Towards Verified and Targeted Explanations through Formal Methods

HANCHEN DAVID WANG, Vanderbilt University, USA
DIEGO MANZANAS LOPEZ, Vanderbilt University, USA
PRESTON K. ROBINETTE, Vanderbilt University, USA
IPEK OGUZ, Vanderbilt University, USA
TAYLOR T. JOHNSON, Vanderbilt University, USA
MEIYI MA, Vanderbilt University, USA

As deep neural networks are deployed in safety-critical domains such as autonomous driving and medical diagnosis, stakeholders need explanations of model behavior that are not only interpretable but also trustworthy with formal guarantees. Existing XAI methods fall short of this requirement: heuristic attribution techniques (e.g., LIME, Integrated Gradients) highlight influential features for individual predictions but offer no mathematical guarantees about decision boundaries, while formal explanation methods verify robustness properties yet remain untargeted, analyzing the nearest boundary regardless of whether it represents a critical risk. In safety-critical systems, however, not all misclassifications carry equal consequences; confusing a "Stop" sign for a "60 kph" sign is far more dangerous than confusing it with a "No Passing" sign. Practitioners therefore lack a principled way to answer a fundamental safety question: *how resilient is a model's classification against a specific, high-risk alternative?* We introduce **VITAX** (Verified and Targeted Explanations), a formal XAI framework that addresses this gap by generating targeted semifactual explanations with mathematical guarantees. For a given input (class y) and a user-specified critical alternative (class t), VITAX performs two key steps: **(1)** it identifies the minimal feature subset most sensitive to the $y \to t$ transition using class-specific sensitivity heuristics, and **(2)** it applies formal reachability analysis to guarantee that perturbing these features by ϵ is insufficient to flip the classification to t. This guarantee constitutes a verified semifactual: "even if these critical features change by ϵ, classification y persists against t." We formalize this reasoning through *Targeted ϵ-Robustness*, a formal property that certifies whether an identified feature subset remains robust under perturbation toward a specific target class. By unifying semifactual explanations, class-specific targeting, and formal verification, VITAX is the first method to provide formally guaranteed explanations of a model's resilience against specific, user-identified alternatives. Our evaluations on image classification (MNIST, GTSRB, EMNIST) and regression (TaxiNet) demonstrate that VITAX achieves significantly higher fidelity (e.g., over 30% improvement) and minimal explanation cardinality compared to existing methods. These results establish VITAX as a scalable and trustworthy foundation for verifiable, targeted XAI.

JAIR Track: Integration of Logical Constraints in Deep Learning

JAIR Associate Editor: Eleonora Giunchiglia

JAIR Reference Format:
Hanchen David Wang, Diego Manzanas Lopez, Preston K. Robinette, Ipek Oguz, Taylor T. Johnson, and Meiyi Ma. 2026. Towards Verified and Targeted Explanations through Formal Methods. *Journal of Artificial Intelligence Research* 86, Article 19 (July 2026), 36 pages. DOI: 10.1613/jair.1.20924

Authors' Contact Information: Hanchen David Wang, ORCID: 0000-0001-5990-5865, wanghanchend@gmail.com, Vanderbilt University, Nashville, Tennessee, USA; Diego Manzanas Lopez, ORCID: 0000-0003-0721-1241, dimanzanas@gmail.com, Vanderbilt University, Nashville, Tennessee, USA; Preston K. Robinette, ORCID: 0000-0002-4906-2179, probinette@google.com, Vanderbilt University, Nashville, Tennessee, USA; Ipek Oguz, ORCID: 0000-0002-1403-2420, ipek.oguz@vanderbilt.edu, Vanderbilt University, Nashville, Tennessee, USA; Taylor T. Johnson, ORCID: 0000-0001-8021-9923, taylor.johnson@vanderbilt.edu, Vanderbilt University, Nashville, Tennessee, USA; Meiyi Ma, ORCID: 0000-0001-6916-8774, meiyi.ma@vanderbilt.edu, Vanderbilt University, Nashville, Tennessee, USA.

DOI: 10.1613/jair.1.20924

1 Introduction

The deployment of Deep Neural Networks in safety-critical domains, such as autonomous driving and medical diagnosis, demands more than high accuracy; it requires trustworthiness. When failures occur, the consequences can be severe, necessitating rigorous methods to understand and verify model behavior. Formal methods in eXplainable AI (Formal XAI) have emerged to address this need, moving beyond heuristic estimations (e.g., LIME (Ribeiro et al. 2016a), Integrated Gradients (Sundararajan et al. 2017)) to provide mathematical guarantees about model decisions (Jiang et al. 2023; Marques-Silva and Ignatiev 2022). More broadly, formal specifications such as temporal logic have been integrated into neural networks to enforce properties in sequential prediction (Ma, Gao, et al. 2020), federated settings (An et al. 2024), and predictive monitoring of cyber-physical systems (Ma, Stankovic, et al. 2021).

However, existing Formal XAI approaches face a critical limitation: they often ignore the asymmetric nature of real-world risk. In safety-critical systems, not all misclassifications are equal. For instance, confusing one speed limit sign for another might be minor, but confusing a 'Stop' sign for a '60 kph' sign is catastrophic. Furthermore, formal verification is computationally expensive. It is therefore crucial to allocate these limited resources effectively, prioritizing the decision boundaries that matter most.

Many current formal explanation methods are inherently constrained by the *nearest* decision boundary in the model's feature space, regardless of whether that boundary represents a critical risk. This limitation is illustrated in Figure 1(A). A traffic sign recognition model correctly identifies a 'Stop' sign (Class 14). The nearest boundary is 'No Passing' (Class 10), a low risk. The critical boundary is '60 kph' (Class 5), a high risk. When applying existing formal methods, the analysis often defaults to the nearest boundary. Methods providing instance-level explanations, such as robust semifactuals and counterfactuals for Human-Neural Multi-Agent Systems (HNMAS) (Figure 1(B)) (Leofante and Lomuscio 2023), generate modified inputs but lack fine-grained feature insights and drift towards the nearest boundary (Class 10). Untargeted formal explanation methods, such as VeriX (Figure 1(C)) (M. Wu et al. 2023), aim to verify robustness against *any* alternative class. Constrained by Class 10, VeriX produces a broad explanation, offering no insight into the resilience against the catastrophic Class 5. These methods fail to answer the crucial question for safety-critical deployment: "How resilient is the model's decision against a specific, high-risk alternative?"

To address this issue, we must distinguish between explanations of *fragility* and explanations of *resilience*. Counterfactual Explanations (CFEs) explain fragility by answering "if-only" questions: "If only these features changed, the decision would flip" (Guidotti 2024). We focus instead on ***semifactual*** explanations, which explain resilience by answering "even-if" questions: "Even if these features change, the decision persists" (Aryal and Keane 2023; E. Kenny and W. Huang 2023).

We introduce VITAX (Verified Targeted Explanations),[1] a novel framework that provides formally verified, targeted semifactual explanations. VITAX addresses the limitations of existing methods by prioritizing verification resources on user-specified, critical decision boundaries. For a given input (class y) and a user-specified critical alternative (class t), VITAX performs two key steps: **(1)** It identifies the minimal feature subset A that is *most sensitive* to the specific $y \to t$ transition. **(2)** It uses formal reachability analysis to *guarantee* that perturbing these sensitive features by a magnitude ϵ is insufficient to flip the classification to t.

As shown in Figure 1(D), VITAX ignores the nearest boundary (Class 10) and focuses analysis on the critical boundary (Class 5). The resulting explanation is precise, highlighting only the features relevant to the Stop → 60 kph transition. The perturbed logits confirm that the analysis is focused on Class 5. This provides a formal guarantee of resilience ("even-if") at the specific boundary that matters most.

Our main contributions are:

[1] Our code is publicly available at https://github.com/AICPS-Lab/formal-xai.

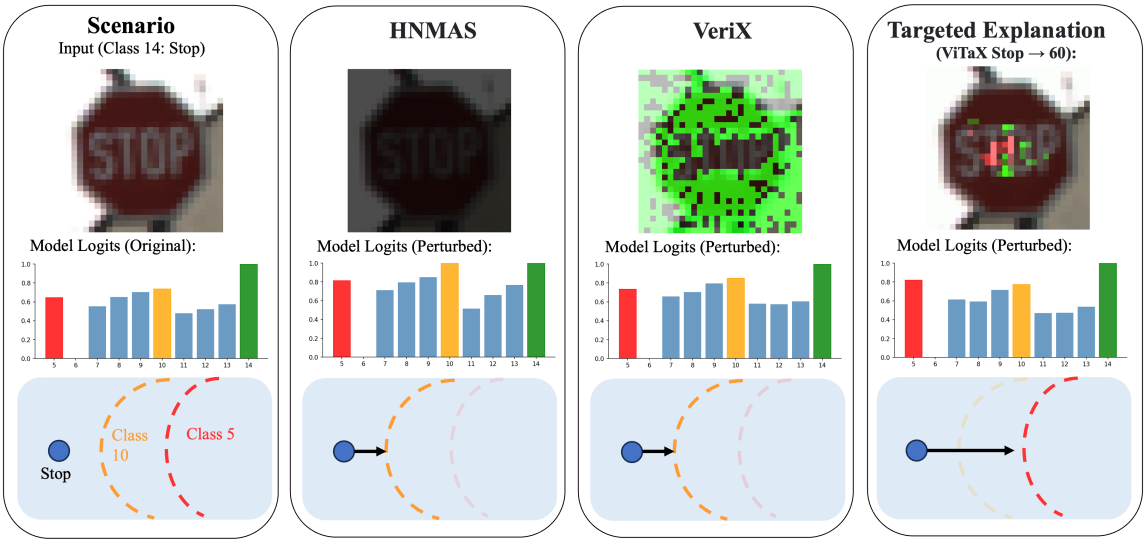


Fig. 1. **The Necessity of Targeted Verification: Prioritizing Critical Boundaries over Nearest Boundaries.** All methods use the same base model (GTSRB CNN). Formal verification is expensive and must be deployed strategically. **(A) Scenario:** A model correctly classifies 'Stop' (Class 14). The decision space has asymmetric risks: Class 10 ('No Pass') is the *nearest* boundary (low risk), while Class 5 ('60 kph') is the *critical* boundary (high risk). Risk levels are defined by the application domain (e.g., traffic safety regulations) and provided by the user; ViTaX then directs verification resources toward the user-specified critical boundary. **(B) HNMAS** (Formal Instance-level) (Leofante and Lomuscio 2023): Provides instance-level semifactual explanations via robustness for multi-agent systems. The analysis is constrained by local geometry, and the evaluation drifts (arrow) towards the nearest boundary (Class 10). **(C) VeriX** (Untargeted Formal) (M. Wu et al. 2023): Provides feature-level explanations of general resilience. The analysis is constrained by the nearest boundary, resulting in a broad explanation and evaluation drift (arrow) towards Class 10. **(D) ViTaX** (Targeted Formal): Provides feature-level explanations of targeted resilience. ViTaX focuses resources (arrow) on the critical boundary, resulting in a precise explanation and evaluation drift towards Class 5. In safety-critical deployment, understanding resilience against high-risk alternatives (Class 5, '60 kph') is essential, even when other boundaries (Class 10, 'No Pass') are closer but less dangerous. ViTaX provides guarantees where they matter most.

- We introduce ViTaX, the first framework to combine semifactual explanations ("even-if" robustness) with class-specific targeting ($y \rightarrow t$ transitions) and formal verification. This addresses a critical gap where existing methods either lack formal guarantees or ignore asymmetric risks in safety-critical decision boundaries.
- We define *Targeted ϵ-Robustness*, a novel formal property for characterizing the robustness of explanations. This property captures the resilience of a model against a specific alternative class under feature perturbations, enabling precise, verified characterization of class-specific decision boundaries.
- We propose a novel integration of sensitivity-driven heuristics and formal verification, utilizing an efficient search strategy ($\mathcal{O}(\log N)$ oracle calls) that makes targeted formal explanations computationally tractable, providing a practical alternative to intractable exact methods.
- We demonstrate on diverse tasks (image classification and regression) that ViTaX significantly outperforms existing heuristic and formal methods (including VeriX, HNMAS, LIME, Anchors) by providing precise, high-fidelity insights into critical decision boundaries, achieving over 30% improvement in explanation fidelity while maintaining minimal cardinality.

Paper Organization

The remainder of this paper is structured as follows. Section 2 lays the formal groundwork, introducing key concepts like standard ϵ-robustness and defining our core notion of *Targeted ϵ-Robustness*. Section 3 then details the VITAX framework, including its heuristic ranking, formal feature search algorithm for identifying explanations that satisfy Targeted ϵ-Robustness, and the soundness of these explanations. Section 4 presents a comprehensive empirical evaluation of VITAX across various datasets and tasks, comparing it against several baseline methods using metrics such as fidelity, cardinality, and robustness. We then situate our work within the broader XAI landscape in Section 5 by reviewing related literature on attribution methods, formal XAI, causality, and explainability metrics. Section 6 further discusses the unique nature and value of VITAX's guarantees, including their connection to semifactual explanations, and thoroughly compares VITAX with contrastive explanation paradigms and research in adversarial robustness. Finally, Section 7 concludes the paper by summarizing our contributions, reiterating the impact of VITAX, and outlining promising directions for future research. Appendices provide supplementary details including solver methods, proofs, additional experimental results, and model specifications.

2 From Standard to Targeted ϵ-Robust Explanations

In this section, we introduce the foundational concepts and formal problem formulation essential for understanding VITAX. Specifically, we formalize the concept of *resilience* from our introduction, providing a mathematical basis for quantifying a model's ability to maintain its decisions under targeted perturbations. These preliminaries establish the basis for our proposed method, which leverages formal reachability analysis to offer specific robustness guarantees pertinent to the identified explanations of neural network behavior. We present these concepts assuming the neural network input is an image with width w, height h, and c color channels, in a classification task with m classes.

Consider a deep learning model $f : \mathbb{R}^{w\times h\times c} \to \mathbb{R}^m$, where the function f maps an $(w \times h \times c)$-dimensional input image to an m-dimensional output vector (representing min-max normalized logits). The inputs and outputs of f define the feature space and the prediction space, respectively. To formally assess the model's output stability under specific input perturbations, we employ a reachability solver, denoted as $\mathcal{V}(f, X_\epsilon, \Phi)$. Here, $X_\epsilon \subseteq \mathbb{R}^{w\times h\times c}$ represents a continuous, infinite set of input images derived from an original input by applying perturbations up to a magnitude of ϵ (detailed in Definition 2.1). The term Φ represents the formal specification, a property we aim to verify, which in our context is a form of decision resilience. The solver evaluates the model f over the entire input set X_ϵ to compute the corresponding reach set of outputs $Y_\epsilon \subseteq \mathbb{R}^m$. That is, $f(X_\epsilon) = Y_\epsilon$, where Y_ϵ encapsulates all possible logit vectors for each class given the perturbed inputs. This output reach set Y_ϵ is then projected into a one-dimensional representation y_ϵ. This projection consists of a set of intervals, one for each class, indicating the lower (l) and upper (u) bounds of the predicted logits for that class. For instance, $y_\epsilon = [y_1, y_2, ..., y_m]$ where $y_1 = [l_1, u_1], y_2 = [l_2, u_2], ..., y_m = [l_m, u_m]$. This projection is illustrated in the rightmost part of Figure 2, where one example might depict a resilient sample (clear separation between the true class interval and others) and another a non-robust sample (overlapping intervals). The reachability solver then formally evaluates whether the specification Φ is satisfied by this projected output y_ϵ. In this work, Φ is primarily concerned with a notion of *ϵ-Robustness*, which we first define in its standard form before introducing our specific refinement for targeted explanations.

Definition 2.1 (Standard *ϵ-Robust Explanation*). Consider an input $x \in \mathbb{R}^{w\times h\times c}$ with true class $y \in \{1, 2, ..., m\}$. For a perturbation size $\epsilon > 0$ and a norm $p \in \{1, 2, \infty\}$, we define the input perturbation set $X_\epsilon = \{x' \in \mathbb{R}^{w\times h\times c} : \|x' - x\|_p \leq \epsilon\}$, where $\|\cdot\|_p$ denotes the $\mathcal{L}_p$ norm. A model is **ϵ-locally robust** at input x if, for all perturbed inputs in X_ϵ, the predicted class remains y. In terms of the projected reach set Y_ϵ, this means the interval corresponding to the true class **y** does not overlap with the interval of any other class **k** (H.-D. Tran, Manzanas Lopez, et al. 2019).

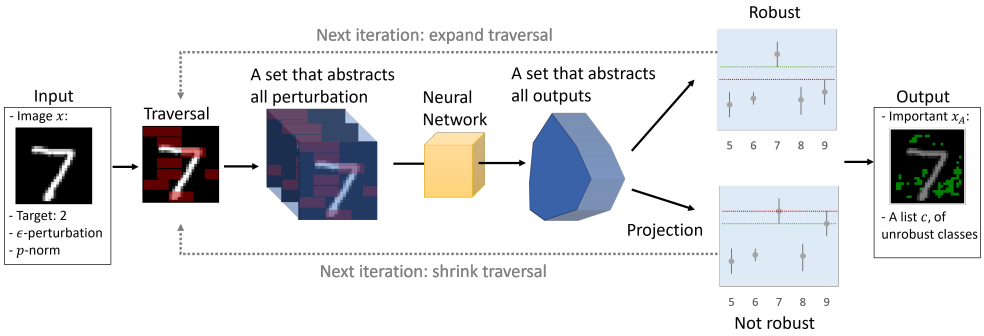


Fig. 2. VITAX takes an initial input image of '7', a target class $\mathbf{t}$ of '2', a perturbation ϵ, and a norm p. Using a heuristic ranking, VITAX identifies a subset of input pixels to be ϵ-perturbed, generating a subset of important input features.

More formally, the lower bound of the logits for the true class, $l_{\mathbf{y}}$, must be strictly greater than the upper bound of the logits for any other class $u_{\mathbf{k}}$, as described in Equation 1. Here, $\mathbf{y}$ is the true class, and $\mathbf{k}$ represents any other class. This standard definition provides a baseline measure of the model's general resilience, quantifying its ability to maintain a decision against untargeted disturbances.

$$\forall \mathbf{k} \in \{1, 2, \ldots, m\} \wedge \mathbf{k} \neq \mathbf{y}, l_{\mathbf{y}} > u_{\mathbf{k}} \tag{1}$$

Definition 2.2 (Targeted ϵ-*Robust Explanation*). Building upon this general notion of resilience, we now define **Targeted ϵ-Robustness**, the formal property central to VITAX. An explanation satisfying this property is defined as a *Targeted ϵ-Robust Explanation*. Consider an input x (true class $\mathbf{y}$) and a specific *target class* $\mathbf{t} \neq \mathbf{y}$. This property verifies the model's semifactual robustness: whether features can be perturbed without flipping the classification. The "targeting" refers to feature selection: subset A is chosen (via the heuristic described in Section 3.2) because its features are most sensitive to potential transitions towards class $\mathbf{t}$.

The model is considered **targeted ϵ-robust** concerning class $\mathbf{t}$ at input x with respect to a feature subset $A \subseteq \{1, 2, \ldots, n\}$ (where $n = w \times h \times c$ is the total number of features), if, when only these features in A are perturbed within $X_{\epsilon,A}$ (i.e., $\|x'_A - x_A\|_p \leq \epsilon$ for features in A, and $x'_F = x_F$ for features $F \notin A$), the model's prediction for the original class $\mathbf{y}$ remains distinguishable from the target class $\mathbf{t}$. Specifically, the lower bound of the projected logits for the true class $\mathbf{y}$ (considering perturbations only on A), denoted $l_{\mathbf{y},A}$, must be strictly greater than the upper bound of the projected logits for the targeted class $\mathbf{t}$, $u_{\mathbf{t},A}$. This property, defined in Equation 2, signifies that perturbing the feature set A alone (up to ϵ) in any direction within the ϵ-ball is formally verified to be *insufficient* to make the target class $\mathbf{t}$ more likely than or indistinguishable from the true class $\mathbf{y}$. This formal guarantee constitutes a verified semifactual statement: "Even if features in A are perturbed by ϵ (in any direction), the classification $\mathbf{y}$ persists and remains distinguishable from $\mathbf{t}$."

$$\exists A \subseteq \{1, 2, \ldots, n\}, \mathbf{t} \neq \mathbf{y}, \quad \text{s.t.} \quad \|x'_A - x_A\|_p \leq \epsilon \wedge l_{\mathbf{y},A} > u_{\mathbf{t},A}. \tag{2}$$

The *targeted* refers to how A is selected (using sensitivity toward $\mathbf{t}$ as a heuristic, Eq. 3), not to the direction of perturbations in the verification. The formal guarantee holds for perturbations in any direction within the ϵ-ball on features A. And an optional dominance constraint ($\forall k \neq y, t : u_{t,A} > u_{k,A}$) can ensure $\mathbf{t}$ is the most prominent alternative, though our evaluation uses only the core condition above (Section 3.2 and Appendix E.1).

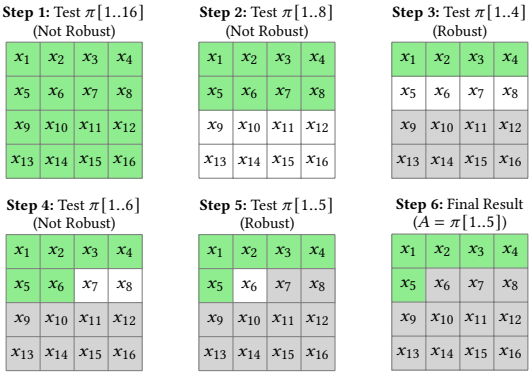


Fig. 3. Computing the explanation towards one particular class using reachability solver for a simple input $x = [1, ..16]$. Green is the current set of features that solver is evaluating. Gray denotes the irrelevant feature and white is undetermined features.

3 VITAX

To quantify a model's decision resilience efficiently, we present a novel framework called VITAX that builds upon ϵ-perturbation-based approaches, as shown in Figure 2. Similar to solving a cardinality minimal problem (Gainer-Dewar and Vera-Licona 2017; Ignatiev and Marques-Silva 2021; Ignatiev, Narodytska, et al. 2019), we incorporate a heuristic ranking function to perform a binary search based on sensitivity, solving the problem in $\log_2(n)$ time. Our approach differs in three key aspects: (1) Instead of evaluating features individually and iteratively, we focus on understanding the correlations between input features; (2) we use ϵ-ball perturbations and the sensitivity of the output with respect to the input to find targeted explanations; (3) we emphasize using a star-based reachability solver to generate explanations that are most *sensitive* to the model. The VITAX framework is described in more detail below.

3.1 A Running Example

Before presenting the algorithm in detail, we first illustrate it via an example. Suppose x is the 16-features input $[x_1, ..x_{16}]$ as shown in Figure 3, having a classification model f, a perturbation magnitude ϵ, and $p = \infty$. For simplicity, in this example, the features are ordered x_1 to x_{16} from most to least sensitive. Initially, we might evaluate the entire set of features for *robustness* with respect to a *target class* $\mathbf{t}$ under ϵ-perturbation. (In practice, Algorithm 1 uses a binary search and doesn't necessarily start with the whole set if a smaller set is already assumed non-robust under large ϵ, or robust if ϵ is very small). Concretely, when a candidate subset of features d (e.g., x_1 to x_{16}) is evaluated, VITAX composes a pre-condition $X_{\epsilon,d}$ representing the ϵ-perturbation applied only to these features. This $X_{\epsilon,d}$ is passed to a reachability solver to compute the output reach set Y_ϵ. A verifier then checks if this reach set implies the post-condition of *Targeted ϵ-Robustness* (Definition 2.2), as per Lines 10 - 12 of Algorithm 1.

Assume the verifier returns that the full set (x_1 to x_{16}) is not robust. Following the binary search logic, VITAX would then test a smaller prefix, say the first half: x_1 to x_8. The features x_9 to x_{16} are temporarily held as outside this current test prefix (as shown on the top middle of Figure 3).

In the third grid of Figure 3, let's assume the solver then returns that this prefix of eight features (x_1 to x_8) is also not robust. This indicates that if a robust prefix exists among the most sensitive features, it must be shorter than eight features. Therefore, the binary search adjusts its upper bound, effectively excluding features x_9 to x_{16} (now colored grey) from consideration as part of the maximal robust prefix because the current search focuses

on prefixes of length less than eight. The algorithm would then proceed to evaluate an even smaller prefix, for instance, x_1 to x_4, while features x_5 to x_8 are held as unspecified for this iteration.

If, subsequently, the subset x_1 to x_4 is found to be robust (resilience holds), the algorithm would then attempt to find a larger robust prefix by exploring features between x_4 and x_8 (e.g., testing x_1 to x_6, then x_1 to x_5). This iterative process continues.

Eventually, VITAX identifies a subset $A = [x_1, x_2, x_3, x_4, x_5]$ as the largest prefix of the ranked features for which Targeted ϵ-Robustness holds (bottom right grid of Figure 3). This means that for this set A, perturbing its features by ϵ (in the direction of sensitivity towards $\mathbf{t}$) formally guarantees that the original class remains distinguishable from the target class $\mathbf{t}$ (as per Definition 2.2), and any minimal expansion of this set A (by including the next most sensitive feature, x_6 in this example) would cause this guarantee to fail.

3.2 Computing VITAX Explanations

VITAX consists of three main stages: 1) **heuristic ranking**, 2) **feature search**, and 3) **final assessment** (as illustrated in Section 3.1). We describe each of these components in more detail below, followed by Algorithm 1.

(1) Heuristic Ranking (Saliency): To effectively analyze a subset of features, we first introduce a heuristic ranking method based on feature sensitivity to produce a traversal order for the reachability solver, inspired by the occlusion and saliency methods (Ghorbani et al. 2019; Simonyan et al. 2014; M. Wu et al. 2023; Zeiler and Fergus 2013). Sensitivity (Saliency) is quantified by the partial derivatives of the network's output with respect to each input feature for a target class. For a neural network f and input vector $x = [x_1, x_2 \ldots, x_n]$, the sensitivity for the i-th feature x_i is defined as:

$$\textsc{HeuristicRanking}(x, \mathbf{t}) = \mathrm{S}(x, \mathbf{t}) = \nabla_x f_{\mathbf{t}}(x) \tag{3}$$

The gradient of the output with respect to each feature measures the influence of each input component on the model's decision. A high magnitude indicates a significant impact, while the sign indicates the change direction. We utilize the sensitivity as a ranking $\pi \in \mathbb{R}^n$ of all features based on the absolute magnitude of their sensitivities. We then rank the feature indices from most sensitive to least sensitive. In the running example (Section 3.1), this corresponds to the assumed ordering x_1 to x_{16} from most to least sensitive. For instance, consider an input $x = [x_1, x_2, x_3]$. We evaluate the sensitivity of each feature to create the heuristic ranking $\pi = [x_3, x_1, x_2]$ s.t. $|S(x_3)| \geq |S(x_1)| \geq |S(x_2)|$. This heuristic is motivated by the intuition that features exhibiting a high gradient magnitude towards the target class $\mathbf{t}$ are most likely to influence a potential transition towards $\mathbf{t}$. Prioritizing such features provides a relevant ordering for efficiently identifying a minimal critical set A that satisfies our Targeted ϵ-Robustness condition.

(2) Feature Search: To identify the minimal subset of the most sensitive features (as ranked by π) that are critical for the transition towards target $\mathbf{t}$, we aim to determine the largest prefix d of π (i.e., $\pi[0:u]$) for which Targeted ϵ-Robustness (Definition 2.2) still holds. Consider a subset of features x_d corresponding to indices $d \subset \{1, 2, \ldots, n\}$. An ϵ-perturbation applied only to this subset is represented as $X_{\epsilon,d} = \{x' : \|x' - x\|_p^d \leq \epsilon \text{ and } x'_k = x_k \text{ for } k \notin d\}$. Our objective is to find the largest set d (largest u) such that $\mathcal{V}(f, X_{\epsilon,d}, \Phi)$ returns true for the specification Φ derived from Definition 2.2.

To efficiently determine the largest prefix d satisfying our formal condition, we employ a binary search strategy over the heuristically ranked features π. This approach is motivated by the need to minimize queries to the computationally intensive formal solver $\mathcal{V}$ (requiring only a logarithmic number of calls) while still guaranteeing the identification of the maximal prefix that meets the specification Φ. The "loss of robustness" occurs when attempting to include an additional feature from π causes this condition to fail. The search process, detailed as steps (2a)-(2d) in Algorithm 1, systematically finds this boundary. Each Step in Figure 3 corresponds to one full iteration of this loop (2b→2c→2d): Step 1 tests $\pi[1..16]$, Step 2 tests $\pi[1..8]$, Step 3 tests $\pi[1..4]$, Step 4 tests $\pi[1..6]$, Step 5 tests $\pi[1..5]$, and Step 6 shows the final result.

Algorithm 1 VITAX (Reachability eXplanation)

Input: neural network f, input x
Parameters: target class $\mathbf{t}$, norm p, perturbation ϵ
Output: important feature x_A, unsatisfied class c

1: **function** VITAX
2: $\pi \leftarrow\!\shortmid$ HEURISTICRANKING$(x, \mathbf{t})$ — **(1)**
3: $I \leftarrow\!\shortmid 0$
4: $J \leftarrow\!\shortmid n$
5: $A, A_{candidate} \leftarrow\!\shortmid \emptyset$ — **(2a)**
6: **while** $I \leq J$ **do**
7: $u \leftarrow\!\shortmid (I + J)/2$
8: $d \leftarrow\!\shortmid \pi[0 : u]$
9: $X^p_{\epsilon,d} \leftarrow\!\shortmid \|x' - x\|^d_p \leq \epsilon$ — **(2b)**
10: $\Phi \leftarrow\!\shortmid l_{\mathbf{y},d} > u_{\mathbf{t},d}$
11: **for** $\mathbf{k} \in \{1, \ldots, m\} \setminus \{\mathbf{y}, \mathbf{t}\}$ **do**
12: $\Phi \leftarrow\!\shortmid \Phi \wedge u_{\mathbf{t},d} > u_{\mathbf{k},d}$ — **(2c)**
13: **end for**
14: (FLAG, c) $\leftarrow\!\shortmid \mathcal{V}(f, X_{\epsilon,d}, \Phi)$
15: **if** FLAG **then**
16: $I \leftarrow\!\shortmid u + 1$
17: $A_{candidate} \leftarrow\!\shortmid \pi[0 : u]$
18: **else** — **(2d)**
19: $J \leftarrow\!\shortmid u - 1$
20: **end if**
21: **end while**
22: $A \leftarrow\!\shortmid A_{candidate}$
23: **return** x_A, c — **(3)**
24: **end function**

(2a) *Initialization (Lines 3-5):* Set search pointers $I = 0$ (start of π) and $J = n$ (end of π). A will store the resulting feature subset.

(2b) *Feature Selection (Lines 7-9):* In each iteration of the binary search, compute the midpoint u of the current interval $[I, J]$. Select the top u features from π as the current candidate subset $d = \pi[0 : u]$. Define the perturbation set $X^p_{\epsilon,d}$ for these features.

(2c) *Verification (Lines 10-12 and following solver call):* Construct the formal specification Φ. The primary condition, Φ_{target}, checks if $l_{\mathbf{y},d} > u_{\mathbf{t},d}$ (from Equation 2), ensuring the original class $\mathbf{y}$ remains distinct from the target class $\mathbf{t}$ when only features in d are perturbed. Moreover, an optional dominance constraint Φ_{others} verifies that $u_{t,d} > u_{k,d}$ for all other classes $k \neq y, t$. This ensures the explanation unambiguously targets $\mathbf{t}$ rather than another alternative $\mathbf{k}$. However, this constraint can be restrictive when multiple classes are closely clustered in logit space. Our evaluation uses only Φ_{target}, providing valid semifactual explanations for robustness against $\mathbf{t}$ without requiring $\mathbf{t}$ to dominate all alternatives. Cases where both constraints apply are analyzed in Appendix E.1. The specification Φ is then verified using the solver $\mathcal{V}(f, X_{\epsilon,d}, \Phi)$.

(2d) *Update (Following **If** statement):* If FLAG is false (meaning Φ is not satisfied for subset d), or specifically if Φ_{target} is not met, it implies that perturbing the current subset d breaks the model's resilience against **t** (or another class **k** became more prominent than **t**). Thus, d is too large or contains features that cause this loss of targeted robustness. The search space is narrowed by setting $J \leftarrow\!\shortmid u - 1$. If FLAG is true, the subset d satisfies the Targeted ϵ-Robustness (and the conditions for other classes **k**). We then try to find an even larger robust subset by searching in the right half: $I \leftarrow\!\shortmid u + 1$. In Figure 3, Step 3 (robust at $\pi[1..4]$) triggers expansion to Step 4, and Step 5 (robust at $\pi[1..5]$) confirms the final boundary.

(3) Final Assessment (Line 22 and Return): The binary search continues until $I > J$. The set $A_{candidate}$ from the last successful verification (largest u for which Φ_{target} held) is the desired set of important features A. This corresponds to Step 6 of Figure 3, where $A = \pi[1..5]$ is the final result. If no such $A_{candidate}$ was found (e.g., even perturbing the single most sensitive feature causes loss of Targeted ϵ-Robustness), A would be empty. The algorithm returns x_A (the features corresponding to indices in A) and $c_{candidate}$ (classes **k** that were not robustly less likely than **t** for the final A). The heatmap for explanation is generated from x_A, showing features whose ϵ-perturbation (in the direction of sensitivity) maintains the condition in Definition 2.2. If the model is inherently non-robust or highly sensitive near input x, VITAX will correctly return a very small or empty feature set A. Conversely, if the model is robust at input x for the given target class **t**, the algorithm will find a large set A, indicating that many features can be simultaneously perturbed by ϵ without flipping the classification to **t**. Therefore, an empty A arises exclusively from non-robustness and serves as a diagnostic signal that the model's decision boundary is fragile at that input. In such cases, reducing ϵ may yield a non-empty explanation, as a smaller perturbation magnitude can better find the robustness guarantee for the most sensitive features. Because the heuristic concentrates the most critical features at the top of the ranking, a smaller maximal robust prefix (lower cardinality) indicates that the vulnerability is tightly isolated, yielding a highly focused, minimal explanation of the decision boundary.

3.3 Formal Guarantees of VITAX Explanations

In formal XAI, *soundness* implies that an explanation accurately reflects the model's reasoning, while *completeness* means it captures all necessary aspects of that reasoning (Haufe et al. 2024; Key 2025; Marques-Silva and Ignatiev 2022; Paul et al. 2024). For VITAX, the explanation x_A is sound if it genuinely represents a feature set satisfying our definition of Targeted ϵ-Robustness. We include the definitions, rigorous proofs, and empirical evaluations for Lemma 1 and Remark 1 and Theorem 3.1 in Appendix B.1, B.2, B.3, respectively. To simplify notation in these proofs, the input is treated as a flattened vector of length $n = w \times h \times c$.

Lemma 1 (Feature Inclusion Property)**.** Given a model $f : \mathbb{R}^n \to \mathbb{R}^m$, for $y = f(x)$ and $i \in \{1, 2, \ldots, n\}$, define the set $X_{\epsilon,i}$ as:

$$X_{\epsilon,i} = \{x' \in \mathbb{R}^n : \|x' - x\|^i \leq \epsilon\} \text{ (perturbation on } i^{th} \text{ feature only)} \tag{4}$$

Then, the reachability property follows:

$$Y_\epsilon = f(X_\epsilon) \supseteq \bigcup_{i=1}^{n} f(X_{\epsilon,i}) \tag{5}$$

The output range resulting from perturbing all features simultaneously up to ϵ is a superset of the union of the output ranges resulting from perturbing each feature i individually.

Remark 1 (Monotonicity of Reach Sets under Feature Perturbation)**.** Given a model $f : \mathbb{R}^n \to \mathbb{R}^m$ and the sets defined as $X_\epsilon = \{x' \in \mathbb{R}^n : \|x' - x\|_p \leq \epsilon\}$ and $X_{\epsilon,i} = \{x' \in \mathbb{R}^n : \|x'_i - x_i\|_p \leq \epsilon\}$ for $i \in \{1, 2, \ldots, n\}$. Suppose we have two subsets of feature indices d and l from $\{1, 2, \ldots, n\}$ such that $|d| < |l|$ and $d \subset l$. Define the output

reach sets when perturbing only features in d and l respectively:

$$Y_{\epsilon,d} = f(X_{\epsilon,d}) \text{ and } Y_{\epsilon,l} = f(X_{\epsilon,l}). \tag{6}$$

Then, the following holds:

$$Y_{\epsilon,d} \subseteq Y_{\epsilon,l} \tag{7}$$

Theorem 3.1 (Soundness of VITAX Explanation). *If the reachability solver, $\mathcal{V}$, used as a sub-routine in Algorithm 1 is sound and complete (i.e., it correctly determines if the property $\Phi(d)$ holds for the input set $X_{\epsilon,d}$ corresponding to feature subset d), then the feature subset A identified by VITAX (and its corresponding x_A) satisfies the following properties:*

***(1) Targeted ϵ-Robustness:** The set A satisfies Definition 2.2 ($\Phi(A)$ holds, meaning $l_{\mathbf{y},A} > u_{\mathbf{t},A}$ and $u_{\mathbf{t},A} > u_{\mathbf{k},A}$ for other classes $\mathbf{k}$).*

***(2) Maximal Robust Set:** A is the largest prefix of the heuristic ranking π for which Φ holds. Formally, if $A = \pi[0:u]$, then $\Phi(\pi[0:u]) = True$ and $\Phi(\pi[0:u+1]) = False$ (or $u = |\pi|$).*

***(3) Heuristic Dependence:** The size $|A|$ depends on the quality of π. A different ranking π' may yield a smaller or larger robust set A'.*

VITAX provides an $\mathcal{O}(\log_2(N))$ solution to the problem of finding a feature set satisfying Φ, where the quality of the solution (minimality of $|A|$) depends on how well π captures feature importance for the $\mathbf{y} \to \mathbf{t}$ transition.

Proof Sketch. Let A_{final} be the feature subset returned by Algorithm 1. The algorithm identifies A_{final} as the largest prefix $\pi[0:u]$ of the heuristically ranked features π for which the solver $\mathcal{V}$ returned 'FLAG = True' when verifying the property $\Phi(A_{final})$. The property $\Phi(d)$ is defined as $\left(l_{\mathbf{y},d} > u_{\mathbf{t},d}\right) \wedge \left(\forall \mathbf{k} \neq \mathbf{y}, \mathbf{k} \neq \mathbf{t} \implies u_{\mathbf{t},d} > u_{\mathbf{k},d}\right)$, which directly embodies Definition 2.2 along with the condition for other classes.

Since $\mathcal{V}$ returned 'FLAG = True' for A_{final} and $\mathcal{V}$ is assumed to be sound (if $\mathcal{V}$ says $\Phi(d)$ holds, it does), it directly implies that $\Phi(A_{final})$ holds. Thus, A_{final} satisfies Targeted ϵ-Robustness (Property 1).

Furthermore, the binary search procedure is designed to find the maximum u (and thus the largest prefix $A_{final} = \pi[0:u]$) for which Φ holds. If a larger prefix $\pi[0:u']$ (with $u' > u$) existed that also satisfied Φ, the search logic (specifically, updating the lower bound $I \leftarrow u+1$ and storing $A_{candidate}$) would have aimed to find it. The algorithm terminates when this process converges, ensuring A_{final} is the largest such prefix for which $\mathcal{V}$ (assumed complete: if $\Phi(d)$ holds, $\mathcal{V}$ returns 'FLAG = True') confirms Φ (Property 2).

Property 3 follows from the algorithm's structure: the binary search operates on the ranking π, and a different ranking would lead to testing different feature subsets in different orders, potentially yielding a different final set A'.

A detailed proof is available in Appendix B.3. □

4 Evaluation

This section presents a comprehensive empirical validation of VITAX. Our evaluation is structured around key research questions designed to assess its performance, scalability, and the unique nature of its formally verified semifactual explanations.

- **RQ1 (Performance):** How does VITAX compare quantitatively against state-of-the-art heuristic, formal, and alternative semifactual methods in terms of fidelity, cardinality, and robustness?
- **RQ2 (Scalability & Applicability):** How does VITAX perform on larger architectures, and how does it generalize to other tasks (e.g., regression)?
- **RQ3 (Qualitative Insight):** What is the qualitative nature of VITAX's targeted semifactual explanations, and how do key parameters and components affect the results?

4.1 Experimental Setup

Datasets and Architectures. We utilize several datasets: MNIST (LeCun et al. 2009), GTSRB (Stallkamp et al. 2012), EMNIST Letters (Cohen et al. 2017), and the TaxiNet regression dataset (Julian et al. 2020). To address concerns regarding scalability, we analyze performance across models of varying complexity (MLP, CNN, ResNet, Inception; details in Appendix H) and discuss the computational trade-offs inherent in formal verification. For primary evaluations, we use 100 correctly predicted test samples.

Baselines. We compare VITAX against a diverse set of methods. Recognizing that standard implementations of LIME and Anchors methods are not inherently designed for targeted semifactual explanations, we specifically adapted these baselines to align their objectives with the semifactual goal of explaining resilience (Details in Appendix D). The comparison set includes:

- **Probabilistic Semifactual (Adapted LIME & Anchors):** We adjusted LIME (Ribeiro et al. 2016b) and Anchors (Ribeiro et al. 2018) to generate probabilistic semifactual explanations by identifying minimal feature sets that ensure prediction persistence with high probability. For LIME, we implemented a two-stage approach (local linear model + greedy search for stability); for Anchors, we formalized its greedy optimization to meet a high precision threshold.
- **Heuristic Semifactual (TSA & Prototype):** TSA (Targeted Semifactual Adversarial) uses iterative optimization (e.g., PGD) to find a perturbation δ that maximizes the target logit while maintaining the original classification via a margin penalty (Aryal and Keane 2023; Chowdhury et al. 2025; E. Kenny and W. Huang 2023; E. M. Kenny and Keane 2021). Prototype identifies the closest existing data sample that shares the original class but is nearest the target class in logit space, using the pixel-wise difference as the explanation (Aryal and Keane 2023; Fernández et al. 2022).
- **Formal (Untargeted):** VeriX (M. Wu et al. 2023), which identifies features sufficient for the original classification against any alternative.
- **Formal (Instance-level):** HNMAS (Leofante and Lomuscio 2023), which generates robust counterfactual instances for multi-agent systems. As it provides instance-level explanations (entire input modifications) rather than feature subsets, standard feature-level metrics like cardinality and fidelity (Eq. 8) are not directly applicable; we include it for qualitative comparison (Figure 1).
- **Algorithmic Baseline:** Brute-Force (linear search using the VITAX solver).

Implementation Details. VITAX utilizes the NNV tool (Lopez et al. 2023; H.-D. Tran, Manzanas Lopez, et al. 2019) as the subroutine solver $\mathcal{V}$. We utilize both the 'Approx-Star' method and the 'CP-Star' method (Hashemi et al. 2025). Experiments are conducted on a workstation equipped with an AMD Ryzen Threadripper PRO 7975WX 32-Core CPU and an NVIDIA RTX 6000 Ada Generation GPU.

Metrics. **Fidelity** (Eq. 8) measures how effectively the explanation influences the model's prediction towards the target class $\mathbf{t}$, while penalizing influence towards other classes $\mathbf{k}$.

$$\text{Fidelity} = \frac{f(x')_{\mathbf{t}} - f(x)_{\mathbf{t}}}{f(x)_{\mathbf{y}}} - \frac{\sum_{\mathbf{k}\neq\mathbf{y},\mathbf{k}\neq\mathbf{t}} \max\left[0, f(x')_{\mathbf{k}} - f(x')_{\mathbf{t}}\right]}{f(x)_{\mathbf{y}}} \tag{8}$$

Here, x' is generated by applying the worst-case adversarial perturbation of magnitude ϵ to the features in A towards the target class $\mathbf{t}$, as determined by the reachability solver.

Cardinality measures the average number of features in A. **Time** measures the average computational time in seconds.

Robustness (Robustness against Noisy Execution - NE (Jiang et al. 2023, 2024)). We adapt the procedure from Anchors (Ribeiro et al. 2018) to evaluate the sufficiency of A. We randomly perturb the *non-explanatory* regions (features outside A) by replacing them with features from target class t samples (500 trials), and measure

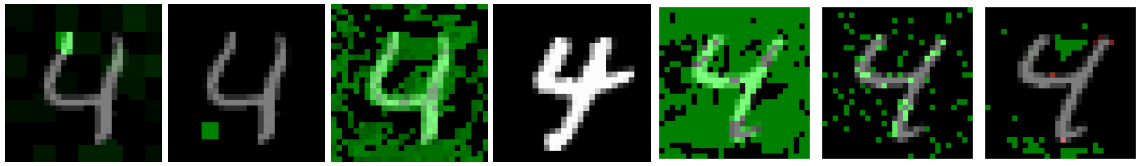


Fig. 4. Qualitative comparison of explanations for an MNIST sample (original '4', target '9' for VITAX). All methods use the same base model (MNIST MLP). VITAX highlights the critical feature set A. Note that this visualization shows the sensitive features (a semifactual insight), not a counterfactual instance that successfully flips the class.

Table 1. Comprehensive Evaluation Results. ↑ indicates higher is better; ↓ indicates lower is better. Best results among heuristic/probabilistic methods are underlined; best results among formal methods are bolded.

	MNIST (MLP)				GTSRB (CNN)			
Method	**Fid. ↑**	**Rob. (NE) ↑**	**Time (s) ↓**	**Card. ↓**	**Fid. ↑**	**Rob. (NE) ↑**	**Time (s) ↓**	**Card. ↓**
Heuristic/Probabilistic Methods								
LIME (Adapted)	0.09	0.93	3.42	44.27	0.32	0.60	4.65	<u>47.11</u>
Anchors (Adapted)	0.11	0.94	4.06	<u>41.59</u>	<u>0.57</u>	0.56	8.73	75.11
TSA	<u>0.52</u>	<u>0.97</u>	0.01	514.37	0.10	0.92	0.02	621.47
Prototype	0.15	0.93	<u>0.00</u>	214.44	0.33	<u>1.00</u>	<u>0.00</u>	782.59
Formal Methods								
VeriX	0.16	**0.99**	699.91	472.30	0.10	**0.93**	1536.34	764.97
VITAX (Ours)	**0.56**	0.97	**11.89**	**121.47**	**0.78**	0.70	**47.23**	**30.57**

the fraction of trials for which the model's prediction *does not flip* to t. A high robustness score indicates that A alone is sufficient to maintain the decision, demonstrating stability despite adversarial perturbation of irrelevant features. Note the intentional duality here: while our formal guarantee (Definition 2.2) certifies safety when perturbing A, the empirical NE metric tests stability by freezing A. Because A captures the features most sensitive to the $\mathbf{y} \to \mathbf{t}$ transition, freezing A anchors the prediction, protecting it against adversarial noise in the background.

4.2 Comparative Analysis (RQ1)

To answer RQ1, we evaluate VITAX against all baselines on MNIST and GTSRB across fidelity, robustness (NE), cardinality, and computational cost. The quantitative results are summarized in Table 1, and Figure 4 provides a qualitative visualization of the explanations generated by each method.

Performance Overview. VITAX demonstrates superior balance of explanation quality and efficiency among formal methods. It achieves the highest fidelity across both datasets (0.56 on MNIST, 0.78 on GTSRB) while simultaneously maintaining the lowest cardinality among formal methods (121.47 on MNIST, 30.57 on GTSRB). This combination is crucial: high fidelity demonstrates that the identified features are highly relevant to the targeted class transition, while low cardinality ensures the explanation is concise and interpretable. VITAX's exceptionally low cardinality on GTSRB (30.57 features) indicates highly focused explanations, demonstrating the effectiveness of binary search with sensitivity-based ranking for identifying minimal critical feature sets.

Comparison with Semifactual Methods. We compare VITAX against several methods adapted or designed for semifactual insights: the probabilistic approaches (LIME, Anchors) and the heuristic search methods (TSA, Prototype). VITAX outperforms all these baselines in achieving the optimal balance of fidelity and cardinality.

VITAX significantly outperforms the adapted probabilistic methods (LIME, Anchors) in fidelity. On MNIST, VITAX achieves 0.56 compared to 0.09 (LIME) and 0.11 (Anchors)—representing over 6× and 5× improvements respectively. On GTSRB, VITAX achieves 0.78 versus 0.32 (LIME) and 0.57 (Anchors)—a 37% improvement over the best probabilistic baseline. This integration of formal verification with targeted heuristic search is far more effective at isolating critical features than methods relying on local surrogate models and sampling.

The heuristic baselines reveal significant challenges in generating effective semifactual explanations without formal guarantees. The optimization-based TSA method achieves competitive fidelity on MNIST (0.52, closest to VITAX among heuristics) but fails dramatically on GTSRB (0.10), while producing extremely high cardinality (514-621 features). This architecture-dependent performance suggests that iterative optimization struggles to maintain the semifactual constraint across different model geometries and decision boundaries.

The Prototype method is exceptionally fast but shows inconsistent performance: low fidelity on MNIST (0.15) despite high robustness (0.93), and moderate fidelity on GTSRB (0.33) with perfect robustness (1.00). Its cardinality remains consistently high (214-782 features). This highlights a fundamental limitation: relying on existing data samples depends heavily on dataset density and distribution, and cannot systematically minimize explanations for targeted transitions.

The critical distinction between VITAX and all these heuristic/probabilistic methods lies in the nature of the guarantee (summarized conceptually in Table 2). While LIME/Anchors provide probabilistic estimates, and TSA/Prototype rely on localized optimization or distance metrics, VITAX provides a formal Targeted ϵ-Robustness guarantee. This formal underpinning ensures trustworthiness by verifying the entire continuous perturbation space, offering mathematical certainty where other methods provide estimations or localized examples.

Table 2. Conceptual Comparison of Semifactual Explanation Generation Methods, illustrated with HELOC dataset (FICO Community 2019). VITAX uniquely combines targeted insights with formal guarantees and scalability.

Method	**Mechanism**	**Example (HELOC)**	**Nature of Guarantee**
LIME/Anchors (Adapted)	Local Surrogate + Probabilistic Sampling. Identifies features that ensure prediction persistence with high probability.	"Maintaining the current 'ExternalRiskEstimate' and 'AverageMInFile' is sufficient to keep the 'Good Risk' status with 95% probability."	Probabilistic & Empirical.
Prototype / TSA	Example-based search or iterative optimization. Finds specific instances or perturbations that do not cross the boundary.	"Even when 'NumInqLast6M' was increased by 2, the status remained 'Good Risk'."	Heuristic / Best Effort.
VeriX	Formal Verification (MILP/SMT). Finds features that guarantee the prediction regardless of other feature values (Sufficiency).	"It's guaranteed that as long as 'ExternalRiskEstimate' < 65, the loan is always 'Good Risk'."	Deterministic & Formal (Untargeted).
VITAX	Formal Reachability + Efficient Search. Proves perturbation of critical features (A) by ϵ is insufficient to cross the boundary towards a specific target (t).	For a 'Good Risk' applicant, it's guaranteed that if you change 'ExternalRiskEstimate' and 'AverageMInFile' by up to 10% in any combination, the result will **not** change to 'Bad Risk'.	Deterministic & Formal (Targeted).

Analysis of Trade-offs and Formal Methods. When considering empirical robustness (NE), VeriX achieves the highest scores (0.99 on MNIST, 0.93 on GTSRB). This is expected, as VeriX aims for general sufficiency—identifying features that maintain the original classification against *any* alternative class, often operating further from specific decision boundaries. In contrast, VITAX operates precisely at the *targeted* boundary between the original and the target class.

VITAX achieves competitive robustness on MNIST (0.97, matching TSA and nearly matching VeriX's 0.99) while delivering dramatically superior fidelity (0.56 vs 0.16 for VeriX, a 3.5× improvement; 0.78 vs 0.10 for VeriX on GTSRB, a 7.8× improvement). This indicates substantially higher relevance to the targeted transition. The lower robustness on GTSRB (0.70) reflects the inherent trade-off: VITAX operates at the targeted boundary to maximize fidelity, accepting reduced general stability for targeted precision. Crucially, VITAX achieves this with dramatically faster runtimes (11.89s vs 699.91s on MNIST; 47.23s vs 1536.34s on GTSRB—representing 59× and 33× speedups respectively) and much lower cardinality. This highlights VITAX's effective balance between formal rigor, targeted explanatory power, and computational feasibility.

Qualitative Analysis. The quantitative advantages are complemented by qualitative differences. For the transition from '4' to '9', VITAX precisely highlights the upper loop closure necessary for the transition—the specific region where adding or modifying strokes would create the characteristic upper circle of a '9' (Figure 4). In contrast, baselines like LIME and Anchors identify broader, less focused regions that overlap with general features of '4', while VeriX focuses on features sufficient for maintaining '4' classification generally, rather than those specific to the boundary with '9'. This demonstrates VITAX's core advantage: providing targeted, boundary-specific insights rather than general attribution.

4.3 Scalability and Applicability (RQ2)

We analyze the scalability of VITAX regarding model architecture complexity and input dimensionality, as well as its applicability beyond classification. A key challenge in formal XAI is the computational cost of the underlying verification solvers when applied to deep architectures or high-dimensional inputs. We evaluate the feasibility of VITAX under these conditions and discuss the computational trade-offs.

Scalability to Complex Architectures. To address concerns about the applicability of VITAX to larger models, we extended our evaluation to five complex architectures on the MNIST dataset: a standard CNN, ResNet-Tiny, ResNet-Small, ResNet-Large, and an Inception model. We evaluated performance across two distinct perturbation budgets: a smaller budget ($\epsilon = 25/255 \approx 0.10$) and a significantly larger budget ($\epsilon = 55/255 \approx 0.22$).

Traditional deterministic reachability analysis solvers (like NNV Approx-Star) often struggle with the scale and complexity (e.g., residual connections, depth) of these architectures. The VITAX framework is agnostic to the underlying solver $\mathcal{V}$. To manage this complexity, we utilized the CP-Star verification backend, which is better suited for these complex network structures. This demonstrates VITAX's capability to integrate different verification strategies to achieve scalability.

The results demonstrate that VITAX successfully generates formally grounded explanations for these complex architectures, including deep ResNets and Inception models, within practical timeframes (under 70 seconds), regardless of the ϵ budget (Table 3). This is a significant demonstration of scalability, highlighting the benefit of VITAX's solver-agnostic design.

Impact of Model Complexity: We observe that Fidelity is generally lower across these models (0.16-0.55) compared to the simpler MLP (0.56 in Table 1). This is expected behavior. Deeper architectures often rely on distributed, hierarchical feature representations and tend to have more robust decision boundaries. Consequently, perturbing a feature subset (A) at the input level results in a smaller relative shift in logits (lower fidelity). This highlights a

Table 3. VITAX Performance on Complex Architectures (MNIST) for two perturbation budgets, $\epsilon \approx 0.10$ (25/255) and $\epsilon \approx 0.22$ (55/255), using the CP-Star Verification Backend. (Averages over 10 samples).

Model	# Params	Acc (%)	Fid. ↑		Rob. (NE) ↑		Time (s) ↓		Card. ↓	
			$\epsilon \approx 0.10$	$\epsilon \approx 0.22$	$\epsilon \approx 0.10$	$\epsilon \approx 0.22$	$\epsilon \approx 0.10$	$\epsilon \approx 0.22$	$\epsilon \approx 0.10$	$\epsilon \approx 0.22$
CNN	15,634	99.0	0.39	0.55	0.99	0.97	56.89	55.24	476.4	237.5
ResNet-Tiny	10,738	97.1	0.30	0.37	0.77	0.82	58.52	57.65	270.3	68.6
ResNet-Small	173,850	99.4	0.27	0.39	0.97	0.92	60.80	58.73	463.3	177.2
ResNet-Large	693,802	99.1	0.29	0.34	0.94	0.86	63.74	61.83	421.1	90.5
Inception	490,954	84.1	0.16	0.16	0.93	0.92	66.71	66.40	124.8	55.2

known limitation of input-space attribution methods: their explanatory impact inherently diminishes on very deep or highly robust models.

Impact of Perturbation Budget (ϵ): The comparison between $\epsilon \approx 0.10$ and $\epsilon \approx 0.22$ confirms the trends observed in RQ3 (Section 4.4). Increasing ϵ consistently leads to significantly lower Cardinality (e.g., 476.4 down to 237.5 for CNN), as the explanation must become more focused to maintain the guarantee under larger perturbations. Concurrently, the larger perturbation often leads to higher Fidelity (e.g., 0.39 up to 0.55 for CNN), as the features in A are perturbed more significantly towards the target class.

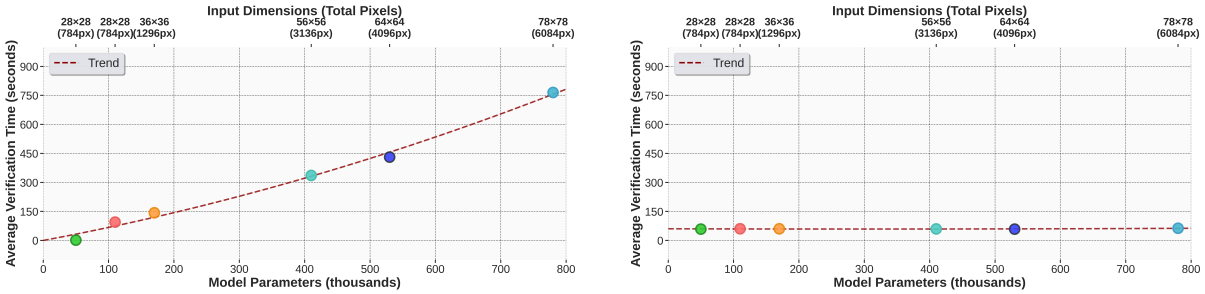
The Robustness (NE) shows some variation with the larger ϵ (e.g., 0.99 down to 0.97 for CNN; 0.77 up to 0.82 for ResNet-Tiny). The pattern is less consistent than in the simpler MLP case, reflecting the complex interplay between model architecture, decision geometry, and perturbation budget. For some models, larger ϵ brings explanations closer to boundaries (reducing NE), while for others, the more focused feature sets (lower cardinality) at higher ϵ may improve stability. This highlights a vital distinction between VITAX's formal guarantee (Targeted ϵ-Robustness) and empirical stability against untargeted noise. The formal guarantee ensures that perturbing A by ϵ does not cross the targeted boundary, but proximity to that boundary affects how robust the prediction is to noise in *other* features (as tested by NE). This contrasts with methods seeking general sufficiency (like VeriX, which achieves 0.99 NE on MNIST MLP in Table 1), which operate further from boundaries.

Scalability to Input Dimensions. To further evaluate scalability regarding input size, we analyzed the impact of increasing image dimensions on computation time. We trained MLP models (using the same hidden layer structure as the baseline MLP) on upscaled versions of the MNIST dataset: 28x28 (784 features), 56x56 (3136 features), and 78x78 (6084 features). We measured the execution time using both the Approx-Star and CP-Star solvers (Figure 5).

As the input dimension increases, the computation time increases for both solvers, reflecting the higher complexity of the search space. Approx-Star (Figure 5a) is significantly faster than CP-Star (Figure 5b) on these MLP architectures. Approx-Star scales from approx. 2s (28x28) to approx. 15s (78x78), whereas CP-Star scales from approx. 55s to 80s. This indicates that while VITAX can handle larger images, the choice of the underlying solver is critical and often architecture-dependent—Approx-Star is preferable for MLPs, while CP-Star may be necessary for complex CNNs/ResNets (as shown previously in Table 3).

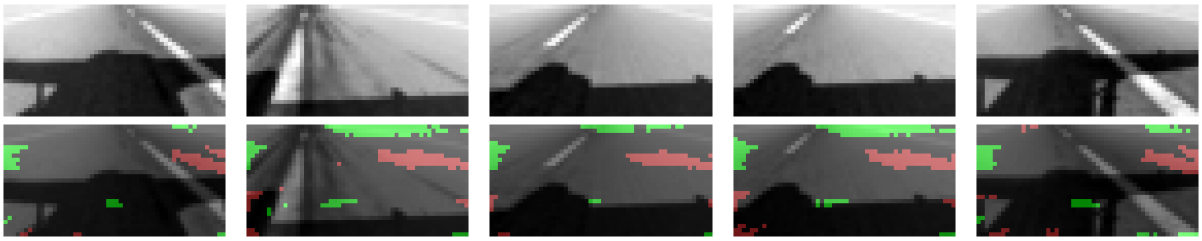
This analysis confirms that the VITAX methodology remains effective for identifying critical, targeted feature sets even on deep networks and larger inputs. We acknowledge the computational cost associated with formal verification and position VITAX as a rigorous tool for precise boundary analysis, where this cost is justified by the need for formal guarantees in safety-critical domains (see Section 6).

Application to TaxiNet Regression Task. To demonstrate broader applicability beyond classification, we applied VITAX to the TaxiNet regression task (Julian et al. 2020) (Figure 6). For regression, VITAX identifies features



(a) Approx Star (b) CP Star

Fig. 5. Scalability analysis on MLP models with varying input image sizes (MNIST 28x28, 56x56, 78x78). Runtime increases with input dimensions, with Approx-Star showing superior efficiency over CP-Star for MLP architectures.



(a) GT: 5.06 (Pred: 5.13) (b) GT: 1.55 (Pred: 1.40) (c) GT: -1.21 (Pred: -1.16) (d) GT: -0.70 (Pred: -0.83) (e) GT: 4.90 (Pred: 5.01)

Fig. 6. Application of VITAX to the TaxiNet regression task. Top row: original input images. Bottom row: VITAX explanations highlighting features relevant to the predicted steering angle (values shown as Ground Truth (Model Prediction)). Green indicates positive perturbations, red indicates negative, relative to their influence on the output.

critical for explaining the model's resilience around its current output value (steering angle). VITAX reveals how the critical feature set A changes based on the targeted output shift—focusing on immediate path projection (e.g., the white lines near the center) for small adjustments, and features further ahead for larger changes. This demonstrates VITAX's utility for understanding the dynamic feature reliance in continuous control models.

4.4 Qualitative Analysis and Ablation Studies (RQ3)

We now examine the qualitative nature of VITAX's explanations and conduct ablation studies on key components—the perturbation magnitude ϵ, the heuristic ranking function, and the underlying solver—to understand their impact on the resulting explanations.

Qualitative Effectiveness. We examine VITAX's application to the GTSRB (Figure 7) and EMNIST (Figure 8) datasets to qualitatively assess the nature of its targeted semifactual explanations. These results demonstrate VITAX's ability to identify minimal, critical feature subsets pertinent to understanding the resilience against specific target classes.

In the GTSRB example (Figure 7), we analyze the resilience of the "Straight" sign against different directional alternatives. When the target is "Turn Right" (Col 2) or "Turn Left" (Col 3), VITAX correctly identifies the specific side of the vertical arrow shaft as the critical feature set A. This aligns with intuition: these are the precise areas

Fig. 7. GTSRB Targeted Explanations. The first row shows the original "Straight" sign and various targets. The second row shows the VITAX semifactual explanations (Feature Set A) critical for the resilience of "Straight" against each specific target.

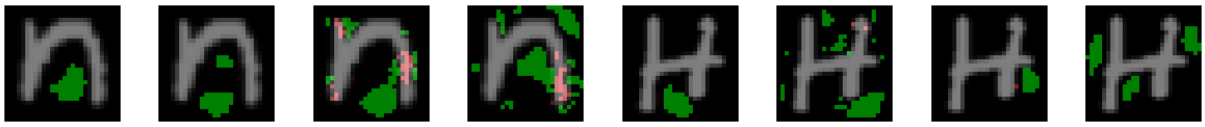


Fig. 8. EMNIST Targeted Explanations. VITAX identifies the semifactual explanation relevant to specific character transitions.

where the arrowheads for left or right turns would manifest. The explanation guarantees that the "Straight" classification persists even if these critical areas are perturbed, defining the boundary for this specific transition.

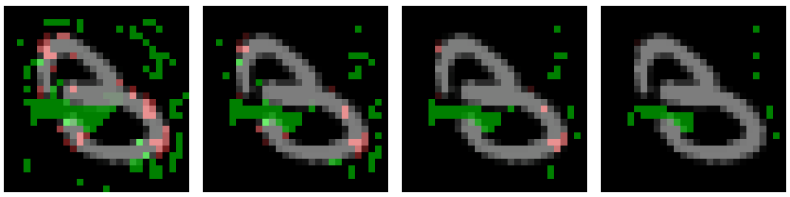
Similarly, on EMNIST (Figure 8), the explanations are highly specific to the target morphology. For the transition from 'n' to 'A' (a), VITAX highlights the center gap where a crossbar would be needed. For 'H' to 'm' (g), it focuses on the upper sections of the vertical lines. This demonstrates the core strength of VITAX: unlike general attribution methods that highlight features supporting the current class, VITAX isolates the features specifically relevant to the boundary with the chosen alternative.

Effect of Perturbation Size (ϵ). The perturbation magnitude, ϵ, is a critical parameter that controls the granularity of the explanation A (Figure 9). It defines the magnitude of change allowed when verifying the Targeted ϵ-Robustness property.

There is an inverse relationship between ϵ and the size of the explanation set A. When ϵ is small (e.g., 0.059), the allowed perturbation is minor. Consequently, a larger set of features A can be included while still guaranteeing that the classification (e.g., '3') persists against the target ('8'). The explanation is broader.

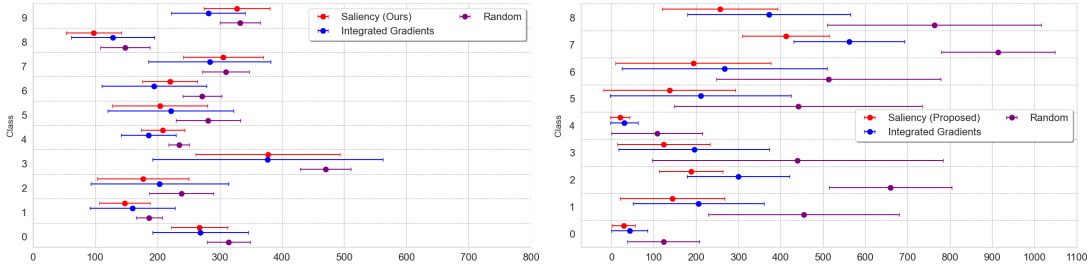
As ϵ increases (e.g., 0.216), the allowed perturbation is significant. To maintain the formal guarantee that the boundary is not crossed, the feature set A must be reduced, focusing only on the most critical features prioritized by the heuristic ranking. The explanation becomes sparser and more focused on the key areas differentiating '3' from '8' (the left-side gaps). This demonstrates how ϵ acts as a "zoom lens," allowing users to explore the decision boundary at varying levels of specificity.

Evaluating Heuristic Ranking Functions. The choice of the heuristic ranking function (π) is crucial for the efficiency and quality (minimality) of the VITAX explanation. While the formal guarantee holds regardless of the



(a) $\epsilon = 0.059$ (b) $\epsilon = 0.098$ (c) $\epsilon = 0.137$ (d) $\epsilon = 0.216$

Fig. 9. VITAX's targeted explanation for transitioning MNIST '3' (original) towards '8' (target) at different perturbation magnitudes ϵ. As ϵ increases, the minimal set A satisfying Targeted ϵ-Robustness tends to become smaller and more focused.



(a) MNIST (b) GTSRB

Fig. 10. Comparison of explanation cardinality (Number of Pixels, x-axis) resulting from different heuristic ranking functions across classes (y-axis). Lower cardinality is better. Sensitivity consistently yields the most minimal explanations. (GTSRB shows the first nine classes for readability).

heuristic (VITAX guarantees A is the maximal robust prefix of π), an effective heuristic guides the search towards smaller feature sets faster. We compared our proposed sensitivity-based approach (gradient saliency towards the target class) against Integrated Gradients (IG) (Sundararajan et al. 2017) and a Random ranking baseline.

The sensitivity-based heuristic consistently leads to explanations with significantly lower cardinality across both MNIST and GTSRB datasets (Figure 10). Random ranking performs the worst, often resulting in very large feature sets, as the search cannot efficiently prioritize relevant features.

The superiority of the sensitivity heuristic stems from its direct alignment with the goal of VITAX: by prioritizing features based on their direct impact on the target class logits, the search efficiently pinpoints the smallest prefix of features that defines the decision boundary. This analysis confirms that a well-aligned heuristic is pivotal for delivering minimal and focused explanations.

Impact of Solver Choice. The performance of VITAX is dependent on the underlying reachability solver. We compared different methods provided by the NNV tool—including Exact Star, Approx-Star, Relax-Star variants, and CP-Star—on a smaller MNIST model (Table 4) to assess the trade-offs between precision and speed.

The Exact Star method provides sound and complete analysis but is computationally expensive (30.71s). Approx-Star, which is sound but incomplete (it over-approximates the reach set), offers a significant speedup (2.04s, approximately 15 times faster). CP-Star, while crucial for complex architectures (Section 4.3), was the slowest among the MLP architectures (52.18s).

Interestingly, the Approx-Star method yields explanations with lower cardinality (111.46 vs 129.66 for Exact Star) while maintaining nearly identical fidelity (0.218 vs 0.220) and robustness (0.59 vs 0.60). Because Approx-Star

Table 4. Comparison of Different Reachability Solver Methods within VITAX (on a smaller MNIST model).

Solver Method	Time (s) ↓	Card. ↓	Fidelity ↑	Rob. (NE) ↑
Exact Star	30.71	129.66	0.220	0.60
Approx Star	2.04	111.46	0.218	0.59
Relax Star 50%	2.06	111.46	0.218	0.59
Relax Star 75%	2.03	109.91	0.217	0.59
Relax Star 85%	2.00	109.91	0.217	0.59
CP Star	52.18	130.28	0.211	0.59

over-approximates the reach set, it is more conservative in certifying robustness. This may force the search algorithm to identify a smaller, more critical feature set A to satisfy the Targeted ϵ-Robustness property under the over-approximation. Relax-Star methods offer marginal speed improvements with slight reductions in fidelity.

This analysis, combined with the findings in Section 4.3, demonstrates the importance of VITAX's solver-agnostic design. For simpler models such as MLPs, Approx-Star generally provides the optimal trade-off, enabling the efficient delivery of formally sound explanations at a significantly reduced computational cost (as also seen in Figure 5). However, the choice of solver is architecture-dependent; for deeper and more complex networks like ResNets or Inception (Table 3), alternative solvers such as CP-Star may be necessary to handle the complexity of the reachability analysis.

5 Related Work

The field of Explainable AI (XAI) aims to render the behavior of complex models transparent. We review the landscape relevant to VITAX, focusing on the evolution from heuristic methods to the emerging demand for formal guarantees and robustness.

5.1 Heuristic XAI: Attribution and Surrogates

Attribution-based methods are pivotal, highlighting influential input data portions that contribute to a model's decision. Examples include Grad-CAM, which uses gradient-based localization (Selvaraju et al. 2020), Integrated Gradients (IG) that assess feature significance via model gradients (Sundararajan et al. 2017), and DeepLIFT (Shrikumar, Greenside, and Kundaje 2019). Game-theoretic approaches like SHAP (SHapley Additive exPlanations) values use game theory to assign importance (Lundberg and Lee 2017). Other techniques include the Input X Gradient method, which multiplies the input by the output gradients (Shrikumar, Greenside, Shcherbina, et al. 2016), and Guided Backpropagation, which enhances standard backpropagation for visualization (Springenberg et al. 2015). These methods, among others (Linardatos et al. 2020; H. D. Wang et al. 2024), collectively enhance neural network interpretability, though they often provide only correlational insights and have been shown to lack stability.

Instead of analyzing the complex model directly, local surrogate models like LIME (Local Interpretable Model-agnostic Explanations) (Ribeiro et al. 2016b) train simpler, interpretable models to approximate the black-box behavior locally. Anchors (Ribeiro et al. 2018) identifies a minimal set of feature conditions sufficient to lock in a prediction. These provide plausible local explanations but lack formal assurances of their fidelity. Some approaches also utilize secondary models or delve into causal effects for classification tasks (Karimi et al. 2020; T. Q. Tran et al. 2022).

5.2 Robustness and Formal Guarantees in XAI

The fragility of standard heuristic methods has motivated a shift towards rigorous explanations. This involves two main thrusts: ensuring the stability of explanations (Robust XAI) and providing mathematical guarantees about their content (Formal XAI).

Regarding robustness of explanations, small input perturbations should not lead to vastly different explanations. Wicker et al. (Wicker et al. 2022) introduced constraints for robust feature attribution (e.g., saliency maps). Similarly, the robustness of CFEs is critical, ensuring that suggested changes remain valid under noise or model shifts (Jiang et al. 2024). Verification techniques have been employed to certify this validity. For instance, Leofante and Lomuscio (Leofante and Lomuscio 2023) introduced an approach to formally verify robust CFEs for human-neural multi-agent systems.

Formal methods in XAI aim to provide rigorous guarantees for explanations (Müller et al. 2022; Singh et al. 2019; Zelazny et al. 2022). Many works define an explanation as a minimal subset of input features critical for a model's decision, such that perturbations outside this subset ideally do not alter the prediction. Examples include abductive explanations (Ignatiev, Narodytska, et al. 2019), sufficient reasoning (Darwiche and Hirth 2020), prime implicants (Shih et al. 2018), and contrastive explanations (Miller 2019). The concept of *bounded* perturbations as explanations has been introduced, particularly in the NLP domain (Malfa et al. 2021). Building on these foundations, VeriX presents an iterative method to find the smallest number of features within continuous ϵ-ball perturbations that constitute an explanation (M. Wu et al. 2023). These methods often prioritize global cardinality minimality, which is generally NP-hard. Such efforts are closely related to the broader field of neural network verification, which often employs traditional software verification techniques like SMT (Katz, Barrett, et al. 2017) and MILP (Tjeng et al. 2017) for exact analysis, though these can face scalability challenges. Consequently, faster, albeit sometimes less precise, methods like abstract interpretation and probabilistic assessments have been developed. Significant research investigates how automated reasoning, through abstraction (Anderson et al. 2019; Gehr et al. 2018; Müller et al. 2022; Singh et al. 2019; H.-D. Tran, Bak, et al. 2020; S. Wang et al. 2018a,b; Wei et al. 2023; H. Wu et al. 2022; Zelazny et al. 2022; H. Zhang et al. 2018; S. Zhang et al. 2024) and search techniques (Ehlers 2017; Ferrari et al. 2022; Geng et al. 2023; X. Huang, Kwiatkowska, et al. 2017; Katz, Barrett, et al. 2017; Katz, D. A. Huang, et al. 2019), can verify network specifications (X. Huang, Kroening, et al. 2020; Liu et al. 2021). Formal logic-based approaches have also been explored for enforcing temporal properties in neural networks (Ma, Gao, et al. 2020) and for predictive monitoring with calibrated uncertainty (Ma, Stankovic, et al. 2021). Recent advances also include neural network-based meta-verification strategies that predict safety properties to reduce computational load (Elboher et al. 2022).

VITAX uniquely intersects these areas and is distinct from existing paradigms in several key ways. **(1) Derivation vs. Stability:** Unlike methods that improve the robustness *of* an existing explanation (e.g., Wicker et al. (Wicker et al. 2022) certify that gradient-based explanations remain stable when inputs or model parameters are perturbed), VITAX uses a formal robustness property (Targeted ϵ-Robustness) to *derive* the explanation itself—the explanation is the direct result of verifying which features satisfy the robustness property. For example, Wicker et al.'s approach would compute a gradient-based explanation (e.g., feature attribution scores) for an image classified as '7', then verify the explanation remains consistent under perturbations. In contrast, VITAX directly identifies the minimal feature set (e.g., specific pixels) that can be perturbed by ϵ without causing a 7→9 flip, with the robustness verification defining the explanation. **(2) Semifactual vs. Counterfactual Guarantees:** VITAX differs fundamentally from approaches focused on robust CFEs (e.g., Leofante and Lomuscio (Leofante and Lomuscio 2023)). While CFE methods guarantee the robustness of a *decision flip*, VITAX guarantees the robustness of the *non-flip* (the semifactual persistence). **(3) Scalability and Targeting:** Unlike Formal XAI methods focused on global minimality and general sufficiency (e.g., VeriX (M. Wu et al. 2023), Ignatiev et al. (Ignatiev, Narodytska,

et al. 2019)), VITAX sacrifices the intractable goal of global minimality for scalability (using $\mathcal{O}(\log_2(N))$ oracle calls) and provides targeted insights into specific class boundaries.

6 Discussion

Our empirical results demonstrate the effectiveness of VITAX in generating precise, formally-backed explanations. In what follows, we examine four aspects of this contribution: (1) the nature and scope of the Targeted ϵ-Robustness guarantee, (2) how VITAX produces verified semifactual explanations, (3) its fundamental distinction from counterfactual explanation paradigms, and (4) its positioning within the broader XAI landscape.

The Nature of the Guarantee for Targeted Semifactual Robustness. A key aspect of VITAX is the nature of the formal guarantee it provides. The Targeted ϵ-Robustness (Definition 2.2) is specifically local to the given input x, the identified feature subset A, and the chosen target class $\mathbf{t}$. This means that for the explanation A, VITAX formally verifies that perturbing these specific features by ϵ does not cause the model's classification for the original class $\mathbf{y}$ to flip to the target class $\mathbf{t}$. This provides a semifactual guarantee: "even if features in A are perturbed by ϵ, the classification remains $\mathbf{y}$."

This type of guarantee differs significantly from formal methods that aim to prove, for instance, the robustness of a model over an entire ϵ-ball neighborhood around x against *any* adversarial perturbation towards *any* incorrect class. Such global or comprehensive robustness certifications provide broad assurances about the model's stability. In contrast, VITAX offers a more nuanced and specific guarantee that is intrinsically tied to the explanation itself. Rather than certifying the model's general invulnerability, VITAX certifies a property of the *explanation's features*: that they represent a point of semifactual robustness against a specific, targeted alternative.

Unlike Formal XAI methods focused on global minimality and general sufficiency (e.g., (Ignatiev, Narodytska, et al. 2019)), which are often NP-hard and thus computationally intractable, VITAX makes a deliberate trade-off. It sacrifices the intractable goal of global cardinality minimality for scalability and targeted insights. The novelty of our approach lies not in the binary search algorithm itself, but in (1) the formulation of a new formal property, Targeted ϵ-Robustness, and (2) the creation of a framework that integrates a sensitivity-driven heuristic with formal verification to make these explanations computationally feasible. By using $\mathcal{O}(\log_2(N))$ oracle calls to a formal solver, VITAX provides a guarantee that is both rigorous and practical: it identifies the maximal robust prefix of a well-aligned heuristic. This offers a trustworthy and scalable alternative for understanding model behavior at specific class boundaries where exact methods would be infeasible.

VITAX as Verified Semifactual Explanations. Methods focused on robust XAI typically aim to ensure the robustness *of* explanations (such as gradient-based feature attributions) against perturbations in the input; they seek to certify that explanations do not change significantly when the input changes slightly (Wicker et al. 2022). In contrast, VITAX uses a robustness property of the *model's output* (Targeted ϵ-Robustness) to *derive* the explanation itself. The explanation (A) is defined by the verification of the model's robustness, rather than being a post-hoc robustness certification of a pre-computed explanation.

VITAX generates semifactual explanations (Aryal and Keane 2023; E. Kenny and W. Huang 2023) that answer "even-if" questions: "Even if features A are perturbed by ϵ, the classification remains $\mathbf{y}$." The key innovation is making semifactual explanations *targeted*: rather than identifying features sufficient for $\mathbf{y}$ against any alternative, we focus on features most relevant to specific class transitions. This targeting occurs in two stages: **(1)** We use sensitivity analysis ($\nabla_x f_t(x)$) to identify features most relevant to the $\mathbf{y} \rightarrow \mathbf{t}$ boundary. **(2)** We formally verify these features satisfy "even-if" robustness (can be perturbed by ϵ without flipping to $\mathbf{t}$). This design provides class-specific semifactual insights that are both efficient to compute ($\mathcal{O}(\log n)$ via binary search on the sensitivity ranking) and formally guaranteed (via reachability analysis).

Relation to Counterfactual Explanations. Our approach differs fundamentally from Counterfactual Explanations (CFEs) (Haufe et al. 2024; Montavon et al. 2018). CFEs identify minimal changes that flip decisions, answering "if-only" questions and explaining model fragility. VITAX answers "even-if" questions and explains robustness against specific alternatives.

It is crucial to clarify that while the visualizations in Figures 4 and 8 highlight features sensitive to a transition, these figures visualize the feature set A, a semifactual insight about which features can change without flipping the class, not an actualized counterfactual instance that successfully flips the class.

VITAX also differs fundamentally from verification-based robust CFEs (Jiang et al. 2024; Leofante and Lomuscio 2023). Robust CFEs verify the robustness of a *flip*: they find a counterfactual x_{cf} where $f(x_{cf}) = \mathbf{t}$, then verify perturbations around x_{cf} maintain the target class $\mathbf{t}$. The specification is: $\Phi_{\text{RCFE}} : \forall x' \in B_\epsilon(x_{cf}), f(x') = \mathbf{t}$. VITAX verifies the robustness of the *non-flip*: verifying that perturbations of features A at the original input x maintain the original class $\mathbf{y}$. The specification is: $\Phi_{\text{VITAX}} : \forall x' \in X_{\epsilon,A}, l_{y,A} > u_{t,A}$. These address complementary questions: *Robust CFEs:* "What change flips to $\mathbf{t}$, and is that flip stable?" *VITAX:* "What features preserve $\mathbf{y}$ against perturbations toward $\mathbf{t}$?"

Therefore, this granular, explanation-centric guarantee is not necessarily "weaker" than broader robustness guarantees but is differently focused, offering specific and verifiable insights into the mechanics of potential class transitions. Such understanding can be critical in applications where the conditions under which a decision *doesn't* change, despite pressures towards an alternative, are as important as understanding why it might flip.

Positioning in the Broader XAI Landscape. Beyond understanding the intrinsic nature of its guarantees, situating VITAX relative to established XAI paradigms further clarifies its contribution. VITAX's approach to generating formally verified, targeted explanations offers valuable perspectives when contrasted with other XAI paradigms, notably contrastive explanations and research into adversarial perturbations (Miller 2019). While traditional contrastive explanations often address "Why P rather than Q?" by showing general distinguishing features or minimal changes to achieve Q, VITAX provides a more specific insight: it identifies a minimal, formally verified feature subset A whose ϵ-perturbation is guaranteed *not* to flip the classification from $\mathbf{y}$ to $\mathbf{t}$. This offers a nuanced understanding of the boundary conditions for the specific $\mathbf{y} \to \mathbf{t}$ potential transition, explaining the semifactual robustness of $\mathbf{y}$ against that particular alternative based on the identified features in A. Similarly, concerning adversarial robustness, VITAX differs from methods seeking *any* minimal perturbation to cause *any* misclassification (Chakraborty et al. 2021).

Limitations. While VITAX demonstrates strong empirical performance and provides formal guarantees, several limitations should be noted. The computational cost of formal verification remains substantial, even with $\mathcal{O}(\log_2(N))$ oracle calls. Verification can require tens of seconds per sample for complex architectures, making VITAX most suitable for offline analysis or safety-critical applications where formal guarantees justify this investment; future work could investigate more efficient solver strategies or parallelized verification to reduce this overhead. The quality of explanations depends on the sensitivity heuristic; while our gradient-based approach performs well empirically, the framework does not provide automatic guidance for selecting appropriate ϵ values, requiring domain expertise. Adaptive ϵ selection strategies guided by the model's local decision geometry represent a promising direction. Our guarantee is inherently local to a specific input x, feature subset A, and target class $\mathbf{t}$, and as observed in Figure 13d in Appendix E.1, may not extend to robustness against all other classes k when decision boundaries are closely clustered; extending the framework to jointly certify robustness across multiple target classes is a natural next step. Finally, while semifactual explanations offer unique insights, further user studies with domain experts are needed to validate whether these targeted insights are actionable for model validation and trust calibration in practice.

Despite these limitations, VITAX represents a meaningful step toward trustworthy, targeted formal explanations for safety-critical AI systems, bridging the gap between scalable XAI and rigorous verification.

7 Conclusion and Future Work

We introduced VITAX, a formal XAI method that generates targeted semifactual explanations for deep learning models. VITAX identifies minimal feature subsets A and provides a Targeted ϵ-Robustness guarantee, formally certifying that perturbing A by ϵ does not flip the classification from $\mathbf{y}$ to a chosen target class $\mathbf{t}$. This explanation-centric guarantee offers precise, verified insights into the boundary conditions for specific class transitions—a semifactual understanding of why a classification persists against a particular alternative.

Our empirical evaluations across image classification (MNIST, GTSRB, EMNIST) and regression (TaxiNet) tasks demonstrate that VITAX achieves higher fidelity, greater minimality, and dramatically faster runtimes compared to existing formal and heuristic XAI methods, while maintaining formal guarantees. By bridging sensitivity-driven heuristics with reachability-based verification in $\mathcal{O}(\log_2(N))$ oracle calls, VITAX offers a practical and scalable approach to formally grounded, targeted explanations.

Several directions remain for future work. First, we plan to explore strategies for identifying collectively influential features that may not individually rank as most sensitive but jointly approach decision boundaries. Second, we aim to extend VITAX to diverse data modalities, including natural language and time-series data. Finally, in close collaboration with domain experts, we intend to apply VITAX to safety-critical domains such as medical image analysis, validating its practical utility for model debugging, trust calibration, and responsible AI deployment.

Acknowledgments

This work was supported in part by the National Science Foundation (NSF) under grants 2443803, 2220401, and 2427711 and the Defense Advanced Research Projects Agency (DARPA) under contract number FA8750-23-C-0518. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or NSF.

References

Z. An, T. T. Johnson, and M. Ma. 2024. "Formal logic enabled personalized federated learning through property inference." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 10. Vol. 38, 10882–10890.

G. Anderson, S. Pailoor, I. Dillig, and S. Chaudhuri. 2019. "Optimization and abstraction: a synergistic approach for analyzing neural network robustness." In: *Proceedings of the 40th ACM SIGPLAN conference on programming language design and implementation*, 731–744.

S. Aryal and M. T. Keane. 2023. "Even if explanations: Prior work, desiderata & benchmarks for semi-factual xai." *arXiv preprint arXiv:2301.11970*.

A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay. 2021. "A survey on adversarial attacks and defences." *CAAI Transactions on Intelligence Technology*, 6, 1, 25–45.

T. F. Chowdhury, V. M. H. Phan, K. Liao, N. Dong, M.-S. To, A. Hengel, J. Verjans, and Z. Liao. 2025. "Looking in the mirror: A faithful counterfactual explanation method for interpreting deep image classification models." *arXiv preprint arXiv:2509.16822*.

G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. 2017. "EMNIST: Extending MNIST to handwritten letters." In: *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2921–2926.

A. Darwiche and A. Hirth. Feb. 2020. *On The Reasons Behind Decisions*. en. (Feb. 2020). Retrieved May 16, 2024 from https://arxiv-org.proxy.library.vanderbilt.edu/abs/2002.09284v1.

R. Ehlers. 2017. "Formal verification of piece-wise linear feed-forward neural networks." In: *Automated Technology for Verification and Analysis: 15th International Symposium, ATVA 2017, Pune, India, October 3–6, 2017, Proceedings 15*. Springer, 269–286.

Y. Y. Elboher, E. Cohen, and G. Katz. 2022. "Neural network verification using residual reasoning." In: *Software Engineering and Formal Methods: 20th International Conference, SEFM 2022, Berlin, Germany, September 26–30, 2022, Proceedings*. Springer, 173–189.

R. R. Fernández, I. M. de Diego, J. M. Moguerza, and F. Herrera. 2022. "Explanation sets: A general framework for machine learning explainability." *Information Sciences*, 617, 464–481.

C. Ferrari, M. N. Muller, N. Jovanovic, and M. Vechev. 2022. "Complete verification via multi-neuron relaxation guided branch-and-bound." *arXiv preprint arXiv:2205.00263*.

FICO Community. 2019. *Explainable Machine Learning Challenge*. https://community.fico.com/s/explainable-machine-learning-challenge. Accessed: 2025-09-30. (2019).

A. Gainer-Dewar and P. Vera-Licona. 2017. "The minimal hitting set generation problem: algorithms and computation." *SIAM Journal on Discrete Mathematics*, 31, 1, 63–100.

T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. T. Vechev. 2018. "AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation." In: *SP*. IEEE Computer Society, 3–18. doi:10.1109/SP.2018.00058.

C. Geng, N. Le, X. Xu, Z. Wang, A. Gurfinkel, and X. Si. 2023. "Towards reliable neural specifications." In: *International Conference on Machine Learning*. PMLR, 11196–11212.

A. Ghorbani, A. Abid, and J. Zou. 2019. "Interpretation of neural networks is fragile." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33, 3681–3688.

R. Guidotti. 2024. "Counterfactual explanations and how to find them: literature review and benchmarking." *Data Mining and Knowledge Discovery*, 38, 5, 2770–2824.

N. Hashemi, S. Sasaki, D. M. Lopez, I. Oguz, M. Ma, and T. T. Johnson. 2025. "Probabilistic Robustness Analysis in High Dimensional Space: Application to Semantic Segmentation Network." *arXiv preprint arXiv:2509.11838*.

S. Haufe, R. Wilming, B. Clark, R. Zhumagambetov, D. Panknin, and A. Boubekki. 2024. "Explainable AI needs formal notions of explanation correctness." *arXiv preprint arXiv:2409.14590*.

X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, and X. Yi. 2020. "A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability." *Computer Science Review*, 37, 100270.

X. Huang, M. Kwiatkowska, S. Wang, and M. Wu. 2017. "Safety verification of deep neural networks." In: *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*. Springer, 3–29.

A. Ignatiev and J. Marques-Silva. 2021. "SAT-based rigorous explanations for decision lists." In: *Theory and Applications of Satisfiability Testing–SAT 2021: 24th International Conference, Barcelona, Spain, July 5-9, 2021, Proceedings 24*. Springer, 251–269.

A. Ignatiev, N. Narodytska, and J. Marques-Silva. 2019. "Abduction-based explanations for machine learning models." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 01. Vol. 33, 1511–1519.

J. Jiang, F. Leofante, A. Rago, and F. Toni. June 2023. "Formalising the Robustness of Counterfactual Explanations for Neural Networks." en. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37, 12, (June 2023), 14901–14909. doi:10.1609/aaai.v37i12.26740.

J. Jiang, F. Leofante, A. Rago, and F. Toni. 2024. "Robust counterfactual explanations in machine learning: A survey." *arXiv preprint arXiv:2402.01928*.

K. D. Julian, R. Lee, and M. J. Kochenderfer. 2020. "Validation of image-based neural network controllers through adaptive stress testing." In: *2020 IEEE 23rd international conference on intelligent transportation systems (ITSC)*. IEEE, 1–7.

A.-H. Karimi, B. Schölkopf, and I. Valera. Oct. 2020. *Algorithmic Recourse: from Counterfactual Explanations to Interventions*. en. arXiv:2002.06278 [cs, stat]. (Oct. 2020). Retrieved Sept. 25, 2023 from http://arxiv.org/abs/2002.06278.

G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. 2017. "Reluplex: An efficient SMT solver for verifying deep neural networks." In: *International Conference on Computer Aided Verification*. Springer, 97–117.

G. Katz, D. A. Huang, et al.. 2019. "The Marabou Framework for Verification and Analysis of Deep Neural Networks." In: *Computer Aided Verification*. Ed. by I. Dillig and S. Tasiran. Springer International Publishing, Cham, 443–452. ISBN: 978-3-030-25540-4.

E. Kenny and W. Huang. 2023. "The utility of "even if" semifactual explanation to optimise positive outcomes." *Advances in Neural Information Processing Systems*, 36, 52907–52935.

E. M. Kenny and M. T. Keane. 2021. "On generating plausible counterfactual and semi-factual explanations for deep learning." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 13. Vol. 35, 11575–11585.

H. Key. 2025. "A SAT-centered XAI method for Deep Learning based Video Understanding." *arXiv preprint arXiv:2503.23870*.

Y. LeCun, C. Cortes, and C. J. Burges. 2009. "The MNIST database of handwritten digits (2010)." *URL http://yann. lecun. com/exdb/mnist*, 7.

F. Leofante and A. Lomuscio. 2023. "Robust explanations for human-neural multi-agent systems with formal verification." In: *European Conference on Multi-Agent Systems*. Springer, 244–262.

P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis. 2020. "Explainable ai: A review of machine learning interpretability methods." *Entropy*, 23, 1, 18.

C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, M. J. Kochenderfer, et al.. 2021. "Algorithms for verifying deep neural networks." *Foundations and Trends® in Optimization*, 4, 3-4, 244–404.

D. M. Lopez, S. W. Choi, H.-D. Tran, and T. T. Johnson. 2023. "NNV 2.0: the neural network verification tool." In: *International Conference on Computer Aided Verification*. Springer, 397–412.

S. M. Lundberg and S.-I. Lee. 2017. "A Unified Approach to Interpreting Model Predictions." en. *Advances in neural information processing systems*, 30.

M. Ma, J. Gao, L. Feng, and J. Stankovic. 2020. "STLnet: Signal temporal logic enforced multivariate recurrent neural networks." *Advances in Neural Information Processing Systems*, 33, 14604–14614.

M. Ma, J. Stankovic, E. Bartocci, and L. Feng. 2021. "Predictive monitoring with logic-calibrated uncertainty for cyber-physical systems." *ACM Transactions on Embedded Computing Systems (TECS)*, 20, 5s, 1–25.

E. L. Malfa, R. Michelmore, A. M. Zbrzezny, N. Paoletti, and M. Kwiatkowska. Aug. 2021. "On Guaranteed Optimal Robust Explanations for NLP Models." en. In: vol. 3. ISSN: 1045-0823. (Aug. 2021), 2658–2665. doi:10.24963/ijcai.2021/366.

J. Marques-Silva and A. Ignatiev. June 2022. "Delivering Trustworthy AI through Formal XAI." en. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36, 11, (June 2022), 12342–12350. Number: 11. doi:10.1609/aaai.v36i11.21499.

T. Miller. 2019. "Explanation in artificial intelligence: Insights from the social sciences." *Artificial intelligence*, 267, 1–38.

G. Montavon, W. Samek, and K.-R. Müller. Feb. 2018. "Methods for Interpreting and Understanding Deep Neural Networks." en. *Digital Signal Processing*, 73, (Feb. 2018), 1–15. arXiv:1706.07979 [cs, stat]. doi:10.1016/j.dsp.2017.10.011.

M. N. Müller, G. Makarchuk, G. Singh, M. Püschel, and M. Vechev. 2022. "PRIMA: general and precise neural network certification via scalable convex hull approximations." *Proceedings of the ACM on Programming Languages*, 6, POPL, 1–33.

S. Paul, J. Yu, J. J. Dekker, A. Ignatiev, and P. J. Stuckey. 2024. "Formal Explanations for Neuro-Symbolic AI." *arXiv preprint arXiv:2410.14219*.

M. T. Ribeiro, S. Singh, and C. Guestrin. 2016a. "" Why should i trust you?" Explaining the predictions of any classifier." In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

M. T. Ribeiro, S. Singh, and C. Guestrin. Aug. 2016b. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. en. arXiv:1602.04938 [cs, stat]. (Aug. 2016). Retrieved Sept. 7, 2022 from http://arxiv.org/abs/1602.04938.

M. T. Ribeiro, S. Singh, and C. Guestrin. Apr. 2018. "Anchors: High-Precision Model-Agnostic Explanations." en. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32, 1, (Apr. 2018). doi:10.1609/aaai.v32i1.11491.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Feb. 2020. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization." en. *International Journal of Computer Vision*, 128, 2, (Feb. 2020), 336–359. arXiv:1610.02391 [cs]. doi:10.1007/s11263-019-01228-7.

A. Shih, A. Choi, and A. Darwiche. May 2018. *A Symbolic Approach to Explaining Bayesian Network Classifiers*. en. (May 2018). Retrieved May 16, 2024 from https://arxiv-org.proxy.library.vanderbilt.edu/abs/1805.03364v1.

A. Shrikumar, P. Greenside, and A. Kundaje. Oct. 2019. *Learning Important Features Through Propagating Activation Differences*. en. arXiv:1704.02685 [cs]. (Oct. 2019). Retrieved Jan. 20, 2023 from http://arxiv.org/abs/1704.02685.

A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje. May 2016. *Not Just a Black Box: Learning Important Features Through Propagating Activation Differences*. en. (May 2016). Retrieved Aug. 23, 2023 from https://arxiv.org/abs/1605.01713v3.

K. Simonyan, A. Vedaldi, and A. Zisserman. Apr. 2014. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. en. arXiv:1312.6034 [cs]. (Apr. 2014). Retrieved Sept. 12, 2022 from http://arxiv.org/abs/1312.6034.

G. Singh, T. Gehr, M. Püschel, and M. Vechev. 2019. "An abstract domain for certifying neural networks." *Proceedings of the ACM on Programming Languages*, 3, POPL, 1–30.

J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Apr. 2015. *Striving for Simplicity: The All Convolutional Net*. arXiv:1412.6806 [cs]. (Apr. 2015). doi:10.48550/arXiv.1412.6806.

J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. 2012. "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition." *Neural networks*, 32, 323–332.

M. Sundararajan, A. Taly, and Q. Yan. July 2017. "Axiomatic Attribution for Deep Networks." en. In: *Proceedings of the 34th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, (July 2017), 3319–3328. Retrieved Aug. 30, 2023 from https://proceedings.mlr.press/v70/sundararajan17a.html.

V. Tjeng, K. Xiao, and R. Tedrake. 2017. "Evaluating Robustness of Neural Networks with Mixed Integer Programming." *arXiv preprint arXiv:1711.07356*.

H.-D. Tran, S. Bak, W. Xiang, and T. T. Johnson. 2020. "Verification of deep convolutional neural networks using imagestars." In: *International conference on computer aided verification*. Springer, 18–42.

H.-D. Tran, D. Manzanas Lopez, P. Musau, X. Yang, L. V. Nguyen, W. Xiang, and T. T. Johnson. 2019. "Star-based reachability analysis of deep neural networks." In: *Formal Methods–The Next 30 Years: Third World Congress, FM 2019, Porto, Portugal, October 7–11, 2019, Proceedings 3*. Springer, 670–686.

H.-D. Tran, P. Musau, D. M. Lopez, X. Yang, L. V. Nguyen, W. Xiang, and T. T. Johnson. Oct. 2019. "Star-Based Reachability Analysis for Deep Neural Networks." In: *23rd International Symposium on Formal Methods (FM'19)*. Springer International Publishing, (Oct. 2019).

H.-D. Tran, N. Pal, D. M. Lopez, P. Musau, X. Yang, L. V. Nguyen, W. Xiang, S. Bak, and T. T. Johnson. Aug. 2021. "Verification of Piecewise Deep Neural Networks: A Star Set Approach with Zonotope Pre-Filter." *Form. Asp. Comput.*, 33, 4–5, (Aug. 2021), 519–545. doi:10.1007/s00165-021-00553-4.

T. Q. Tran, K. Fukuchi, Y. Akimoto, and J. Sakuma. June 2022. "Unsupervised Causal Binary Concepts Discovery with VAE for Black-Box Model Explanation." en. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36, 9, (June 2022), 9614–9622. doi:10.1609/aaai.v36i9.21195.

H. D. Wang, N. Khan, A. Chen, N. Sarkar, P. Wisniewski, and M. Ma. 2024. "MicroXercise: A Micro-Level Comparative and Explainable System for Remote Physical Therapy." In: *2024 IEEE/ACM Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE, 73–84.

S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana. 2018a. "Efficient formal safety analysis of neural networks." *Advances in neural information processing systems*, 31.

S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana. 2018b. "Formal security analysis of neural networks using symbolic intervals." In: *27th USENIX Security Symposium (USENIX Security 18)*, 1599–1614.
D. Wei, H. Wu, M. Wu, P.-Y. Chen, C. Barrett, and E. Farchi. 2023. "Convex bounds on the softmax function with applications to robustness verification." In: *International Conference on Artificial Intelligence and Statistics*. PMLR, 6853–6878.
M. Wicker, J. Heo, L. Costabello, and A. Weller. 2022. "Robust explanation constraints for neural networks." *arXiv preprint arXiv:2212.08507*.
H. Wu, C. Barrett, M. Sharif, N. Narodytska, and G. Singh. 2022. "Scalable verification of GNN-based job schedulers." *Proceedings of the ACM on Programming Languages*, 6, OOPSLA2, 1036–1065.
M. Wu, H. Wu, and C. Barrett. Dec. 2023. "VeriX: Towards Verified Explainability of Deep Neural Networks." en. *Advances in Neural Information Processing Systems*, 36, (Dec. 2023), 22247–22268. Retrieved May 5, 2024 from https://proceedings.neurips.cc/paper_files/paper/2023/hash/46907c2ff9fafd618095161d76461842-Abstract-Conference.html.
M. D. Zeiler and R. Fergus. Nov. 2013. *Visualizing and Understanding Convolutional Networks*. arXiv:1311.2901 [cs]. (Nov. 2013). doi:10.48550/arXiv.1311.2901.
T. Zelazny, H. Wu, C. Barrett, and G. Katz. 2022. "On optimizing back-substitution methods for neural network verification." In: *2022 Formal Methods in Computer-Aided Design (FMCAD)*. IEEE, 17–26.
H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel. 2018. "Efficient neural network robustness certification with general activation functions." *Advances in neural information processing systems*, 31.
S. Zhang, J. Campos, C. Feldmann, D. Walz, F. Sandfort, M. Mathea, C. Tsay, and R. Misener. 2024. "Optimizing over trained GNNs via symmetry breaking." *Advances in Neural Information Processing Systems*, 36.

A Overapproximation Methods for Reach Sets

A.1 Definition of Reach Set and Overapproximation

Given a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, the reach set $f(X)$ from a set of initial states $X \subseteq \mathbb{R}^n$ represents all possible states in $\mathbb{R}^m$ that the system can reach. An overapproximation $\hat{f}(X)$ of this reach set is then defined as:

$$\hat{f}(X) \supseteq f(X)$$

This definition ensures that $\hat{f}(X)$ contains all the values of $f(X)$, along with potentially additional values that provide a conservative estimate. Verifying Deep Neural Networks (DNNs) is an NP-Complete problem (Katz, Barrett, et al. 2017), so computing an overapproximation of the output set simplifies the problem, making it more feasible to compute for larger and more complex problems. Our method can compute an (output) reachable set Y of a DNN f, i.e., $Y = f(X)$, where Y can be computed exactly (sound and complete) or overapproximated (sound and incomplete).

$$\begin{aligned} complete &: \forall y \in Y,\ \exists x \in X \ \mid\ y = f(x) \\ sound &: \forall x \in X,\ \exists y \in Y \ \mid\ y = f(x) \end{aligned}$$

A.2 Methods for Computing Reach Sets

Several methods exist for the overapproximation of reach sets, each with its advantages, computational implications, and limitations. We are using NNV, making use of star sets, an abstraction-based reachability tool with the following available methods:

Exact Star (Sound and Complete). The Exact Star method computes the exact reach set $f(X)$. However, it can be computationally expensive and is often infeasible for high-dimensional systems or very large NNs (H.-D. Tran, Musau, et al. 2019). This method is able to compute the exact reach set for linear and piece-wise linear layers (e.g., ReLU, Convolution, BatchNorm, etc.).

Approximate (Approx) Star (Sound). The Approximate Star method uses star sets to provide a close but approximate representation of the reach set. It simplifies the computation by approximating some of the constraints that define

the reach set on piece-wise linear layers. This method strikes a balance between computational efficiency and accuracy (H.-D. Tran, Pal, et al. 2021).

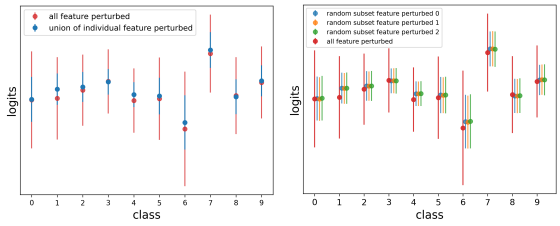
Relaxed Star (Sound). The Relaxed Star method involves increasing the approximation by introducing a relaxation parameter β, which defines how much the approximation can deviate from the actual reach set:

$$\text{Relaxed Star}(\beta) \supseteq \text{Approximate Star} \supseteq \text{Exact Star}$$

This is applicable to piece-wise linear layers, where the β parameter determined the number of constraints to solve using linear programming (LP), and how many to overapproximate. Higher values of β increase the size of the overapproximation (larger number of overapproximated LP problems), thereby reducing the computational load at the cost of accuracy. A relax-star method with $\beta = 0$ is equivalent to approx-star.

B Proofs for Section Soundness of Explanations 3.3

Additional to the proofs, we illustrate the inclusiveness of Remark 1 with the output space in logits as shown in Figure 11 and a toy example where NN has 3 inputs and 2 outputs, and we perturbed all 3 inputs (red), 2 (yellow) or 1 (green) as shown in Figure 12



(a) The range of union of individual feature perturbed (blue) versus all feature perturbed (red), based on Lemma 1.

(b) The range of subset of features (that are subset of the whole set) perturbed (blue, orange, and green) versus all feature perturbed (red), based on Remark 1.

Fig. 11. Empirical evaluation on input space projected into one dimensional space (in output space). As shown in both of the subfigures, it shows the empirical evaluation of Lemma 1 and Remark 1, which satisfies what we claim. We evaluated on both approx star and exact star reachability method.

B.1 Proof for Lemma 1

Given a model $f : \mathbb{R}^n \to \mathbb{R}^m$, for a point $x \in \mathbb{R}^n$, define the set $X_{\epsilon,i}$ for $i \in \{1, 2, \ldots, n\}$ as:

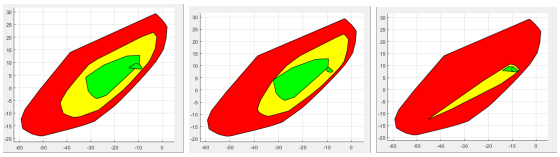
$$X_{\epsilon,i} = \{x' \in \mathbb{R}^n : \|x' - x\|^i \leq \epsilon\} \text{ (perturbation on } i^{th} \text{ feature only)}$$

Let X_ϵ be defined as:

$$X_\epsilon = \{x' \in \mathbb{R}^n : \|x' - x\|_\infty \leq \epsilon\}$$

It holds that $Y_\epsilon = f(X_\epsilon) \supseteq \bigcup_{i=1}^n f(X_{\epsilon,i})$.

Proof. To prove that $f(X_\epsilon) \supseteq \bigcup_{i=1}^n f(X_{\epsilon,i})$, we must show that any element belonging to the set $\bigcup_{i=1}^n f(X_{\epsilon,i})$ also belongs to the set $f(X_\epsilon)$.



(a) Feature of Red: [1, 1, 1] Yellow: [1, 1, 1] Green: [1, 0, 0] (b) Feature of Red: [1, 1, 1] Yellow: [1, 0, 1] Green: [0, 0, 1] (c) Feature of Red: [0, 1, 0] Yellow: [1, 1, 1] Green: [0, 1, 0]

Fig. 12. Soundness and inclusiveness visualization on a toy example. We evaluate the feature space projected into 2 dimension output space. The individual subcaptions show the vector of features perturbed.

Let y_{out} be an arbitrary element in $\bigcup_{i=1}^{n} f(X_{\epsilon,i})$. By the definition of the union of sets, if $y_{out} \in \bigcup_{i=1}^{n} f(X_{\epsilon,i})$, then there exists at least one index $k \in \{1, 2, \ldots, n\}$ such that $y_{out} \in f(X_{\epsilon,k})$.

Since $y_{out} \in f(X_{\epsilon,k})$, there exists an input vector $x^* \in X_{\epsilon,k}$ such that $y_{out} = f(x^*)$.

According to our definition of $X_{\epsilon,k}$:

(1) For all indices $j \neq k$, the j-th component of x^* is $x_j^* = x_j$.
(2) For the k-th index, the k-th component of x^* is x_k^*, and it satisfies $|x_k^* - x_k| \leq \epsilon$.

Now, we need to show that this x^* is also an element of X_ϵ. A vector x' is in X_ϵ if $\|x' - x\|_\infty \leq \epsilon$. This means that for all component indices $j \in \{1, \ldots, n\}$, the condition $|x'_j - x_j| \leq \epsilon$ must be satisfied.

Let's examine the absolute differences of the components of x^* and x:

- For any index $j \neq k$: $|x_j^* - x_j| = |x_j - x_j| = 0$.
- For the index $j = k$: $|x_k^* - x_k| \leq \epsilon$.

Since $\epsilon \geq 0$ (as it represents a magnitude of perturbation), the condition $0 \leq \epsilon$ is true. Thus, for every index $j \in \{1, \ldots, n\}$, we have $|x_j^* - x_j| \leq \epsilon$. This implies that the L_∞ norm of the difference $x^* - x$ satisfies:

$$\|x^* - x\|_\infty = \max_{j \in \{1,\ldots,n\}} |x_j^* - x_j| \leq \epsilon$$

Therefore, x^* is an element of X_ϵ. Since $x^* \in X_\epsilon$ and $y_{out} = f(x^*)$, it follows by the definition of $f(X_\epsilon)$ that $y_{out} \in f(X_\epsilon)$.

As y_{out} was an arbitrary element of $\bigcup_{i=1}^{n} f(X_{\epsilon,i})$, we have shown that every element of $\bigcup_{i=1}^{n} f(X_{\epsilon,i})$ is also an element of $f(X_\epsilon)$. Therefore, $\bigcup_{i=1}^{n} f(X_{\epsilon,i}) \subseteq f(X_\epsilon)$, which is equivalent to the statement $f(X_\epsilon) \supseteq \bigcup_{i=1}^{n} f(X_{\epsilon,i})$.

This completes the direct proof.

□

B.2 Proof for Remark 1

Proof. To prove that $Y_{\epsilon,d} \subseteq Y_{\epsilon,l}$, it is sufficient to show that $X_{\epsilon,d} \subseteq X_{\epsilon,l}$. If this holds, then for any $y \in Y_{\epsilon,d}$, there exists an $x^* \in X_{\epsilon,d}$ such that $y = f(x^*)$. Since $x^* \in X_{\epsilon,d}$ and $X_{\epsilon,d} \subseteq X_{\epsilon,l}$, x^* must also be in $X_{\epsilon,l}$. Therefore, $y = f(x^*) \in f(X_{\epsilon,l}) = Y_{\epsilon,l}$, which proves $Y_{\epsilon,d} \subseteq Y_{\epsilon,l}$.

We will prove $X_{\epsilon,d} \subseteq X_{\epsilon,l}$ by induction on $m = |l \setminus d|$, which is the number of elements in l that are not in d. Since $d \subset l$ and it's implied by the setup ($|d| < |l|$) that l is strictly larger than d, we have $m \geq 1$.

Base Case: $m = 1$. This means l contains exactly one element more than d, and $d \subset l$. So, l can be written as $l = d \cup \{j_0\}$ for some feature index $j_0 \notin d$. Let $x^* \in X_{\epsilon,d}$. By definition of $X_{\epsilon,d}$:

(1) $(x^*)_k = x_k$ for all $k \notin d$.

(2) $|(x^*)_k - x_k| \leq \epsilon$ for all $k \in d$.

We want to show that $x^* \in X_{\epsilon,l} = X_{\epsilon,d\cup\{j_0\}}$. For this, x^* must satisfy:

(a) $(x^*)_k = x_k$ for all $k \notin (d \cup \{j_0\})$.

(b) $|(x^*)_k - x_k| \leq \epsilon$ for all $k \in (d \cup \{j_0\})$.

Verify these conditions:

For condition (a): If $k \notin (d \cup \{j_0\})$, then $k \notin d$ and $k \neq j_0$. Since $k \notin d$, by property (1) of x^*, we have $(x^*)_k = x_k$. So, condition (a) is satisfied.

For condition (b): Consider $k \in (d \cup \{j_0\})$. If $k \in d$: By property (2) of x^*, $|(x^*)_k - x_k| \leq \epsilon$. This part of condition (b) is satisfied. If $k = j_0$: Since $j_0 \notin d$, by property (1) of x^*, we have $(x^*)_{j_0} = x_{j_0}$. Therefore, $|(x^*)_{j_0} - x_{j_0}| = |x_{j_0} - x_{j_0}| = 0$. Since $\epsilon \geq 0$, $0 \leq \epsilon$. This part of condition (b) is also satisfied. Thus, $x^* \in X_{\epsilon,l}$. Since x^* was an arbitrary element of $X_{\epsilon,d}$, we have $X_{\epsilon,d} \subseteq X_{\epsilon,l}$ for $m = 1$. (The specific example where $|d| = 1$ (e.g., $d = \{i\}$) and $|l| = 2$ (e.g., $l = \{i, j\}$ with $i \neq j$) is an instance of this base case, as $m = |l \setminus d| = 1$.)

Inductive Hypothesis (IH): Assume that for some integer $k \geq 1$, the statement holds for $m = k$. That is, for any two sets of feature indices D' and L' such that $D' \subset L'$ and $|L' \setminus D'| = k$, we have $X_{\epsilon,D'} \subseteq X_{\epsilon,L'}$.

Inductive Step: We need to prove the statement for $m = k + 1$. Let d and l be sets of feature indices such that $d \subset l$ and $|l \setminus d| = k + 1$. Since $|l \setminus d| = k + 1 \geq 1$ (as $k \geq 1$, $k + 1 \geq 2$; if $k = 0$ was allowed in IH, $k + 1 \geq 1$), we can pick an element $j^* \in l \setminus d$. Define an intermediate set $l_{int} = l \setminus \{j^*\}$.

Now consider the relationship between d and l_{int}: Since $d \subset l$ and $j^* \in (l \setminus d)$, d does not contain j^*. Thus, $d \subseteq l \setminus \{j^*\}$, so $d \subseteq l_{int}$. The elements in l_{int} that are not in d are $(l \setminus \{j^*\}) \setminus d = (l \setminus d) \setminus \{j^*\}$. So, $|l_{int} \setminus d| = |(l \setminus d) \setminus \{j^*\}| = (k + 1) - 1 = k$.

We have $d \subseteq l_{int}$ and $|l_{int} \setminus d| = k$.

- If $k = 0$: This would mean $l_{int} = d$. Then $|l \setminus d| = 1$. This situation is covered by the Base Case directly, where $X_{\epsilon,d} \subseteq X_{\epsilon,l}$ was proven. (Note: if IH assumes $k \geq 1$, then this $k = 0$ sub-case for $l_{int} \setminus d$ won't occur here as we're proving for $k + 1 \geq 2$).
- If $k \geq 1$: Then $d \subset l_{int}$ (if $d \neq l_{int}$) or $d = l_{int}$ (if $k = 0$, but here we assume $k \geq 1$ for IH). If $d \subset l_{int}$ and $|l_{int} \setminus d| = k$, by the Inductive Hypothesis, we have $X_{\epsilon,d} \subseteq X_{\epsilon,l_{int}}$. If $d = l_{int}$ (which means $k = 0$), then $X_{\epsilon,d} = X_{\epsilon,l_{int}}$ is trivially true. For our IH defined with $k \geq 1$, we must have $d \subset l_{int}$ (unless d was empty and $k = 0$ for $|l_{int} \setminus d|$), and $X_{\epsilon,d} \subseteq X_{\epsilon,l_{int}}$ by IH.

To simplify, since the IH is stated for $k \geq 1$: For $|l_{int} \setminus d| = k \geq 1$, we have $D' = d$ and $L' = l_{int}$, so $X_{\epsilon,d} \subseteq X_{\epsilon,l_{int}}$ by IH. If $k = 0$ (meaning $|l \setminus d| = 1$), this was the base case. The structure of induction ensures k from IH is ≥ 1.

So, $X_{\epsilon,d} \subseteq X_{\epsilon,l_{int}}$ (either because $d = l_{int}$ if $k = 0$ which corresponds to $|l \setminus d| = 1$ covered by base case, or by IH if $k \geq 1$).

Now consider l_{int} and l. We have $l_{int} \subset l$ (since $j^* \in l$ and $j^* \notin l_{int}$) and $l = l_{int} \cup \{j^*\}$, with $j^* \notin l_{int}$. Thus, $|l \setminus l_{int}| = 1$. By our Base Case ($m = 1$), we have $X_{\epsilon,l_{int}} \subseteq X_{\epsilon,l}$.

Combining these results: $X_{\epsilon,d} \subseteq X_{\epsilon,l_{int}}$ and $X_{\epsilon,l_{int}} \subseteq X_{\epsilon,l}$. By the transitivity of set inclusion, $X_{\epsilon,d} \subseteq X_{\epsilon,l}$. This completes the inductive step for $m = k + 1$.

Therefore, by the principle of mathematical induction, for all sets of feature indices d, l such that $d \subset l$ (which implies $m = |l \setminus d| \geq 1$), it holds that $X_{\epsilon,d} \subseteq X_{\epsilon,l}$.

As established at the beginning of the proof, if $X_{\epsilon,d} \subseteq X_{\epsilon,l}$, then $Y_{\epsilon,d} = f(X_{\epsilon,d}) \subseteq f(X_{\epsilon,l}) = Y_{\epsilon,l}$. This concludes the proof. □

B.3 Proof for Theorem 3.1 (Soundness of ViTaX Explanation)

Proof. Let A_{found} be the feature subset (specifically, the prefix $\pi[0:u]$ of the heuristically ranked features π) returned by Algorithm 1 (ViTaX). We need to show two things:

(1) A_{found} satisfies Targeted ϵ-Robustness as defined in Definition 2.2 (which involves $l_{\mathbf{y},A_{found}} > u_{\mathbf{t},A_{found}}$ and the conditions for other classes $\mathbf{k}$).
(2) A_{found} is the largest prefix of π for which this property holds.

The formal property Φ that Algorithm 1 verifies for a candidate prefix $d = \pi[0:u]$ using the solver $\mathcal{V}$ is:

$$\Phi(d) = \left(l_{\mathbf{y},d} > u_{\mathbf{t},d}\right) \wedge \left(\forall \mathbf{k} \in \{1,\ldots,m\}, \mathbf{k} \neq \mathbf{y} \wedge \mathbf{k} \neq \mathbf{t} \implies u_{\mathbf{t},d} > u_{\mathbf{k},d}\right)$$

This $\Phi(d)$ directly corresponds to the conditions required for d to be a Targeted ϵ-Robust Explanation as per Definition 2.2, along with ensuring the target class $\mathbf{t}$ is more prominent than other non-original classes $\mathbf{k}$ within the context of the perturbed subset d.

The reachability solver $\mathcal{V}$ is assumed to be:

- **Sound (for $\mathcal{V}$):** If $\mathcal{V}$ returns 'FLAG = True' for $\Phi(d)$, then $\Phi(d)$ indeed holds for all $x' \in X_{\epsilon,d}$.
- **Complete (for $\mathcal{V}$):** If $\Phi(d)$ holds for all $x' \in X_{\epsilon,d}$, then $\mathcal{V}$ returns 'FLAG = True' for $\Phi(d)$.

1. Proving A_{found} satisfies Targeted ϵ-Robustness:

Algorithm 1 employs a binary search strategy to find a candidate set. The algorithm stores a candidate $A_{candidate} \leftarrow d$ when 'FLAG = True' (i.e., $\Phi(d)$ is satisfied) and then attempts to find a larger robust subset by updating $I \leftarrow u + 1$. The final set A_{found} returned by the algorithm is the largest prefix $d = \pi[0:u]$ for which $\mathcal{V}$ returned 'FLAG = True'.

Since $\mathcal{V}$ returned 'FLAG = True' for A_{found} (otherwise it would not have been selected as the final $A_{candidate}$ that becomes A_{found}), and $\mathcal{V}$ is sound, it directly implies that the property $\Phi(A_{found})$ holds. As $\Phi(A_{found})$ encompasses the conditions of Definition 2.2, A_{found} is an explanation that satisfies Targeted ϵ-Robustness.

2. Proving A_{found} is the largest such prefix of π:

The binary search systematically explores prefixes of π. Let $A_{found} = \pi[0:u_{final}]$.

- By construction, $\Phi(A_{found})$ holds.
- Consider any prefix $\pi[0:u']$ where $u' > u_{final}$. If such a prefix also satisfied Φ, the binary search logic (specifically, the update $I \leftarrow u + 1$ when a robust subset is found, aiming to extend it) would have led to $A_{candidate}$ being updated to this larger prefix, or the search continuing in a range that includes u'. The algorithm terminates when $I > J$, and A_{found} is the $A_{candidate}$ corresponding to the largest u for which Φ was found to hold.
- If we consider the specific feature $\pi[u_{final}+1]$ (the feature immediately following the last feature in A_{found} according to the heuristic ranking), including it to form $A' = \pi[0:u_{final}+1]$ must have resulted in $\mathcal{V}$ returning 'FLAG = False' for $\Phi(A')$. If $\mathcal{V}$ is complete, its returning 'FLAG = False' means $\Phi(A')$ indeed does not hold.

Therefore, the binary search procedure, relying on the soundness and completeness of the solver $\mathcal{V}$ to correctly evaluate Φ for each tested prefix, ensures that A_{found} is the largest prefix of π for which Targeted ϵ-Robustness (as embodied by Φ) holds.

Thus, the explanation A_{found} generated by ViTaX satisfies Targeted ϵ-Robustness and is the maximal such explanation according to the heuristic feature ordering π. This completes the proof. □

C Baselines and Time Complexity

Brute Force Baseline. In principle, if we were to exhaustively test *every* possible combination of N features (i.e., every subset), the time complexity would be $\mathcal{O}\left(2^N \cdot \mathcal{V}(N)\right)$. However, due to computational constraints, we approximate this brute-force approach by sampling a random ordering of the N features multiple times (e.g., 100 runs). For each run, we binarily select features *in the random order* until the verifier considers its robustness (one additional feature will be robust). We record fidelity for each run and select the best result over these 100

trials. While still time-consuming, this procedure is vastly simpler than enumerating 2^N subsets and serves as a straightforward benchmark for comparison with more advanced explanation methods.

Complexities.

- **VITAX**: $\mathcal{O}\big(\log_2(N) \cdot \mathcal{V}(N)\big)$.
- **VeriX**: $\mathcal{O}\big(N \cdot \mathcal{V}(N)\big)$.
- **Brute Force (theoretical)**: $\mathcal{O}\big(2^N \cdot \mathcal{V}(N)\big)$.

D Baseline Adaptation Methodology

To facilitate a fair comparison with the semifactual explanations generated by VITAX, the baseline methods LIME (Ribeiro et al. 2016b) and Anchors (Ribeiro et al. 2018) were adapted from their standard implementations. The core objective was to shift these methods from generating factual explanations (explaining "why this prediction occurred") to generating probabilistic semifactual explanations (explaining "which minimal features are sufficient to maintain this prediction").

D.1 Conceptual Framework

The adaptation aims to identify a minimal subset of features (A_{prob}) such that, when these features are fixed, the model's prediction remains the original class (y) with high probability ($P \geq \tau$, where τ is a precision threshold), even when the remaining features ($\neg A_{prob}$) are randomized or perturbed.

D.2 Adapting Anchors

The original Anchors algorithm is inherently aligned with semifactual reasoning, as it seeks a minimal set of conditions (an "anchor") sufficient to support the current prediction. We characterize this as constructive sufficiency, where the explanation is built up greedily to find the minimal conditions required to maintain the prediction.

Methodology. We formalized the Anchors implementation to explicitly optimize for semifactual precision:

(1) **Greedy Optimization:** The algorithm iteratively adds features to the candidate anchor set.
(2) **Precision Thresholding:** At each step, the precision is evaluated by sampling perturbations in the non-anchor features (see Algorithmic Details below). The search continues until the predefined high precision threshold τ (e.g., 0.95) is met.
(3) **Output:** The minimal feature set that satisfies the precision threshold.

This formalization ensures that the output of Anchors is explicitly a statement of probabilistic sufficiency: "If these features are present, the prediction persists with probability P."

D.3 Adapting LIME

The original LIME algorithm learns a local linear model to approximate the decision boundary, identifying features that most contribute to the prediction. This is fundamentally an attribution method. To generate semifactual insights, we extended LIME with a secondary analysis phase

Methodology: We implemented a two-stage process:

(1) **Stage 1 (Feature Importance Learning):** Standard LIME is executed to learn the local linear model and determine the importance ranking of the features.
(2) **Stage 2 (Minimal Sufficiency Search):** A secondary greedy search is performed using the learned importance scores. This search asks: "Which combination of the most important features is minimally sufficient to preserve the prediction?"

(3) **Stability Validation:** Similar to the Anchors adaptation, the sufficiency of the subset is validated by sampling perturbations in the remaining features until the precision threshold τ is met.

This adaptation shifts LIME from merely identifying contributors to identifying the minimal necessary components for prediction stability, guided by the local approximation.

D.4 Algorithmic Details

The core of both adaptations is the sufficiency search (utilized in Anchors optimization and LIME Stage 2). This involves a greedy selection process and a probabilistic estimation of precision.

Precision Estimation (Stability Validation): To estimate the precision of a candidate feature set A_{cand}, we use Monte Carlo sampling:

(1) Generate N samples (e.g., $N = 500$).
(2) For each sample, the features in A_{cand} are fixed to their original values.
(3) The remaining features ($\neg A_{cand}$) are randomized (e.g., using Bernoulli sampling or replacing with baseline values).
(4) The model prediction is evaluated on the perturbed sample.
(5) Precision is calculated as the fraction of the N samples where the prediction remains the original class y.

D.5 Nature of the Guarantee

It is crucial to note that the guarantees provided by these adapted methods are probabilistic and empirical. They rely on sampling to estimate the stability of the prediction. Unlike VITAX, which uses formal reachability analysis to verify the entire continuous perturbation space, these methods cannot guarantee behavior between the samples and may miss adversarial counterexamples within the perturbation space.

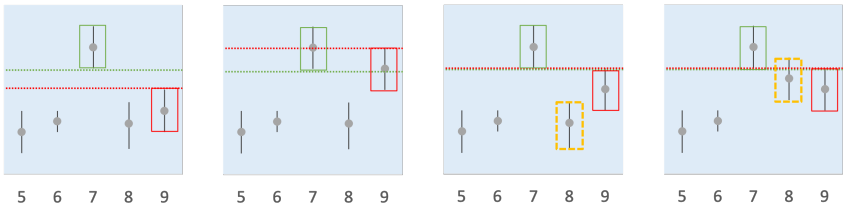
E Evaluation Cont.

To validate the generalizability and robustness of our methodologies, we extended our evaluations to include diverse datasets: GTSRB, the EMNIST Letters, and the TaxiNet dataset. These datasets were selected to represent a broad range of challenges in image recognition, from traffic sign identification to character recognition and autonomous aircraft taxiing scenarios. For each dataset, we applied the same experimental protocols to ensure consistency in comparison. The GTSRB dataset helped us assess the effectiveness of our approach in recognizing and interpreting traffic signs under various lighting and occlusion conditions. In EMNIST Letters, our focus was on evaluating the capability to discern and classify distinct alphabetical characters, which exhibit subtle variations. Meanwhile, TaxiNet provided a unique challenge in navigating complex taxiway environments, testing the adaptability of our algorithms to spatial and contextual variability.

VITAX evaluates the GTSRB classification dataset with CNN model (accuracy 85.35% and structure in Table 7) on all 43 classes. The input image size is $28 \times 28 \times 3$. We use a perturbation magnitude of $\epsilon = \frac{25}{255}$ to generate counterfactual examples across all classes from single sample images. This approach helps explore the targeted explanatory power of perturbations in identifying and delineating class-specific responses.

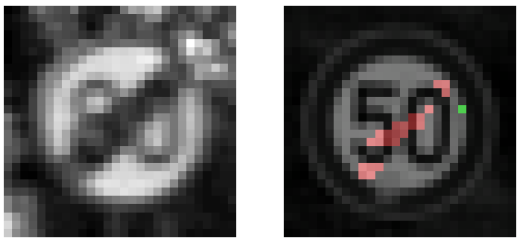
E.1 Not Robust to Other Classes k

Robustness to Some Classes. We train on different combination of number of classes in order to find the optimal way of robustness to **t** and to **k**, as shown in Figure 13c. One thing that we notice is that though there are cases where we are not able to find **t**-targeted explanation that are also robust to all **k**. For instance, labeled-7 sample from MNIST goes to 9 as a targeted explanation. But upper bound of other classes, such as 8, overlaps with 9,



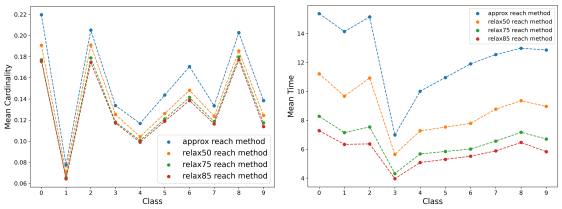
(a) robust (not at the boundary) (b) unrobust to **t** (c) robust to **t** and to **k** (d) robust to **t**, not to **k**

Fig. 13. An example of potentially not robust to other classes k in projected range of logits. Green box is label **y**, red box is target **t**, and yellow box is other classes **k**. In this case, we are arguing, given our Equation 1 and 2, there are potentially cases in (c) and (d). The yellow box **k** could cross the boundary while the subset of feature is robust to **t**.



(a) Speed Limit 80 End: the target we try to perturb to (b) The perturbed of 50 to Speed Limit 80 End

Fig. 14. GTSRB Explanation on 50 to Speed Limit 80 End

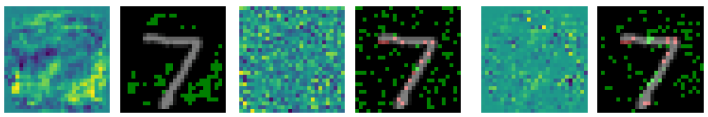


(a) Cardinality for Each Class (b) Time (sec) for Each Class

Fig. 15. X-axis represents the class from 0 to 9 and y-axis represents the mean cardinality value as percentage as shown in 15a and mean time in 15b.

as shown in 13d. So this subset of features would be not robust for other classes but only for class 7. We argue though this might not be most robust or explainable features, we believe there is some usefulness in it.

As shown in Figure 14, the image we are perturbing is '50' and give us very interesting result. Though in the robustness solver, it also returns us that the targeted explanation it generated not robust to image of speed limit '60', but the explanation shows the red line to attempt to find the targeted explanation for speed limit 80 end.



(a) Saliency (Proposed) (b) Random (c) IG

Fig. 16. VITAX with different heuristic functions. For each pair, the left side shows the output of the heuristic function, and the right side shows the corresponding VITAX explanations generated using the heuristic function's output.

Table 5. The table presents the average percentage of cardinality, illustrating the proportion of features that significantly contribute to class transitions under ϵ-perturbation. The columns represent the original sample classes $\mathbf{y}$, while the rows indicate the target explanation classes $\mathbf{t}$. Notably, the '-' symbol denotes cases where the target class is the same as the original class, which are purposefully excluded as they do not necessitate an explanation.

	Target Class $\mathbf{t}$									
Class $\mathbf{y}$	0	1	2	3	4	5	6	7	8	9
0	-	26.4%	21.3%	22.1%	26.8%	24.9%	21.4%	21.8%	24.5%	22.6%
1	11.0%	-	9.6%	7.6%	8.6%	8.7%	7.4%	5.8%	8.3%	8.4%
2	16.0%	13.3%	-	12.7%	18.8%	18.9%	18.0%	11.8%	17.1%	18.8%
3	17.1%	20.6%	14.6%	-	20.9%	13.3%	20.0%	19.7%	14.2%	14.5%
4	16.2%	17.7%	16.2%	17.5%	-	16.2%	14.9%	15.5%	17.0%	9.6%
5	10.7%	12.5%	11.4%	6.3%	13.8%	-	9.8%	12.4%	9.2%	10.2%
6	11.6%	18.4%	15.9%	17.4%	16.2%	13.0%	-	17.8%	16.0%	17.1%
7	18.6%	16.7%	14.6%	12.3%	17.6%	21.3%	24.4%	-	18.9%	11.5%
8	15.6%	18.2%	10.9%	10.8%	18.9%	15.4%	16.5%	15.0%	-	13.5%
9	16.3%	17.5%	17.1%	17.0%	10.0%	16.7%	18.6%	10.0%	17.9%	-

F Impact of Approximation Methods on Reachability

We further analyze the role of different approximation methods in solvers for reachability analysis. Interestingly, we classify an unknown subset of features as unrobust (meaning shrinking down the subset), which differs from (M. Wu et al. 2023), where important (ϵ-robust) features are considered irrelevant, and unknown and unrobust features are seen as important. We use several sound and incomplete reach methods, including approx star, relax star (50% relaxed), relax star (75%), and relax star (85%). Figure 15 shows that as the approximation becomes more relaxed (up to relax 85%), the cardinality percentage decreases. *Intuitively, this decrease indicates that some subsets are important and sensitive to perturbation, influencing the output significantly, while other subsets of features do not contribute to robustness.*

G Cardinality Benchmarking

As show in Table 5, by using our method, we show that VITAX generates useful cardinality for evaluating and benchmarking other explanation methods, and it helps quantify feature importance variability across class transitions, indicating the proportion of features significantly contributing to explanations under ϵ-perturbation. Table 5 shows original sample classes on the vertical axis and target classes $\mathbf{t}$ on the horizontal axis, with a dashed line denoting the absence of explanation of itself. We randomly selected 100 correctly predicted test samples and calculated the average percentage of cardinality in MNIST using the same model, quantifying feature importance

variability across class transitions. For example, explaining from class '0' to target classes '2', '3', '6', '7', and '9' shows consistent cardinality around 22%, while transitioning from class '1' to other target classes shows a reduction from 22% to around 8%. The use of cardinality as a benchmark promotes a standardized approach for researchers, aiding in the selection and evaluation of features influenced by ϵ-perturbation. *This reduction suggests that cardinality proportion is vital for understanding sample robustness across classes, making it an potentially effective metric for evaluating and benchmarking explanation methods.*

H Model Specifications

Table 6. Structure for the MNIST MLP

Type	Parameters	Activation
Input	$28 \times 28 \times 1$	-
Flatten	-	-
Fully Connected	128	ReLU
Fully Connected	64	ReLU
Fully Connected	10	softmax

Table 7. Structure for CNN

Type	Parameters	Activation
Input	$28 \times 28 \times 3(1)$	-
Convolution	$3 \times 3 \times 8$	ReLU
AVG Pool	2×2	-
Flatten	-	-
F.C.	43(9)(26)	softmax

Table 8. Structure of the TaxiNet MLP

Type	Parameters	Activation
Input	$27 \times 54 \times 1$	-
Flatten	-	-
Fully Connected	20	ReLU
Fully Connected	1	-

Table 9. Structure of the small MLP

Type	Parameters	Activation
Input	$28 \times 28 \times 1$	-
Flatten	-	-
Fully Connected	15	ReLU
Fully Connected	10	softmax

Table 10. Structure of the MLP Dense

Type	Parameters	Activation
Input	$h \times w \times c$	-
Flatten	-	-
Fully Connected	10	ReLU
Fully Connected	10	ReLU
Fully Connected	# classes	softmax

Table 11. Structure of the MLP Dense Large

Type	Parameters	Activation
Input	$h \times w \times c$	-
Flatten	-	-
Fully Connected	30	ReLU
Fully Connected	30	ReLU
Fully Connected	# classes	softmax

Table 12. Structure of the CNN Dense

Type	Parameters	Activation
Input	$h \times w \times c$	-
Convolution	$3 \times 3 \times 4$	ReLU
Convolution	$3 \times 3 \times 4$	ReLU
Flatten	-	-
Fully Connected	20	ReLU
Fully Connected	# classes	softmax

Received 02 November 2025; accepted 03 March 2026