Models That Are Interpretable But Not Transparent

Chudi Zhong
Duke University
UNC-Chapel Hill

Panyu Chen
Duke University

Cynthia Rudin
Duke University

Abstract

Faithful explanations are essential for machine learning models in high-stakes applications. Inherently interpretable models are well-suited for these applications because they naturally provide faithful explanations by revealing their decision logic. However, model designers often need to keep these models proprietary to maintain their value. This creates a tension: we need models that are *interpretable*—allowing human decision-makers to understand and justify predictions, but not *transparent*, so that the model's decision boundary is not easily replicated by attackers. Shielding the model's decision boundary is particularly challenging alongside the requirement of completely faithful explanations, since such explanations reveal the true logic of the model for an entire subspace around each query point. This work provides an approach, *FaithfulDefense*, that creates model explanations for logical models that are completely faithful, yet reveal as little as possible about the decision boundary. FaithfulDefense is based on a maximum set cover formulation, and we provide multiple formulations for it, taking advantage of submodularity.

1 Introduction

Many organizations rely on machine learning (ML) models and their employees' livelihoods rely on those models being proprietary; the competitive value of these models depends on their secrecy. At the same time, these companies would like to provide explanations for each prediction. Explanations are important for many reasons, including accountability, troubleshooting of the inputs, sanity checking, and allowing the individual subject to the scores to understand the decision and refute them (Rudin, 2019). Explanations are also required by law for high stakes decisions in the European Union (European Parliament and Council of the European Union, 2016; Goodman and Flaxman, 2017) and in the United States, via the Equal Credit Opportunity Act (ECOA). According to Rohit Chopra, the Director of the Consumer Financial Protection Bureau in the United States, "Companies are not absolved of their legal responsibilities when they let a black-box model make lending decisions"..."The law gives every applicant the right to a specific explanation if their application for credit was denied, and that right is not diminished simply because a company uses a complex algorithm that it doesn't understand." (Bureau, 2022). That is, to satisfy legal requirements, explanations must be completely *faithful*, meaning that the explanations must reveal the reasoning process that is actually used in the predictive model to make the prediction. *Inherently interpretable* models provide faithful explanations, but such explanations could release substantial information about the model; lenders would not use a model if it were easily reverse-engineered.

How might an organization balance between the two goals of keeping models secret and revealing information within explanations? This tension between keeping models interpretable and proprietary is embodied by the *model extraction attack* (Tramèr et al., 2016). The attacker queries an ML model to obtain predictions. The attacker may not know the exact model type or the true data distribution used to train the model. After enough queries, the attacker can use the collected query-label pairs to train a surrogate model that achieves accuracy very close to that of the confidential model. It is hard enough to prevent information leakage from these attacks by revealing only predictions, but the need to reveal explanations could make these attacks much more powerful.

In this paper, we provide a model form that addresses this tension. We propose a novel explanation gen-

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

eration method for logical machine learning models, *FaithfulDefense*, that ensures the released explanations are always faithful and yet does not disclose the true decision boundary used by the underlying model. Our approaches also allow organizations to customize the level of detail revealed in explanations. FaithfulDefense is derived as a maximum set cover formulation. If its solution does not use the full "length" budget allocated to it by the user, extra terms can be appended to preserve the disclosure of information about the model while still maintaining faithful explanations. Since maximum set cover is submodular, we also produce a greedy solution that also works well in practice. An empirical evaluation with 3 different attacker strategies and 6 different explanation methods demonstrates that our proposed method is effective in safeguarding the underlying model. An implementation of the algorithm is available at: `https://github.com/chudizhong/FaithfulDefense`.

2 Related Work

Interpretability: Understanding how predictions are made allows humans to identify and rectify potentially serious problems (Ashoori and Weisz, 2019; Brundage et al., 2020; Lo Piano, 2020; Rudin and Wagstaff, 2014; Thiebes et al., 2020). While there is an abundance of work in explainable artificial intelligence (XAI), through providing simpler approximations to black-box models (Bastani et al., 2017; Lakkaraju et al., 2019) or local approximations to black boxes (Lundberg et al., 2020; Lundberg and Lee, 2017; Ribeiro et al., 2016; Simonyan et al., 2014; Sundararajan et al., 2017), such techniques are unsuitable for high-stakes decisions because their explanations are often unfaithful, incomplete or misleading (Lakkaraju and Bastani, 2020; Laugel et al., 2019; Rudin, 2019; Adebayo et al., 2018). Such unfaithful explanations can exacerbate problems with lack of trust. Interpretable machine learning, on the other hand, focuses on developing inherently interpretable models. These models reflect exactly how they make decisions, and the reasoning is always faithful. Logical models such as decision sets and decision trees are important types of interpretable models that have existed since the beginning of artificial intelligence. Numerous algorithms have been developed for optimizing them (Wang et al., 2017; Aglin et al., 2020; Angelino et al., 2018; Dash et al., 2018; Demirović et al., 2022; Lin et al., 2020; Rudin and Shaposhnik, 2023), and modern versions of these algorithms can find sparse models with accuracy comparable to that of black box counterparts. These types of models have a long precedent in high-stakes decisions because they provide logical rules that faithfully describe, for instance, why someone's loan was denied.

Model extraction: Model extraction means acquiring information from an unknown target model that goes beyond simple outputs to a set of input queries. An *attacker* who extracts information from the model might aim to train its own surrogate model, having performance no worse than that of the target model (Tramèr et al., 2016; Orekondy et al., 2019; Shi et al., 2017; Correia-Silva et al., 2018; Chandrasekaran et al., 2020; Shi et al., 2018a; Teitelman et al., 2020).

The notion of providing an explanation clashes with the model extraction paradigm. If an explanation is required with each prediction, the attackers job is potentially much easier since the explanation could reveal the predictions of an entire portion of the input space. Recent studies delve into the role of explanations in model extraction attacks (Yan et al., 2022, 2023; Wang et al., 2022; Ezzeddine et al., 2024; Miura et al., 2024; Oksuz et al., 2023; Nguyen et al., 2023). Most of these works consider explanations derived from XAI algorithms like LIME (Ribeiro et al., 2016) and SHAP (Lundberg et al., 2020) or counterfactual explanations. As discussed, posthoc explanations are not faithful. They are also generally incomplete, in that they reveal possibly a few variables that might be important to the prediction, but do not reveal how these variables are used to form the prediction. Explanations from inherently interpretable models would be far more valuable to an attacker because they explain the full reasoning process that led to a decision. Given that high-stakes domains like finance generally require inherently interpretable models, and should require complete and faithful explanations, we should be far more concerned by the prospect of attackers in the setting of inherently interpretable models.

The literature, however, focuses on the protection of individuals whose scores are computed by the model, while overlooking the need to protect the companies that develop and use these models. This gap in protection can have serious consequences: companies may avoid using inherently interpretable models out of concern for the increased risk of model extraction attacks. If companies cannot secure their proprietary interpretable models, they may continue to rely on black-box models with unfaithful explanations.

Hence, we want models that are inherently *interpretable* – with completely faithful explanations – but not *transparent* – we do not want attackers to approximate or see the full model with only a few queries.

In order to gather the most valuable data to build a surrogate model, attackers can use generative models for generating artificial data for querying the target model (Mosafi et al., 2019; Kariyappa et al., 2021; Shi et al., 2018b; Yuan et al., 2022; Truong et al., 2021;

Sanyal et al., 2022), or active learning (Tramèr et al., 2016; Chandrasekaran et al., 2020; Pal et al., 2019, 2020; Pengcheng et al., 2018; Shi et al., 2018a; Reith et al., 2019; Wang and Lin, 2022; Xie et al., 2022). The attacker will eventually gather enough information to build an accurate surrogate model, and our goal is to slow down this process so the model's decision boundary remains protected for as long as possible. Ideally, we want the explanations to provide little more information than if they were absent.

3 Problem Setting

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be the dataset, where $\mathbf{x}_i \in \mathbb{R}^p$ are features and $y_i \in \{0, 1\}$ denotes binary labels. This reflects high-stakes decision-making in, for instance, loan and insurance approvals. A model f can be trained to learn relations between features $\{\mathbf{x}_i\}$ and outcomes $\{y_i\}$. We consider logical models that use "if-else" conditions, specifically decision sets. Decision sets encompass other logical models, including decision trees and decision lists (Rudin et al., 2022). Sparse decision sets are inherently interpretable and are particularly easy to understand and troubleshoot, but are also particularly difficult to protect from attackers when the true model logic is employed in explanations. Sparse generalized additive models with piecewise constant shape functions – which are currently the most common models for financial risk scoring – can be converted efficiently into decision sets, as described in Appendix D.

Algorithm 1 ModelExtraction(max_q)

Require: a maximum number of queries max_q
1: $Q \leftarrow \{\}, label \leftarrow \{\}, E \leftarrow \{\}$
2: **for** $t \in \{1, ..., max\_q\}$ **do**
3: $q \leftarrow$ AttackerGenerateQuery$(Q, label, E)$
4: $qhat, e \leftarrow$ DefenderStrategy(q)
5: Q.add(q), $label$.add$(qhat)$, E.add(e)
6: Train a surrogate model f' using Q, $label$ and E.

An ML model extraction attack occurs when the attacker has query access to the target model f and attempts to learn a surrogate model f' that closely approximates or matches the performance of f by sequentially asking queries. Conversely, the defender must provide the label of the query and give an explanation if the query has a positive label (e.g., loan denied, insurance claim denied, job application denied). The process is shown in Algorithm 1. In this work, we stand on the position of the defender to address the challenge of how to provide meaningful and faithful explanations while preventing attackers from training an accurate surrogate model with a small number of queries.

4 Methodology

In this section, we first introduce our defense method, detailing how to generate an explanation for a query predicted to be positive by the model f. We then illustrate how attackers leverage these explanations to launch more potent attacks.

4.1 How a Defender Can Generate Explanations

We assume f is trained on proprietary dataset $\mathcal{D}$ and belongs to an inherently interpretable model class, such as decision sets. A decision set, also known as a "disjunction of conjunctions," "disjunctive normal form" (DNF), or an "OR of ANDs" is comprised of an unordered collection of rules, where each rule is a conjunction of conditions. A positive prediction is made if at least one of the rules is satisfied. Decision sets encompass decision trees and decision lists: any decision tree or decision list can be written as a decision set. A decision set is inherently interpretable and the satisfied rule can be directly used as a faithful explanation. However, generating explanations in this way discloses too much information. We now introduce a more frugal method to provide faithful explanations.

Given a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$, we can turn it into a binary dataset by considering all categories for categorical features and binning continuous features using thresholds. Let $C = \{c_1, c_2, ..., c_m\}$ denote the full set of possible binary features. It can also be viewed as a collection of conditions. For example, a c_j could be "income <5K", "age $\leq$ 20", "saving < 5K", etc. The binarized dataset is denoted as $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_i\}_{i=1}^n$ where $\tilde{\mathbf{x}}_i \in \{0, 1\}^m$ is a binary vector of length m. Each rule in f is a subset of C. Let β be a set of conditions. We define $\text{cap}(\beta, \tilde{\mathbf{x}}_i) = 1$ if $\tilde{\mathbf{x}}_i$ satisfies all conditions in β, and 0 otherwise. The collection of samples from the dataset captured by β can be denoted as $S_\beta(\tilde{\mathbf{X}}) = \{\tilde{\mathbf{x}}_i : \text{cap}(\beta, \tilde{\mathbf{x}}_i) = 1\}$.

Let us define a faithful explanation for query q.

Definition 1. *(Faithful explanation e for query q): Let q be a query such that its predicted label $f(q)$ is 1, where an explanation is required. Denote a rule in f that q satisfies as f_r, where $f_r \subseteq C$. Then, $e \subseteq C$ is a faithful explanation for q if $f_r \subseteq e$.*

Intuitively, an explanation is faithful if it reveals part of the model. The explanation does not need to reveal *all* rules in the model, or even an entire rule, to be faithful. For instance, if a rule in the model is "income <5K," then an explanation that "income <5K and age $\leq$ 20" is faithful, and yet does not reveal that the entire "income <5K" subgroup has label 1. Providing an explanation that reveals only part of the model's origi-

nal rule, rather than the full rule ("income <5K and age $\leq$ 20" instead of "income <5K") is identical to the process of (1) creating an equivalent underlying model with the rule split in two (i.e., replacing "income <5K" in the model with the pair of rules "income <5K and age $\leq$ 20," "income <5K and age > 20"), and then (2) providing a rule in this new model as the explanation ("income <5K and age $\leq$ 20"). Thus, our definition of a faithful explanation takes into account this option for the modeler of creating an equivalent decision set before providing the explanation. Importantly, a faithful explanation does not need to be "complete," meaning it need only reveal one reason for the decision, not all possible reasons. E.g., both "income <5K" and "employed = no" are true, but we need only give an explanation involving one of them. (The reason would come with a notification that there may be other reasons why the loan was denied, to prevent a misinterpretation that the explanation is complete.)

We define $|e|$ as the size of the explanation, i.e., the number of conditions in the explanation. Let $\text{supp}(e, \tilde{\mathbf{X}})$ denote the number of samples captured by e, i.e.,

$$\text{supp}(e, \tilde{\mathbf{X}}) = \sum_{i=1}^{n} \text{cap}(e, \tilde{\mathbf{x}}_i) = |S_e(\tilde{\mathbf{X}})|.$$

$\text{supp}(e, \tilde{\mathbf{X}})$ can be used to estimate the released information. There is a tradeoff between $\text{supp}(e, \tilde{\mathbf{X}})$ and $|e|$. Typically, $\text{supp}(e, \tilde{\mathbf{X}})$ monotonically decreases as $|e|$ increases, e.g., the size 2 rule "income <5K and age$\leq$ 20" has less support than the size 1 rule "income <5K."

Our goal is to find a simple explanation e, meaning the explanation can be described succinctly, i.e., $|e|$ is no larger than a max length l, and yet captures few positively predicted samples in the training set. Therefore, the optimization problem can be written in the following form:

$$\min_{\text{faithful explanation } e} \text{supp}(e, \tilde{\mathbf{X}}) \text{ s.t. } |e| \leq l. \quad (1)$$

Since f is a decision set, we can slightly modify Problem (1). Let e consist of two parts, i.e., $e = \{e^{\text{base}}, e^{\text{add}}\}$, where $e^{\text{base}} = f_r$ is a set of conditions used by one of the rules in f that q satisfies, and e^{add} is the set of additional conditions satisfied by q other than the conditions used by the satisfied rule, denoted as $C_q \subseteq C \backslash e^{\text{base}}$. E.g., if q obeys "income $< 10k$," and income is not one of the conditions in e^{base}, we may choose to add it to e^{add} to narrow the explanation. Since e^{base} is fixed, we only need to optimize e^{add}. Then, the modified optimization problem is

$$\min_{e^{\text{add}} \subseteq C_q} \quad \text{supp}(\{e^{\text{base}}, e^{\text{add}}\}, \tilde{\mathbf{X}}) \quad \text{s.t. } |e^{\text{add}}| \leq l. \quad (2)$$

Theorem 1. *Let q be the query with $f(q) = 1$, e^{base} be the set of conditions used by the rule in f that q satisfies, and $C_q \subseteq C \backslash e^{base}$ be the additional conditions satisfied by q. Problem 2 of selecting a subset $e^{add} \subseteq C_q$ such that the intersection of the selected set of samples covers the minimum number of elements in $\tilde{\mathbf{X}}$ is equivalent to the maximum coverage problem: selecting a subset $e^{add} \subseteq C_q$ such that the union of selected samples cover the maximum of the complement set $\tilde{\mathbf{X}}^c$.*

This theorem indicates that Problem 2 is the same as the maximum coverage problem, and we need only optimize e^{add}, the set of additional conditions beyond the base model to include within the explanation, though this is also intuitive.

The maximum coverage problem is NP-hard (Karp, 2010) but a submodular problem (Nemhauser et al., 1978), and we use three methods to find e for each query q.

Method I: FaithfulDefense Greedy. We choose the condition c_j in each step such that $S_{\neg c_j}(\tilde{\mathbf{X}}_{\text{base}})$ is maximized. The method approximates the optimal solution in a factor of $1 - \frac{1}{e}$ (Nemhauser et al., 1978). If the greedy solution does not use all the length budget, i.e., $|e^{\text{add}}| < l$, we randomly append extra conditions to use the full budget.

Method II: FaithfulDefense IP. We can find an optimal solution for Problem (2) by solving an integer programming (IP) problem. The IP formulation is shown below. Let $\mathbf{u} \in \{0,1\}^n, \mathbf{v} \in \{0,1\}^m$ be the binary vectors. Despite IP being NP-hard, the problem is usually solved in seconds by a solver.

$$\max_{\mathbf{u},\mathbf{v}} \quad \sum_{\{i:\tilde{\mathbf{x}}_i \text{not covered}\}}^{n} u_i \quad (3)$$

$$s.t. \quad \sum_{j=1}^{m} v_j \leq l, \quad \sum_{j:x_{ij}=1} v_j \geq u_i \quad (4)$$

$$u_i \in \{0,1\} \ \forall i \in \{1, ..., n\} \quad (5)$$

$$v_j \in \{0,1\} \ \forall j \in \{1, ..., m\} \quad (6)$$

Equation 3 aims to maximize the sum of the covered samples. Equation 5-6 indicate that if $u_i = 1$, $\tilde{\mathbf{x}}_i$ is covered by at least one of the $\neg c_j$'s and if $v_j = 1$, $\neg c_j$ is selected), respectively.

Method III: FaithfulDefense IP-RA. If the IP solution does not use the full length budget, we randomly append extra conditions to use the entire budget.

Algorithm 2 shows how the defender safeguards the model while providing a faithful explanation.

4.2 How an Attacker Can Use Explanations

Given explanations in addition to query labels, attackers will use these explanations to generate more

Algorithm 2 FaithfulDefense(q)

Require: Base model f, dataset $\tilde{\mathbf{X}}$, conditions C, query q, history of explanations E

if $f(q) = 0$ **then**
  return $f(q), \emptyset$ ▷ No need for explanation.
else
  if q can be explained by an $e \in E$ **then**
    $e \leftarrow$ FindFromHistory(E, q) ▷ Return only the explanation shown previously.
  else
    $e \leftarrow$ GenerateExplanation$(q, f, \tilde{\mathbf{X}}, C)$. ▷ Solve Problem 2 by IP, IP-RA or greedy method.
  return $f(q), e$ ▷ Returns low support and small size explanations

informative queries. A straightforward attacker strategy is generating an unasked query that lies beyond the boundaries marked by past explanations. This strategy does not fully take advantage of these explanations. An alternative strategy is to explore a high-density area close to the decision boundary. Let us explain it in detail.

Given a query q and its explanation e, the attacker can generate candidate queries by adjusting one of the feature values close to and outside the boundary marked by e. For example, if e involves k features, the attacker can generate at least k candidate queries. However, this could lead to too many candidate queries if $|e|$ is large. Given the limited number of queries the attacker can ask, it would be reasonable to only consider candidate queries that explore regions determined by more important features. So the next question is how to determine which features are more important. The attacker can track the number of times each feature appears in past explanations E. Features with higher counts can be viewed as more important than others. Then the attacker can select the top k features in e and generate new queries that differ from q by adjusting the value of one of the selected features. If a feature j used in e appears most often in past explanations, the new value assigned to feature j is positioned close to, yet beyond, the boundary provided by e. For categorical features, q_j^{new} is set to $q_j \pm 1$ to transition to another category. For continuous features, q_j^{new} is adjusted by boundary$(e, j) \pm \delta$. For instance, if the explanation for feature j is "$j \leq \$5000$," then the new value for j could be set to 5001. In cases where feature j is bounded by e on both sides, the attacker can generate two candidate queries. All other features retain the same values as q. Thus, a new candidate query is formulated as $q^{new} = [q_1, ..., q_{j-1}, q_j^{new}, q_{j+1}, ...q_p]$. The attacker then adds all new candidate queries into a query pool and rearranges them based on the importance of the perturbed feature.

This perturbation-based query generation method is inspired by (Oksuz et al., 2023), which adopts the perspective of attackers upon receiving LIME explanations (Ribeiro et al., 2016). In their setup, a LIME explanation is provided for each query, reporting weights for each feature. They then slightly adjust the value of one of the features with a high weight as a candidate query. However, our explanations differ from LIME in three ways: (1) our explanations are always faithful, following the rules used to make predictions, while LIME explanations are local approximations of the underlying model; (2) our explanations do not assign weights to features since features are not weighted in the ground truth model, so the explanation itself cannot be used to rank features; and (3) our explanations are provided only for queries predicted to be positive, while in (Oksuz et al., 2023), a LIME explanation is provided for each query.

Meanwhile, since our explanations are faithful, they can also be used as part of the attacker's surrogate model for prediction. For instance, upon the arrival of a new sample, the attacker initially verifies whether the sample aligns with any of the explanations. If such a match is found, a positive prediction is made directly. Otherwise, a prediction is generated based on the surrogate model f', i.e.,

$$\hat{y}_i = \text{cap}(E, \mathbf{x}_i) \vee f'(\mathbf{x}_i). \tag{7}$$

5 Experiments

In this section, we conduct experiments to show that (1) FaithfulDefense is efficient in that it reveals less information about the dataset than baselines (Section 5.1), (2) FaithfulDefense can provide explanations within seconds (Section 5.2), (3) FaithfulDefense requires more queries by the attacker to achieve performance similar to f on the test set (Section 5.3), and (4) FaithfulDefense always provides completely faithful explanations (Section 5.4).

Since loan decisions are a key situation where faithful explanations are required and where model owners keep models secret, we use three credit datasets, the Fair Isaac (FICO) credit risk dataset (FICO et al., 2018) used for the Explainable ML Challenge and German credit dataset from UCI (Dua and Graff, 2017). We use the merged training/testing sets of a loan approval prediction problem dataset from Kaggle (Chatterjee, 2018). Each dataset is divided into train and test sets with an 80:20 split. Details are in Appendix B. We use fast sparse rule sets (FastSRS) (anonymized for review, 2024) on the training set to train the underlying interpretable model that we aim to protect.

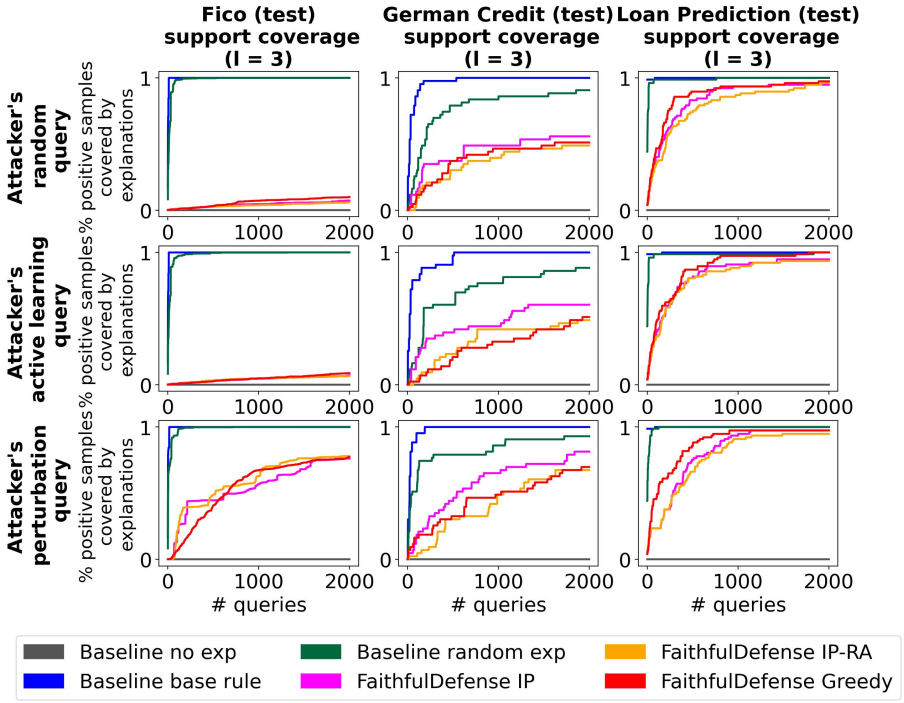

Figure 1: Number of queries vs. the proportion of positive samples covered by explanations. Lower curves are better. FaithfulDefense (red, orange, pink curves) captures fewer positive samples in test sets for all three datasets using three different querying strategies. LIME was only used for perturbation queries since attackers perturb the LIME explanations to generate the next query; LIME explanations are only applicable to the attacker's perturbation-based querying strategy. LIME explanations are incompatible with other attacker querying strategies. (max length $l = 3$)

Since, as discussed in the related work section, we know of no previous method that provides fully faithful explanations that is resilient to model extraction attacks, we compare FaithfulDefense with three natural baselines: (1) returning the rule matched by the query in model f (2) randomly appending l conditions, (3) providing no explanation. We also use LIME explanations (Ribeiro et al., 2016) as another baseline. We were able to coerce LIME into comparison because its explanations share a format (e.g., if x1>5 and x2<3, predict yes) similar to ours. Nevertheless, it's important to notice that LIME does not faithfully provide the reasoning process of the underlying model.

We cannot include Shapley value-based explanation methods as a baseline in our comparison because they return only feature importances: Shapley values provide *no* explanation of the underlying model's calculations, only what variables are estimated to be important in those calculations; they provide *no* explanations about any observation other than the one being considered (e.g., "the loan was denied because some of your features are more important than others"). To be faithful, the user would need to know *how* the variables are combined, not just how important they might be.

We assume the attacker knows the marginal distribution of each feature but not the joint distribution. There are three query generation strategies that are reasonable for the attacker: (1) randomly generate an unasked query outside the boundaries marked by past explanations, (2) use an agnostic active learning algorithm called importance weighted active learning (iwal) (Beygelzimer et al., 2010; Chandrasekaran et al., 2020) to find an informative query that is outside the explanation boundaries, and (3) generate queries by adjusting the value of important features as described in Section 4.2. For each dataset, we allow the attacker to ask 2000 queries. Note that only perturbation-based queries are generated if LIME is used to provide explanations. We use the absolute value of coefficients in LIME explanations as the measure of variable importance.

5.1 How much information do the explanations leak?

Figure 1 demonstrates that *FaithfulDefense* (red, orange, and pink curves) **reveals less information about the test set than the baseline defense strategies** (i.e., captures fewer positive samples) for

all three datasets using three different querying strategies, as the three curves consistently fall below the green and blue curves. This figure also indicates that the perturbation-based querying method is usually more aggressive than the other two attacker methods. Training results are in Appendix C.

5.2 How much time does each method take to generate explanations?

Figure 2 shows the difference in time consumption between our FaithfulDefense methods and LIME for generating explanations. **The time taken by our FaithfulDefense Greedy is generally very fast, usually less than 0.1 second.** In most cases, FaithfulDefense IP and FaithfulDefense IP-RA take longer than the greedy method but are faster than LIME. Sometimes they can take longer because a solver is used to find the optimal solution. In practice, if the IP method cannot easily find the optimal solution, it would be better to use FaithfulDefense Greedy instead. Returning base rules as explanations and using random explanations are instantaneous. More time consumption results are in Appendix C.

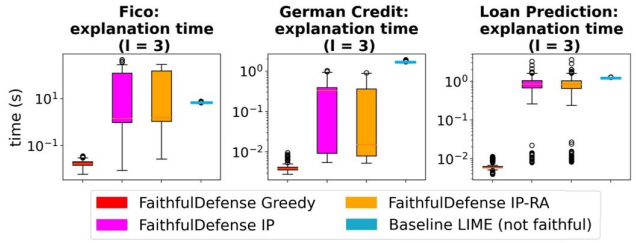

Figure 2: Time consumption of generating explanations when the attacker uses the perturbation strategy.(max length $l = 3$).

5.3 How well does the attacker's surrogate model perform?

The attacker can train a surrogate model using the collected query-label pairs. In real situations, the attacker may not know the exact model class and architecture of the underlying model. However, they may infer that the underlying model is a logical model based on the explanations provided. Therefore, we consider three different model classes (CART, Breiman et al. 1984, Random Forest, Breiman 2001, and Gradient Boosted Tree, Friedman 2001) that the attacker might use to train a surrogate model f'. Since all explanations except LIME are faithful, they can also be used for prediction in addition to generating more informative queries. Given a test set, Eq (7) is used to make predictions. We compare the performance based on the similarity between predictions made by the surrogate model and the underlying model on the test set, i.e., $\frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} 1[f(\mathbf{x}_i^{\text{test}}) == \text{cap}(E, \mathbf{x}_i^{\text{test}}) \vee f'(\mathbf{x}_i^{\text{test}})]$. A new surrogate model is trained on the cumulative set once 50 more query-label pairs are obtained. This allows us to track how many queries are needed to achieve a similar performance as the underlying model.

Figure 3 compares the test performance among different explanation methods. CART with a depth of 5 is used to train the surrogate model. In this figure, a higher attacker's error indicates better defense. In other words, more queries are needed to achieve similar performance to the underlying model f. As observed, our method (red, orange, and pink curves) is usually above the green and blue curves on the FICO and German credit datasets even when 2000 query-label pairs are used, indicating that **our explanations are more effective in protecting the underlying model**. The phenomenon is less obvious in the loan prediction dataset. The results imply that our defense methods outperform the baselines more significantly in larger datasets. Sometimes, we even observe overlap between the red, orange, and pink curves and the dark gray curve, suggesting that **providing explanations using FaithfulDefense is nearly equivalent to providing no explanations**.

The bottom row in Figure 3 shows that sometimes when the attacker uses the perturbation-based querying strategy, FaithfulDefense is better than or similar to LIME in the 1000 queries and then becomes slightly worse. This is because explanations given by our FaithfulDefense are always faithful (see Table 1) and can be used as part of the surrogate model for prediction but LIME explanations are permitted to be unfaithful. More results are in Appendix C.

5.4 How faithful the explanations are?

We use the false positive rate (FPR) as a measure of the faithfulness of our explanations because an explanation is considered faithful when it aligns with the (partial) true model. In our framework, explanations are provided specifically when the true model predicts a positive outcome. Consequently, for samples that match the explanation, we expect them to consistently receive a positive prediction. A non-zero FPR would indicate a misalignment between the explanation and the true model. Table 1 shows the false positive rate on the test set when explanations themselves are used to make predictions. **All explanation methods except LIME have completely faithful explanations** as their FPR are always 0, while LIME explanations are not faithful given FPR greater than 0.

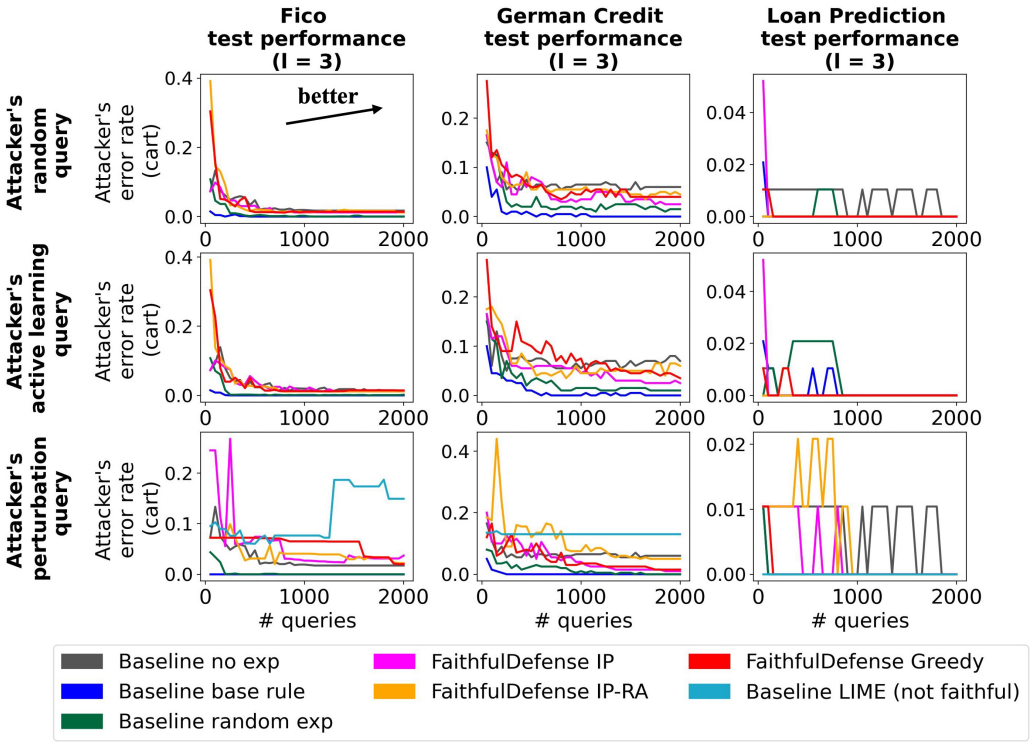

Figure 3: Comparison of test performance between base and surrogate models. CART is used by the attacker to train the surrogate model. (max length $l = 3$).

Table 1: False positive rate (FPR) on the test set when only explanations are used to make predictions. All explanation methods except LIME have completely faithful explanations, so their FPRs are zero.

FPR	FICO	German Credit	Loan
Base rule	0	0	0
Random exp	0	0	0
FaithfulDefense IP	0	0	0
FaithfulDefense IP_RA	0	0	0
FaithfulDefense Greedy	0	0	0
LIME	19.78%	15.71%	3.33%

6 Discussion

Our goal is to provide models that are *interpretable*, meaning that they have completely faithful explanations and could be used in high-stakes situations, but not *transparent*, meaning that they provide some level of protection for the model's formula, which enables organizations to protect their efforts and intellectual property. There is no perfect protection from a large attack, in the sense that even without the requirement of explanations, an attacker could create a surrogate model based on queries alone, given enough of them. Explanations make this problem worse in that, unless they are extremely specific to the query ("people with exactly your credit history all had their loan applications denied") they tend to reveal whole subspaces. In this work, we aimed to strike a balance between providing faithful explanations and protecting models.

Our approach can be directly used in practice for an immense number of applications, and would be valuable to those creating models for credit scoring, car insurance, health insurance, advertising or news agencies that may reject ads or articles based on content restrictions, social media companies that can reject posts, employment, rental housing applications, and utilities (water/natural gas/electricity).

Our faithful explanations can benefit users in several ways: they can verify their data to ensure the decision is accurate, appeal the decision if the model's logic appears to be faulty, consider legal action against the decision-maker, make improvements to their situation, or reach out to decision-makers for further information.

We believe this work may open an interesting exchange within the research community about how much information should be shared with end-users regarding the explanation behind their decisions from the lens of protecting intellectual property.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Gaël Aglin, Siegfried Nijssen, and Pierre Schaus. Learning optimal decision trees using caching branch-and-bound search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3146–3153, 2020.

Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research*, 18:1–78, 2018.

anonymized for review. Fast sparse rule sets. unpublished, 2024.

Maryam Ashoori and Justin D. Weisz. In AI we trust? Factors that influence trustworthiness of AI-infused decision-making processes. *arXiv e-print arXiv:1912.02675*, 2019.

Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpretability via model extraction. *4th Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML 2017)*, 2017.

Alina Beygelzimer, Daniel J Hsu, John Langford, and Tong Zhang. Agnostic active learning without constraints. In *Advances in Neural Information Processing Systems*, volume 23, 2010.

Leo Breiman. Random forests. *Machine Learning*, 45 (1):5–32, 2001.

Leo Breiman, Jerry H. Friedman, R Olshen, and CJ Stone. *Classification and Regression Trees*. Wadsworth, 1984.

Miles Brundage, Shahar Avin, Jasmine Wang, Haydn Belfield, Gretchen Krueger, and Gillian Hadfield et al. Toward trustworthy AI development: Mechanisms for supporting verifiable claims. *arXiv e-print arXiv:2004.07213*, 2020.

Consumer Financial Protection Bureau. CFPB acts to protect the public from black-box credit models using complex algorithms. `https://www.consumerfinance.gov/about-us/newsroom/cfpb-acts-to-protect-the-public-from-black-box-credit-models-using-complex-algorithms/`, may 2022.

Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. Exploring connections between active learning and model extraction. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1309–1326, 2020.

Debdatta Chatterjee. Loan prediction problem dataset. `https://www.kaggle.com/datasets/altruistdelhite04/loan-prediction-problem-dataset`, 2018.

Jacson Rodrigues Correia-Silva, Rodrigo F Berriel, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Copycat CNN: Stealing knowledge by persuading confession with random non-labeled data. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018.

Sanjeeb Dash, Oktay Günlük, and Dennis Wei. Boolean decision rules via column generation. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Emir Demirović, Anna Lukina, Emmanuel Hebrard, Jeffrey Chan, James Bailey, Christopher Leckie, Kotagiri Ramamohanarao, and Peter J Stuckey. Murtree: Optimal decision trees via dynamic programming and search. *Journal of Machine Learning Research*, 23(26):1–47, 2022.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

European Parliament and Council of the European Union. General data protection regulation, 2016. URL `https://data.europa.eu/eli/reg/2016/679/oj`.

Fatima Ezzeddine, Omran Ayoub, and Silvia Giordano. Knowledge distillation-based model extraction attack using private counterfactual explanations. *arXiv preprint arXiv:2404.03348*, 2024.

FICO, Google, Imperial College London, MIT, University of Oxford, UC Irvine, and UC Berkeley. Explainable Machine Learning Challenge. `https://community.fico.com/s/explainable-machine-learning-challenge`, 2018.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57, 2017.

TJ Hastie and RJ Tibshirani. *Generalized Additive Models*, volume 43. CRC Press, 1990.

Sanjay Kariyappa, Atul Prakash, and Moinuddin K Qureshi. Maze: Data-free model stealing attack using zeroth-order gradient estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13814–13823, 2021.

Richard M Karp. *Reducibility among combinatorial problems*. Springer, 2010.

Himabindu Lakkaraju and Osbert Bastani. "How do I fool you?" Manipulating user trust via misleading black box explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 79–85, 2020.

Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 131–138, 2019.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, (IJCAI)*, pages 2801–2807, 7 2019.

Jimmy Lin, Chudi Zhong, Diane Hu, Cynthia Rudin, and Margo Seltzer. Generalized and scalable optimal sparse decision trees. In *International Conference on Machine Learning*, pages 6150–6160, 2020.

Jiachang Liu, Chudi Zhong, Margo Seltzer, and Cynthia Rudin. Fast sparse classification for generalized linear and additive models. In *Proceedings of Artificial Intelligence and Statistics (AISTATS)*, 2022.

Samuele Lo Piano. Ethical principles in machine learning and artificial intelligence: cases from the field and possible ways forward. *Humanities and Social Sciences Communications*, 7(1):1–7, 2020.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158, 2012.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56–67, 2020.

Takayuki Miura, Toshiki Shibahara, and Naoto Yanai. Megex: Data-free model extraction attack against gradient-based explainable AI. In *Proceedings of the 2nd ACM Workshop on Secure and Trustworthy Deep Learning Systems*, page 56–66, 2024.

Itay Mosafi, Eli Omid David, and Nathan S Netanyahu. Stealing knowledge from protected deep neural networks using composite unlabeled data. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14:265–294, 1978.

Truc Nguyen, Phung Lai, Hai Phan, and My T Thai. Xrand: Differentially private defense against explanation-guided attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11873–11881, 2023.

Abdullah Caglar Oksuz, Anisa Halimi, and Erman Ayday. Autolycus: Exploiting explainable AI (XAI) for model extraction attacks against decision tree models. *arXiv preprint arXiv:2302.02162*, 2023.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4954–4963, 2019.

Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, and Vinod Ganapathy. A framework for the extraction of deep neural networks by leveraging public data. *arXiv preprint arXiv:1905.09165*, 2019.

Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, and Vinod Ganapathy. Activethief: Model extraction using active learning and unannotated public data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 865–872, 2020.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Li Pengcheng, Jinfeng Yi, and Lijun Zhang. Query-efficient black-box attack by active learning. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1200–1205, 2018.

Robert Nikolai Reith, Thomas Schneider, and Oleksandr Tkachenko. Efficiently stealing your machine learning models. In *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*, pages 198–210, 2019.

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.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Cynthia Rudin and Yaron Shaposhnik. Globally-consistent rule-based summary-explanations for machine learning models: application to credit-risk evaluation. *Journal of Machine Learning Research*, 24 (16):1–44, 2023.

Cynthia Rudin and Kiri L. Wagstaff. Machine learning for science and society. *Machine Learning*, 95(1), 2014.

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistic Surveys*, 16:1–85, 2022.

Sunandini Sanyal, Sravanti Addepalli, and R Venkatesh Babu. Towards data-free model stealing in a hard label setting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15284–15293, 2022.

Yi Shi, Yalin E Sagduyu, and Alexander Grushin. How to steal a machine learning classifier with deep learning. In *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–5, 2017.

Yi Shi, Yalin E Sagduyu, Kemal Davaslioglu, and Jason H Li. Active deep learning attacks under strict rate limitations for online api calls. In *2018 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–6, 2018a.

Yi Shi, Yalin E Sagduyu, Kemal Davaslioglu, and Jason H Li. Generative adversarial networks for black-box api attacks with limited training data. In *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 453–458, 2018b.

K Simonyan, A Vedaldi, and A Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328, 2017.

Daniel Teitelman, Itay Naeh, and Shie Mannor. Stealing black-box functionality using the deep neural tree architecture. *arXiv preprint arXiv:2002.09864*, 2020.

Scott Thiebes, Sebastian Lins, and Ali Sunyaev. Trustworthy artificial intelligence. *Electronic Markets*, 2020.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction APIs. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 601–618, 2016.

Jean-Baptiste Truong, Pratyush Maini, Robert J Walls, and Nicolas Papernot. Data-free model extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4771–4780, 2021.

Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. A Bayesian framework for learning rule sets for interpretable classification. *Journal of Machine Learning Research*, 18(70):1–37, 2017.

Yixu Wang and Xianming Lin. Enhance model stealing attack via label refining. In *2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pages 1040–1043, 2022.

Yongjie Wang, Hangwei Qian, and Chunyan Miao. Dualcf: Efficient model extraction attack from counterfactual explanations. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1318–1329, 2022.

Yi Xie, Mengdie Huang, Xiaoyu Zhang, Changyu Dong, Willy Susilo, and Xiaofeng Chen. Game: Generative-based adaptive model extraction attack. In *European Symposium on Research in Computer Security*, pages 570–588, 2022.

Anli Yan, Ruitao Hou, Xiaozhang Liu, Hongyang Yan, Teng Huang, and Xianmin Wang. Towards explainable model extraction attacks. *International Journal of Intelligent Systems*, 37(11):9936–9956, 2022.

Anli Yan, Ruitao Hou, Hongyang Yan, and Xiaozhang Liu. Explanation-based data-free model extraction attacks. *World Wide Web*, 26(5):3081–3092, 2023.

Xiaoyong Yuan, Leah Ding, Lan Zhang, Xiaolin Li, and Dapeng Oliver Wu. ES attack: Model stealing against deep neural networks without data hurdles. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(5):1258–1270, 2022.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] Please see Sections 4 and 5.

(b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] Please see Section 4.

(c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] We will provide the code in the supplement.

2. For any theoretical claim, check if you include:

(a) Statements of the full set of assumptions of all theoretical results. [Yes] Please see Section 4.

(b) Complete proofs of all theoretical results. [Yes] Please see Appendix.

(c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] A summary of datasets and experimental setup are discussed in the Appendix. We will provide the code in the supplement.

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] Please see Appendix.

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] Please see Section 5 and Appendix.

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] Please see Appendix.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [Yes]

(b) The license information of the assets, if applicable. [Not Applicable]

(c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable] All datasets and baselines we use are publicly available.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Models That Are Interpretable But Not Transparent Supplementary Materials

A Theorems and proofs

Theorem 1. *Let q be the query with $f(q) = 1$, e^{base} be the set of conditions used by the rule in f that q satisfies, and $C_q \subseteq C \backslash e^{base}$ be the additional conditions satisfied by q. Problem 2 of selecting a subset $e^{add} \subseteq C_q$ such that the intersection of the selected set of samples covers the minimum number of elements in $\tilde{\mathbf{X}}$ is equivalent to the maximum coverage problem: selecting a subset $e^{add} \subseteq C_q$ such that the union of selected samples cover the maximum of the complement set $\tilde{\mathbf{X}}^c$.*

Proof. Let $\tilde{\mathbf{X}}_{\text{base}}$ be the samples captured by e^{base}. The objective can be written as

$$\min_{e^{\text{add}} \subseteq C_q} \text{supp}(\{e^{\text{base}}, e^{\text{add}}\}, \tilde{\mathbf{X}}) = \min_{e^{\text{add}} \subseteq C_q} \left| S_{e^{\text{base}}}(\tilde{\mathbf{X}}) \cap S_{e^{\text{add}}}(\tilde{\mathbf{X}}) \right| = \min_{e^{\text{add}} \subseteq C_q} \left| S_{e^{\text{add}}}(\tilde{\mathbf{X}}_{\text{base}}) \right|.$$

Minimization of the intersection is equivalent to maximizing the complement, i.e,

$$\max_{e^{\text{add}} \subseteq C_q} \left| \left(S_{e^{\text{add}}}(\tilde{\mathbf{X}}_{\text{base}}) \right)^c \right| = \max_{e^{\text{add}} \subseteq C_q} \left| \left(\cap_{c_j \in e^{\text{add}}} S_{c_j}(\tilde{\mathbf{X}}_{\text{base}}) \right)^c \right|$$

which, by De Morgan's law,

$$= \max_{e^{\text{add}} \subseteq C_q} \left| \left(\cup_{c_j \in e^{\text{add}}} S_{c_j}(\tilde{\mathbf{X}}_{\text{base}})^c \right) \right|.$$

Since c_j are binary conditions, the complement of samples captured by c_j is the collection of samples captured by $\neg c_j$, i.e.,

$$\max_{e^{\text{add}} \subseteq C_q} \left| \left(\cup_{c_j \in e^{\text{add}}} S_{\neg c_j}(\tilde{\mathbf{X}}_{\text{base}}) \right) \right|.$$

Therefore, Problem (2) is the same as the maximum coverage problem. □

B Datasets and experimental setup

We use three credit datasets FICO credit risk dataset (FICO et al., 2018), German credit dataset from UCI (Dua and Graff, 2017), and a loan approval prediction problem dataset from Kaggle (Chatterjee, 2018). Details about these datasets are in Table 2.

Dataset	Samples	Features	Description
FICO	10459	23	whether someone would default on a loan
German credit	1000	20	predict the approval of credit applications
Kaggle loan prediction problem	480	11	predict the approval of loan applications

Table 2: Dataset summary

All experiments are run on a 2.7Ghz (768GB RAM 48 cores) Intel Xeon Gold 6226 processor. Cplex Studio 22.1 is used for FaithfulDefense IP and FaithfulDefense IP-RA. The time limit is set to 180 seconds. To train the surrogate model, we use CART, Random Forest, and GBDT from scikit-learn (Pedregosa et al., 2011). We set CART's max_depth to 5 and use the default setting for other hyperparameters.

C More experimental results

In this section, we show more experimental results.

Interpretable models are as accurate as complex models. We compare models produced by FastSRS, Random Forest, and GBDT on the original datasets in Table 3. FastSRS can achieve performance comparable to complex models, indicating that interpretable models can replace complex models in real applications. This motivates us to develop models that are interpretable but not transparent for practical use.

Dataset	FastSRS	Random Forest	GBDT
FICO	0.713	0.725	**0.73**
German credit	0.755	**0.765**	0.755
Kaggle loan prediction problem	**0.844**	0.802	0.833

Table 3: Test accuracy of interpretable models versus complex models on the original datasets. FastSRS achieves test accuracy comparable to Random Forest and GBDT.

How Many Positive Training Points are Covered by the Explanation. Figure 4 demonstrates that our FaithfulDefense (red, orange, and pink curves) reveals less information (i.e., captures fewer positive samples) in training sets for all three datasets using three different querying strategies, as the red, orange, and pink curves consistently fall below the green and blue curves.

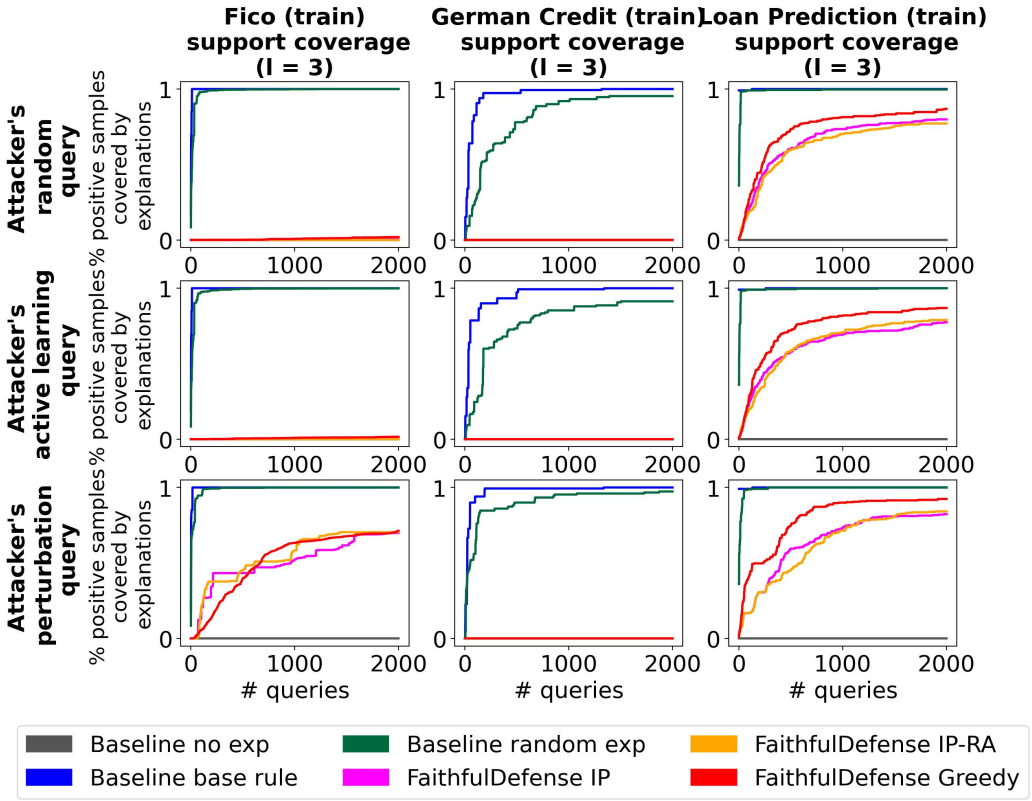

Figure 4: Number of queries vs. the proportion of positive samples covered by explanations on the training set. (max length $l = 3$).

Explanation Generation Timing. Figure 5 shows the explanation generation time on three datasets when three different querying strategies are used. We observe that the time taken by our FaithfulDefense Greedy is generally fast, usually less than 0.1 second for different querying strategies, while FaithfulDefense IP and FaithfulDefense IP-RA have higher time costs.

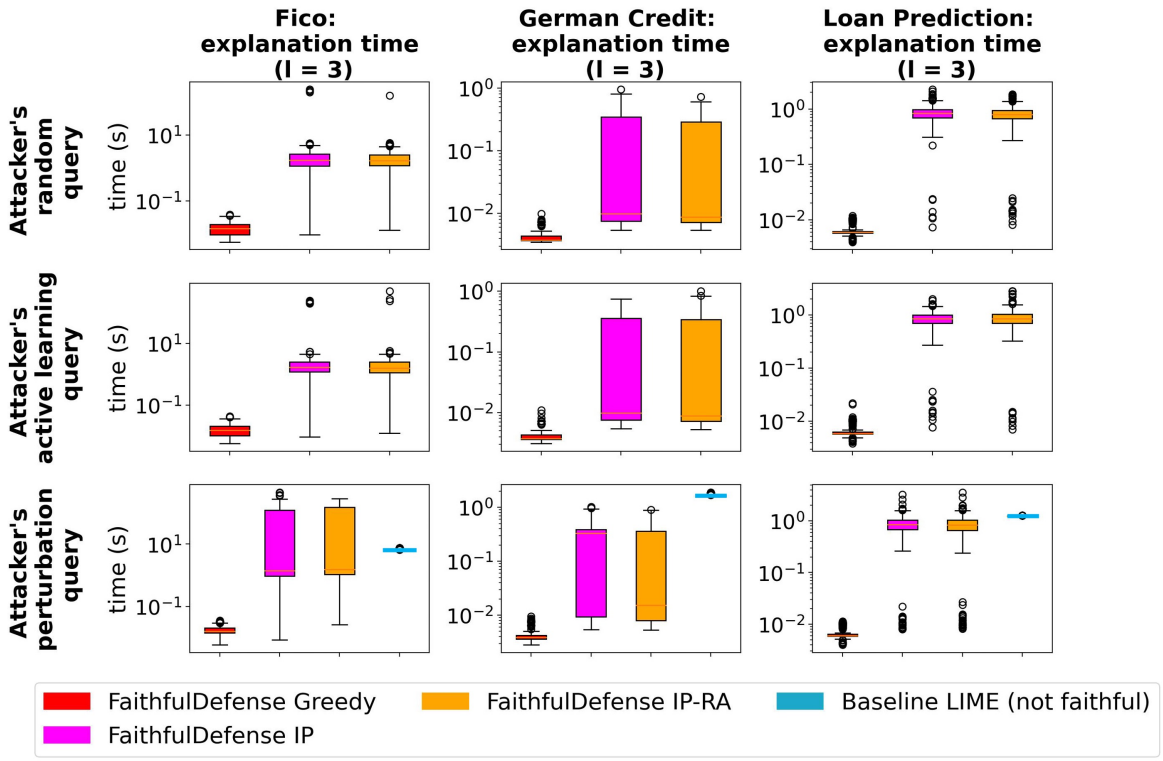

Figure 5: Time consumption of generating explanations. (max length $l = 3$)

Test Accuracy of Surrogate Models. Due to the space limitation, we only showed the test performance of surrogate models trained using CART in the main paper. Figure 6 shows the test performance when Random Forest and GBDT are used to train the surrogate model. We use the default setting for these two model configurations. Test performance using Random Forest is similar to that of CART, where providing our explanations usually requires more queries compared with the other two baselines and sometimes providing our explanations yields test performance close to providing no explanations. GBDT is more powerful than CART and Random Forest; however, Random Forest and GBDT are black box models. They cannot be used to replace the interpretable models to return faithful explanations.

Impact of Changing Max Length. We also study how the value of max length l influences information leakage and the attacker's performance. Figure 7 shows the number of queries versus support coverage when l is set to 5 and 7. For all explanation methods, support coverage decreases as l increases on the test set since more conditions are used in the explanation. Our FaithfulDefense outperforms the baselines, as the red, orange, and pink curves are always below the green and blue curves. Additionally, our FaithfulDefense outperforms the baselines on surrogate model performance. As shown in Figure 8-10, the red, orange, and pink curves for the FICO and German credit datasets are generally above the green and blue curves, regardless of the attacker's querying strategy or the model class used to train the surrogate model.

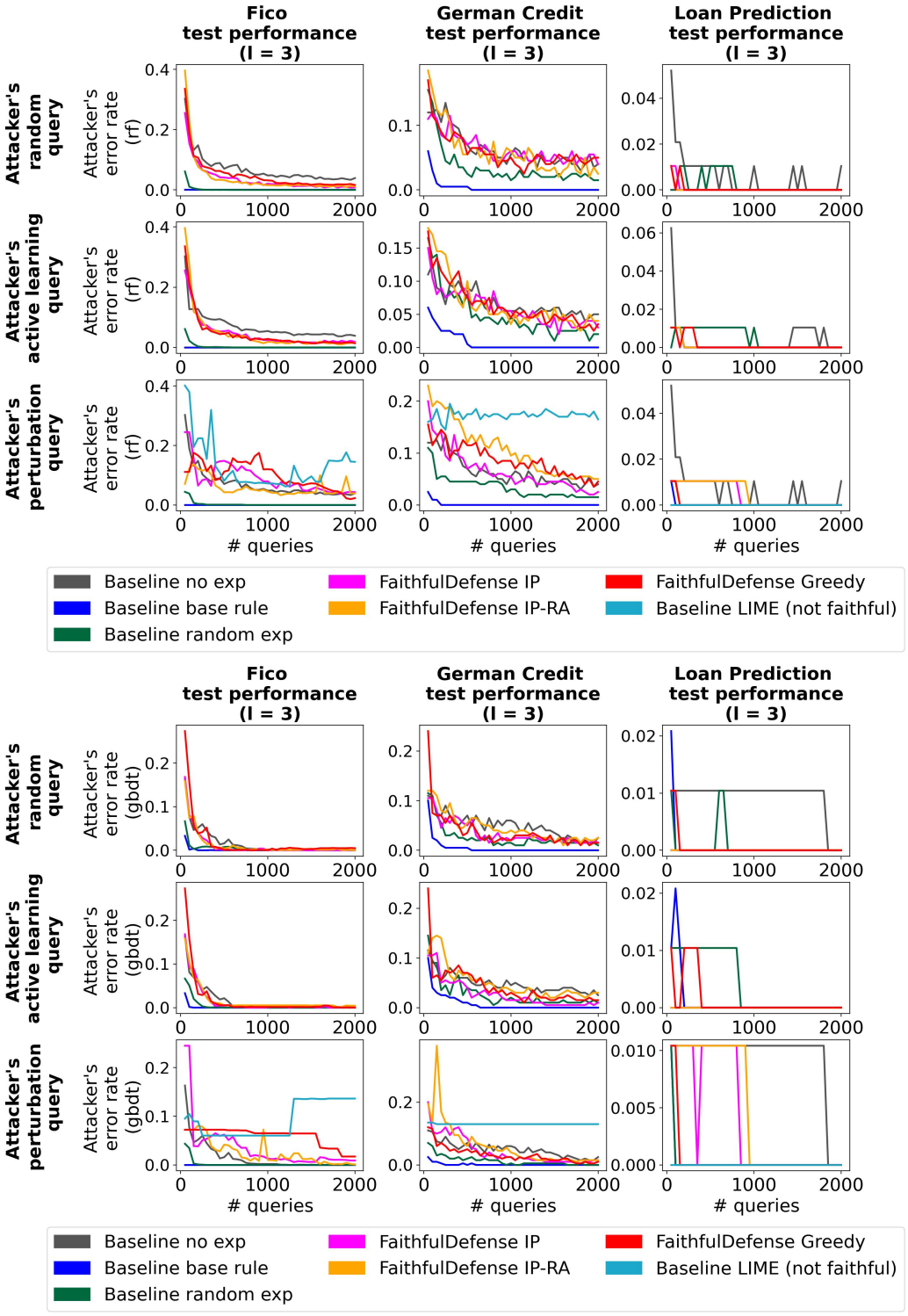

Figure 6: Comparison of test performance between base and surrogate models. Random Forest (top) and GBDT (bottom) are used by the attacker to train the surrogate model. (max length $l = 3$).

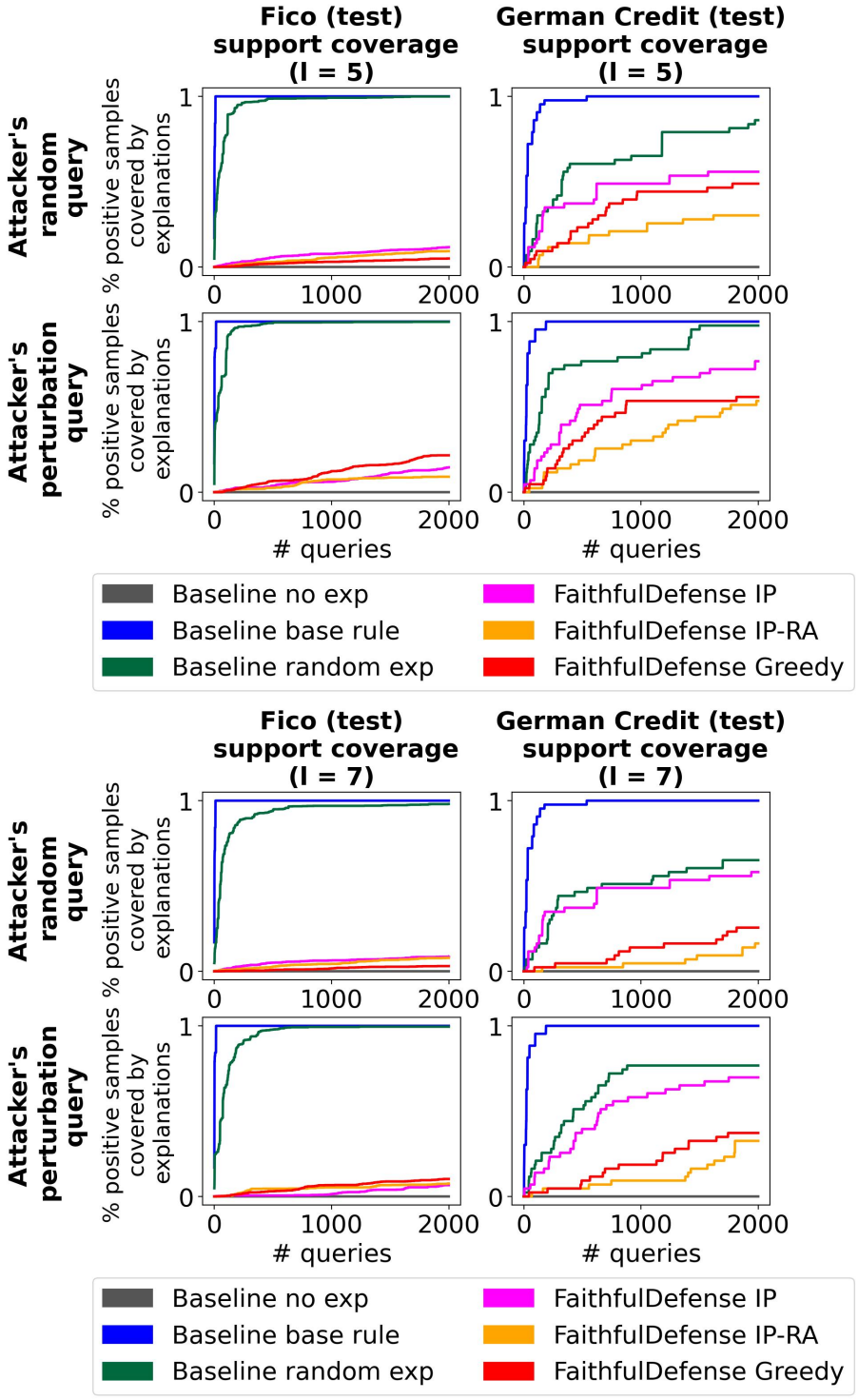

Figure 7: Number of queries vs. the proportion of positive samples covered by explanations on the test set (max length $l = 5$ and $l = 7$).

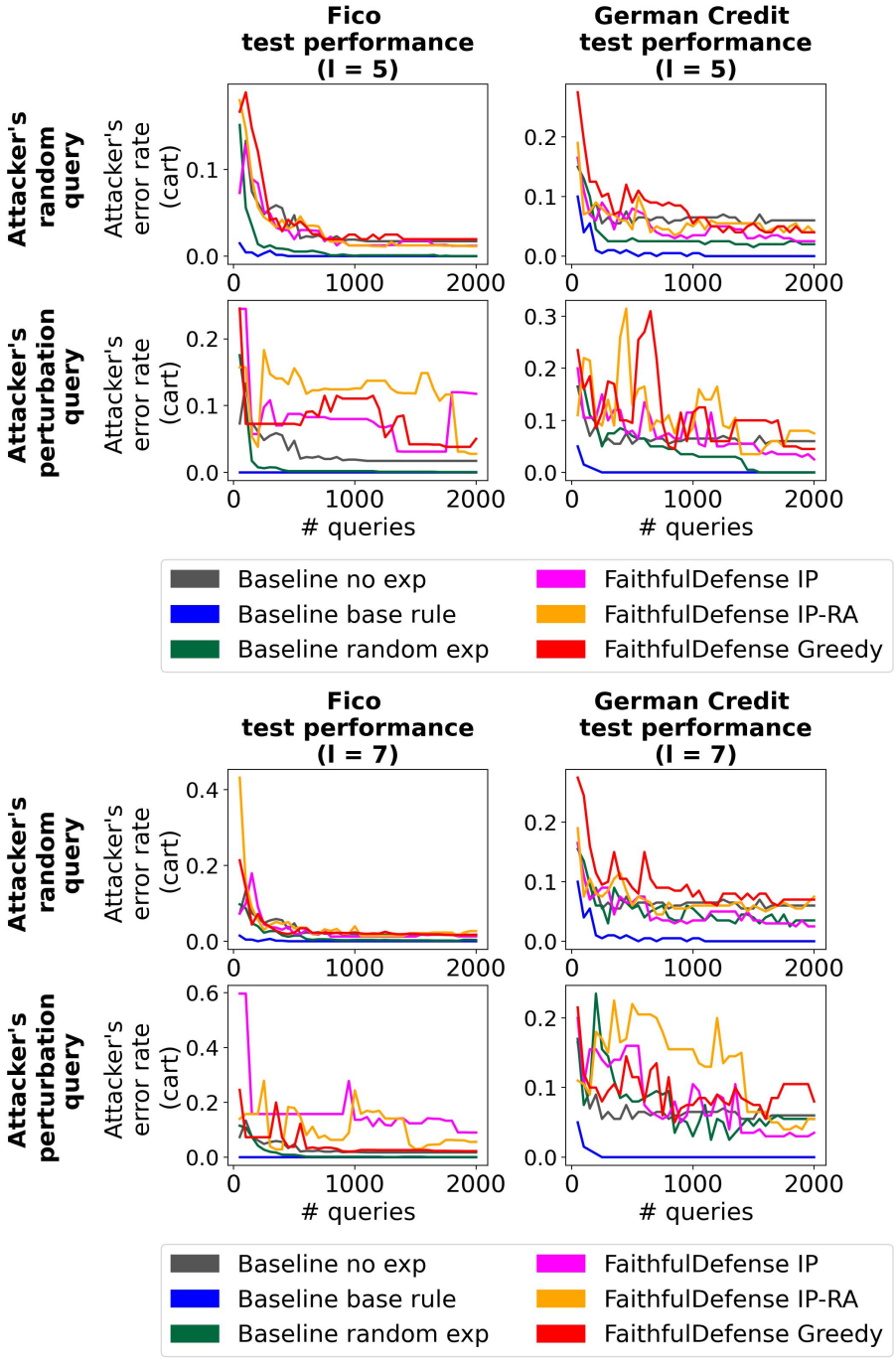

Figure 8: Comparison of test performance when max length $l = 5$ and $l = 7$ with respect to the CART model.

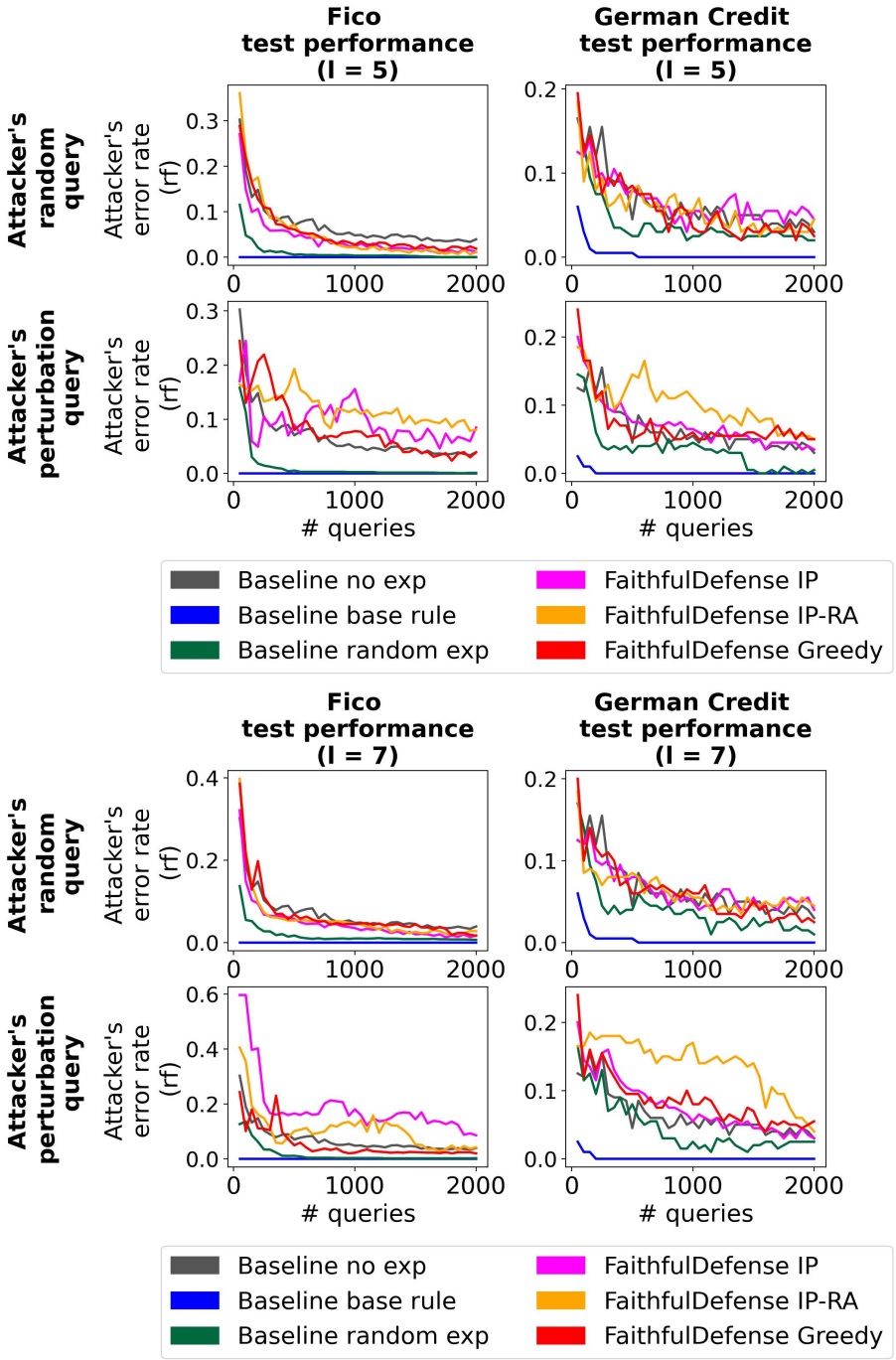

Figure 9: Comparison of test performance when max length $l = 5$ and $l = 7$ with respect to the Random Forest model.

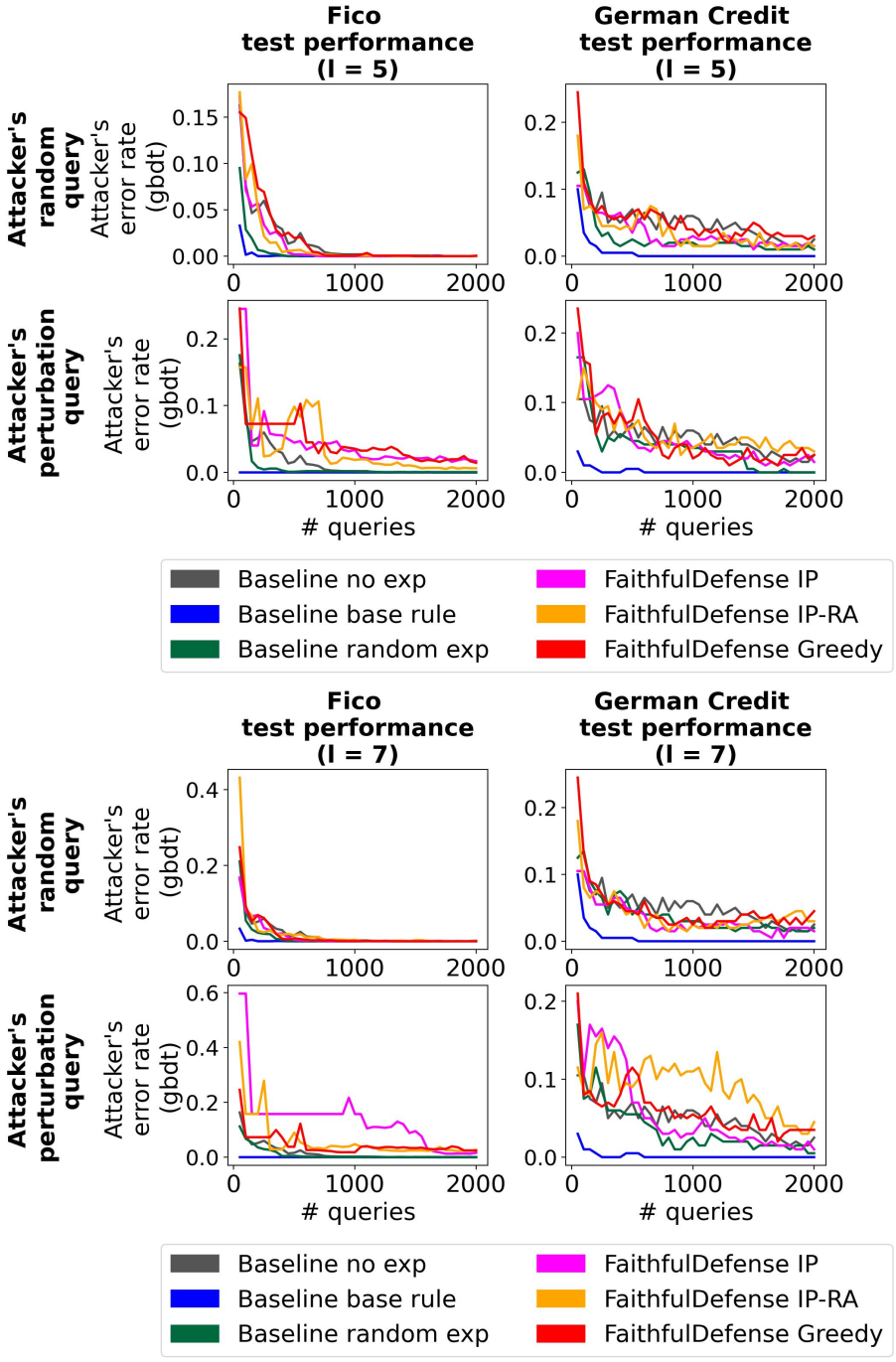

Figure 10: Comparison of test performance when max length $l = 5$ and $l = 7$ with respect to the GBDT model.

D FaithfulDefense for other model classes

Generalized additive models (Hastie and Tibshirani, 1990) linearly combine flexible component functions for each feature:

$$g(E[y]) = \omega_0 + f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p), \tag{8}$$

where x_j indicates the jth feature, the f_j's are learned univariate component functions that are possibly nonlinear, and $g(\cdot)$ is a link function, e.g., the identity function for regression or the inverse logistic function for classification. Each shape function f_j operates on only one feature x_j, thus the shape functions can directly be plotted. This makes GAMs interpretable since the entire model can be easily visualized.

In practice, each continuous feature is usually divided into bins (Lou et al., 2012; Liu et al., 2022), thereby its shape function can be viewed as a set of step functions, i.e.,

$$f_j(x_j) = \sum_{k=0}^{B_j-1} \omega_{j,k} \cdot \mathbf{1}[b_{j,k} < x_j \leq b_{j,k+1}], \tag{9}$$

where $\{b_{j,k}\}_{k=0}^{B_j}$ are the bin edges of feature j, leading to B_j total bins.

Sparsity regularization, i.e., the ℓ_0 penalty on the number of steps, is often used to encourage generalization and avoid constructing overly complicated models. A sparse GAM model with piecewise constant shape functions can be efficiently converted into logic models. For example, given p shape functions and each shape function has B_j bins, the GAM model can be converted into a multi-split decision tree with at most depth p. In reality, not all leaves have to reach the depth p. If concatenating bins from other shape functions will not change the leaf prediction, then we can stop early. Any decision tree can then be converted into decision sets by extracting leaf paths with positive predictions.

E Relationship to recourse

Our proposed FaithfulDefense does not aim to provide recourse. There are three issues in providing recourse discussed below: (1) Recourse has technical challenges in being too prescriptive; (2) Recourse reveals a tremendous amount about the model, making it difficult to keep it non-transparent; (3) Laws require explanations, but not recourse. This could easily be due to the two issues listed above. Let us discuss this in more depth below.

(1) Recourse technical challenges and being too prescriptive. Recourse requires being able to change the features, and knowing costs for each possible change to the features.

- Many features may not be able to be changed. For loan decisions, the features are typically based on credit history and job status, and there is generally no way for loan applicants to change those. Thus, recourse may simply not be possible.

- In other situations where features could be changed, there is a cost for changing each feature in the recourse framework, but those costs might be unknown or too high. How much would it cost the loan applicant to change jobs to make more per month? It is not clear they would be able to do it at all let alone having the bank know the cost for the applicant to do that. Thus, any possible recourse given to the user may not estimate these costs correctly and may not be actionable in reality (i.e., may be too prescriptive).

(2) Recourse reveals a lot about the model. If we told the user that changing feature 1 would change the decision, they would then know that feature 1 is actually used in the model, whereas our approach hides that information in order to keep the model non-transparent. So, any defensive algorithm would fail to keep the model's variables as a secret. Since the bank would not use a model that is easily revealed, it most likely would resort to a black box with unfaithful explanations again. Another option is to provide an unfaithful recourse, but this would defeat the point of providing the recourse in the first place.

(3) The obligation of loan lenders to give an explanation for each denial is sourced from "Right to explanation" and other similar laws (e.g., in the Code of Federal Regulations, the "Form of Equal Credit Opportunity Act (ECOA) notice and statement of specific reasons"). Those laws only govern the right to explanation, not the right to recourse, possibly due to the issues mentioned above in recourse not actually being possible, and, if possible, being impractical to compute effectively.