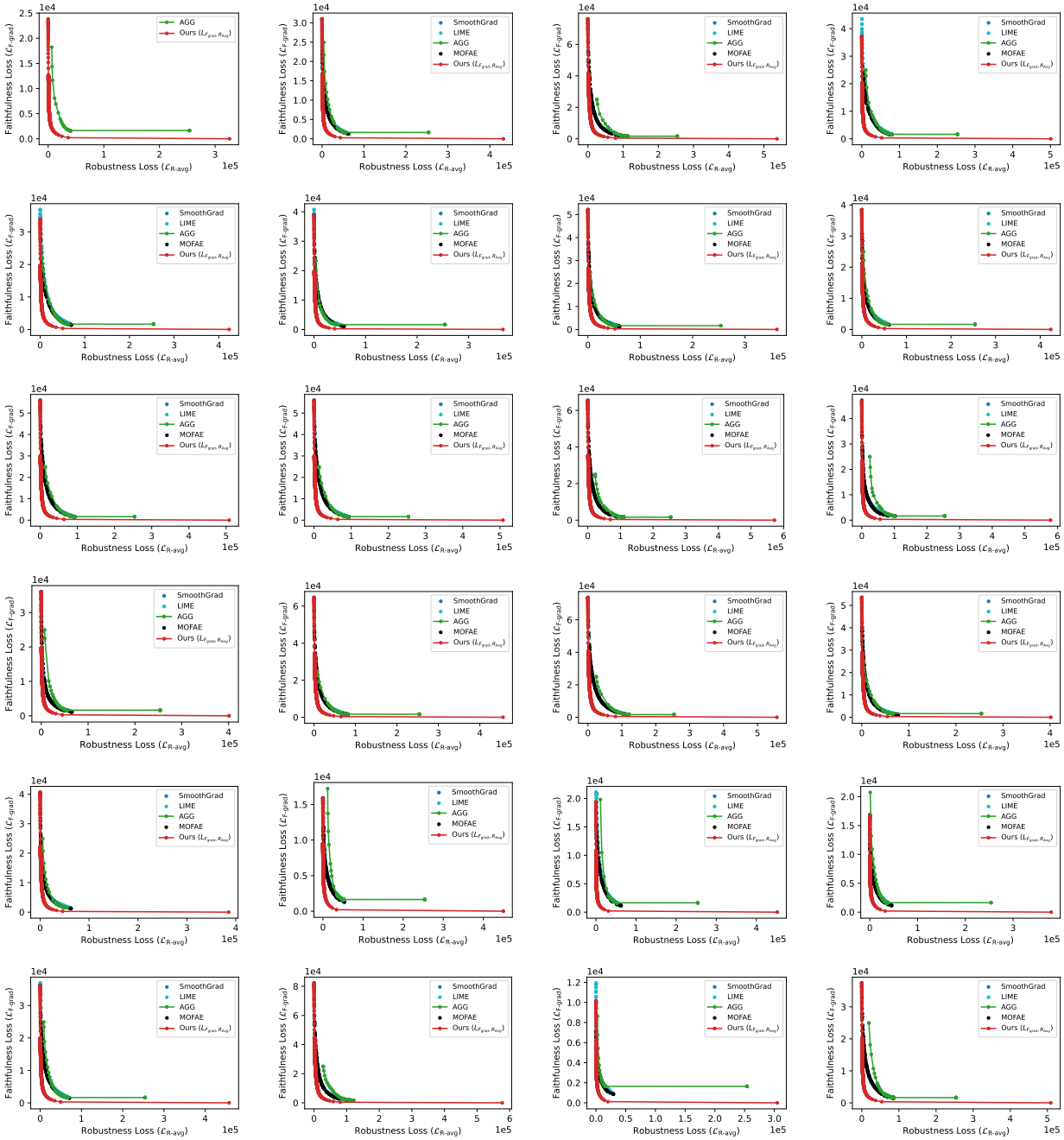


Figure 8: **Faithfulness vs. Robustness.**We observe that our method provides explanations that are Pareto-optimal and capable of managing trade-offs between properties. In contrast, the baselines produce explanations that are not optimal and concentrated in a limited region of the Pareto front(upper left corner).

C.3 NEURAL NETWORK

We trained neural network models with one hidden layer using the sol [1989] dataset. The results show that our method provides more optimal solutions than the baselines. Furthermore, they highlight the limitations of AGG and MOFAE, as these methods are highly dependent on the quality of the base explanations used to generate the explanations.


SmoothGrad
LIME
AGG
MOFAE
Ours
Faithfulness Loss
Robustness Loss
1e4
1e5



D QUALITATIVE COMPARISON

We compared POE with baselines based on agreement on the top-1 most important feature, where a score closer to 1 indicates stronger agreement. Our method demonstrates consistently high agreement across varying trade-off hyperparameter values (λ), whereas the baselines show comparably lower agreement scores regardless of changes to their respective hyperparameters. In addition, we also compared POE with baselines based on cosine similarity and L_2 distance, results are shown in Figure 10 and 11.

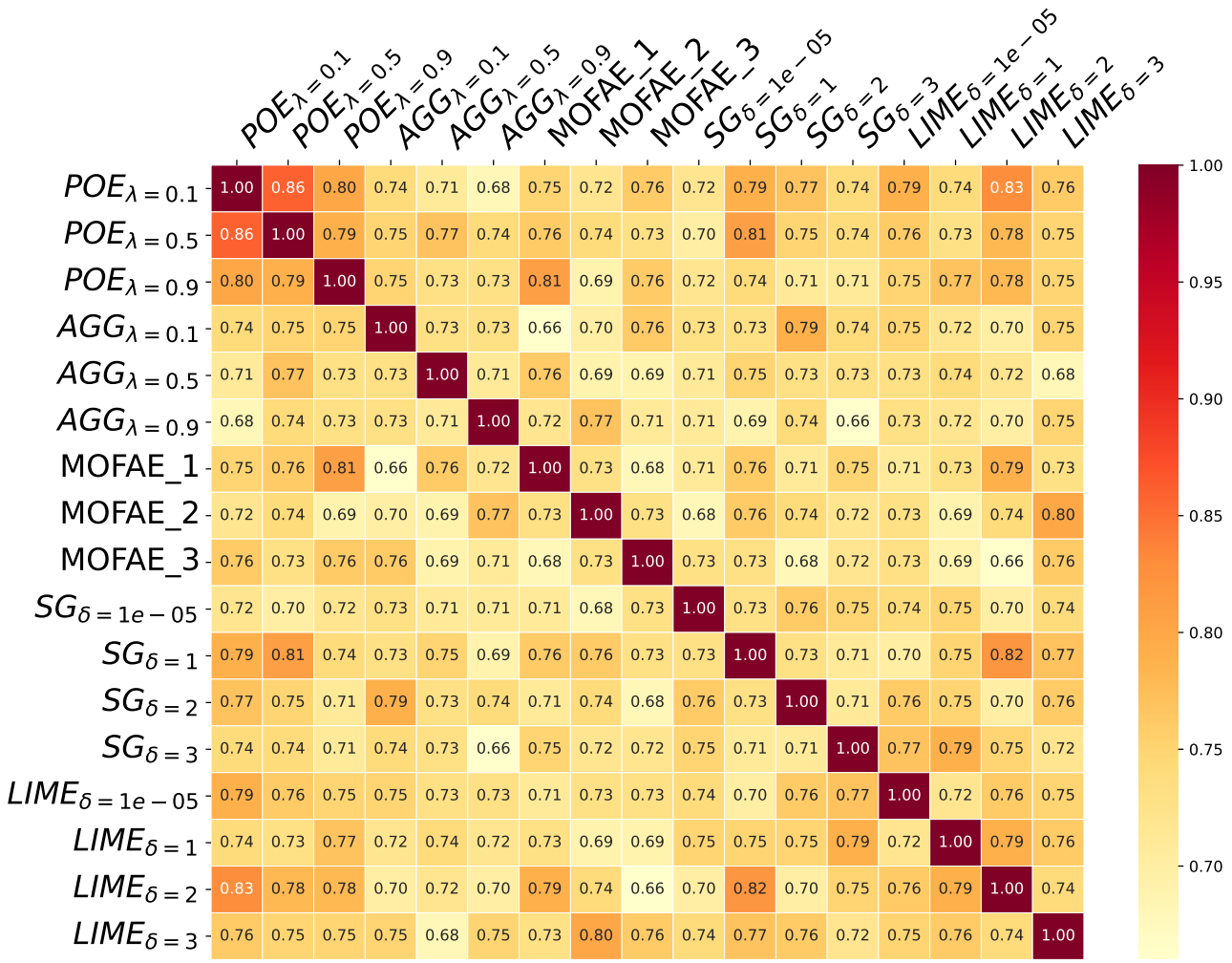


Figure 9: Comparison of POE with baselines using Agreement on the top feature for a cubic function.

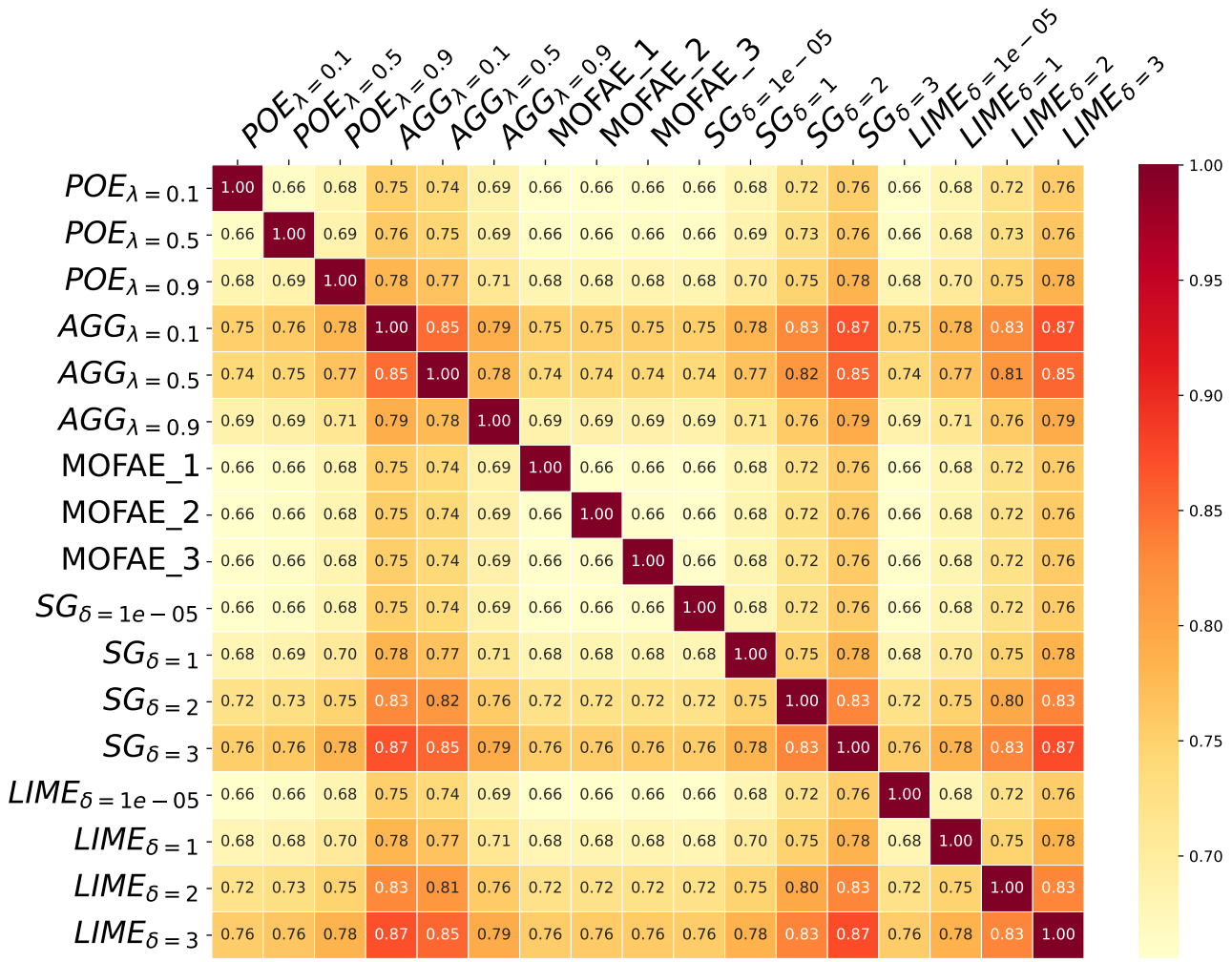


Figure 10: Comparison of POE with baselines with respective hyperparameters using cosine similarity for a cubic function

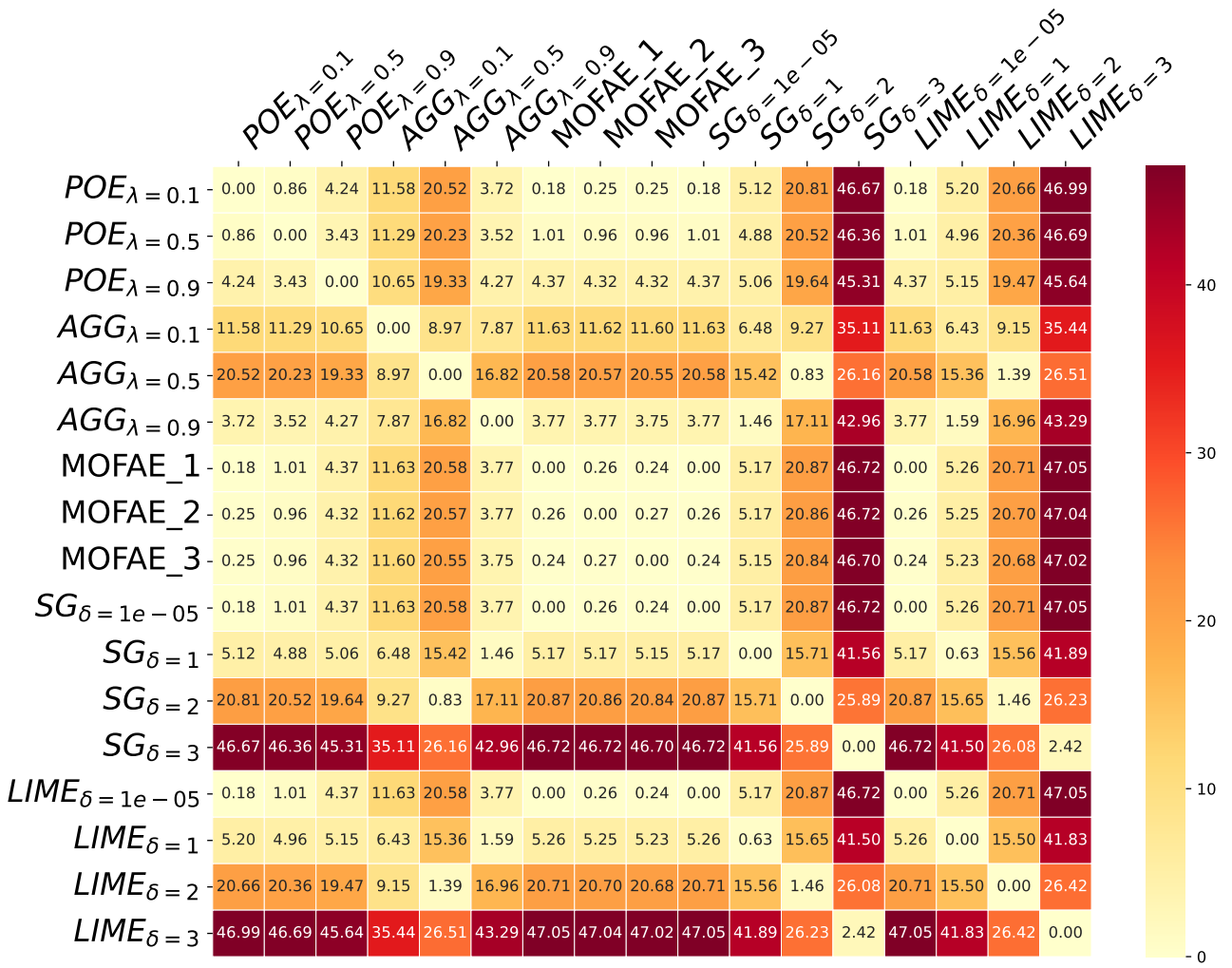


Figure 11: Comparison of POE with baselines with respective hyperparameters using L_2 distance for a cubic function.

E SCALABILITY OF TRANSDUCTIVE SETTING

In the transductive setting, our method is faster, providing an explanation in approximately 0.00 seconds per lambda for lower-dimensional inputs. In comparison, the baselines require substantially more time: SmoothGrad (0.39 seconds), LIME (0.13 seconds), AGG (33.01 seconds), and MOFAE (40.51 seconds) per respective hyperparameter. This difference becomes more pronounced with higher-dimensional inputs, such as images.

The scalability of our method, as well as that of some baseline methods like AGG and MOFAE, remains a limitation in this version of the work. Our primary goal was to demonstrate the generality and usefulness of the idea of treating explanations as quantities. Developing more efficient algorithms for scalability, including amortization schemes, is an important direction for future research.

F FURTHER QUALITATIVE EVALUATION

In Figure 5, we argued that our approach can control the balance between faithfulness, robustness and smoothness while highlighting that SmoothGrad can only manage the trade-off between faithfulness and robustness (missing smoothness). Here, we present a similar figure but for a different initial image (Figure 12) to show that this pattern is not limited to a single instance.

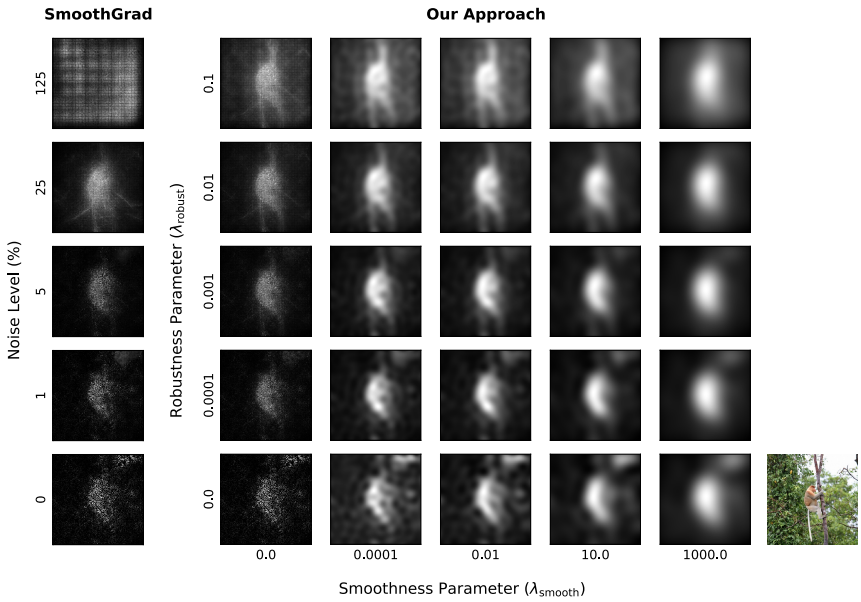


Figure 12: Comparison of our approach vs. SmoothGrad for a different input image when explaining image-based models.

G CHOICE OF INDUCING POINTS

Our inductive approach required specifying a set of inducing points to be used as training data in GP inference. In Figure 4, we investigated how the number of these inducing points affect the results of our method and concluded that solutions approach to their transductive counterparts exponentially fast as the number of points increase. During those experiments, we sampled points uniformly at random from the target function's input domain Ω as our inducing points and used the same set of inducing points for each query point. However, this leaves the question of whether one can be more efficient with their choice of inducing points than uniform sampling. For instance, given a particular query point $\boldsymbol{x}^*$, would it be as effective to sample fewer points around $\boldsymbol{x}^*$ as oppose to covering the entire input domain Ω?

Setup. We consider the same functions[1] as in Figure 4 but with $D = 1$ so that we can compute solutions using a very large density of inducing points over the whole input domain ($N = 1000$, uniformly spread over interval $\Omega = [-10, 10]$). Using these inducing points, we generate explanations for 100 query points, uniformly spread over interval $[-5, 5]$ (a narrower range to avoid any edge artifacts). These explanations optimize for faithfulness and robustness induced by a Gaussian kernel with length scale $\Lambda = 0.25$), where $\lambda_{\text{Faithful}} = 1$ and $\lambda_{\text{Robustness}} = 0.01$. Treating these explanations as ground-truth, we consider three other explanations generated using different inducing points: (i) *Global* uses a smaller number of points, $N = 100$, but still uniformly spread over the whole input domain; (ii) *Local* also uses $N = 100$ points but spread over a narrower interval $\Omega' = [\boldsymbol{x}^* - R, \boldsymbol{x}^* + R]$ around each query point $\boldsymbol{x}^*$; and (iii) *Global+Local* uses the union of both point sets.

Our GP-based approximation to the inductive problem, stated in Proposition 1, requires inducing points $\{\boldsymbol{x}_n\}$ to be distributed uniformly over Ω (as in *Global*). We use importance re-weighting to account for the distribution shift from *Global* to *Local* or *Global+Local*. When inducing points are distributed according to an arbitrary density $p(\boldsymbol{x}_n)$, each sample $\boldsymbol{x}_n$ is assigned the importance weight $(1/|\Omega|)/p(\boldsymbol{x}_n)$. For instance, if inducing points are sampled uniformly over $\Omega' \subset \Omega$ instead of Ω (as in *Local*), then each inducing point is now weighted more heavily with weighting factor $|\Omega'|/|\Omega| > 1$.

Results. In Figure 13, we report the mean square error of *Global*, *Local*, and *Global+Local* as the range of local inducing points R varies relative to the kernel length scale Λ. We see that $R \ll \Lambda$ leads to worse accuracy than *Global*, inducing points fail to cover parts of the input domain that are similar to the query point as dictated by our kernel. As long as $R \gtrapprox \Lambda$, *Local* achieves smaller error than *Global* despite using the same number of inducing points. This is because a majority of the inducing points used by *Global* would have a low similarity to a given query point; *Local* avoids this by using inducing points in the immediate vicinity of each query point. However, this efficiency diminishes as R keeps increasing and the inducing points become more spread out.

While *Local* makes more efficient use of inducing points, the fact that the set of inducing points is different for each query

[1]That is with the exception of *quasi-periodic with exponent* as it oscillates more frequently than the period between inducing points considered in this set of experiments.

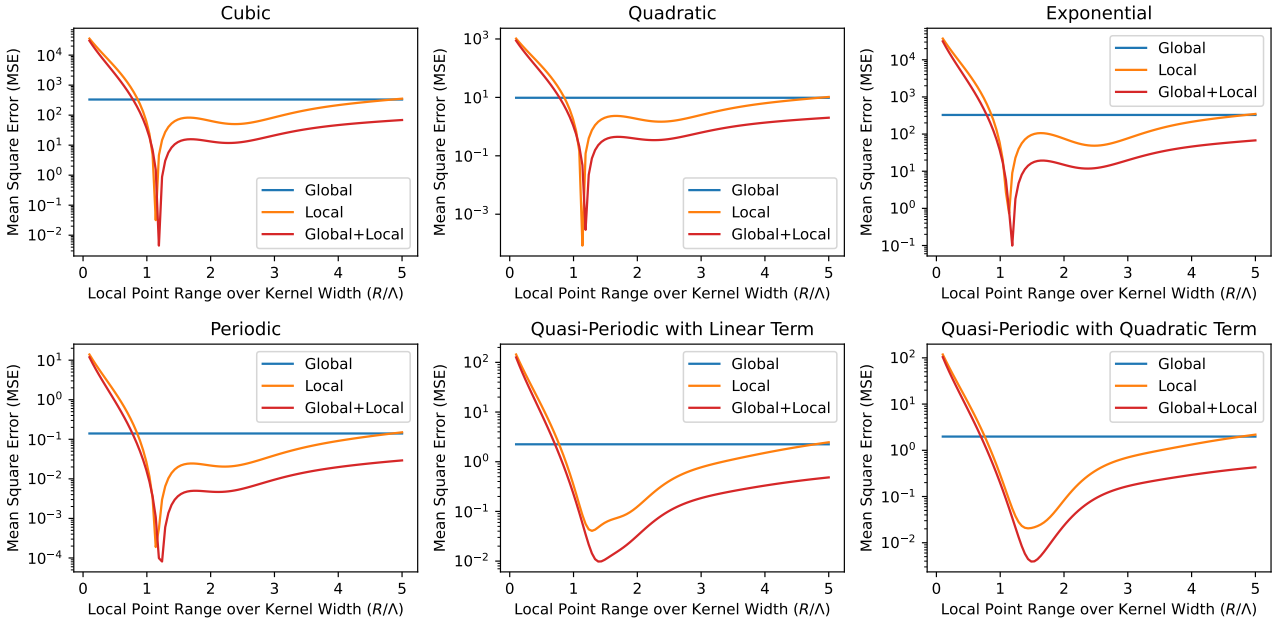


Figure 13: Comparison of *Global*, *Local*, and *Global+Local* inducing points. As long as the range of local points are wider than the kernel width, they achieve lower error than *Global* using the same number of inducing points. This efficiency is reduced as the local point range gets wider and wider.

point requires performing GP inference from scratch for each individual query point (unlike *Global* where the GP posterior can be computed once for the fixed set of inducing points). Therefore, we give the following practical advice: If query points already cover the input domain densely, *Global* is likely to be the computationally more efficient strategy (despite requiring more computation upfront). However, if query points are sparse relative to the size of the input domain (for instance, if the input domain is very high dimensional), than *Local* with a range R on the same magnitude as the kernel length scale Λ is likely to be the more efficient strategy instead.