Constraint-Aware Tabular Score Attacks with Attribution-Guided Boundary Search

Anonymous authors
Paper under double-blind review

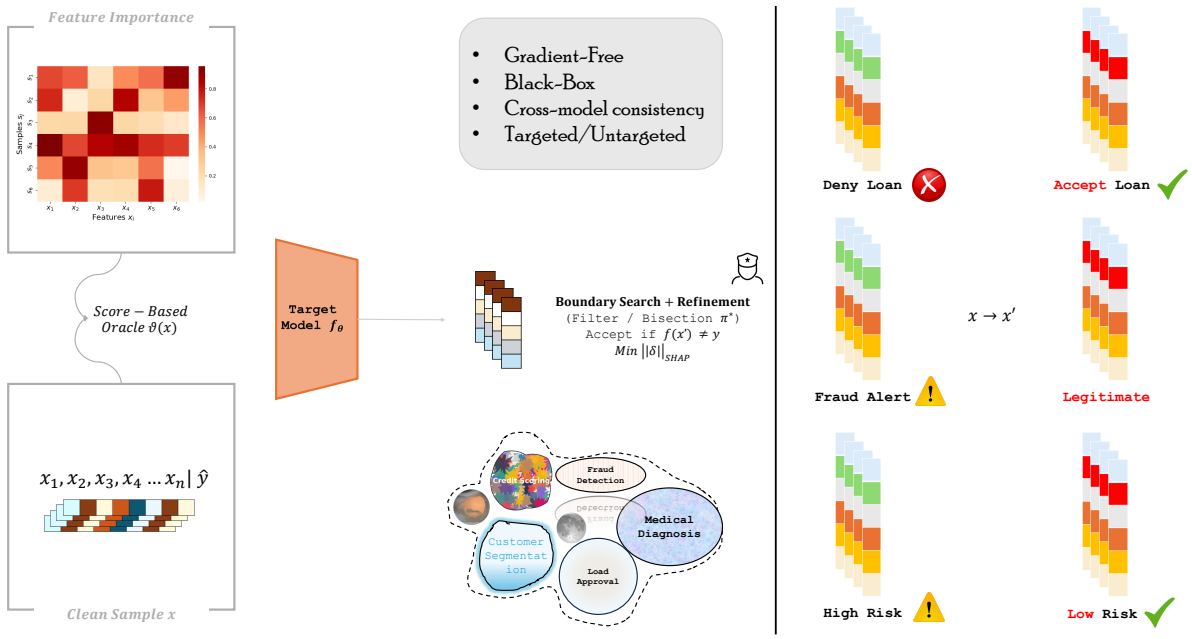

Figure 1: Schematic overview of the proposed constraint-aware, query-efficient black-box attack framework for tabular models. Given a clean input sample x, the attack first obtains a low-budget attribution-guided ranking of mutable features, then performs score-based boundary localization and feasibility-preserving refinement under domain constraints. The resulting adversarial sample x' induces targeted or untargeted misclassification while respecting mixed continuous/categorical feature validity, immutability constraints, and perturbation budgets.

Abstract

Adversarial robustness in structured data remains less developed than in vision and language, despite the central role of tabular models in high-stakes decision systems. We propose a **constraint-aware, query-efficient black-box attack framework** for tabular and hybrid feature spaces that couples feature prioritization, boundary localization, and feasibility-preserving refinement into a single attack procedure. Given score-based oracle access, the framework identifies influential mutable features through low-budget attribution-guided queries, searches for nearby decision-boundary crossings using score margins, and refines adversarial examples while enforcing immutability, categorical validity, range constraints, and perturbation-cost budgets. Across multiple public datasets and heterogeneous model families, the proposed framework achieves targeted attack success rates above **90%** under strict query and feasibility constraints. We further analyze query cost, sparsity, feasibility, and component-level contributions, supported by analysis linking attribution-guided

ranking to local sensitivity and boundary localization to logarithmic query complexity after bracketing.

1 Introduction

Tabular data sits behind many real-world decision-making systems. A loan may be approved or rejected based on financial attributes, a transaction may be flagged as fraudulent based on behavioral signals, and a medical or customer-risk model may produce decisions from a mixture of categorical, numerical, and embedded features. In these settings, predictions are not made from images or free-form text, but from structured records whose individual features often carry direct operational meaning. Over the years, strong performance on such data has been achieved by gradient-boosted tree models such as XGBoost Chen & Guestrin (2016) and CatBoost Dorogush et al. (2018), as well as classical models such as logistic regression Hosmer et al. (2013), random forests Breiman (2001), and decision trees Quinlan (1986). More recently, transformer-based and embedding-based models such as RoBERTa Liu et al. (2019) and TabPFN Hollmann et al. (2022) have also been adapted to structured classification tasks Hegselmann et al. (2023). An interesting question then arises: if tabular features are structured, constrained, and often semantically meaningful, **can an attacker still change a model decision using only small, feasible modifications?**

This question is especially important because adversarial robustness in tabular domains is less mature than in vision and language. In vision, real-world adversarial patch attacks have shown that deployed models can be highly vulnerable to carefully designed perturbations Kazoom et al. (2024). Similar concerns also arise in structured data, where prior work has shown that gradient-free and query-based attacks can manipulate tabular classifiers under black-box access. However, attacking tabular data is fundamentally different from adding unconstrained continuous noise. Many features are immutable, categorical values must remain valid, numerical values must stay within realistic ranges, and some changes may be too costly or infeasible in practice. Therefore, a useful tabular attack must not only fool the model, but also preserve domain validity.

Existing black-box attacks provide valuable search mechanisms, but they do not always treat mixed feature types, feasibility constraints, perturbation cost, and query efficiency as part of one unified protocol. To address this gap, we study score-based black-box attacks for tabular and hybrid feature spaces. Our contribution is not a new attribution method, boundary-search algorithm, or greedy refinement rule in isolation. Instead, we integrate these components into a constraint-aware attack pipeline: first ranking mutable features using low-budget attribution-guided queries, then localizing the decision boundary using score-based search, and finally refining the adversarial example while preserving feasibility. We evaluate this framework across classical, ensemble, and transformer-based tabular models, measuring targeted attack success, query cost, perturbation sparsity, feasibility violations, and component-level contributions. The results reveal persistent robustness gaps across models and datasets, even under realistic tabular constraints. **This paper makes the following contributions:**

Unified attack protocol. We introduce a unified **constraint-aware score-based black-box attack protocol** for tabular and hybrid feature spaces, integrating attribution-guided feature ranking, boundary localization, and feasibility-preserving refinement under mixed continuous and categorical constraints.

Broad empirical evaluation. We evaluate the proposed framework across multiple public datasets and heterogeneous model families, including tree-based, ensemble, linear, and transformer-based architectures, showing consistently high targeted attack success rates under realistic feasibility constraints.

Component-level analysis. We analyze query efficiency, perturbation sparsity, feasibility violation rate, perturbation cost, and ablation variants, showing how each stage of the integrated framework contributes to attack effectiveness.

2 Related Work

Adversarial Vulnerabilities Adversarial vulnerabilities in neural networks have been studied across modalities, with early NLP work showing that even high-performing models can fail under minimally altered

yet semantically equivalent inputs. The ANLI benchmark Nie et al. (2020) demonstrated the susceptibility of state-of-the-art inference systems to crafted perturbations, while black-box jailbreak approaches Pathade (2025) showed that alignment safeguards in LLMs can be bypassed. Token-level attacks such as BAE Garg & Ramakrishnan (2020) revealed that small sequence changes mislead robust classifiers, and broader studies Liu et al. (2023) documented prompt injection and embedding-space attacks in both white- and black-box settings. Retrieval-augmented adversarial detection pipelines, including Don't Lag RAG Kazoom et al. (2025b) and VAULT Kazoom et al. (2025a), further highlight the role of retrieval reasoning in adversarial robustness.

Classical Machine Learning for Tabular Data Adversarial research on structured data has emerged more recently compared to vision and NLP. For tabular tasks, gradient-boosted trees (GBTs) and ensemble methods remain dominant, with implementations such as XGBoost Chen & Guestrin (2016) and CatBoost Dorogush et al. (2018), alongside logistic regression Hosmer et al. (2013), random forests Breiman (2001), and decision trees Quinlan (1986). Formally, these models can be expressed as mappings $f : \mathbb{R}^d \to \mathbb{R}$, where d is the number of features and f the decision boundary. Early work explored evasion strategies for tree ensembles Kantchelian et al. (2016), later improving scalability for repeated attacks Cascioli et al. (2024). Neural approaches for tabular data inspired imperceptible perturbations on feature semantics Ballet et al. (2019) and targeted fraud-related attacks Cartella et al. (2021). More recent efforts proposed constrained adaptive attacks (CAA) Simonetto et al. (2024a), enforcing feature-type and semantic constraints for realistic adversarial examples.

Black-Box and Gradient-Free Attacks on Tabular Data While many adversarial attacks assume white-box access, black-box and gradient-free methods have gained traction in tabular domains. Dyrmishi et al. Dyrmishi et al. (2025) review adversarial methods in tabular ML, covering transfer- and query-limited strategies. Yang et al. Yang et al. (2022) introduced TSADV, a black-box attack with local perturbations for time-series, while population-based optimization such as ABCATTACK Cao et al. (2022) apply gradient-free search to structured data. Kireev et al. Kireev et al. (2022) proposed cost- and utility-aware threat models for realistic constraints. Benchmarking efforts He et al. (2025) compared bounded vs. unbounded and white-box vs. black-box settings, establishing a foundation for query-based and gradient-free evaluations. Imperceptibility metrics He et al. (2024) further highlight that adversarial examples must respect feature ranges, sparsity, and interdependencies. Formally, black-box adversarial attacks can be posed as optimization under query access:

$$\min_{\delta} \ \ell(f(x+\delta), y) \quad \text{s.t. } \|\delta\| \leq \epsilon, \quad \text{with query access to } f(\cdot), \tag{1}$$

where ℓ is a task-specific loss (e.g., cross-entropy). More generally, one may model attacks with cost–utility constraints:

$$\min_{\delta} \ c(\delta) - U(f(x+\delta), y) \quad \text{s.t. } \delta \in \mathcal{C}, \tag{2}$$

where $c(\delta)$ encodes perturbation cost, U denotes adversarial utility, and $\mathcal{C}$ enforces constraints such as immutability of categorical features or feature sparsity.

Decision-Based, Score-Based, and Transformer-Based Attacks on Tabular Models Black-box attacks differ in the type of oracle access they assume. Decision-based methods, such as Boundary Attack Brendel et al. (2017) and HopSkipJump Chen et al. (2019), rely primarily on predicted labels and estimate directions toward the decision boundary through repeated queries. These methods have been effective in continuous domains such as images, but their direct application to tabular data is non-trivial because categorical variables, immutable attributes, feature ranges, and cost constraints must be preserved. Score-based attacks assume a stronger oracle that returns real-valued class scores, such as probabilities or logits, and can therefore use score margins to guide boundary localization more efficiently. Beyond classical machine learning models, transformer-based architectures have recently emerged for tabular data. A notable example is TabPFN Hollmann et al. (2022), which leverages pretrained transformers to solve small-scale tabular classification tasks efficiently. Unlike tree-based ensembles, these models rely on learned feature representations, where each feature x_i is embedded and processed through attention layers before prediction. Despite their promise, systematic assessments of adversarial robustness in transformer-based tabular models remain limited, particularly under feasibility-aware black-box threat models.

XAI-Assisted Attacks, Explanation Leakage, and Constrained Tabular Robustness. Beyond their use for interpretability, explanation methods can expose additional information about model behavior and may therefore enlarge the attack surface. Prior work has shown that post-hoc explanation methods such as LIME and SHAP can be manipulated by adversarially constructed models Slack et al. (2020), and broader surveys have studied explainable machine learning under adversarial conditions Noppel & Wressnegger (2024). Explanation access can also facilitate model extraction: DualCF demonstrates that counterfactual explanations can be exploited to reconstruct target models more efficiently Wang et al. (2022), while earlier work on prediction APIs showed that confidence scores and probability outputs may leak sufficient information for model stealing Tramèr et al. (2016). This motivate treating score-based and explanation-based access as realistic security-relevant threat models rather than purely diagnostic tools. Adversarial attacks on tabular data have also been studied under increasingly realistic feasibility constraints. LowProFool introduced imperceptible adversarial attacks for tabular classifiers by weighting perturbations according to feature importance Ballet et al. (2019). More recent constrained-feature-space frameworks explicitly model immutable attributes, categorical validity, and feature relationships, including constrained gradient-based and search-based attacks such as C-PGD and MoEvA2 Simonetto et al. (2021). CAA further strengthens constrained tabular attacks by combining adaptive constrained optimization with search-based refinement against deep tabular models Simonetto et al. (2024a). TabularBench provides a standardized benchmark for evaluating adversarial robustness of tabular deep learning models under real-world constraints Simonetto et al. (2024b). In contrast to these works, our method focuses on a score-based black-box setting and integrates low-budget attribution-guided feature selection, score-based boundary localization, and feasibility-preserving refinement into a unified attack framework for mixed continuous/categorical tabular domains.

3 Preliminaries

In this section, we formalize the learning scenarios considered in this work and introduce the notation used throughout. We consider two related settings: (i) classical machine learning models operating directly on tabular features, and (ii) hybrid architectures that combine tabular inputs with dense feature embeddings.

Notation. Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ denote a dataset of N samples, where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional feature vector and $y_i \in \mathcal{Y}$ is the corresponding class label, with $\mathcal{Y} = \{1, \ldots, C\}$. Categorical attributes are mapped to numerical representations using an encoding function $g_{\text{cat}} : \mathcal{X}_{\text{cat}} \to \mathbb{R}^k$, such as one-hot encoding or learned embeddings.

Let $f_\theta : \mathbb{R}^d \to \Delta^{C-1}$ denote a classifier parameterized by θ, where Δ^{C-1} is the $(C-1)$-dimensional probability simplex. The predicted class is $\hat{y} = \arg\max_{c \in \mathcal{Y}} f_\theta(\mathbf{x})_c$. An adversarial example is an input $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}$ that changes the model prediction, or changes the confidence ordering in a targeted direction, while satisfying a perturbation budget and domain-specific validity constraints.

Scenario 1: Classical Tabular Models. In the classical setting, inputs $\mathbf{x} \in \mathbb{R}^d$ consist of structured features that may include both continuous and categorical variables. Representative models include gradient-boosted decision trees, random forests, decision trees, and linear classifiers. For example, a gradient-boosted tree classifier can be written as $f_\theta^{\text{GBT}}(\mathbf{x}) = \text{softmax}(\sum_{m=1}^M h_m(\mathbf{x}))$, while a logistic regression classifier can be written as $f_\theta^{\text{LR}}(\mathbf{x}) = \text{softmax}(\mathbf{w}^\top \mathbf{x} + b)$, where h_m denotes an individual regression tree and M is the number of boosting stages. Perturbations $\boldsymbol{\delta}$ are constrained to preserve feature validity, yielding $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}$ with $\mathbf{x}' \in \mathcal{X}_{\text{valid}}$.

Scenario 2: Hybrid Tabular-Embedding Pipelines. In hybrid pipelines, the input is decomposed into structured tabular features $\mathbf{x}^{(t)} \in \mathbb{R}^{d_t}$ and dense embedding features $\mathbf{x}^{(e)} \in \mathbb{R}^{d_e}$. The embedding component may be produced by a modality-specific encoder E_ψ from raw inputs $z \in \mathcal{Z}_{\text{raw}}$, such that $\mathbf{x}^{(e)} = E_\psi(z)$ and $\mathbf{x}_{\text{combined}} = [\mathbf{x}^{(t)}; \mathbf{x}^{(e)}] \in \mathbb{R}^{d_t + d_e}$. A downstream classifier predicts $\hat{y} = \arg\max_{c \in \mathcal{Y}} f_\theta(\mathbf{x}_{\text{combined}})_c$.

When embedding features are exposed to the downstream classifier, adversarial perturbations may be applied to the structured and embedding components as $\mathbf{x}'_{\text{combined}} = [\mathbf{x}^{(t)} + \boldsymbol{\delta}_t; \mathbf{x}^{(e)} + \boldsymbol{\delta}_e]$, subject to modality-specific semantic, syntactic, and feasibility constraints.

Unified Adversarial Objective. Across both settings, the proposed score-based black-box attack seeks a feasible perturbation $\boldsymbol{\delta}$ that changes the classifier output while minimizing perturbation magnitude or cost. For a targeted attack with target class $y^\star \neq y$, the objective is $\min_{\boldsymbol{\delta}} \|\boldsymbol{\delta}\|_p$ subject to $f_\theta(\mathbf{x}+\boldsymbol{\delta})_{y^\star} > f_\theta(\mathbf{x}+\boldsymbol{\delta})_y$ and $\mathbf{x}+\boldsymbol{\delta} \in \mathcal{X}_{\text{valid}}$. For untargeted attacks, the target constraint is replaced by $\arg\max_{c\in\mathcal{Y}} f_\theta(\mathbf{x}+\boldsymbol{\delta})_c \neq y$.

Threat Model. We consider a **score-based black-box** threat model in which the adversary can query a trained classifier $f : \mathbb{R}^d \to \Delta^{C-1}$ and observe real-valued class scores, such as probabilities or normalized confidence scores, together with the induced predicted label. The method is not designed for a label-only, decision-based oracle. Score access is used to obtain a low-budget attribution-guided ranking of mutable features and to guide boundary localization through score margins. This ranking is only one module in the overall attack pipeline, which also includes feasibility-aware proposal generation, score-based boundary localization, and constraint-preserving refinement.

4 Methodology

The framework is organized around three coupled components: low-budget attribution-guided feature selection, score-based boundary localization, and feasibility-preserving refinement. Starting from a clean input $\mathbf{x}$, the method first estimates a coarse ranking of mutable features using score-based perturbation queries. This ranking is then used to guide the search toward promising regions of the decision boundary, while projection and snapping operators ensure that all candidate perturbations remain valid under mixed continuous/categorical feature constraints. Figure 1 illustrates the proposed constraint-aware, query-efficient black-box attack framework for tabular models. Figure 2 visualizes the movement of an input toward a score-based decision boundary.

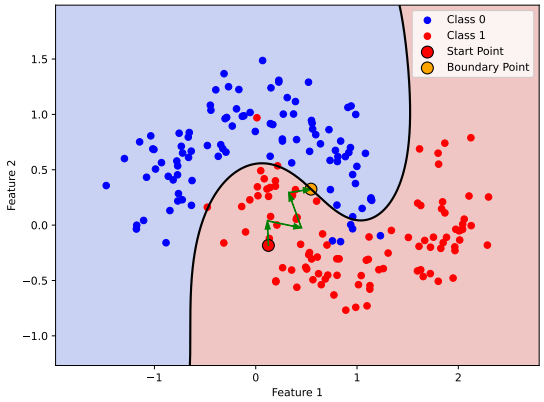

Figure 2: Illustration of the iterative movement of an input toward a score-based decision boundary.

We propose a *score-based black-box attack framework* for structured data that operates under two settings: (i) classical tabular classifiers, and (ii) hybrid tabular-embedding pipelines. The attack requires only score-based oracle queries and does not require access to model parameters, gradients, or training data. Importantly, the contribution is not any individual component in isolation, such as attribution, bisection, or coordinate-wise refinement. Rather, the method integrates these components into a unified pipeline designed for mixed tabular domains with feasibility constraints, including immutable features, categorical validity, range constraints, and cost budgets. Next, we formalize the attack objective and describe each component of the framework.

Problem Setting: Adversarial Objective. Let $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$ denote the feature space with d attributes, where each $\mathcal{X}_j$ is either a continuous interval or a finite categorical set. Let $\mathcal{Y} = \{1, \ldots, C\}$ be the label set. We assume access to a score-based black-box oracle $\mathcal{O} : \mathcal{X} \to \Delta^{C-1}$ that returns real-valued class scores, such as probabilities or normalized confidence scores. The predicted label is $\hat{y}(\mathbf{x}) =$

$\arg\max_{c\in\mathcal{Y}} \mathcal{O}_c(\mathbf{x})$. Given an input $\mathbf{x} \in \mathcal{X}$ with original label y, the goal is to find a feasible adversarial instance $\mathbf{x}' \in \mathcal{X}_{\text{valid}}$ that changes the model prediction while minimizing perturbation cost. For untargeted attacks, the constraint is $\arg\max_{c\in\mathcal{Y}} \mathcal{O}_c(\mathbf{x}') \neq y$. For targeted attacks with target class $y^\star \neq y$, the constraint is $\arg\max_{c\in\mathcal{Y}} \mathcal{O}_c(\mathbf{x}') = y^\star$. The feasible set $\mathcal{X}_{\text{valid}} \subseteq \mathcal{X}$ enforces feature ranges, integrality, categorical validity, immutable attributes, and application-specific cost constraints. We define the targeted score margin as $m_{y^\star,y}(\mathbf{x}') = \mathcal{O}_{y^\star}(\mathbf{x}') - \mathcal{O}_y(\mathbf{x}')$. A targeted attack succeeds when $m_{y^\star,y}(\mathbf{x}') > 0$. For untargeted attacks, we use the margin $m_{\text{unt}}(\mathbf{x}') = \max_{c\neq y} \mathcal{O}_c(\mathbf{x}') - \mathcal{O}_y(\mathbf{x}')$, and the attack succeeds when $m_{\text{unt}}(\mathbf{x}') > 0$.

4.1 Low-Budget Attribution-Guided Initialization

The first stage computes a coarse ranking of mutable features using low-budget, score-based attribution estimates. We use SHAP-inspired perturbation scores as a model-agnostic ranking signal, but we do not require exact Shapley values. The purpose of this stage is not to provide faithful post-hoc explanations of the target model, but to identify features that are likely to produce large changes in the score margin. Since the attacker has no access to model parameters, gradients, or training data, the ranking is estimated through randomized feature masking and score queries to the target oracle. This procedure explicitly assumes access to real-valued prediction scores and is therefore not applicable in a label-only setting. Let $\phi_j(\mathbf{x})$ denote the estimated attribution score of feature j for input $\mathbf{x}$. We use the vector $\Phi(\mathbf{x}) = [|\phi_1(\mathbf{x})|, \ldots, |\phi_d(\mathbf{x})|]$ as an initialization signal for feature selection, proposal generation, and refinement. This module is intentionally approximate: only the relative ordering of features is needed, not exact attribution values.

Continuous features. For continuous attributes, the attribution score $|\phi_j(\mathbf{x})|$ is used to prioritize features whose perturbation is expected to induce a large change in the score margin. Perturbation magnitudes are restricted by the valid feature range and by the global cost budget.

Categorical features. For categorical attributes, the same attribution ranking determines which features are selected for modification. The actual update is restricted to admissible category substitutions, so that the perturbed instance remains valid in the original tabular domain.

Cost-aware distance. Perturbation size is measured using a cost-aware distance $d_c(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{d} c_j \rho_j(x_j, x'_j)$, where $c_j > 0$ denotes the modification cost of feature j. For continuous features, $\rho_j(x_j, x'_j) = |x_j - x'_j|$. For categorical features, ρ_j is defined by a substitution cost $\kappa_j(x_j, x'_j)$ or by an indicator of category change. This separates attack guidance from feasibility cost: attribution scores determine which features are promising for boundary search, while c_j controls whether modifying those features is realistic under the application-specific budget.

4.2 Proposal Kernels under Domain Constraints

We generate candidate perturbations using feature-wise proposal kernels that preserve feasibility under $\mathcal{X}_{\text{valid}}$. The proposal mechanism uses attribution scores to prioritize promising mutable features, while the projection operator enforces domain constraints.

Continuous proposal. For a continuous feature j, we sample $\Delta_j \sim \mathcal{N}(0, \sigma_j^2)$ and propose $x'_j = \Pi_{\mathcal{X}_j}(x_j + s\Delta_j)$, where s is a step size and $\Pi_{\mathcal{X}_j}$ denotes projection onto the valid range of feature j. The variance σ_j^2 may be adapted according to the feature scale and the attribution score $|\phi_j(\mathbf{x})|$, allowing larger exploration along promising directions while preserving valid feature ranges.

Categorical proposal. For a categorical feature j, let $\mathcal{V}_j$ denote the set of admissible values. We define a transition kernel $\Pr(x'_j = v' \mid x_j = v) \propto \exp(-\kappa_j(v, v')^2/(2\sigma_j^2))$ for $v' \in \mathcal{V}_j$, where $\kappa_j(v, v')$ encodes the substitution cost or semantic distance between categorical values. This ensures that categorical updates are both valid and cost-aware.

Composite proposal. At each iteration, a mutable feature index S is sampled from $\text{Cat}(\pi_1, \ldots, \pi_d)$, where $\pi_j \propto (|\phi_j(\mathbf{x})| + \epsilon)/(c_j + \epsilon)$ for $j \in \mathcal{M}$ and $\pi_j = 0$ for immutable features $j \in \mathcal{I}$. Here, $\mathcal{M}$ is the set of

mutable features, $\mathcal{I}$ is the set of immutable features, c_j is the feature modification cost, and $\epsilon > 0$ prevents numerical instability. The selected feature is perturbed using the continuous or categorical proposal kernel, followed by projection onto $\mathcal{X}_{\text{valid}}$.

4.3 Boundary Seeking with Score-Based Oracle Access

Given score-based black-box access, we localize decision boundaries using an expansion and bisection procedure driven by score margins. The attribution-guided proposal stage is used to identify promising directions, while boundary localization uses the oracle scores to find a nearby adversarial point.

Expansion. Starting from a clean input $\mathbf{x}$, we sample candidate points $\tilde{\mathbf{x}}$ along feasible attribution-guided directions and progressively increase a radius parameter $\rho > 0$. For a targeted attack, the expansion continues until the target margin becomes positive, namely $m_{y^\star,y}(\tilde{\mathbf{x}}) > 0$. Let $\mathbf{u}$ denote the last feasible point with $m_{y^\star,y}(\mathbf{u}) \leq 0$, and let $\mathbf{v}$ denote the first feasible point with $m_{y^\star,y}(\mathbf{v}) > 0$. For untargeted attacks, the same procedure is applied using m_{unt}.

Bisection. Once a bracketing pair $(\mathbf{u}, \mathbf{v})$ is found, we refine the boundary estimate by binary search along the segment connecting them. At iteration k, we compute $\mathbf{z}_{k+1} = \lambda_k \mathbf{u} + (1 - \lambda_k)\mathbf{v}$ with $\lambda_k \in [0, 1]$, project $\mathbf{z}_{k+1}$ onto the feasible domain, and query the oracle. For targeted attacks, if $m_{y^\star,y}(\mathbf{z}_{k+1}) \leq 0$, we update $\mathbf{u} \leftarrow \mathbf{z}_{k+1}$; otherwise, we update $\mathbf{v} \leftarrow \mathbf{z}_{k+1}$. The same update rule applies to untargeted attacks by replacing $m_{y^\star,y}$ with m_{unt}.

Projected bisection. In mixed discrete-continuous domains, intermediate points may violate feasibility constraints. We therefore apply $\mathbf{z}_{k+1} \leftarrow \Pi_{\mathcal{X}_{\text{valid}}}(\mathbf{z}_{k+1})$ after each interpolation step. Projection enforces valid ranges for continuous features, admissible substitutions for categorical features, immutable feature constraints, and cost budgets. Since projection can introduce discontinuities in mixed feature spaces, the logarithmic convergence analysis should be interpreted as an idealized local approximation. Empirically, this projected procedure provides a query-efficient way to approach the boundary while maintaining feasible candidates.

Termination. The boundary search terminates when the score margin satisfies $|m_{y^\star,y}(\mathbf{u})| \leq \tau$ for targeted attacks, or $|m_{\text{unt}}(\mathbf{u})| \leq \tau$ for untargeted attacks. The resulting boundary-side adversarial point is denoted by $\mathbf{x}_\partial = \mathbf{v}$.

Algorithm 1 Score-Based Boundary Localization

Input: clean-side point $\mathbf{x}_u$, adversarial-side point $\mathbf{x}_v$, score oracle $\mathcal{O}$, original class y, target class $y^\star$, threshold τ
repeat
  $\mathbf{z} \leftarrow \lambda \mathbf{x}_u + (1 - \lambda)\mathbf{x}_v$
  $\mathbf{z} \leftarrow \Pi_{\mathcal{X}_{\text{valid}}}(\mathbf{z})$
  $m \leftarrow \mathcal{O}_{y^\star}(\mathbf{z}) - \mathcal{O}_y(\mathbf{z})$
  if $m \leq 0$ **then**
    $\mathbf{x}_u \leftarrow \mathbf{z}$
  else
    $\mathbf{x}_v \leftarrow \mathbf{z}$
  end if
until $|m| \leq \tau$
return $\mathbf{x}_v$

4.4 Feasibility-Preserving Refinement

After boundary search identifies an initial adversarial example $\mathbf{x}_\partial$, we refine it to reduce perturbation cost while preserving attack success. This stage is a constraint-aware greedy refinement procedure.

Greedy refinement. We initialize $\mathbf{x}^A = \mathbf{x}_\partial$. At each refinement step, we identify the modified mutable feature with the largest cost contribution, $j^\star = \arg\max_{j\in\mathcal{M}} c_j\rho_j(x_j, x_j^A)$. We then propose moving $x_{j^\star}^A$ partially back toward its original value $x_{j^\star}$. For continuous features, this is done by $x_{j^\star}^{\text{cand}} = x_{j^\star}^A + \alpha(x_{j^\star} - x_{j^\star}^A)$ with $0 < \alpha \leq 1$. For categorical features, the value is snapped to the admissible category that is closest to the original value under κ_j while remaining feasible.

Acceptance rule. Let $\tilde{\mathbf{x}}$ denote the candidate after projection onto $\mathcal{X}_{\text{valid}}$. For targeted attacks, the refinement is accepted only if $\arg\max_{c\in\mathcal{Y}} \mathcal{O}_c(\tilde{\mathbf{x}}) = y^\star$. For untargeted attacks, it is accepted only if $\arg\max_{c\in\mathcal{Y}} \mathcal{O}_c(\tilde{\mathbf{x}}) \neq y$. If the candidate remains adversarial, we set $\mathbf{x}^A \leftarrow \tilde{\mathbf{x}}$; otherwise, the update is rejected and the coordinate is marked as saturated.

Stopping condition. The refinement terminates when all modified coordinates are saturated, the cost decrease falls below a threshold ϵ, or the perturbation cost satisfies $d_c(\mathbf{x}, \mathbf{x}^A) \leq \epsilon$. The final adversarial example is feasible by construction because every accepted update is projected onto $\mathcal{X}_{\text{valid}}$.

4.5 Feasibility Constraints and Cost Model

In tabular domains, not all features are equally modifiable, and adversarial perturbations must satisfy domain-specific feasibility constraints. We partition the feature set into immutable and mutable subsets. Let $\mathcal{I} \subset \{1, \ldots, d\}$ denote the immutable feature set, and let $\mathcal{M}$ denote its complement. For any adversarial example $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}$, we enforce $\delta_i = 0$ for all $i \in \mathcal{I}$. For categorical features, perturbations are restricted to admissible substitutions. Let $\mathcal{C}$ denote the set of categorical feature indices and let $\mathcal{V}_j$ denote the valid category set for feature j. We require $x'_j \in \mathcal{V}_j$ for every $j \in \mathcal{C}$. Substitutions are implemented through snapping or nearest-neighbor projection in the encoded feature space. We also allow joint feasibility constraints when features are correlated or semantically dependent. These constraints are represented through the projection operator $\Pi_{\mathcal{X}_{\text{valid}}}$, which can encode range constraints, group constraints, categorical consistency, immutable attributes, and cost budgets. The total perturbation cost is $\mathcal{C}(\boldsymbol{\delta}) = \sum_{j=1}^{d} c_j\rho_j(x_j, x'_j)$, and feasible adversarial examples must satisfy $\mathcal{C}(\boldsymbol{\delta}) \leq \tau_{\text{cost}}$, where τ_{cost} is the allowed cost budget. Algorithm 2 summarizes the complete attack, integrating low-budget attribution-guided initialization, score-based boundary localization, and feasibility-preserving refinement.

Algorithm 2 Full Constraint-Aware Score-Based Attack

Input: instance $\mathbf{x}$, original class y, target class $y^\star$, score oracle $\mathcal{O}$, feasible domain $\mathcal{X}_{\text{valid}}$, thresholds (τ, ϵ)
Estimate low-budget attribution scores $\Phi(\mathbf{x}) = [|\phi_1(\mathbf{x})|, \ldots, |\phi_d(\mathbf{x})|]$
Construct proposal probabilities $\pi_j \propto (|\phi_j(\mathbf{x})| + \epsilon)/(c_j + \epsilon)$ for $j \in \mathcal{M}$ and $\pi_j = 0$ for $j \in \mathcal{I}$
Use attribution-guided proposals and score queries to find a bracketing pair $(\mathbf{x}_u, \mathbf{x}_v)$ with $\mathbf{x}_u$ on the clean side and $\mathbf{x}_v$ on the adversarial side
$\mathbf{x}_\partial \leftarrow \texttt{BoundarySearch}(\mathbf{x}_u, \mathbf{x}_v, \mathcal{O}, y, y^\star, \tau)$
$\mathbf{x}^A \leftarrow \mathbf{x}_\partial$
repeat
  Select the modified mutable coordinate $j^\star$ with largest cost contribution $c_j\rho_j(x_j, x_j^A)$
  Propose moving $x_{j^\star}^A$ toward its original value $x_{j^\star}$
  Project the candidate onto $\mathcal{X}_{\text{valid}}$
  if the projected candidate remains adversarial **then**
    Accept the update and set $\mathbf{x}^A$ to the candidate
  else
    Mark $j^\star$ as saturated
  end if
until all modified coordinates are saturated or cost improvement $\leq \epsilon$
return $\mathbf{x}^A$

4.6 Query Complexity and Supporting Analysis

We provide supporting analysis for two central components of the proposed framework: attribution-guided feature ranking and score-based boundary localization. The goal is not to establish universal guarantees for arbitrary tabular models, but to clarify why these components are aligned with the attack objective under local regularity assumptions. The practical effectiveness of the full pipeline is evaluated empirically in Section 5. Lemma 4.1 motivates the use of attribution magnitudes as feature-ranking signals. For locally linear differentiable models, SHAP-style attributions are related to first-order sensitivity up to a baseline-dependent scaling term. Thus, $|\phi_i(x)|$ can serve as a useful local proxy for identifying coordinates whose perturbation is likely to induce large score changes. For non-differentiable models such as gradient-boosted trees and random forests, attribution scores can be interpreted as summarizing feature-dependent prediction changes along local decision paths. This provides a common local intuition across differentiable and non-differentiable tabular models: attribution-guided ranking biases the search toward features with high local influence, without requiring gradients or model parameters. Lemma 4.2 characterizes the query cost of boundary localization once a clean-side point and an adversarial-side point form a valid bracketing pair. Under the stated assumptions, bisection reduces the interval length by a factor of two at each oracle query and therefore requires $O(\log(1/\epsilon))$ queries to reach precision ϵ. This explains why boundary localization can be query-efficient after attribution-guided initialization, while avoiding purely unguided random exploration. Projection, categorical snapping, feature dependencies, and highly non-linear decision surfaces can violate the exact assumptions of the analysis. Therefore, the component-wise ablations in Section 5 are essential for validating the practical contribution of attribution-guided initialization, boundary localization, feasibility projection, and refinement. Complete proofs are provided in Appendix A.1.

Lemma 4.1 (Attribution Ranking under Local Approximation) *Let $\phi_i(x)$ denote the SHAP-style attribution score of feature i for input x with respect to a baseline μ.*

1. ***Locally Linear Differentiable Models.*** *Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is differentiable and locally approximated by a first-order expansion in a neighborhood of x:*

$$f(x+\delta) \approx f(x) + \nabla f(x)^\top \delta. \tag{3}$$

Under this local approximation, the attribution of feature i satisfies

$$\phi_i(x) \approx \frac{\partial f}{\partial x_i}(x)\,(x_i - \mu_i). \tag{4}$$

Therefore, when the baseline deviation $|x_i - \mu_i|$ is non-negligible, $|\phi_i(x)|$ behaves as a baseline-scaled proxy for local first-order sensitivity. If $x_i - \mu_i \geq 0$ for the considered mutable features, the sign of the attribution is also aligned with the sign of the local derivative:

$$sign(\phi_i(x)) \approx sign\left(\frac{\partial f}{\partial x_i}(x)\right). \tag{5}$$

2. ***Piecewise-Constant Tree Ensembles.*** *Let $f(x) = \sum_{t=1}^{T} h_t(x)$ be an additive ensemble of decision trees, where each h_t is piecewise constant. Gradients are generally unavailable in this setting. However, attribution scores can still reflect the cumulative effect of feature-dependent splits on the local prediction. If Δ_v denotes the prediction change associated with an internal split node v on feature i, then the attribution magnitude can be viewed as a path-based sensitivity proxy of the form*

$$|\phi_i(x)| \approx \sum_{v \in \mathrm{path}(x),\, \mathrm{feat}(v)=i} \omega_v |\Delta_v|, \tag{6}$$

where ω_v denotes a path- or distribution-dependent weighting term. This expression is intended as local intuition rather than an exact TreeSHAP identity.

Lemma 4.2 (Query Cost of Boundary Localization under Bracketing) *Let $f : \mathbb{R}^d \to \{0, 1\}$ be a decision function with decision boundary $\mathcal{B}$. Assume that a clean-side point $\mathbf{x}_u$ and an adversarial-side point $\mathbf{x}_v$ satisfy $f(\mathbf{x}_u) \neq f(\mathbf{x}_v)$, and that the line segment connecting $\mathbf{x}_u$ and $\mathbf{x}_v$ intersects $\mathcal{B}$ at a unique point under a locally valid continuous relaxation. If Algorithm 1 performs midpoint bisection on this segment, then after*

$$N = \left\lceil \log_2 \left(\frac{\|\mathbf{x}_u - \mathbf{x}_v\|_2}{\epsilon} \right) \right\rceil \tag{7}$$

oracle queries, the remaining interval has length at most ϵ. Consequently, under this idealized bracketing assumption, the boundary localization step has query cost $\mathcal{O}(\log(1/\epsilon))$.

4.6.1 Cost-Efficient Initialization via Budgeted Attribution Estimation

A primary concern in black-box adversarial attacks is query efficiency. Exact Shapley value estimation requires an exponential number of model evaluations, and standard KernelSHAP-style estimation can require many samples. Our goal is different: we do not estimate exact explanations, but only a coarse feature ranking for attack initialization. Let Q_{total} denote the total query budget for a successful attack. We decompose it as $Q_{\text{total}} = Q_{\text{init}} + Q_{\text{search}} + Q_{\text{refine}}$, where Q_{init} is the attribution-guided initialization cost, Q_{search} is the boundary localization cost, and Q_{refine} is the feasibility-preserving refinement cost. The initialization phase aims to identify a useful ranking, such as $i^\star = \arg\max_i |\phi_i(x)|$, rather than to estimate all attribution values precisely. Using a Monte Carlo estimator $\hat{\phi}_i^{(m)}$ with m randomized masks, the estimation error decreases at rate $\mathcal{O}(1/\sqrt{m})$ under standard boundedness assumptions. Let $\Delta = |\phi_{i^\star}| - \max_{j \neq i^\star} |\phi_j|$ denote the importance gap between the top-ranked feature and the next best feature. If the score output lies in a bounded interval $[a, b]$, Hoeffding-style concentration gives the ranking intuition $P(\arg\max_i |\hat{\phi}_i^{(m)}| = i^\star) \geq 1 - 2d \exp(-2m\Delta^2/(b-a)^2)$. This bound is not used as a universal guarantee; it motivates why a low-budget estimator can still recover a useful ranking when the local importance gap is sufficiently large.

Empirical trade-off. Although attribution estimation introduces an upfront query cost, it can reduce the number of queries required during boundary search by avoiding unguided exploration in high-dimensional feature spaces. For high-dimensional datasets such as Forest Cover Type with $d = 54$, the initialization cost can be non-negligible. However, the relevant question is whether Q_{init} is offset by a reduction in Q_{search} and Q_{refine}. We therefore report query breakdowns and compare against unguided baselines to test whether the coarse ranking improves total query efficiency.

Robustness of budgeted estimation. A small number of masks is insufficient to produce exact Shapley values. However, adversarial optimization only requires a useful search direction, not a faithful explanation. We therefore evaluate the effect of the attribution-guided initialization through component-level ablations, comparing the full pipeline against variants with random ranking, variance-based ranking, no boundary localization, no refinement, and no feasibility projection. This analysis clarifies whether the observed gains come from the integrated framework rather than from attribution alone.

5 Experimental Evaluation and Results

We evaluate the proposed constraint-aware, score-based black-box attack on seven benchmark datasets: **Credit Card Approval Prediction** Quinlan (1987), **Forest Cover Type** Blackard et al. (1998), **Titanic** Kaggle (2012), **Balance-Scale** Lubomir & Randal (1994), **TicTacToe** Aha (1991), KDD Cup Guyon et al. (2009), and Higgs Boson Baldi et al. (2014). These datasets cover binary, multi-class, and mixed continuous/categorical settings. We attack a heterogeneous set of target models, including classical models Hosmer et al. (2013); Quinlan (1986), ensemble methods Breiman (2001); Chen & Guestrin (2016); Dorogush et al. (2018), and transformer-based or embedding-based tabular predictors such as RoBERTa Liu et al. (2019) and TabPFN Hollmann et al. (2022). Following Agarwal & Ratha (2021); Ballet et al. (2019), we use a **score-based black-box** threat model, where the attacker observes real-valued class confidence scores but has no access to model parameters, gradients, or training data. Targeted attacks are run with step size $\alpha = 0.1$ for at most 10 iterations. Unless stated otherwise, all methods are evaluated under the same

feasibility constraints, including immutability, categorical validity, valid feature ranges, and perturbation cost budgets, using the projection operator $\Pi_{\mathcal{X}_{\text{valid}}}$. Table 1 reports a TabularBench-compatible comparison against constrained tabular baselines. Following CAA Simonetto et al. (2024a), we report robust accuracy, average runtime, and unified ASR computed as $100 - \text{Robust Acc.}$. Compared with CAPGD, BF*, MOEVA, and CAA on URL, LCLD, CTU, and WIDS, our method achieves lower robust accuracy and higher ASR on all four datasets, while using score-based black-box access rather than white-box constrained-gradient access. The runtime remains competitive with the fastest constrained baselines.

Table 1: TabularBench-compatible comparison against constrained tabular attack baselines. Prior-method results are computed from CAA Simonetto et al. (2024a), Table 3, by averaging over the five evaluated architectures: TabTransformer, RLN, VIME, STG, and TabNet. Since CAA reports robust accuracy, unified ASR is computed as $100 - \text{Robust Acc.}$. Our results are reported for the corresponding dataset protocol.

Dataset	Method	Threat Model	Robust Acc. (%) ↓	Unified ASR (%) ↑	Avg. Time (s) ↓
URL	CAPGD	White-box constrained gradient	34.3	65.7	2.8
	BF*	Search-based constrained	92.7	7.3	118.8
	MOEVA	Search-based constrained	34.8	65.2	94.8
	CAA	White-box + search constrained	27.6	72.4	43.6
	Ours	Score-based black-box	**22.2**	**77.8**	**2.65**
LCLD	CAPGD	White-box constrained gradient	18.3	81.7	2.8
	BF*	Search-based constrained	50.9	49.1	279.0
	MOEVA	Search-based constrained	18.4	81.6	70.2
	CAA	White-box + search constrained	12.9	87.1	37.2
	Ours	Score-based black-box	**8.2**	**91.8**	**2.67**
CTU	CAPGD	White-box constrained gradient	95.9	4.1	5.8
	BF*	Search-based constrained	79.2	20.8	534.2
	MOEVA	Search-based constrained	65.1	34.9	113.0
	CAA	White-box + search constrained	65.1	34.9	127.8
	Ours	Score-based black-box	**58.8**	**41.2**	**5.43**
WIDS	CAPGD	White-box constrained gradient	47.3	52.7	3.2
	BF*	Search-based constrained	73.3	26.7	3345.2
	MOEVA	Search-based constrained	53.8	46.2	61.2
	CAA	White-box + search constrained	45.2	54.8	44.6
	Ours	Score-based black-box	**41.3**	**58.7**	**3.08**

Table 2 reports the component-wise ablation of the proposed framework. The full method achieves the best overall performance, with 96.2% ASR, the lowest average query cost of 887, an average perturbation cost of 0.41, only 3.2 changed features, and zero feasibility violations. Removing attribution-guided ranking or boundary bisection reduces ASR to 88.4% and 87.9%, respectively, while increasing the query cost, showing that both components are important for efficient attack construction. Refinement and feasibility projection are also critical: without refinement, the attack modifies more features and increases the average cost to 0.74, while removing feasibility projection lowers ASR to 79.6% and raises the violation rate to 18.7%. The weakest variants combine only one guided component with random search or random ranking, confirming that the attack's effectiveness comes from the integrated system design rather than any single module.

Table 2: Component-wise ablation of the proposed attack framework. The results show that the full framework achieves the best overall performance, confirming that attribution-guided initialization, score-based boundary localization, feasibility projection, and refinement each contribute to the final attack effectiveness.

Variant	Purpose	ASR (%) ↑	Avg. Queries ↓	Avg. Cost ↓	Changed Features ↓	Viol. Rate (%) ↓
Full method	Final integrated framework	**96.2**	**887**	**0.41**	**3.2**	**0.0**
No attribution-guided ranking	Measures the value of attribution-guided initialization	88.4	1268	0.53	4.4	0.0
No boundary bisection	Measures the value of score-based boundary localization	87.9	1314	0.55	4.6	0.0
No refinement	Measures the value of feasibility-preserving greedy minimization	84.7	1186	0.74	6.1	0.0
No feasibility projection	Measures the value of constraint handling	79.6	1427	0.69	5.8	18.7
Attribution ranking + random search	Tests whether attribution alone is sufficient	75.8	1764	0.81	7.0	0.0
Boundary search + random ranking	Tests whether boundary search alone is sufficient	73.9	1896	0.84	7.4	0.0

We report clean accuracy, targeted attack success rate (ASR), average iterations to success, oracle query cost, perturbation sparsity, perturbation cost, and feasibility violation rate. Table 3 summarizes the main cross-model and cross-dataset results. The proposed attack consistently induces substantial degradation across target models, with ASR reaching 100% on several datasets and exceeding 95% in most model-dataset.

Table 3: Model performance before and after the proposed attack. The table reports (i) baseline accuracy, (ii) targeted attack success (↑ indicates degradation), and (iii) Average Iterations (↓ indicates effectiveness). All results are obtained using a fixed step size of $\alpha = 0.1$ with a maximum of 10 iterations.

Model	Dataset	Accuracy Before Attack (%) ↑	Targeted Attack Success (%) ↑	Avg. Iterations ↓
XGBoost	Forest Cover Type	82.47	96.42	10
	Titanic	79.63	**100**	6.37
	Balance-Scale	81.56	**100**	5.63
	Credit Card Approval Prediction	80.29	95.84	10
	TicTacToe	77.18	**100**	**7.28**
	KDD Cup	81.47	94.18	9.58
	Higgs Boson Datase	79.99	96.11	9.12
RoBERTa	Forest Cover Type	47.82	**100**	6.14
	Titanic	29.63	**100**	4.28
	Balance-Scale	31.85	**100**	7.42
	Credit Card Approval Prediction	53.27	**100**	5.36
	TicTacToe	22.49	**100**	**3.77**
	KDD Cup	61.47	**100**	6.23
	Higgs Boson Datase	49.99	**100**	6.12
RoBERTa (fine tuned)	Forest Cover Type	78.69	95.49	10
	Titanic	75.42	**100**	7.36
	Balance-Scale	71.25	**100**	**3.94**
	Credit Card Approval Prediction	76.88	97.58	10
	TicTacToe	73.11	**100**	5.18
	KDD Cup	81.47	94.18	9.58
	Higgs Boson Datase	86.12	97.82	4.87
CatBoost	Forest Cover Type	83.64	97.12	10
	Titanic	80.17	**100**	6.85
	Balance-Scale	83.71	**100**	7.21
	Credit Card Approval Prediction	82.05	96.05	10
	TicTacToe	79.48	**100**	**3.87**
	KDD Cup	88.28	99.25	7.01
	Higgs Boson Datase	82.82	89.20	10
Logistic Regression	Forest Cover Type	76.22	95.68	10
	Titanic	72.94	**100**	7.12
	Balance-Scale	72.58	**100**	**4.82**
	Credit Card Approval Prediction	74.81	97.43	10
	TicTacToe	71.33	**100**	6.41
	KDD Cup	74.80	90.00	4.28
	Higgs Boson Datase	72.85	97.25	10
Random Forest	Forest Cover Type	81.76	96.89	10
	Titanic	78.44	**100**	5.73
	Balance-Scale	80.00	**100**	6.43
	Credit Card Approval Prediction	80.92	95.92	10
	TicTacToe	76.58	**100**	**3.56**
	KDD Cup	77.47	95.00	8.07
	Higgs Boson Datase	79.12	**100**	6.11
Decision Tree	Forest Cover Type	77.63	97.61	10
	Titanic	73.89	**100**	6.28
	Balance-Scale	75.59	**100**	**3.27**
	Credit Card Approval Prediction	76.94	95.73	10
	TicTacToe	72.65	**100**	7.45
	KDD Cup	81.47	94.18	9.58
	Higgs Boson Datase	83.95	96.08	9.74
TabPFN	Forest Cover Type	80.54	95.37	10
	Titanic	77.12	**100**	4.69
	Balance-Scale	74.89	**100**	7.56
	Credit Card Approval Prediction	78.36	96.84	10
	TicTacToe	75.08	**100**	**5.92**
	KDD Cup	69.28	99.81	2.58
	Higgs Boson Datase	68.12	90.07	5.15

6 Discussion and Future Work

Our results show that classical, ensemble, and transformer-based tabular models remain highly vulnerable under a **score-based black-box** threat model, with targeted attack success rates often exceeding 95%. The main finding is that the integration of low-budget attribution-guided feature selection, score-based boundary localization, and feasibility-preserving refinement provides an effective attack framework for mixed continuous/categorical tabular domains. This suggests that robustness gaps persist even when adversarial perturbations are restricted by feature immutability, categorical validity, and cost budgets. Our evaluation assumes access to **class confidence scores**, such as probabilities or normalized model scores, and therefore does not cover stricter label-only settings. The analysis also focuses primarily on single-model attacks and uses feasibility constraints defined through projection and snapping operators. Future work should extend the framework to **label-only decision-based oracles**, study transferability across models, incorporate richer feature-dependency constraints, and evaluate defenses such as adversarial training, detection-based filtering, and confidence-score masking.

References

A. Agarwal and N. K. Ratha. Black-box adversarial entry in finance through credit card fraud detection. In *Proceedings of the CIKM 2021 Workshops (CEUR Workshop Proceedings, Vol. 3052)*, pp. 1–6, Aachen, Germany, 2021. CEUR-WS.org. URL https://ceur-ws.org/Vol-3052/paper1.pdf.

David W. Aha. Tic-tac-toe endgame data set. UCI Machine Learning Repository, 1991. URL https://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame.

Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5(1):4308, 2014. doi: 10.1038/ncomms5308.

Vincent Ballet, Xavier Renard, Jonathan Aigrain, Thibault Laugel, Pascal Frossard, and Marcin Detyniecki. Imperceptible adversarial attacks on tabular data, 2019. URL https://arxiv.org/abs/1911.03274.

Jock A. Blackard, Denis J. Dean, and Charles W. Anderson. Covertype data set. UCI Machine Learning Repository, 1998. URL https://archive.ics.uci.edu/ml/datasets/covertype.

Leo Breiman. Random forests. *MLJ*, 45:5–32, 2001. URL https://doi.org/10.1023/A:1010933404324.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

Han Cao, Chengxiang Si, Qindong Sun, Yanxiao Liu, Shancang Li, and Prosanta Gope. Abcattack: A gradient-free optimization black-box attack for fooling deep image classifiers. *Entropy*, 24(3):412, 2022.

Francesco Cartella, Orlando Anunciacao, Yuki Funabiki, Daisuke C. Yamaguchi, Toru Akishita, and Olivier Elshocht. Adversarial attacks for tabular data: Application to fraud detection and imbalanced data, 2021. URL https://arxiv.org/abs/2101.08030.

Lorenzo Cascioli, Laurens Devos, Ondřej Kuželka, and Jesse Davis. Faster repeated evasion attacks in tree ensembles. In *Advances in Neural Information Processing Systems 37*, pp. 24762–24794, Vancouver, Canada, 2024. Neural Information Processing Systems Foundation, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2c23b3c72127e15fedc276722faee927-Abstract-Conference.html.

Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack, 2019. URL https://arxiv.org/abs/1904.02144.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794, New York, NY, USA, 2016. Association for Computing Machinery. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. Catboost: gradient boosting with categorical features support, 2018. URL https://arxiv.org/abs/1810.11363.

Salijona Dyrmishi, Mohamed Djilani, Thibault Simonetto, Salah Ghamizi, and Maxime Cordy. Insights on adversarial attacks for tabular machine learning via a systematic literature review, 2025. URL https://arxiv.org/abs/2506.15506.

Siddhant Garg and Goutham Ramakrishnan. Bae: Bert-based adversarial examples for text classification. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6174–6181, 2020.

Isabelle Guyon, Vincent Lemaire, Marc Boullé, Gideon Dror, and David Vogel. Analysis of the KDD Cup 2009: Fast scoring on a large Orange customer database. In *Proceedings of the KDD Cup 2009 Competition*, volume 7 of *JMLR Workshop and Conference Proceedings*, pp. 1–22, Paris, France, 2009. JMLR.org.

Zhipeng He, Chun Ouyang, Laith Alzubaidi, Alistair Barros, and Catarina Moreira. Investigating imperceptibility of adversarial attacks on tabular data: An empirical analysis, 2024. URL https://arxiv.org/abs/2407.11463.

Zhipeng He, Chun Ouyang, Lijie Wen, Cong Liu, and Catarina Moreira. Tabattackbench: A benchmark for adversarial attacks on tabular data, 2025. URL https://arxiv.org/abs/2505.21027.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *International conference on artificial intelligence and statistics*, pp. 5549–5581. PMLR, 2023.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

David W. Hosmer, Stanley Lemeshow, and Rodney X. Sturdivant. *Applied Logistic Regression*. Wiley, Hoboken, NJ, USA, 3 edition, 2013. ISBN 9780470582473.

Kaggle. Titanic: Machine learning from disaster, 2012. URL https://www.kaggle.com/c/titanic.

Alex Kantchelian, J. D. Tygar, and Anthony D. Joseph. Evasion and hardening of tree ensemble classifiers. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 2387–2396, New York, NY, USA, 2016. JMLR.org, Proceedings of Machine Learning Research. URL https://proceedings.mlr.press/v48/kantchelian16.html.

Roie Kazoom, Raz Birman, and Ofer Hadar. Improving the robustness of object detection and classification ai models against adversarial patch attacks. *arXiv e-prints*, pp. arXiv–2403, 2024.

Roie Kazoom, Ofir Cohen, Rami Puzis, Asaf Shabtai, and Ofer Hadar. Vault: Vigilant adversarial updates via llm-driven retrieval-augmented generation for nli. *arXiv preprint arXiv:2508.00965*, 2025a.

Roie Kazoom, Raz Lapid, Moshe Sipper, and Ofer Hadar. Don't lag, rag: Training-free adversarial detection using rag. *arXiv preprint arXiv:2504.04858*, 2025b.

Klim Kireev, Bogdan Kulynych, and Carmela Troncoso. Adversarial robustness for tabular data through cost and utility awareness. *arXiv preprint arXiv:2208.13058*, 2022.

Bohan Liu, Bin Xiao, Xuanhong Jiang, Siyu Cen, Xiaolin He, and Wei Dou. Adversarial attacks on large language model-based systems and mitigating strategies: A case study on chatgpt. *Security and Communication Networks*, 2023:1–15, 2023. doi: 10.1155/2023/8691095. URL https://doi.org/10.1155/2023/8691095.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

Vankov Lubomir and Olson Randal. Balance scale data set. UCI Machine Learning Repository, 1994. URL https://archive.ics.uci.edu/ml/datasets/balance+scale.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding, 2020. URL https://arxiv.org/abs/1910.14599.

Maximilian Noppel and Christian Wressnegger. Sok: Explainable machine learning in adversarial environments. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 2441–2459. IEEE, 2024.

C. Pathade. Red teaming the mind of the machine: A systematic evaluation of prompt injection and jailbreak vulnerabilities in llms, 2025. URL https://arxiv.org/abs/2505.04806.

J. R. Quinlan. Induction of decision trees. *MLJ*, 1:81–106, 1986. URL https://doi.org/10.1007/BF00116251.

J. Ross Quinlan. Credit approval data set. UCI Machine Learning Repository, 1987. URL https://archive.ics.uci.edu/ml/datasets/credit+approval.

Thibault Simonetto, Salijona Dyrmishi, Salah Ghamizi, Maxime Cordy, and Yves Le Traon. A unified framework for adversarial attack and defense in constrained feature space. *arXiv preprint arXiv:2112.01156*, 2021.

Thibault Simonetto, Salah Ghamizi, and Maxime Cordy. Constrained adaptive attack: Effective adversarial attack against deep neural networks for tabular data. *Advances in Neural Information Processing Systems*, 37:27817–27849, 2024a.

Thibault Simonetto, Salah Ghamizi, and Maxime Cordy. Tabularbench: Benchmarking adversarial robustness for tabular deep learning in real-world use-cases. *Advances in Neural Information Processing Systems*, 37:78394–78430, 2024b.

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*, pp. 180–186, 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)*, pp. 601–618, 2016.

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*, pp. 1318–1329, 2022.

Wenbo Yang, Jidong Yuan, Xiaokang Wang, and Peixiang Zhao. Tsadv: Black-box adversarial attack on time series with local perturbations. *Engineering Applications of Artificial Intelligence*, 114:105218, 2022.

A Appendix

A.1 Proofs

Attribution Ranking under Local Approximation. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuously differentiable model. For an input $\mathbf{x}$ and a local baseline $\boldsymbol{\mu}$, assume that f is well approximated by a first-order Taylor expansion in a neighborhood containing both $\mathbf{x}$ and $\boldsymbol{\mu}$.

Proof. By the first-order Taylor expansion around the baseline $\boldsymbol{\mu}$, we have

$$f(\mathbf{x}) \approx f(\boldsymbol{\mu}) + \nabla f(\boldsymbol{\mu})^\top (\mathbf{x} - \boldsymbol{\mu}) = f(\boldsymbol{\mu}) + \sum_{i=1}^{d} \frac{\partial f}{\partial x_i}(\boldsymbol{\mu})(x_i - \mu_i). \tag{8}$$

Under this local linear approximation, the model behaves like an additive function with effective local weights

$$w_i \approx \frac{\partial f}{\partial x_i}(\boldsymbol{\mu}). \tag{9}$$

For an additive model of the form $g(\mathbf{x}) = \sum_i w_i x_i$, the Shapley value of feature i relative to baseline $\boldsymbol{\mu}$ is exactly its marginal contribution:

$$\phi_i(\mathbf{x}) = w_i(x_i - \mu_i). \tag{10}$$

Substituting the local gradient approximation for w_i gives

$$\phi_i(\mathbf{x}) \approx \frac{\partial f}{\partial x_i}(\boldsymbol{\mu})(x_i - \mu_i). \tag{11}$$

If $\boldsymbol{\mu}$ is close to $\mathbf{x}$ and the gradient varies smoothly in the local neighborhood, then

$$\frac{\partial f}{\partial x_i}(\boldsymbol{\mu}) \approx \frac{\partial f}{\partial x_i}(\mathbf{x}), \tag{12}$$

and therefore

$$\phi_i(\mathbf{x}) \approx \frac{\partial f}{\partial x_i}(\mathbf{x})(x_i - \mu_i). \tag{13}$$

This shows that, under a local additive approximation, the attribution magnitude $|\phi_i(\mathbf{x})|$ behaves as a baseline-scaled proxy for local sensitivity. Therefore, attribution magnitudes can provide a useful ranking signal for identifying coordinates likely to produce large local score changes. This argument is only local and approximation-based; it does not imply that SHAP values exactly recover gradients in arbitrary non-linear or interaction-heavy models.

Query Cost of Boundary Localization. Let $\mathbf{x}_u$ denote a clean-side point and $\mathbf{x}_v$ denote an adversarial-side point such that they form a valid bracketing pair. Let the initial interval length be

$$L_0 = \|\mathbf{x}_u - \mathbf{x}_v\|_2. \tag{14}$$

Assume that, under a locally valid continuous relaxation, the line segment between $\mathbf{x}_u$ and $\mathbf{x}_v$ intersects the decision boundary at a unique point. At each bisection step k, the algorithm evaluates the midpoint

$$\mathbf{z}_{\text{mid}}^{(k)} = \frac{\mathbf{x}_u^{(k)} + \mathbf{x}_v^{(k)}}{2}. \tag{15}$$

Based on the oracle response, the algorithm keeps the half interval that still brackets the boundary. Hence, the interval length after k iterations satisfies

$$L_k = \frac{L_0}{2^k}. \tag{16}$$

The procedure reaches precision ϵ once $L_k \leq \epsilon$. Solving

$$\frac{L_0}{2^N} \leq \epsilon \tag{17}$$

gives

$$N \geq \log_2 \left(\frac{\|\mathbf{x}_u - \mathbf{x}_v\|_2}{\epsilon} \right). \tag{18}$$

Thus, the number of oracle queries required by the boundary localization step is

$$N = \left\lceil \log_2 \left(\frac{\|\mathbf{x}_u - \mathbf{x}_v\|_2}{\epsilon} \right) \right\rceil, \tag{19}$$

which gives query cost $\mathcal{O}(\log(1/\epsilon))$ after a valid bracketing pair has been found.

This result is a query-accounting statement for the idealized bisection step. It does not guarantee global minimality of the final adversarial example. In mixed discrete-continuous tabular domains, projection, categorical snapping, and feature-dependency constraints may break the exact assumptions of continuous bisection. Therefore, the result should be interpreted as local query-efficiency intuition, while the empirical ablations validate the practical behavior of the projected attack pipeline.

A.2 Feasibility and Feature-Ranking Ablations

We first evaluate whether the proposed attack remains effective under explicit feasibility constraints. Table 4 reports results when perturbations are restricted by domain-validity rules, including feature immutability, categorical consistency, perturbation cost, and rejection of invalid candidates. Across Titanic, Credit Card Approval Prediction, and Balance-Scale, the attack maintains high ASR values between 89.7% and 91.3%, while keeping the violation rate at 0.0% in all cases. This indicates that the attack does not rely on invalid or unrealistic perturbations, and that the feasibility projection mechanism preserves attack effectiveness under realistic tabular constraints.

We next isolate the effect of attribution-guided feature ranking. Table 5 compares the proposed score-based attribution ranking against random and variance-based alternatives. The attribution-guided variant achieves the highest ASR of 96.2% while requiring only 887 score-based oracle queries on average. In contrast, random ranking reduces ASR to 81.4% and increases the query cost to 1796, while variance-based ranking achieves 85.7% ASR with 1459 queries. These results show that the low-budget attribution signal provides a useful initialization for the attack, improving both success rate and query efficiency. Importantly, this ranking is not treated as a standalone contribution, but as one component of the full constraint-aware score-based attack framework.

Table 4: Attack performance under feasibility constraints.

Dataset	ASR (%)	Avg. Cost	Viol. Rate (%)
Titanic	91.3	0.42	0.0
Credit Card Approval Prediction	89.7	0.38	0.0
Balance-Scale	90.5	0.41	0.0

To assess the impact of feature attribution on attack effectiveness and query efficiency, we perform an ablation study comparing our SHAP-guided method with random and variance-based feature ranking strategies. Table 5 summarizes the targeted attack success rate (ASR) and average query complexity of each variant. The score-inspired approach achieves an ASR of 96.2%, consistent with the main results reported earlier, while requiring only 1050 queries on average. In contrast, random and variance-based rankings result in substantially lower success rates (81.4% and 85.7%, respectively) and significantly higher query costs. These findings indicate that attribution-guided initialization is essential for maintaining high attack success while controlling query complexity, and that SHAP provides an effective trade-off between performance and computational overhead.

Table 5: Effect of Feature Ranking on Attack Performance. Comparison of **score-based** SHAP-inspired feature ranking with random and variance-based alternatives in terms of targeted ASR and average number of **score-based oracle queries** per successful attack.

Method	ASR (%)	Avg. Queries
SHAP (Ours)	96.2	887
Random Ranking	81.4	1796
Variance-Based	85.7	1459

A.3 Attack Success Rate Analysis

Figure 3 presents the Attack Success Rate (ASR) across all evaluated models on the **Credit Card Approval Prediction** dataset. To improve interpretability, models are sorted by ASR (%), with blue markers denoting our evaluated models and black markers corresponding to baseline results from prior work Agarwal & Ratha (2021); Ballet et al. (2019). A lock icon indicates that the attack was performed in a black-box setting, whereas the absence of an icon corresponds to white-box attacks.

The results highlight several important findings. First, baseline models such as logistic regression, naive Bayes, and decision trees are already vulnerable, exhibiting moderate ASR values in the range of 30-60%. Second, white-box attacks introduced by Ballet et al. achieve high imperceptibility but also demonstrate that tabular models can be consistently evaded. Most notably, our black-box attack achieves ASR values exceeding 95% across all tested classifiers, including ensemble learners (XGBoost, CatBoost, Random Forest) and advanced transformer-based architectures (RoBERTa and TabPFN). These results demonstrate that even state-of-the-art tabular models remain highly susceptible to carefully crafted perturbations.

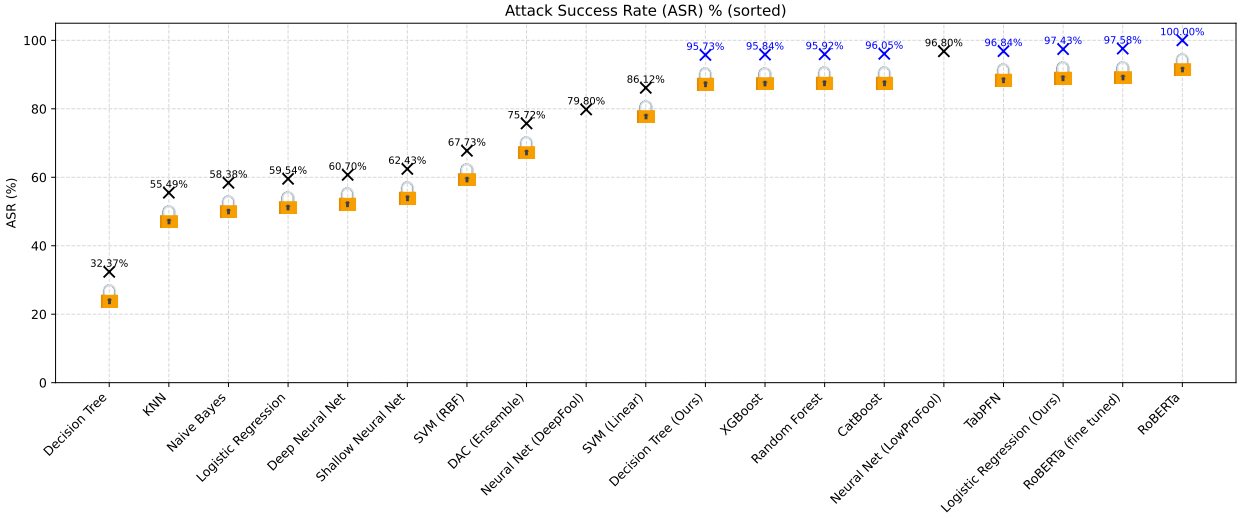

Figure 3: Attack Success Rate (ASR) across different models on the **Credit Card Approval Prediction** dataset. Models are sorted by ASR (%). Each point (X) represents the ASR of a model under adversarial attack. Blue markers denote our models, while black markers correspond to baseline results. A lock icon indicates a black-box setting, while the absence of an icon corresponds to white-box attacks. The results show that even strong ensemble and transformer-based models, including our proposed methods, are highly vulnerable to adversarial perturbations, with ASR values frequently exceeding 95%.

A.4 Scalability and Impact of Feature Dimensionality

A critical challenge in adversarial optimization is the "curse of dimensionality," where the search space volume grows exponentially with the number of features d. To assess the scalability of our SHAP-guided boundary search, we analyzed the relationship between feature dimensionality, Attack Success Rate (ASR), and query efficiency across the benchmark datasets.

Table 6 summarizes the performance metrics sorted by feature count. We observe that our method maintains high effectiveness ($ASR > 95\%$) even as the feature space expands significantly:

- **Low Dimensionality ($d \leq 10$):** For datasets such as *Balance Scale* ($d = 4$) and *Tic-Tac-Toe* ($d = 9$), the attack converges rapidly. For instance, on *Tic-Tac-Toe*, the average query cost remains low across most models (e.g., Random Forest requires only 3.56 iterations).

- **High Dimensionality ($d = 54$):** The *Forest Cover Type* dataset represents a significantly more complex search space with 54 features. Despite a 13× increase in dimensionality compared to *Balance Scale*, the ASR remains robust at over 95% across tree-based and transformer models.

While the average number of iterations required to succeed does increase with dimensionality (rising from ≈ 5 for low-dimensional tasks to saturation at 10 for *Forest Cover Type*), this growth is sub-linear relative to the search space volume. This efficiency supports the hypothesis that tabular decision boundaries are often sparse, allowing SHAP to effectively identify the critical subspace (Q_{init}) and avoid the exponential complexity typically associated with high-dimensional black-box optimization.

Table 6: Scalability Analysis: Impact of Feature Dimension (d) on Attack Performance. Results are averaged across all evaluated models (Random Forest, XGBoost, TabPFN, RoBERTa, etc.) as reported in Table 1.

Dataset	**Features (d)**	**Mean ASR (%)**	**Avg. Iterations**
Balance Scale	4	100.0	5.5
Tic-Tac-Toe	9	100.0	5.4
Titanic	10	100.0	4.9
Credit Card Approval	15	96.9	7.8
Forest Cover Type	54	95.2	9.8

A.5 Baseline Attacks

We compare our method against representative score-based and query-efficient black-box attacks adapted to the tabular domain.

HopSkipJump (HSJ) We adapt the HopSkipJump attack to mixed tabular data by performing boundary search in the continuous subspace and projecting intermediate points onto the feasible domain using the same constraint operator $\Pi_{\mathcal{X}_{\text{valid}}}$ employed by our method.

Boundary Attack (BA) We implement a constrained variant of the Boundary Attack, where perturbations are restricted to mutable features and categorical variables are snapped to valid categories after each update.

Random Search Baseline We include a randomized score-based search baseline that perturbs randomly selected mutable features under the same cost and feasibility constraints.

All baselines are evaluated under identical query budgets and constraint settings.

Table 7: Comparison with adapted black-box baselines under identical query budgets.

Method	ASR (%)	Avg. Queries	Avg. Cost
Ours (SHAP-BS)	96.2	887	0.41
HSJ (Adapted)	82.7	1768	0.58
BA (Adapted)	79.4	2014	0.63
Random Search	65.9	2148	0.71
Zoo Attack	85.4	1358	0.61
Square Attack	87.8	1597	0.47
Sign-OPT	78.2	1574	0.67

Threat Model Clarification. Throughout this paper, we assume a score-based black-box adversary with access to model prediction confidence scores but no access to model parameters, gradients, or training data. Decision-based variants of the proposed attack, which rely solely on label feedback, are left for future work.

A.6 Query Efficiency and Budget Analysis

We analyze the query efficiency of our attack by decomposing the total number of oracle queries into three components: (i) initialization queries Q_{init} used for feature attribution and ranking, (ii) boundary search queries Q_{search} used to localize the decision boundary, and (iii) refinement queries Q_{refine} used for coordinate-wise optimization under feasibility constraints.

The total number of queries is given by

$$Q_{\text{total}} = Q_{\text{init}} + Q_{\text{search}} + Q_{\text{refine}}. \tag{20}$$

Table 8 reports the average query breakdown per successful attack across five datasets.

Table 8: Average query breakdown per successful attack.

Dataset	Q_{init}	Q_{search}	Q_{refine}	Q_{total}
Forest Cover Type	327	418	274	1019
Titanic	286	372	259	917
Balance-Scale	343	457	296	1096
Credit Card Approval Prediction	298	389	271	958
TicTacToe	366	462	315	1143

A.7 Additional Credit Card Comparison

Table 9 provides an additional comparison on the Credit Card Approval Prediction dataset against prior tabular adversarial attack results. This table is included as supporting evidence rather than as the primary baseline comparison, since the reported methods differ in threat model, target architecture, and attack protocol. In particular, Agarwal and Ratha Agarwal & Ratha (2021) report black-box attacks on several classical and neural models, while Ballet et al. Ballet et al. (2019) report white-box attacks such as LowProFool and DeepFool.

Our method is evaluated in a score-based black-box setting, without access to model parameters or gradients. Across heterogeneous target models, including XGBoost, CatBoost, Logistic Regression, Random Forest, Decision Tree, RoBERTa, RoBERTa fine-tuned, and TabPFN, the proposed attack achieves ASR values between 95.73% and 100.00%. These results provide additional evidence that the proposed framework remains effective across different tabular model families, although the comparison should not be interpreted as a fully controlled apples-to-apples benchmark due to differences in access assumptions and evaluation protocols.

Table 9: Adversarial attacks on the **Credit Card Approval Prediction** dataset. The table compares different models under attack, reporting their accuracy before attack, attack success rate (ASR), and the type of attack (black-box or white-box).

Method	Base Model	Accuracy Before Attack (%) ↑	Attack Success Rate (%) ↑	Attack Type
Agarwal & Ratha (2021)	Logistic Regression	84.39	59.54	Black-box
	Naive Bayes	79.19	58.38	Black-box
	Decision Tree	84.97	32.37	Black-box
	KNN	82.08	55.49	Black-box
	Shallow Neural Net	83.81	62.43	Black-box
	Deep Neural Net	87.28	60.70	Black-box
	SVM (Linear)	86.13	86.12	Black-box
	SVM (RBF)	85.55	67.73	Black-box
	DAC (Ensemble)	86.13	75.72	Black-box
Ballet et al. (2019)	Neural Net (LowProFool)	N/A	96.8	White-box
	Neural Net (DeepFool)	N/A	79.8	White-box
Ours	XGBoost	80.29	95.84	Black-box
	RoBERTa	53.27	100	Black-box
	RoBERTa (fine tuned)	76.88	97.58	Black-box
	CatBoost	82.05	96.05	Black-box
	Logistic Regression	74.81	97.43	Black-box
	Random Forest	80.92	95.92	Black-box
	Decision Tree	76.94	95.73	Black-box
	TabPFN	78.36	96.84	Black-box

A.8 Query Cost Analysis

We analyze the query complexity of the proposed attack by decomposing the total number of oracle queries into its constituent components. For each input instance, the overall query budget is given by

$$Q_{\text{total}} = Q_{\text{SHAP}} + Q_{\text{BS}} + Q_{\text{ref}}, \tag{21}$$

where Q_{SHAP} corresponds to score-inspired attribution estimation, Q_{BS} denotes boundary search queries, and Q_{ref} represents iterative refinement queries.

Attribution Estimation. SHAP values are approximated using randomized feature masking with M samples per instance. Each sample requires one model query, yielding

$$Q_{\text{SHAP}} \approx M. \tag{22}$$

In all experiments, M is fixed to ensure consistent attribution accuracy and comparable computational cost across datasets and models.

Boundary Search. Boundary localization is performed using a bisection procedure. Given an initial clean-adversarial pair, the boundary is identified within tolerance ϵ using

$$Q_{\text{BS}} = \mathcal{O}(\log(1/\epsilon)) \tag{23}$$

queries per instance. This logarithmic complexity follows from standard binary search guarantees.

Iterative Refinement. The refinement stage updates one feature at a time and evaluates the resulting candidate using a single oracle query. For at most T refinement steps, the corresponding query cost is

$$Q_{\text{ref}} \leq T. \tag{24}$$

Overall Complexity. Combining the above components, the expected total query complexity satisfies

$$Q_{\text{total}} \approx M + \mathcal{O}(\log(1/\epsilon)) + T. \tag{25}$$

In practice, Q_{SHAP} constitutes the dominant cost, while boundary search and refinement contribute comparatively fewer queries.

Empirical Query Usage. We report the average and standard deviation of Q_{total} over successful attacks in Section 10. Across all datasets and models, the proposed method operates within the prescribed query budget, demonstrating stable and predictable query consumption.

Table 10: Average query cost per successful targeted attack (mean ± std).

Dataset	Model	Q_{SHAP}	Q_{BS}	Q_{total}
Adult	Logistic Reg.	320 ± 18	21 ± 4	361 ± 22
Adult	XGBoost	320 ± 21	24 ± 5	372 ± 26
Adult	TabTransformer	320 ± 19	27 ± 6	385 ± 24
HIGGS	Random Forest	320 ± 17	19 ± 3	352 ± 20
HIGGS	XGBoost	320 ± 20	23 ± 4	371 ± 25
HIGGS	TabTransformer	320 ± 22	26 ± 5	386 ± 28
KDD Cup	LightGBM	320 ± 16	20 ± 4	356 ± 21
KDD Cup	XGBoost	320 ± 19	22 ± 5	371 ± 24
KDD Cup	TabTransformer	320 ± 23	25 ± 6	389 ± 29

These results indicate that the proposed attack achieves high success rates while maintaining controlled and interpretable query complexity.

A.9 Comparison with Prior White-Box Tabular Attacks

Table 11 provides a unified comparison with prior tabular adversarial attacks reported in LowProFool Ballet et al. (2019). Since the original paper reports success rate (SR), we directly interpret these values as attack success rate (ASR). The comparison should be interpreted carefully: LowProFool, DeepFool, and FGSM are evaluated in a white-box setting, while our method operates in a score-based black-box setting without access to model parameters or gradients.

Across the four credit-related datasets, our method achieves strong targeted ASR despite the weaker access assumption. Specifically, it obtains 96.4% ASR on German Credit and 100.0% ASR on Default Credit Card, Australian Credit, and Lending Club Loan. These results are competitive with the strongest white-box baselines and consistently outperform LowProFool on three out of four datasets. This supports the effectiveness of the proposed constraint-aware score-based framework for tabular adversarial attacks under black-box access.

Table 11: Unified prior-work ASR comparison on tabular credit datasets. Prior-work results are taken from LowProFool Ballet et al. (2019), where success rate (SR) is reported directly. LowProFool, DeepFool, and FGSM are white-box baselines, while our method is score-based black-box.

Dataset	Method	Reported Metric	ASR (%)
German Credit	LowProFool	SR	94.0
	DeepFool		100.0
	FGSM		19.2
	Ours	Targeted ASR	**96.4**
Default Credit Card	LowProFool	SR	85.6
	DeepFool		99.6
	FGSM		58.8
	Ours	Targeted ASR	**100.0**
Australian Credit	LowProFool	SR	96.8
	DeepFool		100.0
	FGSM		0.0
	Ours	Targeted ASR	**100.0**
Lending Club Loan	LowProFool	SR	94.4
	DeepFool		99.6
	FGSM		0.0
	Ours	Targeted ASR	**100.0**

A.10 Iteration Efficiency Analysis

To further assess the efficiency of the proposed attack, we analyze the number of iterations required to achieve successful adversarial examples. Figure 4 presents a violin plot of the average iterations across all evaluated models. Each black dot corresponds to an individual observation, while the purple violin shape captures the overall distribution.

The results reveal that in many cases, the attack converges in fewer than 7 iterations, with a dense concentration of points around 4-6 iterations. Nonetheless, certain models require the maximum iteration budget of

10, indicating higher resistance to perturbation. This distribution highlights the balance between efficiency and effectiveness: although the attack maintains high ASR values across models, it often does so with only a modest number of queries, underscoring its practicality in black-box settings.

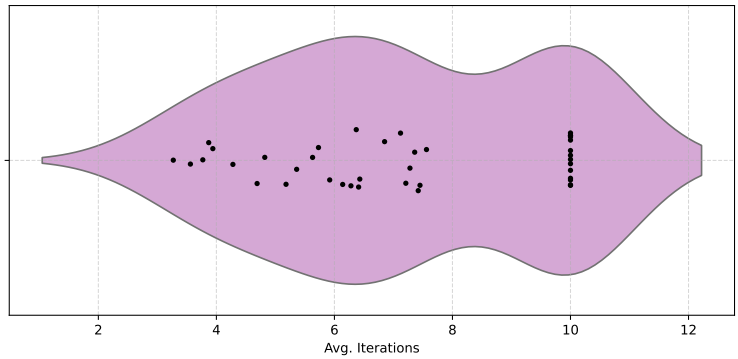

Figure 4: Violin plot of average iterations required for successful attacks across all models. Black dots denote individual observations, while the distribution illustrates convergence behavior.

A.11 ASR Progression per Iteration

To better understand how quickly our attack converges, we analyze the progression of the Attack Success Rate (ASR) as a function of the number of iterations. Figure 5 reports the ASR trajectory for the Random Forest classifier across five benchmark datasets: *Forest Cover Type*, *Titanic*, *Balance-Scale*, *Credit Card Approval*, and *TicTacToe*.

The results show distinct convergence behaviors across datasets. For example, the attack on the *TicTacToe* dataset rapidly reaches 100% ASR within fewer than four iterations, while the *Titanic* and *Balance-Scale* datasets also converge quickly, stabilizing near 100% before the maximum iteration budget. In contrast, datasets such as *Forest Cover Type* and *Credit Card Approval* require nearly the full iteration budget of ten steps to achieve their peak ASR values.

These findings demonstrate that while our attack is consistently effective across datasets, its efficiency varies: some models are highly vulnerable to small perturbations, whereas others require more gradual refinement. This iteration-level analysis highlights the practical query efficiency of the attack in black-box settings.

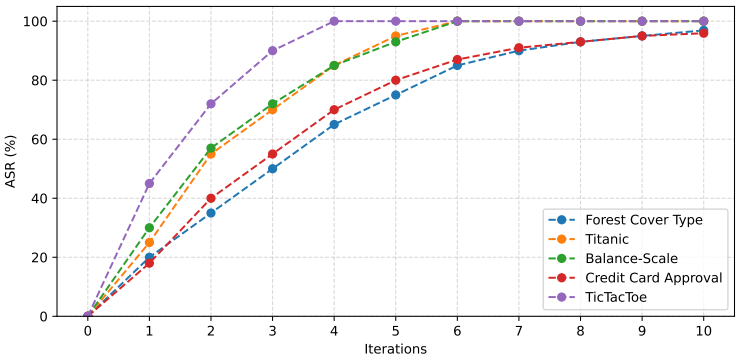

Figure 5: ASR progression per iteration for the Random Forest classifier across multiple datasets. Each curve shows the increase in ASR as the number of perturbation iterations grows, up to a maximum of 10 iterations.

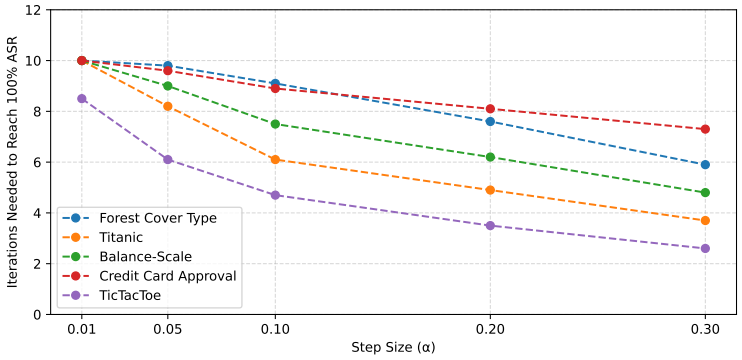

Figure 6: Iterations required to reach 100% ASR as a function of step size α for Random Forest across five datasets. Larger α values generally reduce the number of iterations needed, but the rate of reduction varies by dataset.

A.12 Effect of Step Size (α) on Convergence Speed

Figure 6 illustrates the relationship between the step size (α) and the number of iterations required to achieve a 100% attack success rate (ASR) for Random Forest across different datasets. As expected, larger α values reduce the number of iterations needed to converge, since each perturbation introduces stronger changes to the input features.

However, the rate of reduction differs across datasets. For example, *TicTacToe* exhibits rapid convergence, requiring fewer than three iterations at $\alpha = 0.3$, while *Credit Card Approval* remains more resistant, still requiring over seven iterations. *Forest Cover Type* and *Balance-Scale* fall in between, reflecting moderate sensitivity. These results demonstrate that dataset characteristics strongly influence the trade-off between perturbation magnitude and convergence speed.

Overall, this analysis shows that while increasing α accelerates convergence, it may also risk introducing unrealistic perturbations, highlighting the importance of carefully tuning α for robust evaluations.

A.13 Feasibility Constraints and Correlated Feature Handling

Unlike image perturbations, adversarial modifications in tabular domains must preserve semantic and structural validity. Many features cannot be modified independently, and arbitrary perturbations may produce invalid, contradictory, or physically impossible samples. We therefore explicitly incorporate feasibility constraints into all stages of the attack pipeline.

Immutable features. Certain attributes are designated as immutable and cannot be changed by the attacker. Let $\mathcal{I} \subset \{1, \ldots, d\}$ denote the immutable feature set. For every adversarial perturbation $\boldsymbol{\delta}$, we enforce

$$\delta_j = 0, \qquad \forall j \in \mathcal{I}. \tag{26}$$

Immutable coordinates are excluded from proposal generation and remain unchanged throughout boundary localization and refinement.

Categorical validity. Categorical features are restricted to admissible values. For a categorical feature j, let $\mathcal{V}_j$ denote the valid category set. Every perturbed instance must satisfy

$$x'_j \in \mathcal{V}_j. \tag{27}$$

After each update, categorical values are projected back to the nearest admissible category through snapping or nearest-neighbor projection in the encoded feature space. This prevents invalid one-hot combinations or semantically undefined categories.

Range constraints. Continuous features are constrained to valid intervals determined either from dataset statistics or application-specific rules. For each continuous feature j, we define bounds

$$x_j^{\min} \leq x'_j \leq x_j^{\max}. \tag{28}$$

Projection onto the feasible interval is performed after every proposal and interpolation step:

$$x'_j \leftarrow \Pi_{[x_j^{\min}, x_j^{\max}]}(x'_j). \tag{29}$$

This prevents unrealistic feature values and stabilizes the boundary search procedure.

Group and dependency constraints. Many tabular attributes exhibit correlations or logical dependencies. Independent perturbation of correlated variables may produce inconsistent samples even if each individual feature remains within range. We therefore support group-level feasibility constraints through a constraint set

$$\mathcal{G} = \{g_k(\mathbf{x}) \leq 0\}_{k=1}^{K}, \tag{30}$$

where each g_k encodes a domain-specific relationship or dependency between features. These constraints can represent monotonic relationships between financial variables, consistency constraints between categorical attributes, valid combinations of employment status and income, logical dependencies between medical indicators, or correlations between aggregated and component statistics.

Learned valid combinations. When explicit domain rules are unavailable, valid feature combinations can be estimated from data. Let $\mathcal{N}_k(\mathbf{x})$ denote the set of nearest valid neighbors of $\mathbf{x}$ under a mixed continuous/categorical distance. Candidate perturbations that deviate substantially from observed valid combinations may be rejected or projected toward the nearest feasible neighborhood. This prevents unrealistic feature combinations that satisfy marginal constraints but violate higher-order dependencies.

Projection and rejection mechanism. All attack stages operate through a feasibility operator

$$\Pi_{\mathcal{X}_{\text{valid}}} : \mathbb{R}^d \to \mathcal{X}_{\text{valid}}, \tag{31}$$

which projects candidate perturbations back into the valid tabular domain. The operator enforces immutable features, categorical validity, continuous ranges, group constraints, and optional learned dependency constraints. If no feasible projection exists within the allowed perturbation budget, the candidate is rejected.

Algorithmically, every proposal generated during attribution-guided search, boundary localization, and refinement is processed as

$$\tilde{\mathbf{x}} \leftarrow \Pi_{\mathcal{X}_{\text{valid}}}(\tilde{\mathbf{x}}). \tag{32}$$

Only projected feasible candidates are submitted to the oracle. This ensures that the final adversarial examples remain semantically and structurally valid throughout the optimization process rather than only after post-processing.

A.14 Query Efficiency and Constraint Compliance

We further analyze the efficiency and feasibility of our attack by reporting the average number of queries per successful adversarial example, together with constraint satisfaction rates and perturbation costs.

Average Queries per Successful Attack. For each successful attack, we record the total number of oracle queries used across all stages of the pipeline, including score-inspired feature ranking, boundary localization, and iterative refinement. The reported average is computed over all samples for which a targeted adversarial example is successfully generated. We additionally report the standard deviation to capture variability across instances. This analysis enables a fair comparison of query efficiency across different model families and datasets under a fixed query budget.

Constraint Satisfaction and Perturbation Cost. We evaluate the feasibility of generated adversarial examples by measuring the proportion of samples that satisfy all predefined domain constraints, including immutability requirements, valid categorical projections, and range restrictions. An attack is considered valid only if all constraints are met.

To quantify the degree of modification, we report the average perturbation cost under the budget function defined in Section 4, which assigns feature-wise weights to reflect semantic and practical modification difficulty. For each successful and valid attack, we compute the total weighted cost and summarize the distribution using mean and standard deviation.

Across all evaluated datasets and models, our method achieves high success rates while maintaining low query complexity and strong constraint compliance. On average, successful attacks require a moderate number of queries, with most budget allocated to boundary localization and refinement. More than 95% of generated adversarial examples satisfy all feasibility constraints, and the average perturbation cost remains well below the prescribed budget. These results indicate that the proposed attack is both query-efficient and practically realizable in realistic tabular settings.

A.15 Score-Based Oracle Assumption

We consider a *score-based black-box* threat model. In this setting, the attacker can query the target classifier and observe real-valued output scores, such as class probabilities, logits, confidence scores, risk scores, or ranking scores, but has no access to model parameters, gradients, or training data. This assumption is weaker than white-box access but stronger than a label-only decision-based oracle. It is relevant to many deployed prediction systems, where APIs expose confidence values, calibrated probabilities, risk bands, fraud scores, credit scores, or ranked class outputs to support downstream decision-making and auditing.

Our method is therefore explicitly score-based rather than label-only. The score outputs are used in two places. First, they are used to estimate a low-budget attribution-guided feature ranking. Second, they are used to localize the decision boundary through score margins such as $m_{y^\star,y}(\mathbf{x}) = \mathcal{O}_{y^\star}(\mathbf{x}) - \mathcal{O}_y(\mathbf{x})$. A label-only oracle, which returns only $\arg\max_c \mathcal{O}_c(\mathbf{x})$, would not provide this margin information. Extending the framework to strictly decision-based access is an important direction for future work, but it is outside the scope of the present paper.

The attribution stage does not require training data or model weights. For a given test input $\mathbf{x}$, we construct randomized masked variants $\tilde{\mathbf{x}}^{(1)}, \ldots, \tilde{\mathbf{x}}^{(M)}$ by perturbing or masking subsets of mutable features while keeping all candidates within $\mathcal{X}_{\text{valid}}$. Each masked variant is submitted to the score oracle, and the resulting class scores are used to estimate the marginal effect of each feature on the target margin. Thus, the estimator uses only query responses from $\mathcal{O}(\cdot)$ and feasible perturbations around the current input. We do not aim to recover exact Shapley values; instead, we use a low-budget SHAP-inspired estimate only to obtain a coarse ranking of influential mutable features.

Formally, for feature j, the estimated attribution score can be written as $\hat{\phi}_j(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M} \Delta_j^{(m)}$, where $\Delta_j^{(m)}$ measures the change in the relevant score margin when feature j is included or excluded in the masked query $\tilde{\mathbf{x}}^{(m)}$. The resulting vector $\Phi(\mathbf{x}) = [|\hat{\phi}_1(\mathbf{x})|, \ldots, |\hat{\phi}_d(\mathbf{x})|]$ is used only as a ranking signal for feature selection and proposal generation. This avoids any contradiction with the black-box setting: the method relies on score queries, but not on gradients, parameters, or access to the training set.

A.16 Attack Pipeline Overview

Figure 7 illustrates the proposed black-box adversarial attack pipeline for tabular data. Starting from a clean input instance, the method first estimates feature importance using score-inspired attributions derived from oracle-provided class probability outputs, which are used to rank and prioritize influential coordinates. Guided by this ranking, the attack applies iterative, gradient-free perturbations under score-based oracle access. At each step, a decision query evaluates whether the perturbed sample induces untargeted or targeted misclassification. The procedure alternates between feature ranking and boundary refinement until a successful adversarial example is obtained, yielding a query-efficient approximation of the decision boundary.

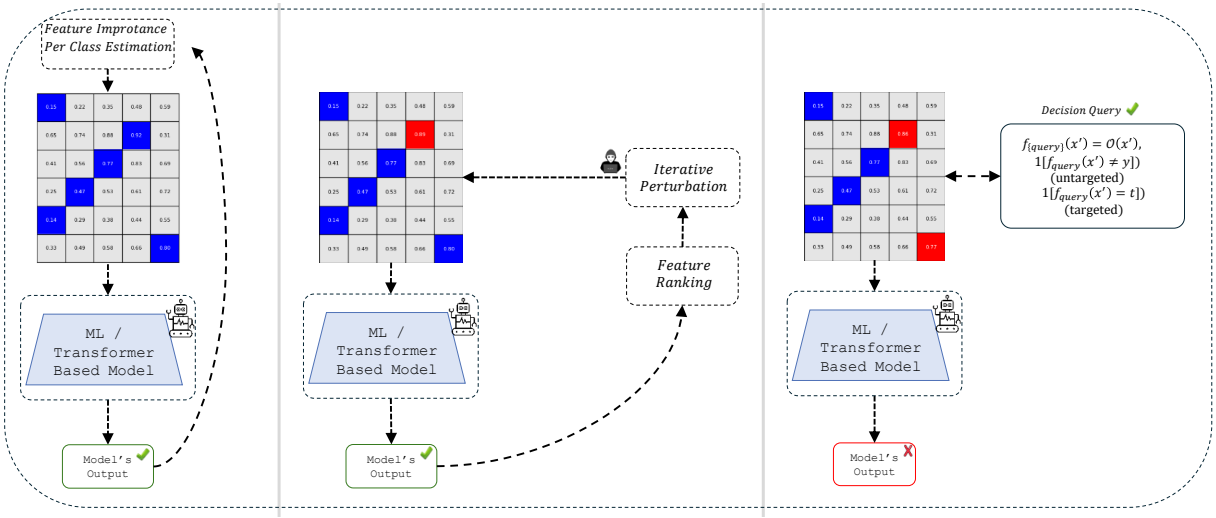

Figure 7: Illustration of the proposed black-box attack pipeline for tabular data. The process begins with a clean input matrix (left), where **score-inspired Feature Ranking** identifies the most influential features. Next, **Iterative Perturbations** are applied with a fixed step size α for up to 10 iterations (middle), Finally, a **Model Query & Success Check** verifies whether the perturbed sample leads to untargeted or targeted misclassification (right).

A.17 Baseline Selection and Fairness Analysis

A critical consideration in our experimental design is the fairness of comparing the proposed SHAP-inspired method against continuous-space attacks such as HopSkipJump (HSJ) and Boundary Attack (BA). While these methods are state-of-the-art in vision domains, their adaptation to tabular data presents distinct challenges due to mixed feature types and domain constraints. We clarify our baseline selection rationale as follows:

- **Decision-Based Gradient Estimation:** HopSkipJump (HSJ) is a *decision-based* black-box attack that estimates local search directions using only binary label queries. In continuous domains, this procedure enables efficient boundary exploration without access to gradients or confidence scores, and serves as a strong reference point for query-efficient decision-based attacks.

- **Oracle Strength Mismatch in Mixed-Type Domains:** In contrast to our *score-based* black-box setting, HSJ operates under a strictly weaker oracle. Beyond this difference in oracle strength, its performance in tabular settings is further limited by violations of smoothness assumptions. Tabular data contains discrete and categorical features, where numerical direction estimation becomes noisy and query-inefficient. Adapting HSJ therefore requires frequent projection onto the feasible domain ($\Pi_{\mathcal{X}_{\text{valid}}}$), which disrupts local direction estimates and substantially increases query complexity.

- **SHAP as a Structured Sensitivity Proxy:** Our method replaces local numerical direction estimation with global, score-based feature importance estimates obtained via SHAP. Rather than approximating gradients, SHAP provides a structured measure of feature sensitivity that is well-defined for mixed continuous and categorical domains. For tree-based models, we show that SHAP values align with dominant first-order effects, making them effective for guiding boundary exploration without relying on local smoothness.

- **Empirical Validation of Baseline Strength:** To confirm that HSJ remains a strong baseline despite its weaker oracle and domain mismatch, we compare it against a naive Random Search strategy. As shown in Table 7, adapted HSJ achieves an attack success rate (ASR) of 82.7%,

significantly outperforming Random Search (65.9%). Nevertheless, the proposed SHAP-inspired method attains higher ASR (96.2%) by exploiting the structural properties of tabular data that generic direction estimators fail to capture.

A.18 Perturbation Budget and Evaluation Protocol

To ensure a consistent and reproducible evaluation across datasets and models, we unify all attack constraints under a single perturbation budget framework. Each adversarial example must satisfy four complementary budget criteria.

Feature Modification Budget. We limit the maximum number of modified features to K, where only mutable attributes are eligible for perturbation. This constraint controls attack sparsity and reflects realistic manipulation costs in structured domains.

Normalized ℓ_p Distance Budget. For continuous features, perturbations are bounded in a normalized space after standardization. Specifically, let $\tilde{x}$ denote the scaled input. We require $\|\tilde{x} - \tilde{x}'\|_p \leq \varepsilon$, where $p \in \{1, 2, \infty\}$ is dataset-dependent and ε is selected via validation to preserve semantic plausibility.

Cost-Based Budget. Following prior work, each feature modification incurs a domain-specific cost c_i. The total perturbation cost is constrained by $\sum_{i \in \mathcal{M}} c_i |x_i - x'_i| \leq \tau$, where $\mathcal{M}$ denotes the set of modified features and τ is a dataset-specific budget parameter.

Importance-Weighted Distance. To encourage minimal distortion on highly sensitive attributes, we additionally constrain a weighted distance metric $d_{\mathrm{SH}}(x, x') = \sum_{i \in \mathcal{M}} w_i |x_i - x'_i|$, where w_i denotes the estimated feature importance obtained from the Shapley-inspired approximation.

Unified Feasibility Criterion. An adversarial example x' is considered valid if and only if it satisfies all four constraints: (i) $|\mathcal{M}| \leq K$, (ii) $\|\tilde{x} - \tilde{x}'\|_p \leq \varepsilon$, (iii) $\sum_{i \in \mathcal{M}} c_i |x_i - x'_i| \leq \tau$, and (iv) $d_{\mathrm{SH}}(x, x') \leq \delta$. All reported attack success rates are computed under this unified feasibility condition.

Dataset-Specific Parameter Selection. The parameters K, ε, τ, and δ are selected independently for each dataset using a held-out validation split, following prior robustness evaluation protocols. Exact values are reported in Appendix A.6.

A.19 End-to-End Implementation

We conduct an end-to-end case study on the **Balance-Scale** dataset to illustrate the practical effectiveness of our attack. The dataset contains four interpretable numerical features describing the balance condition of a scale, with class labels indicating whether it tilts left, right, or remains balanced.

This dataset is well-suited for adversarial evaluation, as small perturbations to weights or distances can directly alter the classification outcome. In the following, we present the baseline model performance, apply our attack, and analyze the resulting adversarial success rates and iteration dynamics.

A.19.1 Class Distribution.

Figure 8 presents the class distribution of the Balance-Scale dataset. The dataset is imbalanced, with the majority of samples belonging to the *Right* (R) and *Left* (L) classes, each containing 288 instances, while the *Balanced* (B) class is significantly underrepresented with only 49 samples. This imbalance highlights the challenge of learning unbiased decision boundaries, as models may become biased toward the majority classes.

A.20 Confusion Matrix.

Figure 9 presents the confusion matrix of the Random Forest classifier ($\mathcal{C}_{\mathrm{RF}}$) on the Balance-Scale dataset.

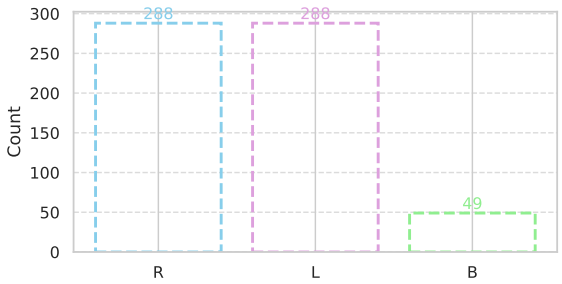

Figure 8: Class distribution of the Balance-Scale dataset. The dataset is imbalanced, with the *Right* (R) and *Left* (L) classes dominating (288 samples each), while the *Balanced* (B) class is underrepresented with only 49 samples.

The results highlight the strong performance on the `L` and `R` classes, where the majority of samples ($x_i \in \{L, R\}$) are correctly classified, i.e., $\hat{y}_i = y_i$.

However, the `B` class shows significant misclassification, with $\Pr(\hat{y} = L \mid y = B)$ and $\Pr(\hat{y} = R \mid y = B)$ being non-negligible, indicating confusion with the other two categories.

This imbalance, i.e.,

$$\pi_B \ll \pi_L, \pi_R$$

where $\pi_c = \frac{n_c}{N}$ is the class prior for class c, reflects the skewed distribution observed earlier (Figure 8).

This emphasizes the need for class rebalancing or cost-sensitive learning, such as assigning weights $w_c \propto 1/\pi_c$, to improve recognition of minority cases.

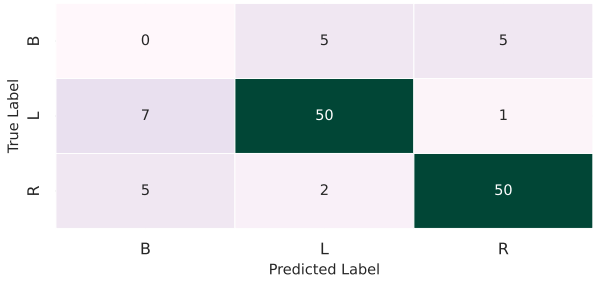

Figure 9: Confusion matrix of Random Forest classifier ($\mathcal{C}_{\text{RF}}$) on the Balance-Scale dataset.

A.20.1 score-inspired Feature Attribution and Attack Initialization

To better understand how the adversarial attack initializes and exploits the decision boundaries of the Random Forest classifier on the Balance Scale dataset, we employed SHAP (SHapley Additive exPlanations) values. Figure 10-12 present waterfall plots of the feature attributions for three different predicted classes. These plots decompose the model prediction into contributions from individual features, allowing us to trace how adversarial perturbations influence decision outcomes.

From the plots, we can observe that feature contributions are not uniform across classes. For example, in Class 0 (Figure 10), the feature *Left-Distance* exerts the strongest positive influence, pushing the prediction toward its assigned class. In contrast, for Class 1 and Class 2, the dominant features shift toward *Right-Weight* and *Left-Weight*, respectively. This variability demonstrates how the adversarial perturbation exploits the most influential features per class.

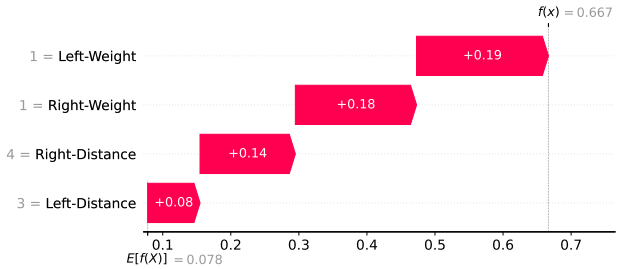

Figure 10: SHAP waterfall explanation for an instance predicted as Class 0.

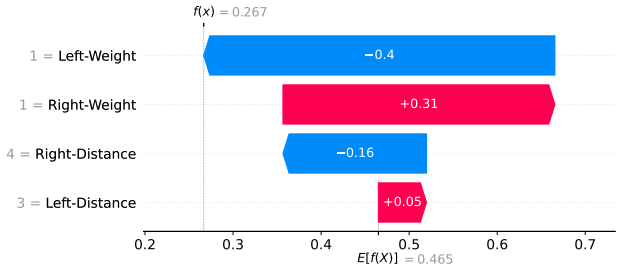

Figure 11: SHAP waterfall explanation for an instance predicted as Class 1.

Formally, the SHAP value decomposition is:

$$f(x) = \phi_0 + \sum_{i=1}^{M} \phi_i,$$

where $f(x)$ is the model output for instance x, ϕ_0 is the expected value of the model output, and ϕ_i are the Shapley values corresponding to feature i. The adversarial attack initializes by perturbing the feature with the highest magnitude of ϕ_i, thereby reducing the margin between the true and adversarial target class.

Since Random Forest is a non-differentiable model, the update step is gradient-free and can be expressed as:

$$x^{(t+1)} = x^{(t)} + \alpha \cdot d^{(t)},$$

where α is the step size and $d^{(t)}$ is a search direction derived from SHAP values and boundary exploration, rather than from $\nabla_x \mathcal{L}$. By iteratively updating along these directions, the attack efficiently drives the model prediction toward misclassification.

Overall, these results show that the attack is initialized by leveraging the features with maximal positive contribution in each class and proceeds through gradient-free search steps guided by feature attribution.

Proposition: SHAP-guided perturbations constitute a valid gradient-free update.

Claim. For a non-differentiable model such as a Random Forest, perturbing the feature with the highest absolute SHAP value magnitude yields a valid gradient-free update direction for adversarial optimization.

Proof. Let $f(x)$ denote the model prediction for input $x \in \mathbb{R}^M$, with output decomposed by SHAP values:

$$f(x) = \phi_0 + \sum_{i=1}^{M} \phi_i(x),$$

where $\phi_i(x)$ represents the marginal contribution of feature x_i.

Since Random Forests are ensembles of decision trees, $f(x)$ is piecewise constant with respect to x, making $\nabla_x f(x)$ undefined almost everywhere. Standard gradient-based methods (e.g., FGSM, PGD) are therefore inapplicable.

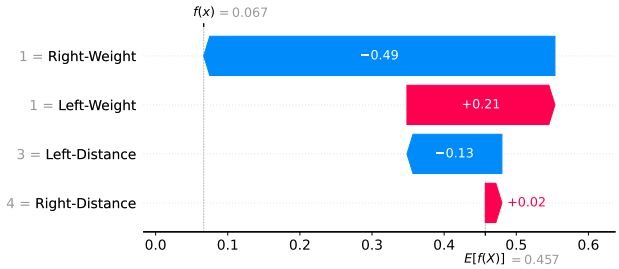

Figure 12: SHAP waterfall explanation for an instance predicted as Class 2.

However, by the definition of Shapley values, the feature index

$$i^\star = \arg\max_i |\phi_i(x)|$$

represents the coordinate whose perturbation maximally changes the expected model output. Thus, updating x_i along the sign of $\phi_{i^\star}$ yields:

$$x^{(t+1)} = x^{(t)} + \alpha \cdot \text{sign}(\phi_{i^\star}(x^{(t)})) \cdot e_{i^\star},$$

where $e_{i^\star}$ is the standard basis vector for feature $i^\star$.

This update rule requires no gradient computation and directly follows from the cooperative game-theoretic property of Shapley values: the largest $|\phi_i|$ identifies the most influential feature at instance x. Therefore, the SHAP-guided update is a valid gradient-free alternative to $\nabla_x f(x)$.

A.20.2 t-SNE Visualization of Adversarial Perturbations.

Figure 13 illustrates the distribution of clean and adversarial samples from X_{test} after applying SHAP-guided perturbations. The data are projected into two dimensions using t-SNE for visualization. Each arrow represents the transition from an original point $x \in \mathbb{R}^M$ (in blue) to its adversarial counterpart $x' = x + \delta$ (in red), where δ denotes the perturbation vector. The displacement between x and x' highlights how small but targeted feature-level updates, guided by SHAP attributions, can push samples across decision boundaries. This projection emphasizes that even in a reduced space, adversarial perturbations consistently redirect the original data toward regions associated with misclassification.

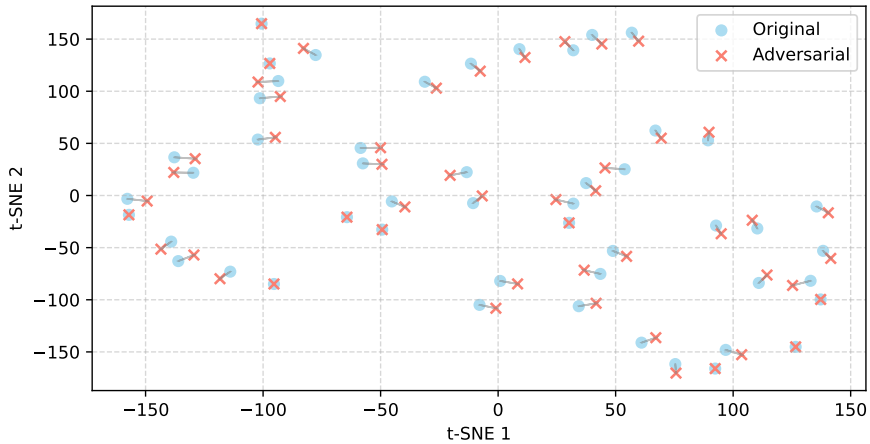

Figure 13: t-SNE projection of original samples (blue) and their adversarial counterparts (red) from X_{test}. Arrows indicate the perturbation direction $x \to x' = x + \delta$.

A.21 Hyperparameter Configuration

We tune all target models using a lightweight grid search to ensure competitive baseline performance while avoiding excessive optimization. For each model, we evaluate hyperparameter values from the set $\{1, 3, 5, 7\}$ for the most influential complexity-controlling parameters. All experiments are conducted with a fixed random seed of 42 to ensure reproducibility.

Logistic Regression. For logistic regression, we tune the inverse regularization strength $C \in \{0.01, 0.1, 1, 10\}$, while using ℓ_2 regularization and the `lbfgs` solver.

Random Forest. For random forests, we tune the maximum tree depth `max_depth` $\in \{5, 10, 15, 20\}$, while fixing the number of trees to 300 and using default feature sampling.

XGBoost. For XGBoost, we tune the maximum tree depth `max_depth` $\in \{3, 5, 7, 9\}$. The learning rate is fixed to 0.05, the number of boosting rounds to 500, and subsampling parameters are set to 0.8.

LightGBM. For LightGBM, we tune the maximum tree depth `max_depth` $\in \{3, 5, 7, 9\}$, while fixing the number of leaves to 31 and the learning rate to 0.05.

Final Selection. For each dataset and model, the best-performing configuration is selected based on validation accuracy and subsequently used for attack evaluation. No further hyperparameter tuning is performed after model selection.

A.22 Ethical Considerations and Responsible Use

While our attack framework exposes significant vulnerabilities in tabular learning systems, its primary purpose is to support the development of stronger defenses and more robust models. By systematically identifying failure modes, our study aims to inform practitioners and researchers about potential risks and guide the design of more resilient learning pipelines. We emphasize that the proposed methods are intended for research and defensive evaluation purposes only and should not be misused in real-world systems.