Brittle Features of Device Authentication

Washington Garcia, Joseph Choi, Kevin Butler, Somesh Jha†

University of Florida
University of Wisconsin-Madison†

AFOSR CoE Annual Meeting
October 15, 2019
Duke University

Introduction

Systems often communicate within contested environments.

- Standard networking protocols offer a spoofing attack surface
- Open problem within network security
- Mitigation: Device Authentication

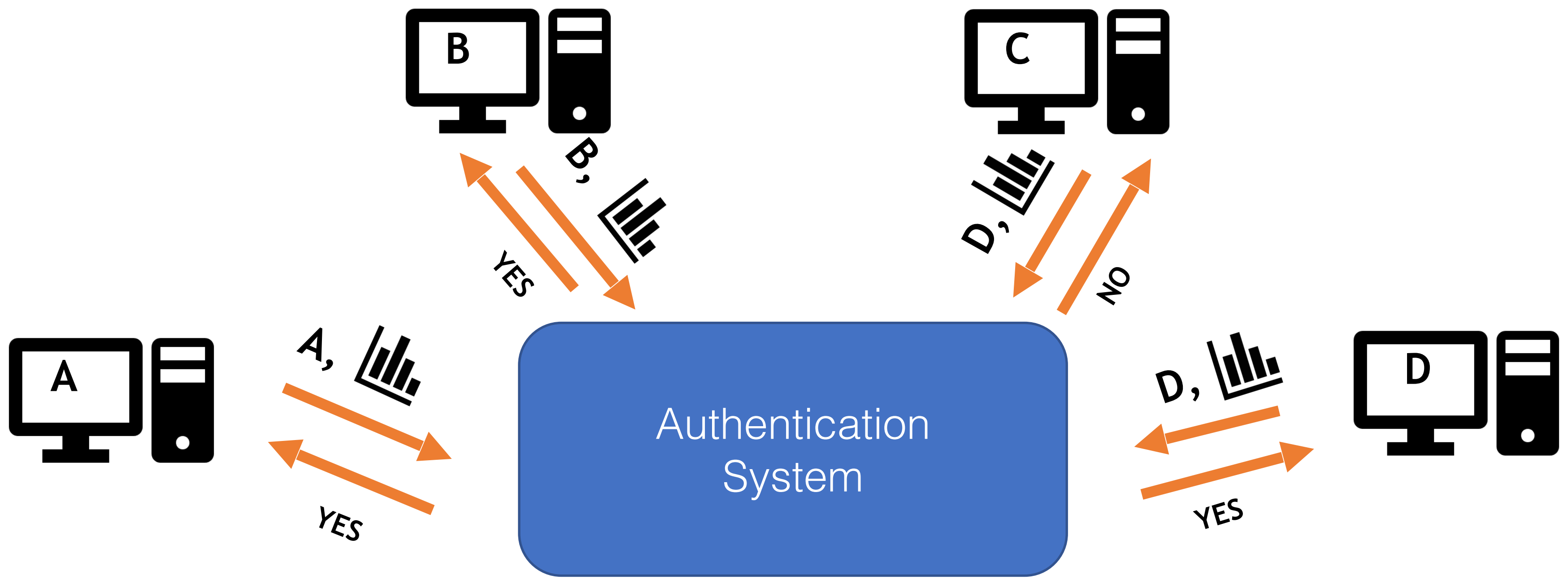
Device Authentication

Grant features or capabilities of a contested network to only **certain** devices

Define the Authentication System (AS):
Performs attestation of (device, sample) pairs.

Previous work: implement the attestation using machine learning.

- Map device samples → devices
- Return YES if matching, NO otherwise



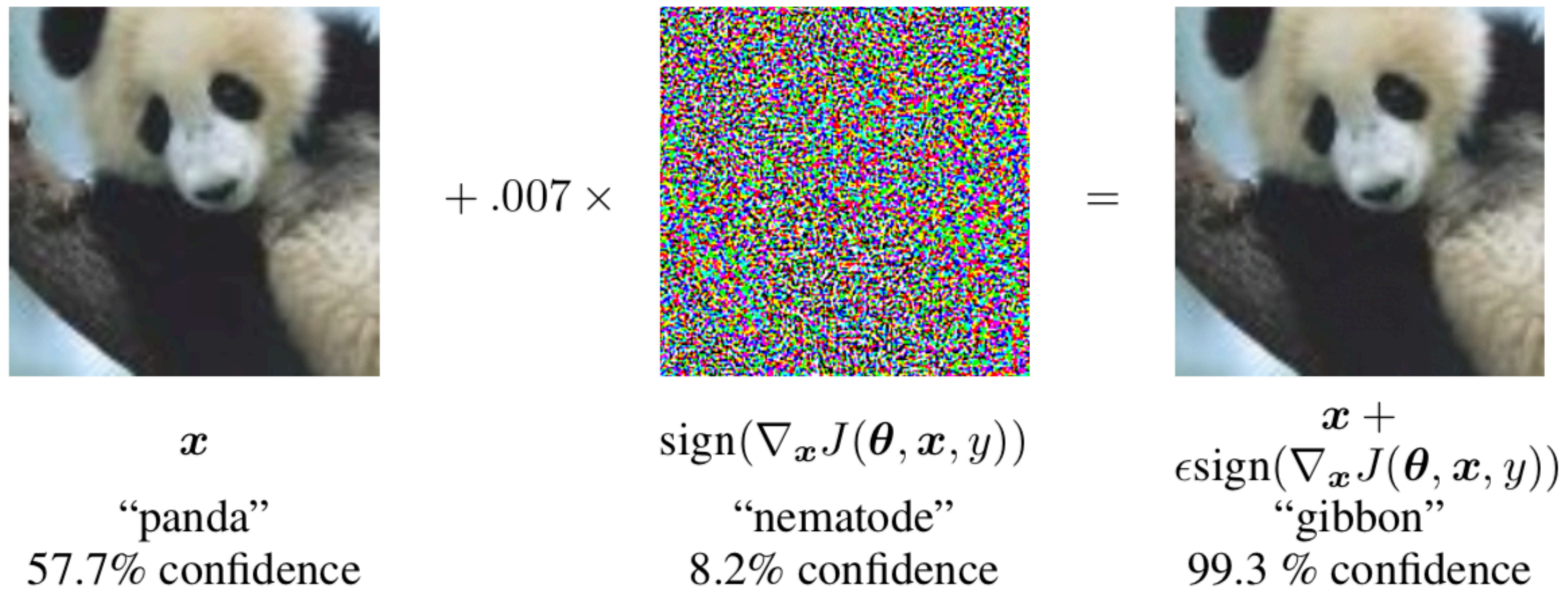
Pitfalls of Feature Extractors

Reduce high-dimensional samples to binary decision

- What could go wrong?

Previous Adversarial Machine Learning (AML) work:

Models exhibit “blind spots”



[Goodfellow ICLR’15]

[Hendrycks CoRR’19]

Takeaway: high-dimensional feature extractors are insufficiently calibrated

Subverting Authentication

Can we subvert authentication systems using previous techniques?

- A target's information is secret and hidden (otherwise you would already have access)
- Information returned from authentication systems is limited (response $\in \{YES, NO\}$)

Short answer: Yes, despite these setbacks

Problem Formulation

Refine previously defined Authentication System (AS)

- Set of credentials: $u \in U$

  Analogous to "usernames" registered with AS

- Define the underlying mapping of submitted samples to users: $F : X \longrightarrow U$

  Mapping performs classification necessary for AS to yield a response

- Treat AS as a function: $AS(u, \mathbf{x}) = y$ for decision $y \in \{YES, NO\}$

Adversarial Capabilities

Introduce adversary *A*:

- Adversary ***A*** is allowed to know the dimensionality *d* of a feature extractor F relies on:

$$F : g(X)^d \longrightarrow U$$

- ***A*** knows some subset of usernames: $U_{\mathscr{A}} \subset U$
- In fact, ***A*** can register their own samples with AS: $X_{\mathscr{A}}$

All other principals (users) of the system:

- Define the set of benign principals known to AS: $\mathscr{V} = \{v \in U : v \neq \mathscr{A}\}$

  … and their samples: $X_{\mathscr{V}}$ with $X_{\mathscr{A}} \cap X_{\mathscr{V}} = \varnothing$

Restrictive-Query Threat Model

Adversary wishes to impersonate some victim, gaining access to resources:

- Denote victim as $v \in \mathscr{V}$
- ***A*** eventually crafts an adversarial sample $\mathbf{x}^*$ such that $AS(v, \mathbf{x}^*) = YES$
- Use a reasonable amount of queries to avoid detection

Intermediate samples $\mathbf{x}'$ are iteratively crafted until $\mathbf{x}^*$ is found.

Henceforth the adversary has achieved *Masquerade* (M)

Strategy

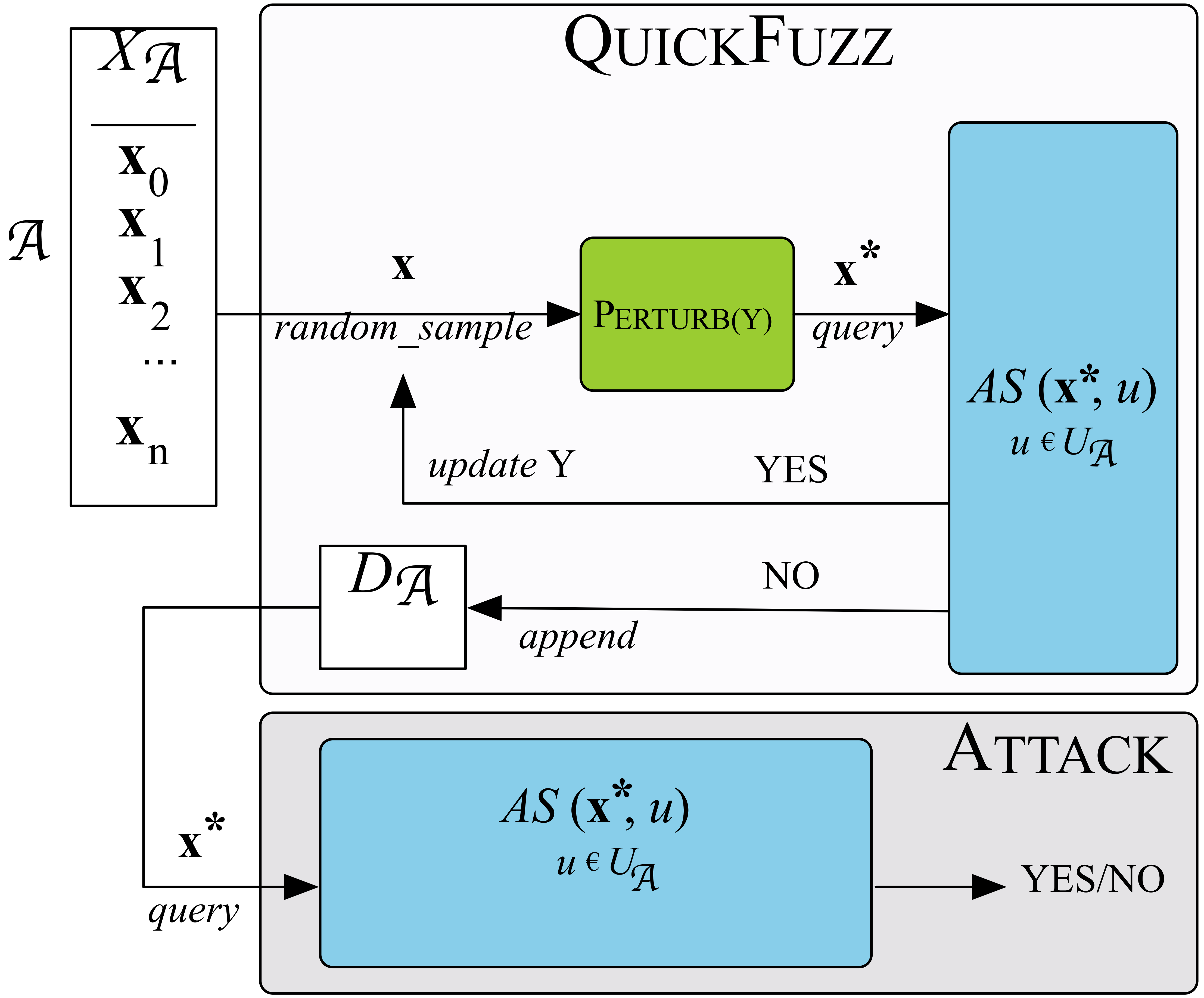
Adversary wishes to impersonate some victim, gaining access to resources

- ***A*** is performing an untargeted exploratory attack
- Target: integrity of resources protected by AS
- No access to weights, data, training algorithm, or confidence scores of AS

Strategy: Construct an algorithm for query-efficient fuzzing through the feature extractor

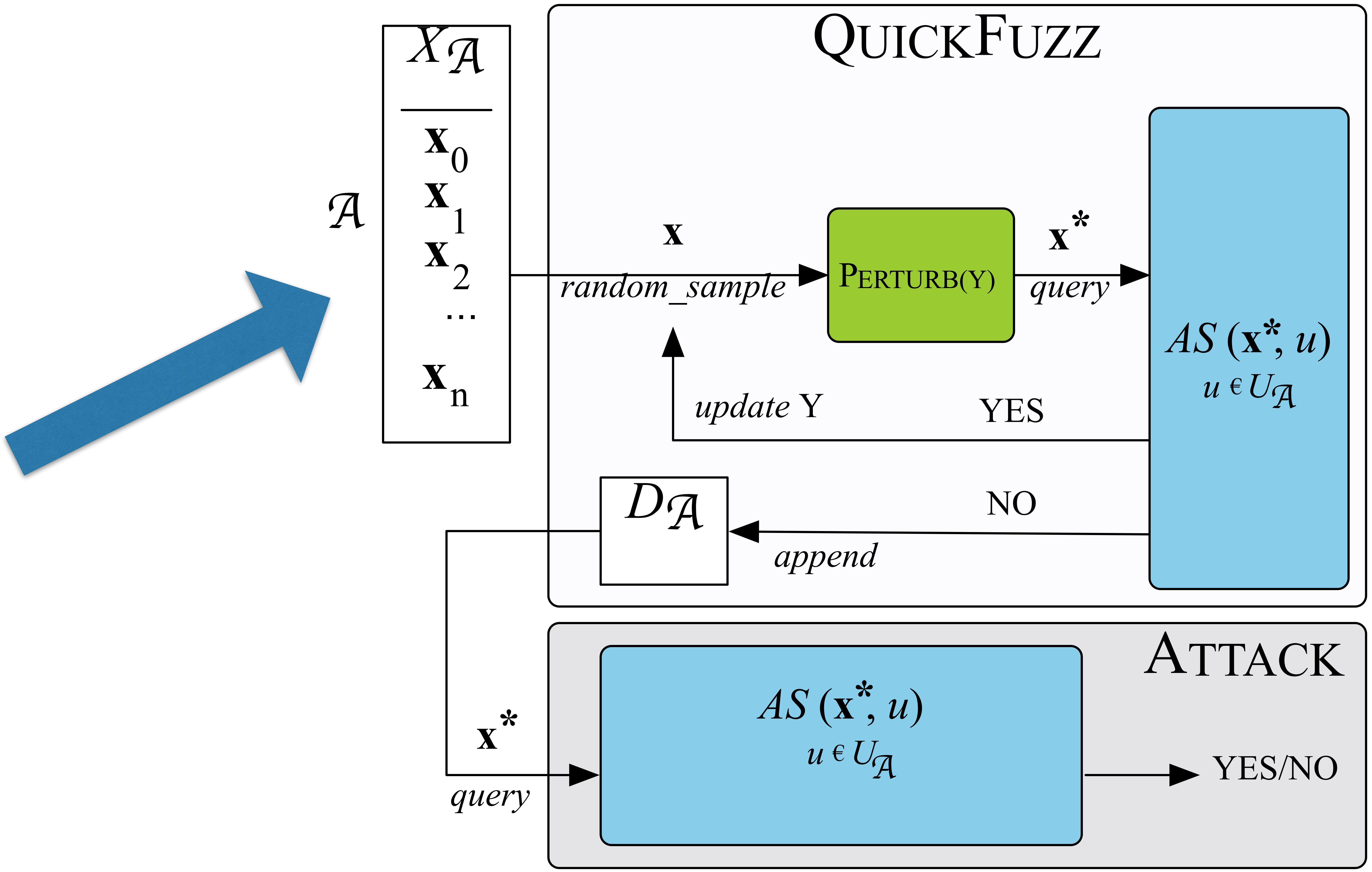
Query-Efficient Fuzzing

Strategy: Query-efficient fuzzing through the feature extractor



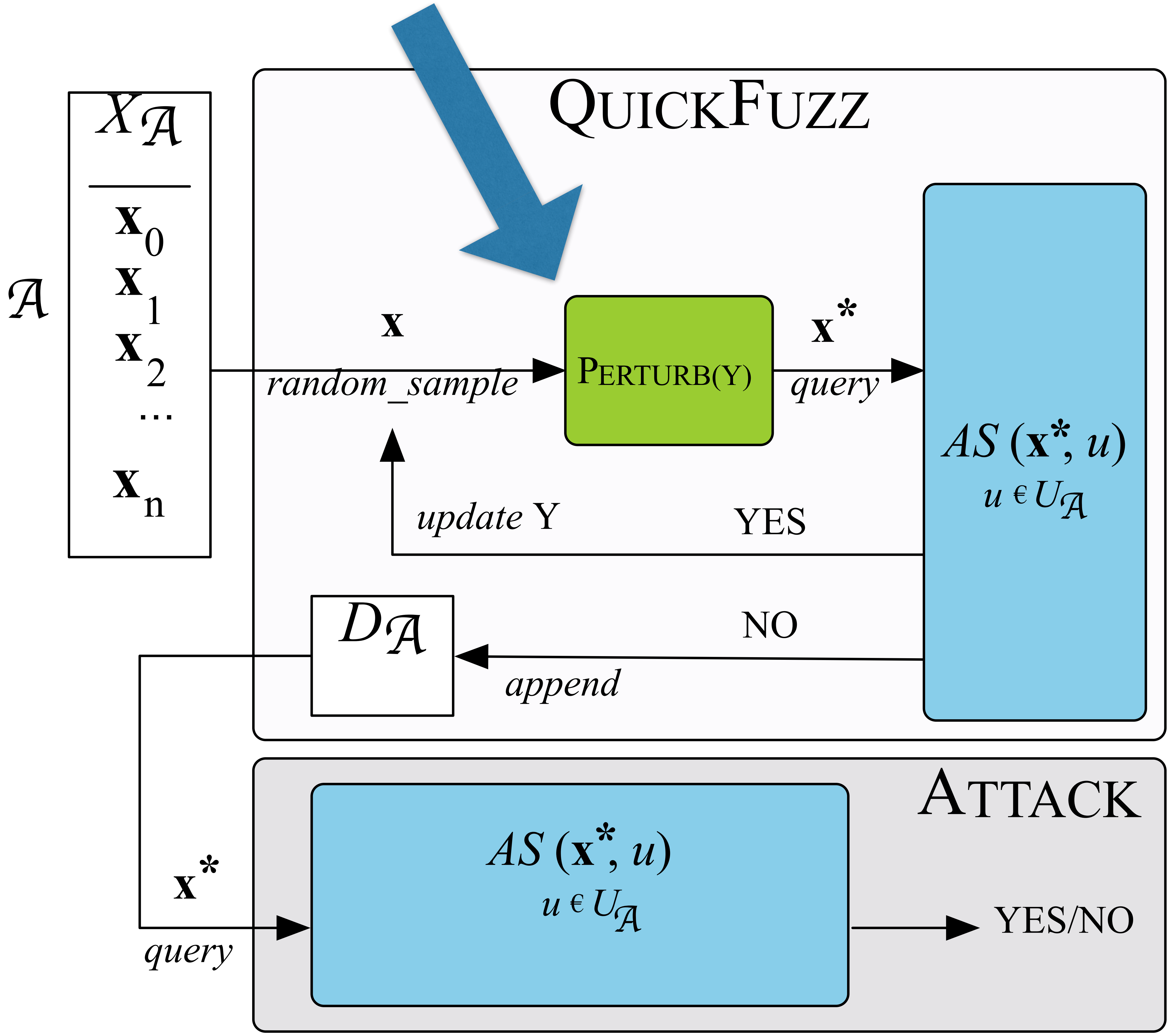
Query-Efficient Fuzzing

Strategy: Query-efficient fuzzing through the feature extractor



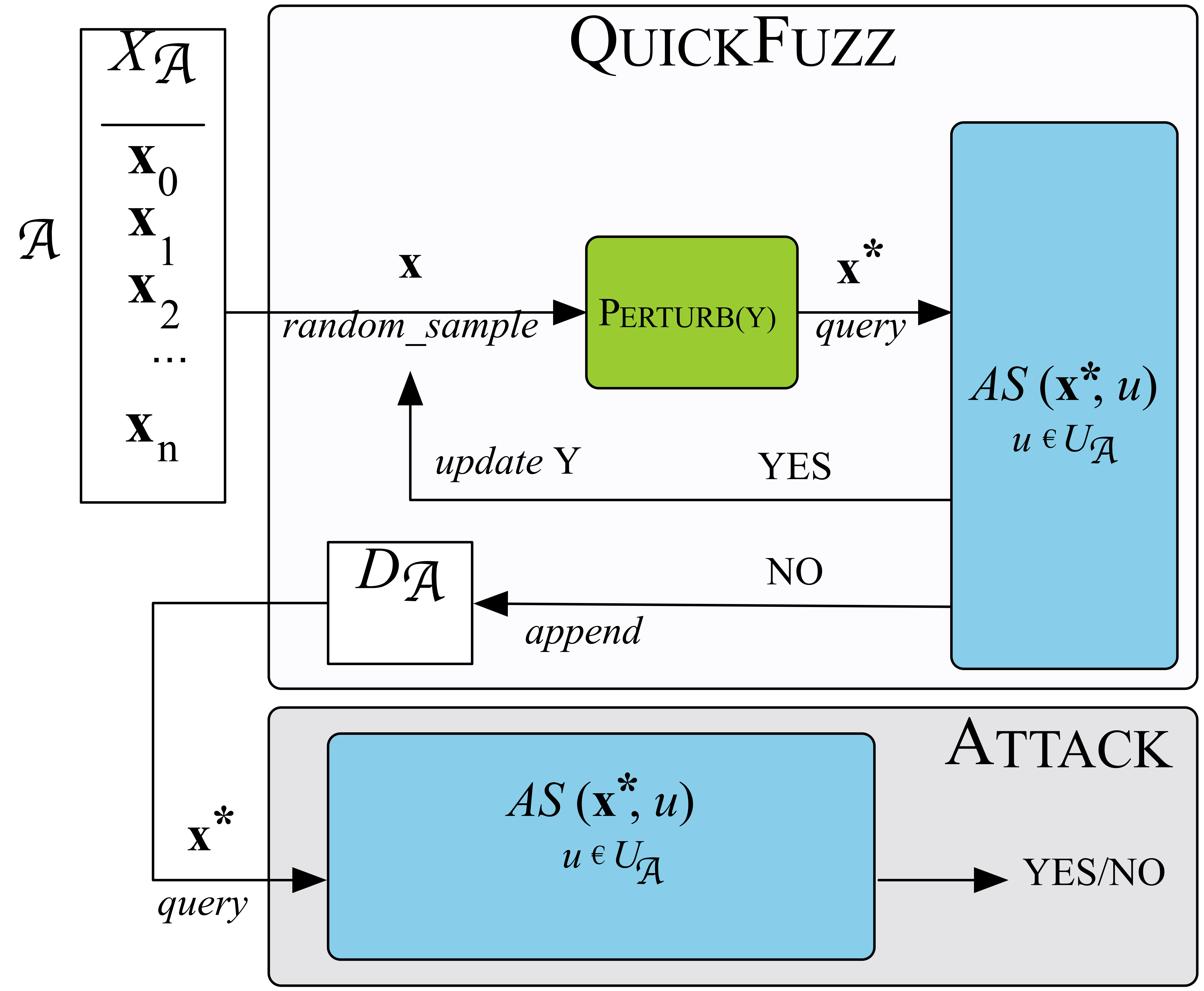
Query-Efficient Fuzzing

Strategy: Query-efficient fuzzing through the feature extractor



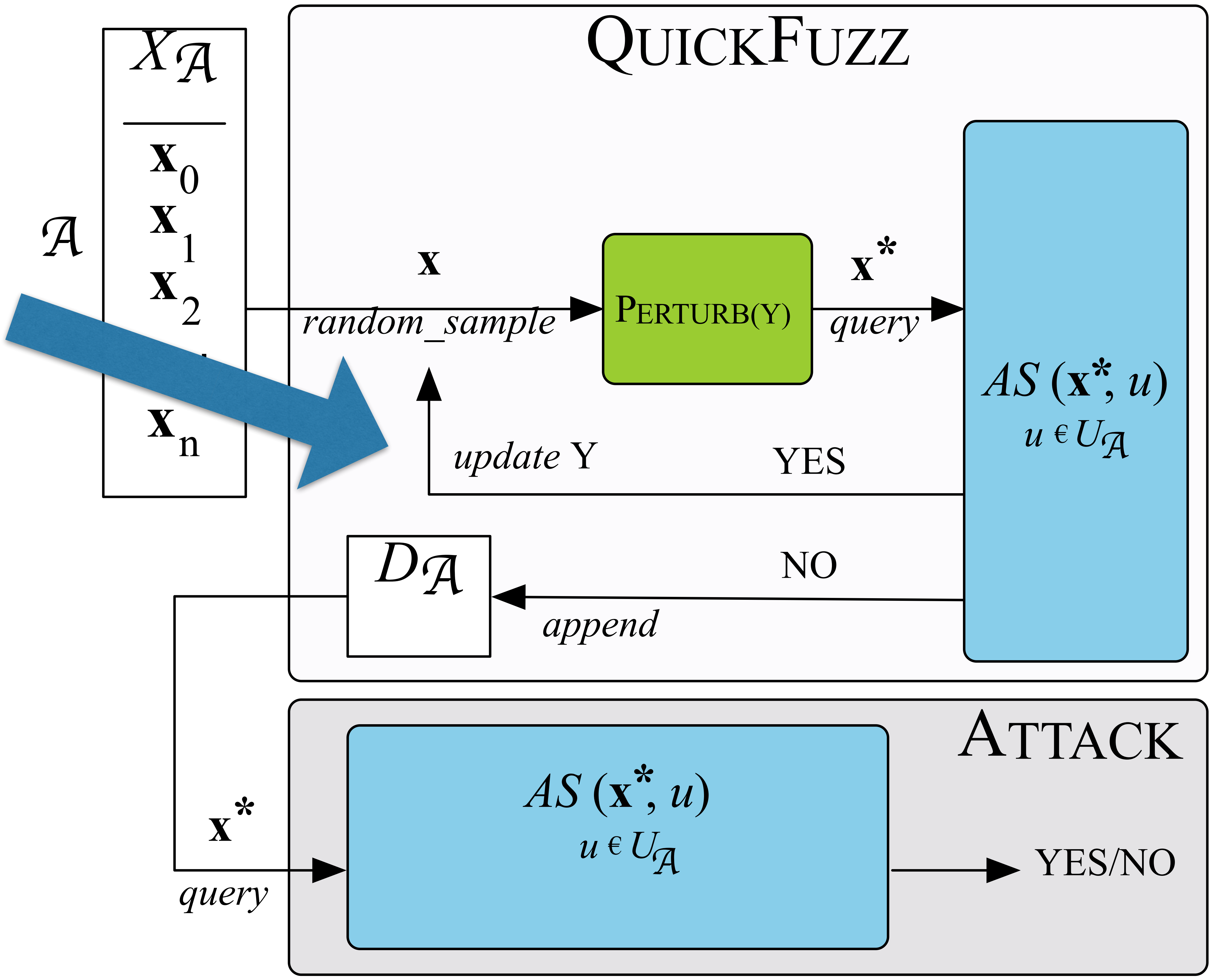
Query-Efficient Fuzzing

Strategy: Query-efficient fuzzing through the feature extractor



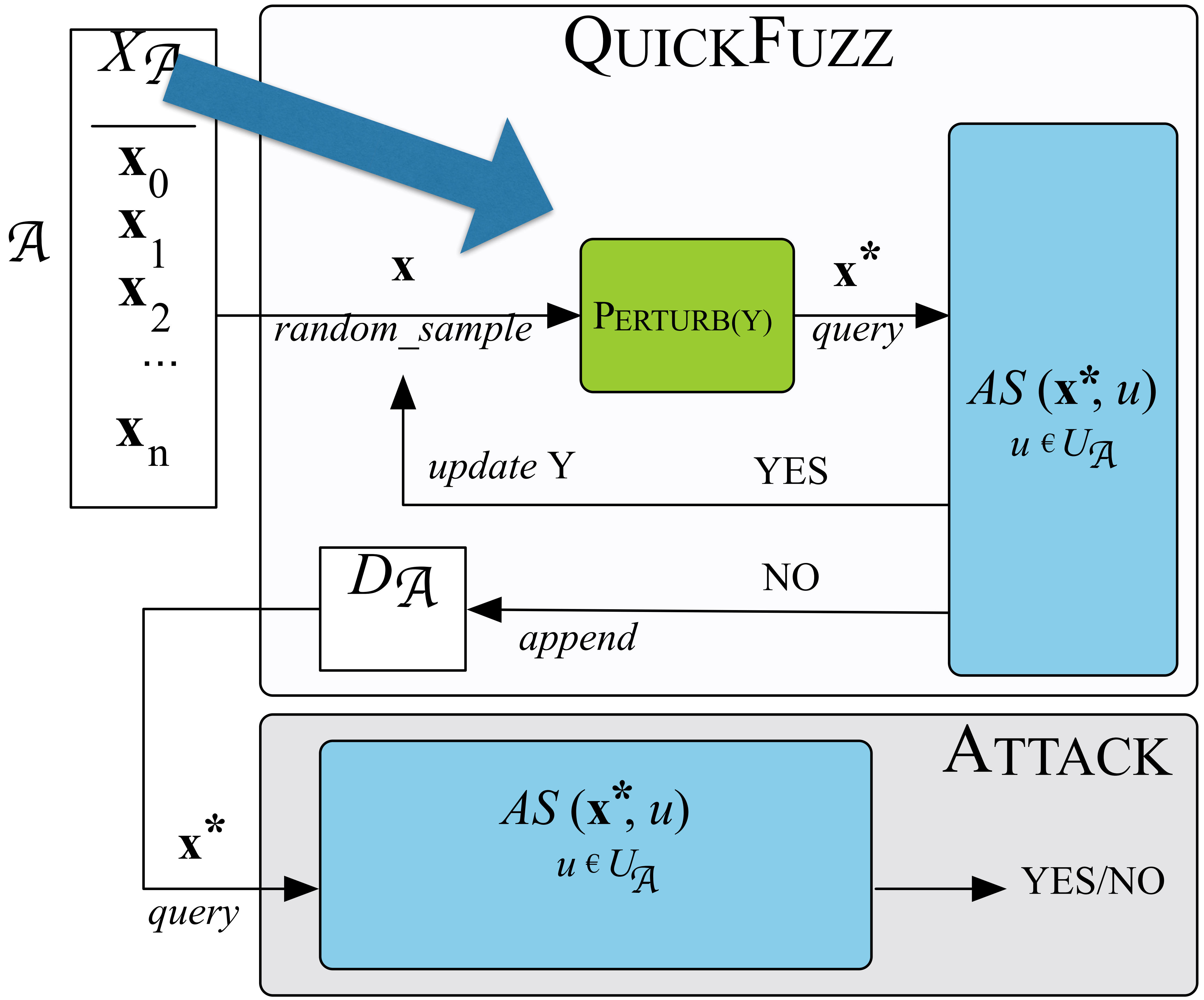
Query-Efficient Fuzzing

Strategy: Query-efficient fuzzing through the feature extractor



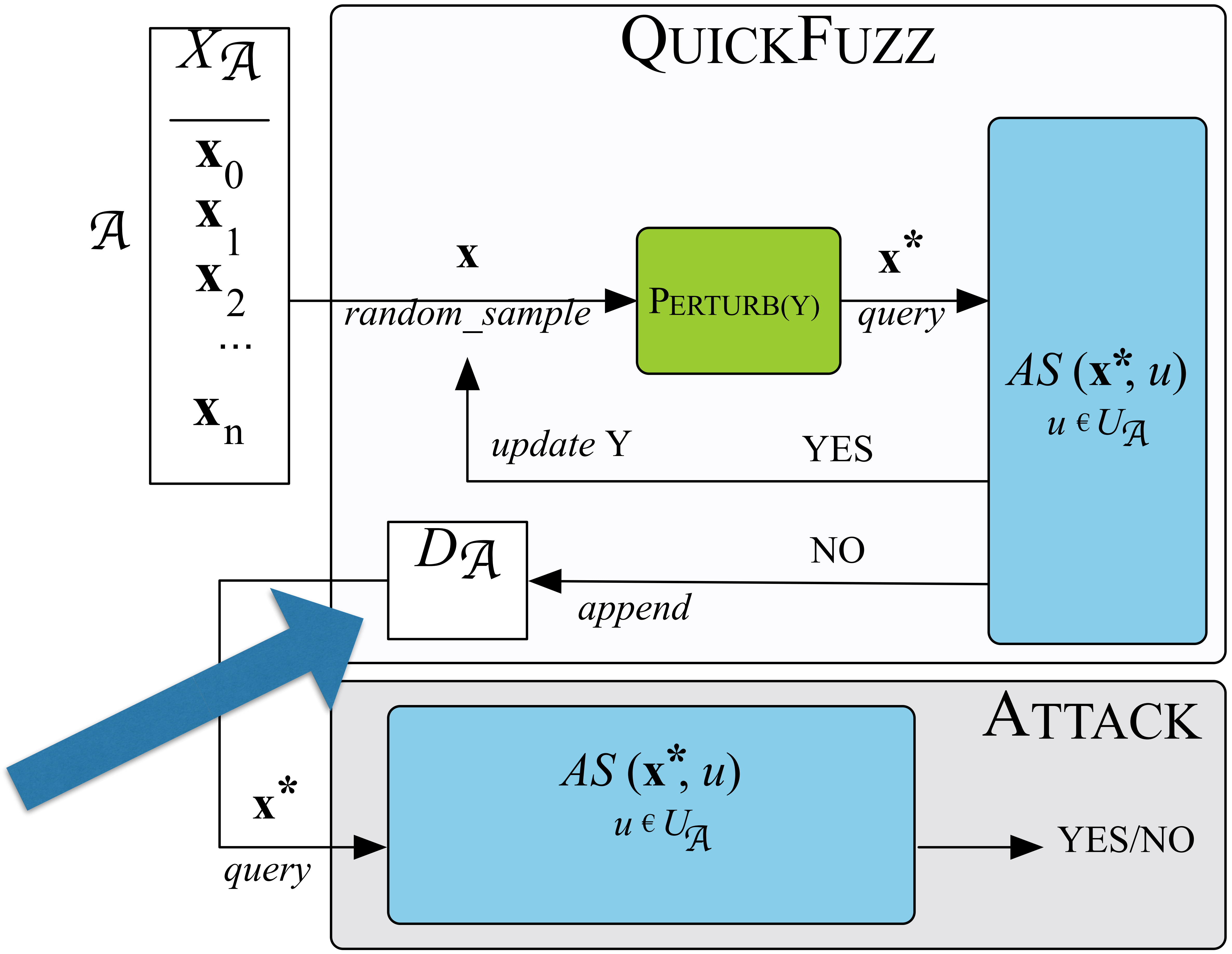
Query-Efficient Fuzzing

Strategy: Query-efficient fuzzing through the feature extractor



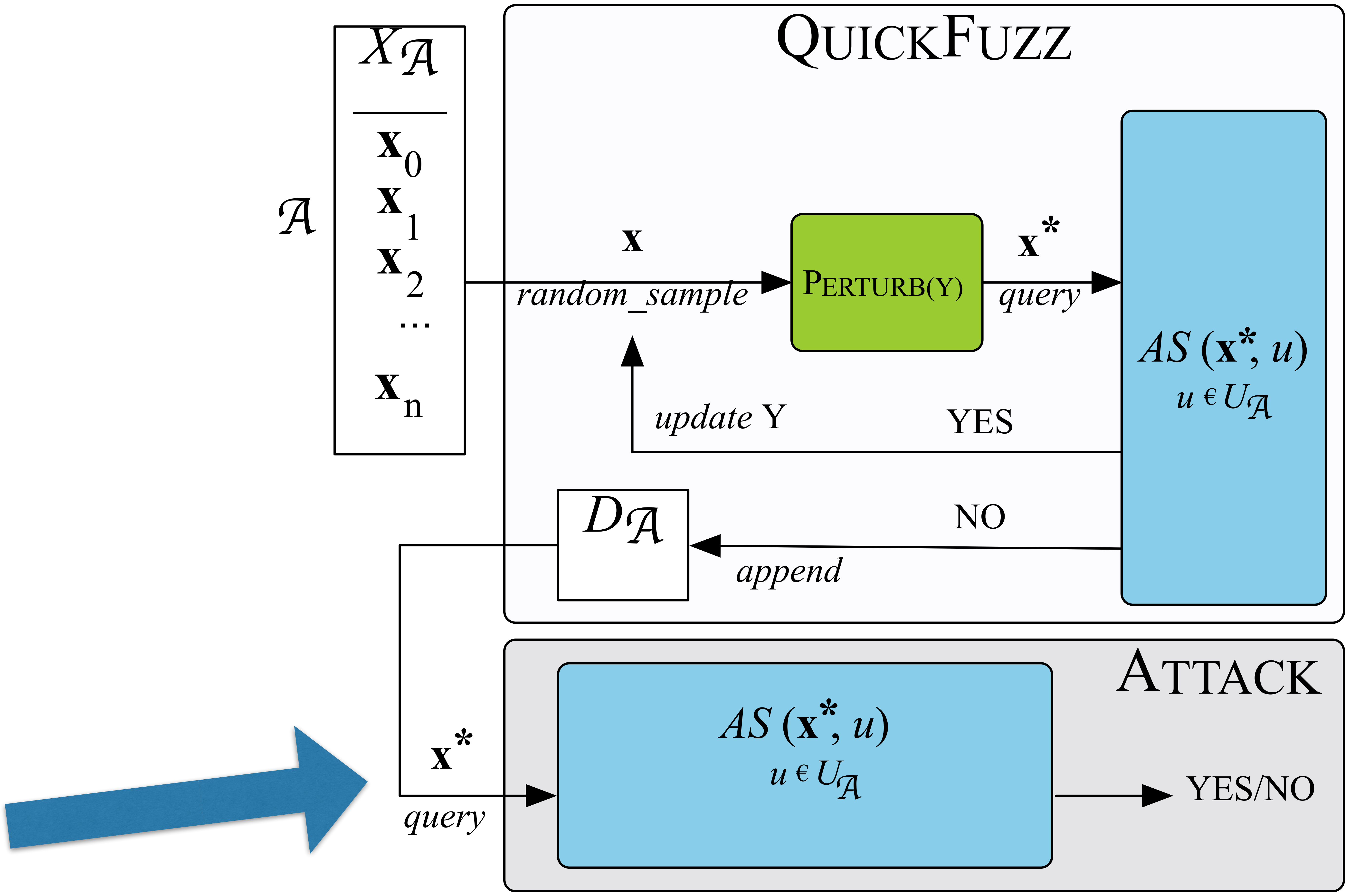
Query-Efficient Fuzzing

Strategy: Query-efficient fuzzing through the feature extractor

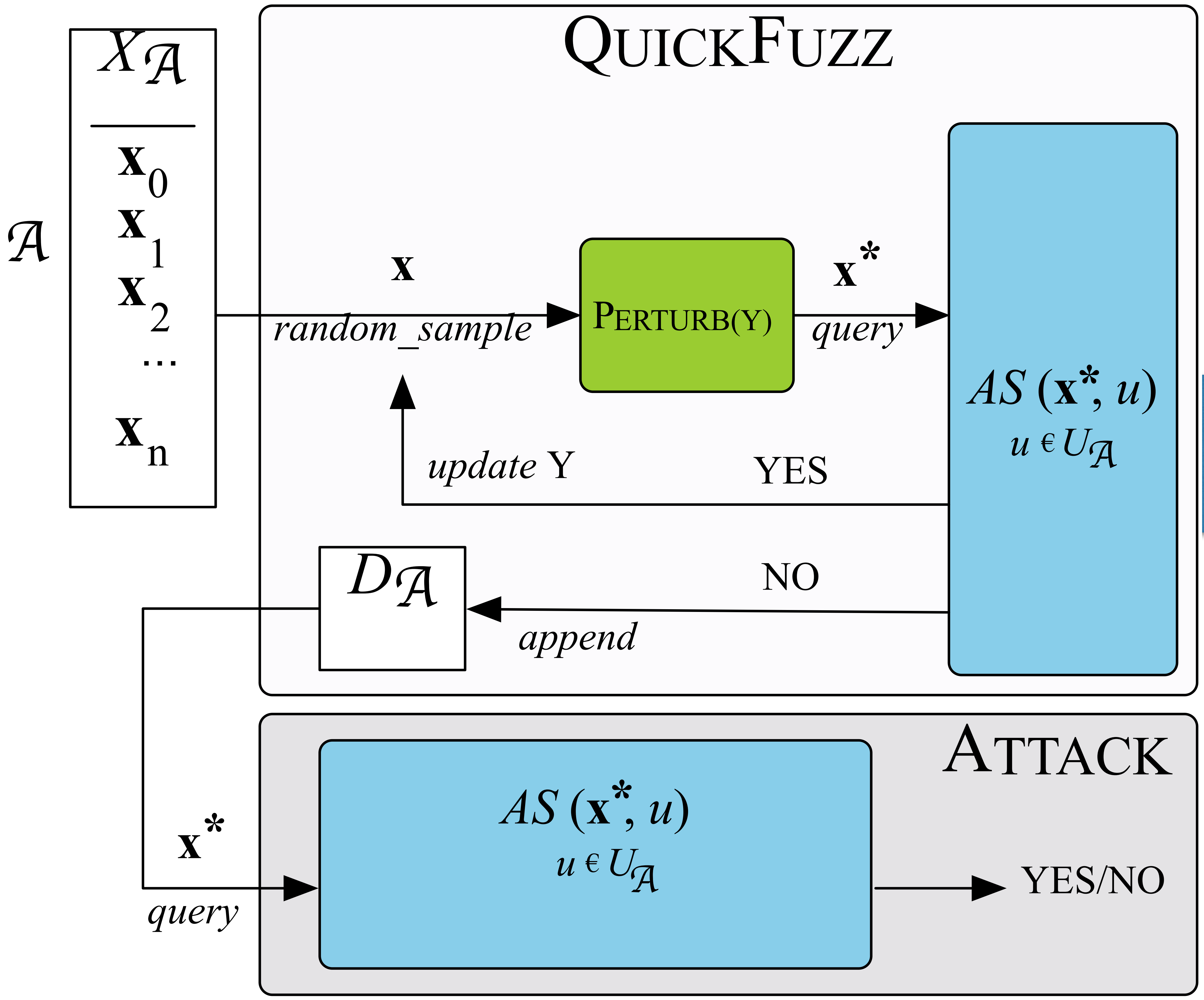


Query-Efficient Fuzzing

Strategy: Query-efficient fuzzing through the feature extractor



Query-Efficient Fuzzing



Algorithm 1 QUICKFUZZ, adversarial sample crafting algorithm for system authentication.

Input: $X_{\mathcal{A}}$, a set of initial samples owned by $\mathcal{A}$. σ, an arbitrary upper bound on number of adversarial samples to create, AS, the victim authentication system, and subroutine *query*, a generic interface available to A for querying AS.

$D_{\mathcal{A}} \leftarrow \varnothing$
$R \leftarrow$ YES
$\Upsilon \leftarrow 0$ ▷ (Distortion parameter)
while $|D_{\mathcal{A}}| < \sigma$ **do**
  ▷ (Loop until we meet sufficient distortion)
  while $R =$ YES **do**
    increment(Υ)
    $\mathbf{x} \leftarrow$ *random_sample*$(X_{\mathcal{A}})$
    $\mathbf{x}^* \leftarrow$ PERTURB$(\mathbf{x}, \Upsilon)$
    $R \leftarrow$ *query*$(AS, \mathcal{A}, \mathbf{x}^*)$
  end while
  $\mathbf{x}^* \leftarrow$ PERTURB$(\mathbf{x}, \Upsilon)$
  $R \leftarrow$ *query*$(AS, \mathcal{A}, \mathbf{x}^*)$
  if $R =$ NO: *append*$(D_{\mathcal{A}}, \mathbf{x}^*)$
end while
return $D_{\mathcal{A}}$

Measuring Effects

Adversary wishes to reach *Masquerade*. How likely is this w.r.t their knowledge?

- Consider a $|U| \times |U|$ matrix S of all possible adversary-victim pairs in the system
- For simplicity, the adversary has full knowledge, $U_{\mathscr{A}} = U$ and acts alone.
- Then $S_{i,j}$ denotes that adversary $\boldsymbol{A}_i$ was successful against victim v_j
- AS is more vulnerable if these pairs are scattered throughout S

Measuring Effects

Adversary wishes to reach *Masquerade*. How likely is this w.r.t their knowledge?

- Consider a $|U| \times |U|$ matrix S of all possible adversary-victim pairs in the system
- For simplicity, the adversary has full knowledge, $U_{\mathscr{A}} = U$ and acts alone.
- Then $S_{i,j}$ denotes that adversary $\boldsymbol{A}_i$ was successful against victim v_j
- AS is more vulnerable if these pairs are scattered throughout S

$$P(M) = \frac{|\{S_{i,j} > 0 : i \neq j\}|}{|\{\mathbf{1}_{i,j}^{|U|\times|U|} : i \neq j\}|}$$

Measuring Distortion

How much distortion is necessary to be successful?

- Calculate distortion ϵ to offer intuition over different methods
- Relative change between $\mathbf{x}$ and best attack sample $\mathbf{x}^*$

$$\epsilon = \frac{||\mathbf{x} - \mathbf{x}^*||_2}{||\mathbf{x}||_2}$$

- Denote average change for some attack strategy as the average $\bar{\epsilon}$

Experimental Validation

Implement attack against three proposed device authentication systems:

1. USB-Fingerprinting (USB-F) - End-host authentication based on USB enumeration timings. [Bates NDSS'14]

    Classifier: Random Forest trained in One vs. Rest style

Experimental Validation

Implement attack against three proposed device authentication systems:

1. USB-Fingerprinting (USB-F) - End-host authentication based on USB enumeration timings. [Bates NDSS'14]

   Classifier: Random Forest trained in One vs. Rest style

2. GTID - Device-agnostic identification based on inter-arrival times of network packets. [Radhakrishnan TDSC'15]

   Classifier: Ensemble of Artificial Neural Networks (ANNs)

Experimental Validation

Implement attack against three proposed device authentication systems:

1. USB-Fingerprinting (USB-F) - End-host authentication based on USB enumeration timings. [Bates NDSS'14]

   Classifier: Random Forest trained in One vs. Rest style

2. GTID - Device-agnostic identification based on inter-arrival times of network packets. [Radhakrishnan TDSC'15]

   Classifier: Ensemble of Artificial Neural Networks (ANNs)

3. WDTF - Device authentication based on probe request traffic of IEEE 802.11 wireless devices. [Dalai WPC'17]

   Classifier: Kernel derived from hand-crafted features

Attack Scenarios

Evaluate using different attack scenarios:

1. Baseline - Legitimate test set data, lower bound of robustness for each system
2. Random - ***A*** constructs samples randomly following a Gaussian distribution.
3. Greedy Adversary - ***A*** wields QuickFuzz algorithm, and stops as soon as a victim is found.
4. Exploratory Adversary - ***A*** wields QuickFuzz and exhausts some fixed *query budget*.

Attack Effectiveness

Research Question 1: Is the attack effective?

USB-F

	Accuracy	Recall
Bates et al. [6]	94-99%	-
Our Baseline	100%	100%
Random	100%	100%
Greedy $\mathcal{A}$	85%	33%
Exploratory $\mathcal{A}$, $\bar{p} = 100$	83%	33%
Exploratory $\mathcal{A}$, $\bar{p} = 200$	80%	33%
Exploratory $\mathcal{A}$, $\bar{p} = 300$	76%	22%

GTID

	Accuracy	Recall
Uluagac et al. [37]	99%	94%
Our Baseline	97%	85%
Random	86%	6%
Greedy $\mathcal{A}$	87%	13%
Exploratory $\mathcal{A}$, $\bar{p} = 100$	78%	13%
Exploratory $\mathcal{A}$, $\bar{p} = 200$	74%	13%
Exploratory $\mathcal{A}$, $\bar{p} = 300$	74%	13%

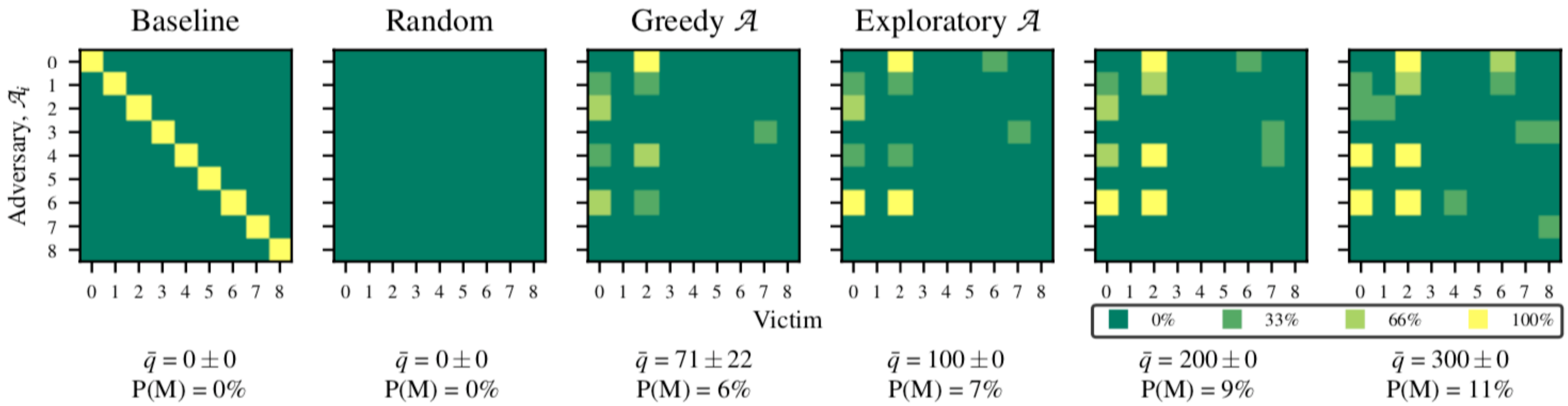
WDTF

	Accuracy	Recall
Our Baseline	98%	97%
Random	73%	47%
Greedy $\mathcal{A}$	87%	75%
Exploratory $\mathcal{A}$, $\bar{p} = 100$	81%	75%
Exploratory $\mathcal{A}$, $\bar{p} = 200$	81%	75%
Exploratory $\mathcal{A}$, $\bar{p} = 300$	81%	75%

Attack Effectiveness

Research Question 2: How many queries to affect integrity?

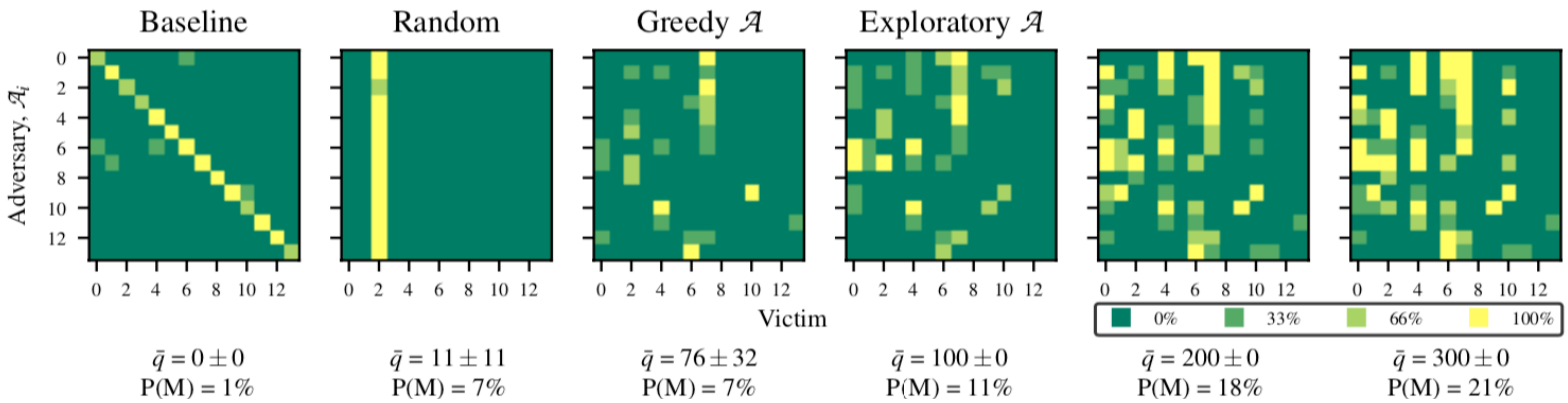
USB-F



Attack Effectiveness

Research Question 2: How many queries to affect integrity?

GTID



Attack Effectiveness

Research Question 2: How many queries to affect integrity?

WDTF

Feature Exploration

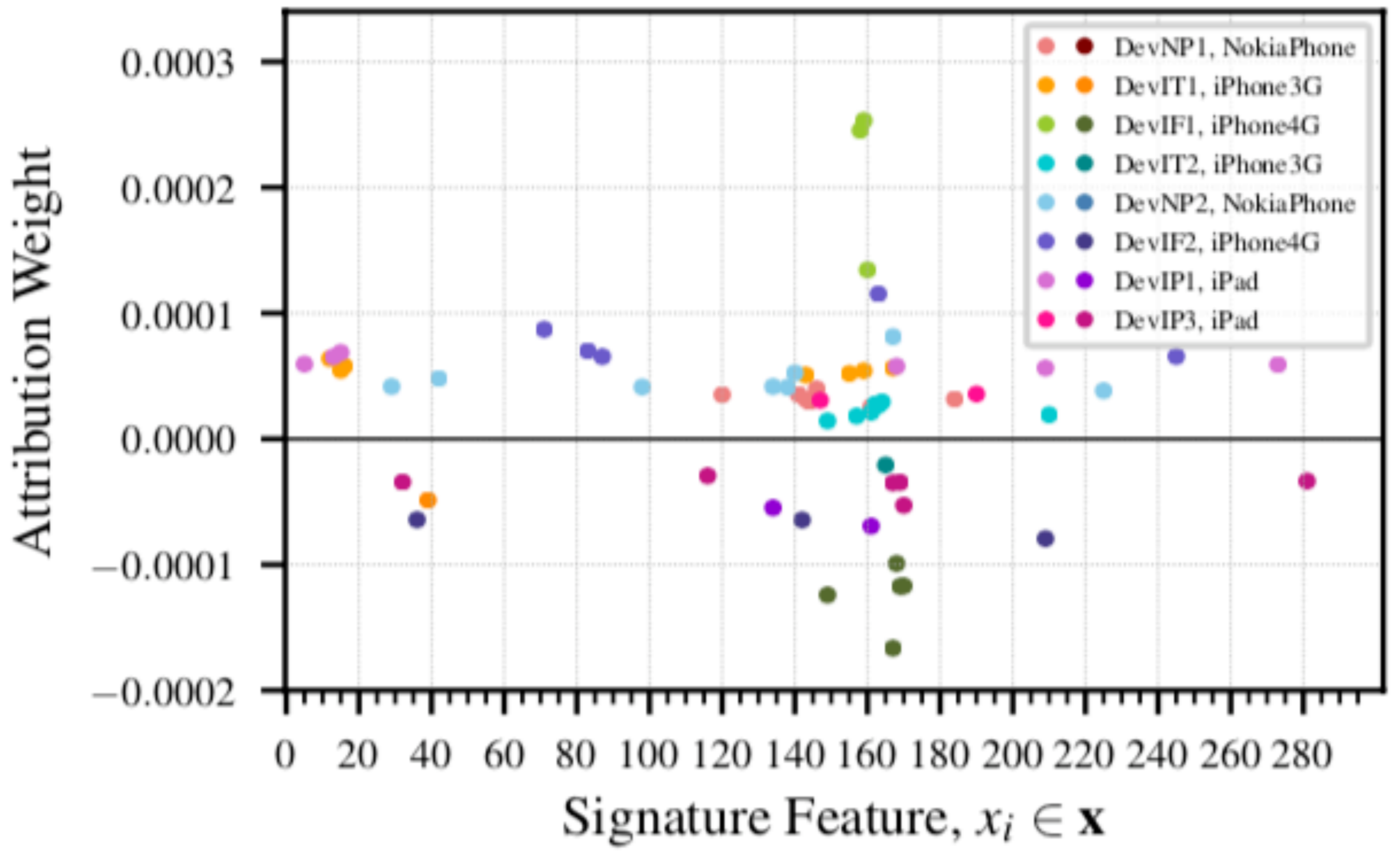
Research Question 3: Do certain features contribute to brittle performance?

- Use XAI technique (LIME) to analyze each decision space. [Ribeiro KDD'16]

Feature Exploration

Research Question 3: Do certain features contribute to brittle performance?

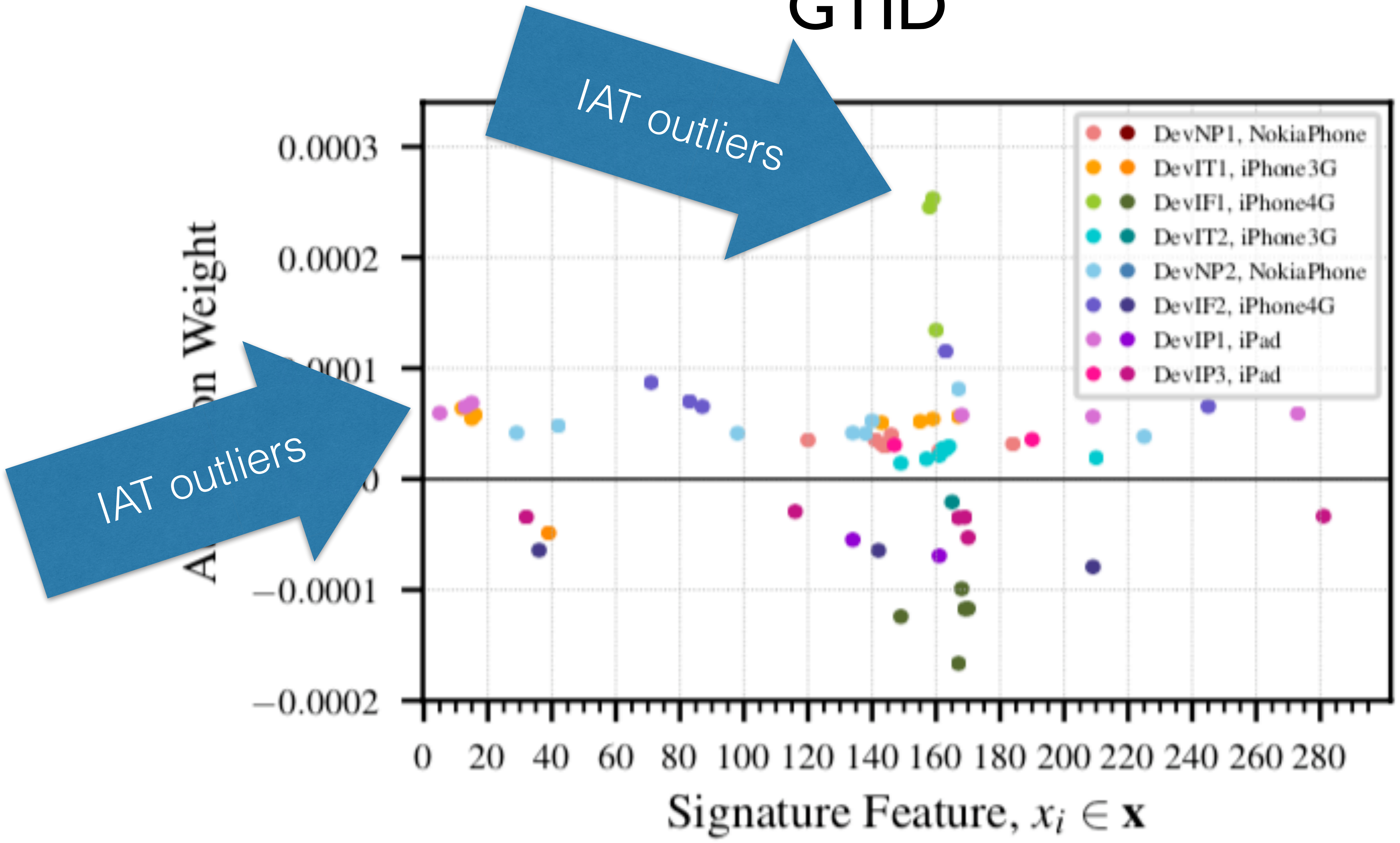
GTID



Feature Exploration

Research Question 3: Do certain features contribute to brittle performance?

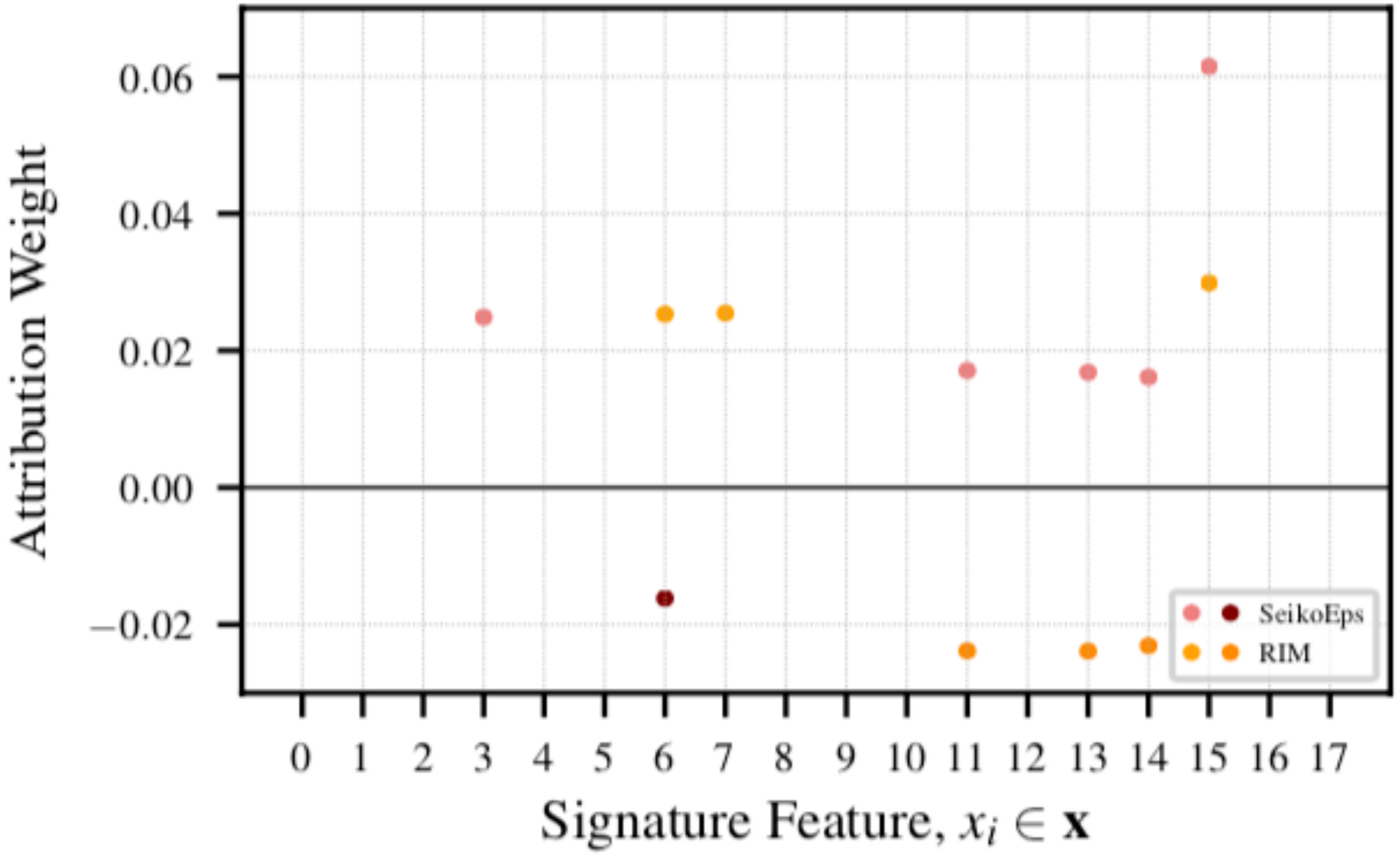
GTID



Feature Exploration

Research Question 3: Do certain features contribute to brittle performance?

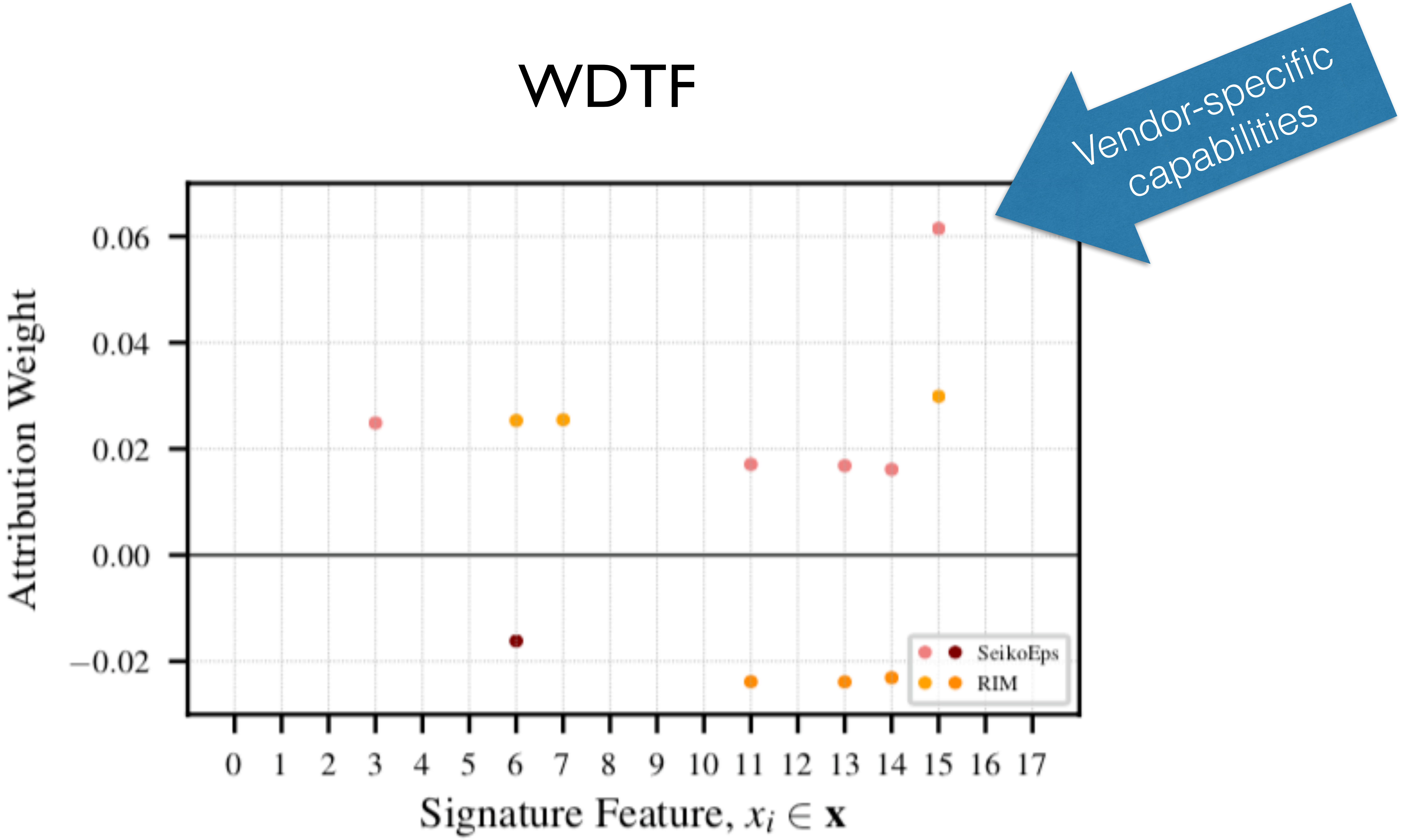
WDTF



Feature Exploration

Research Question 3: Do certain features contribute to brittle performance?

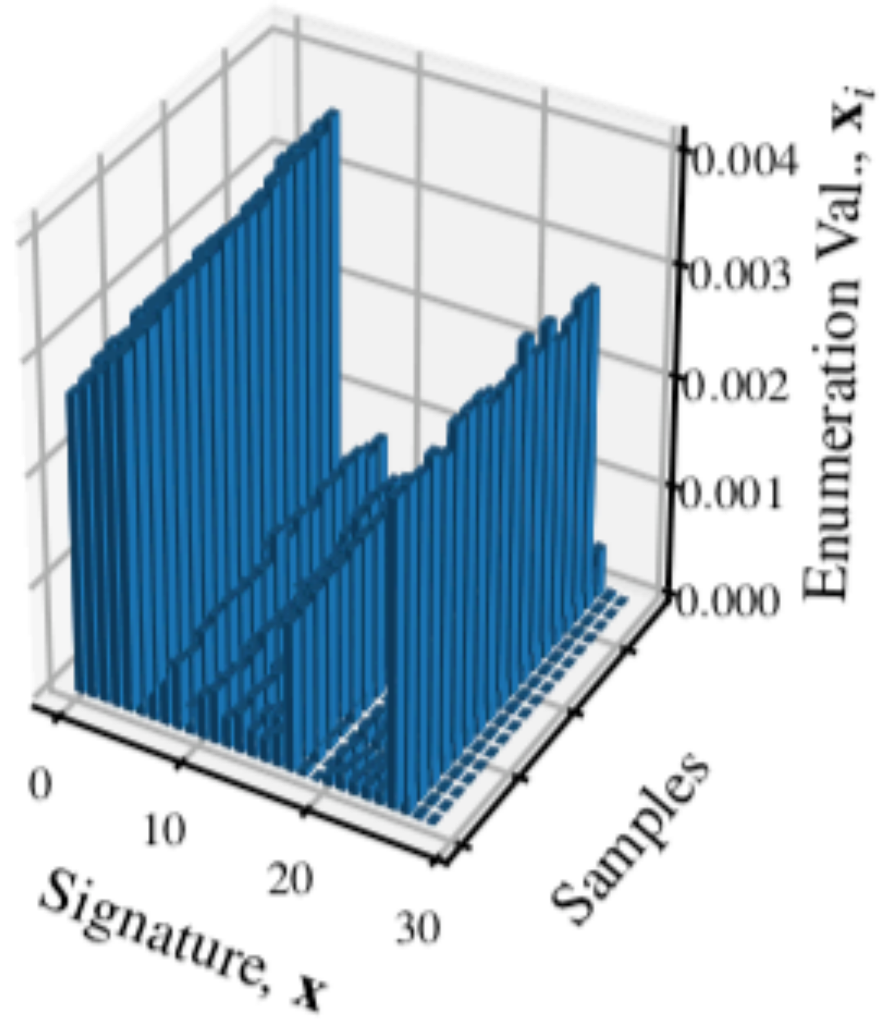
WDTF

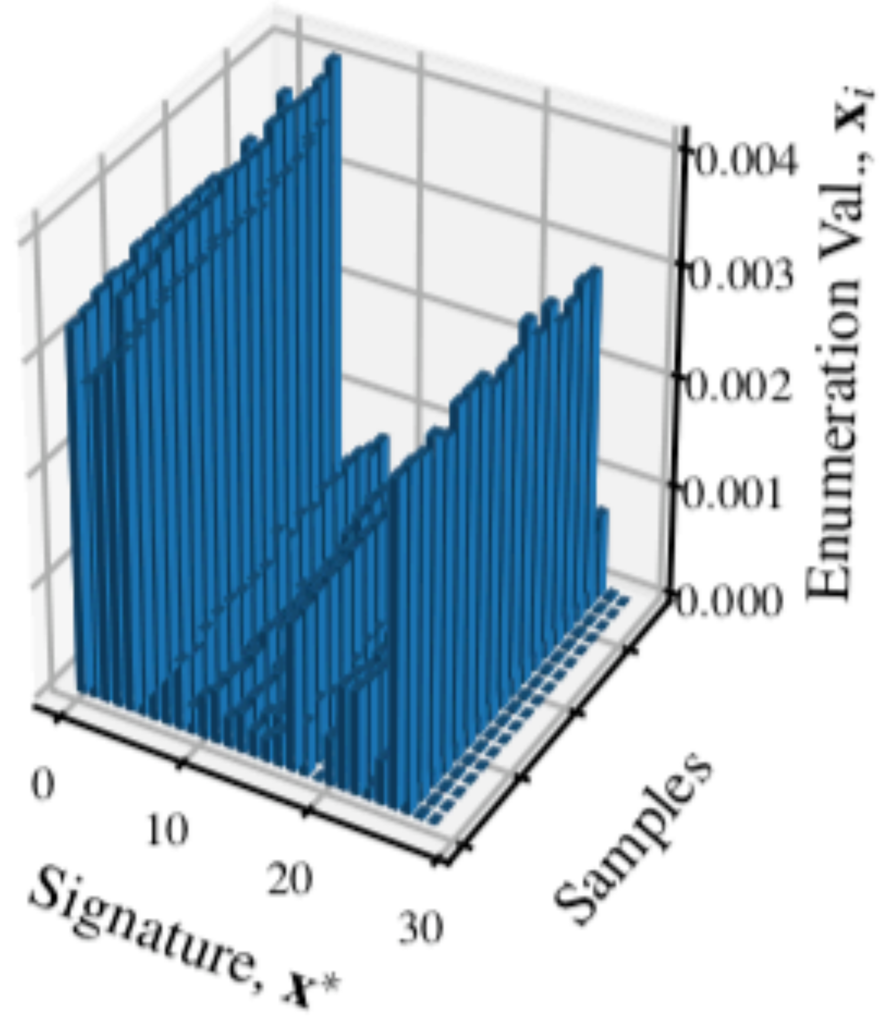


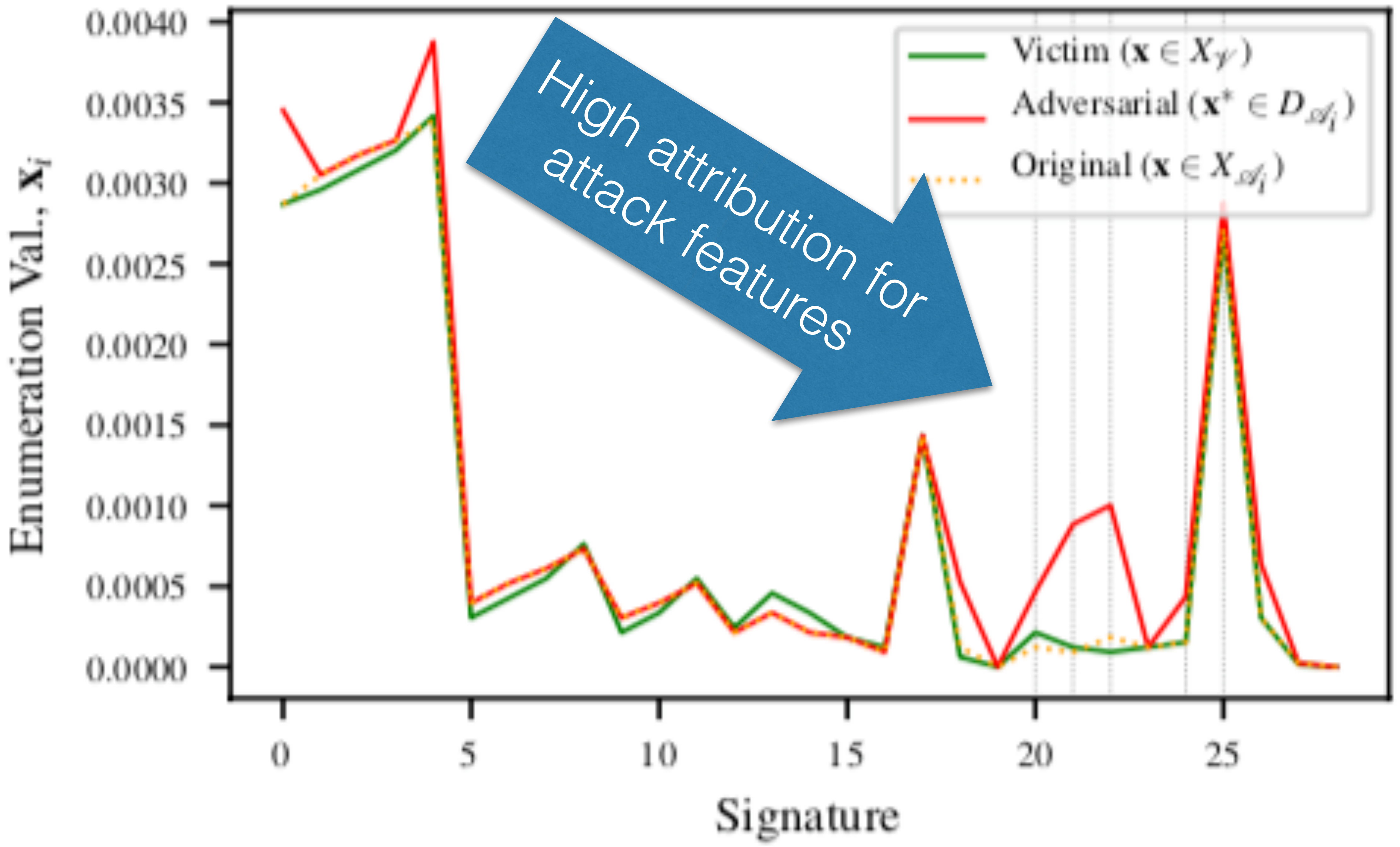
Feature Exploration

Research Question 4: What do attack data distributions look like?

USB-F





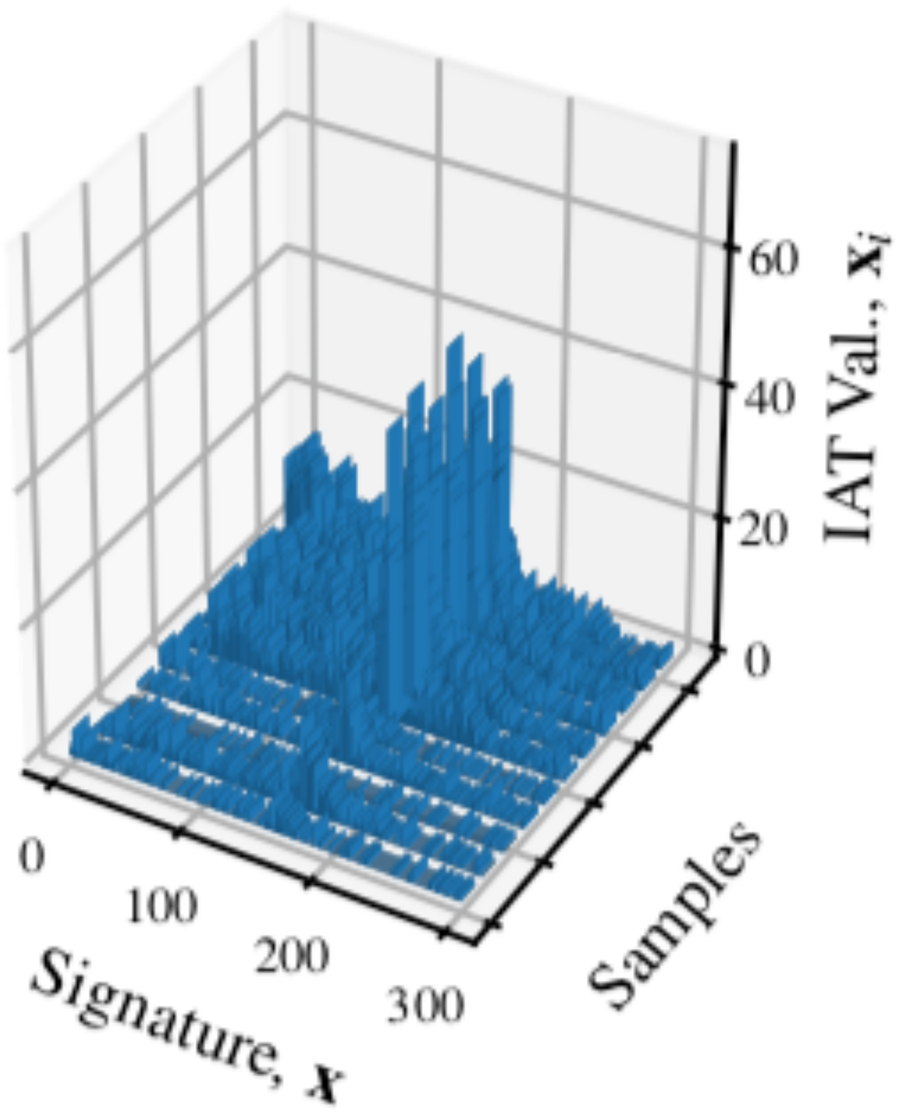


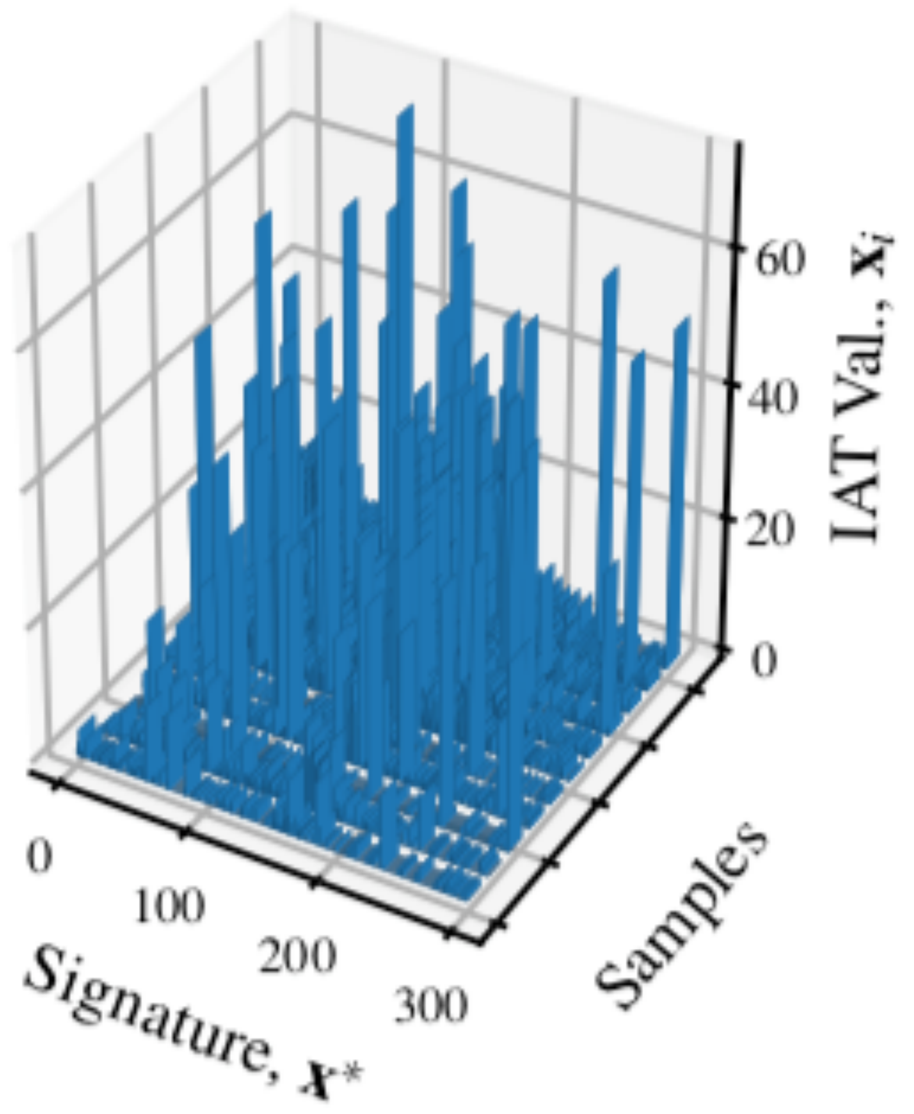
Feature Exploration

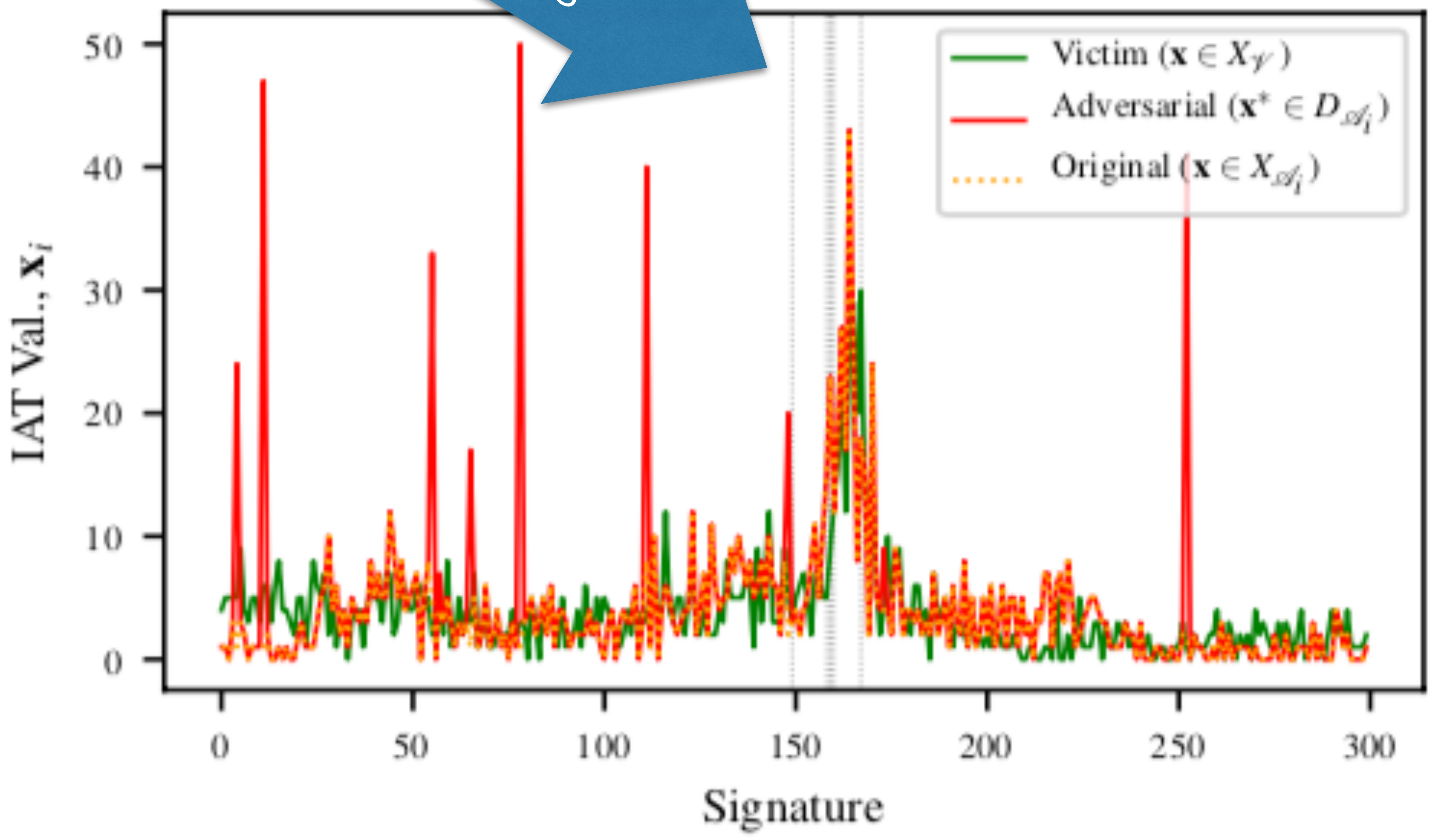
Research Question 4: What do attack data distributions look like?

GTID

High attribution for IAT outliers

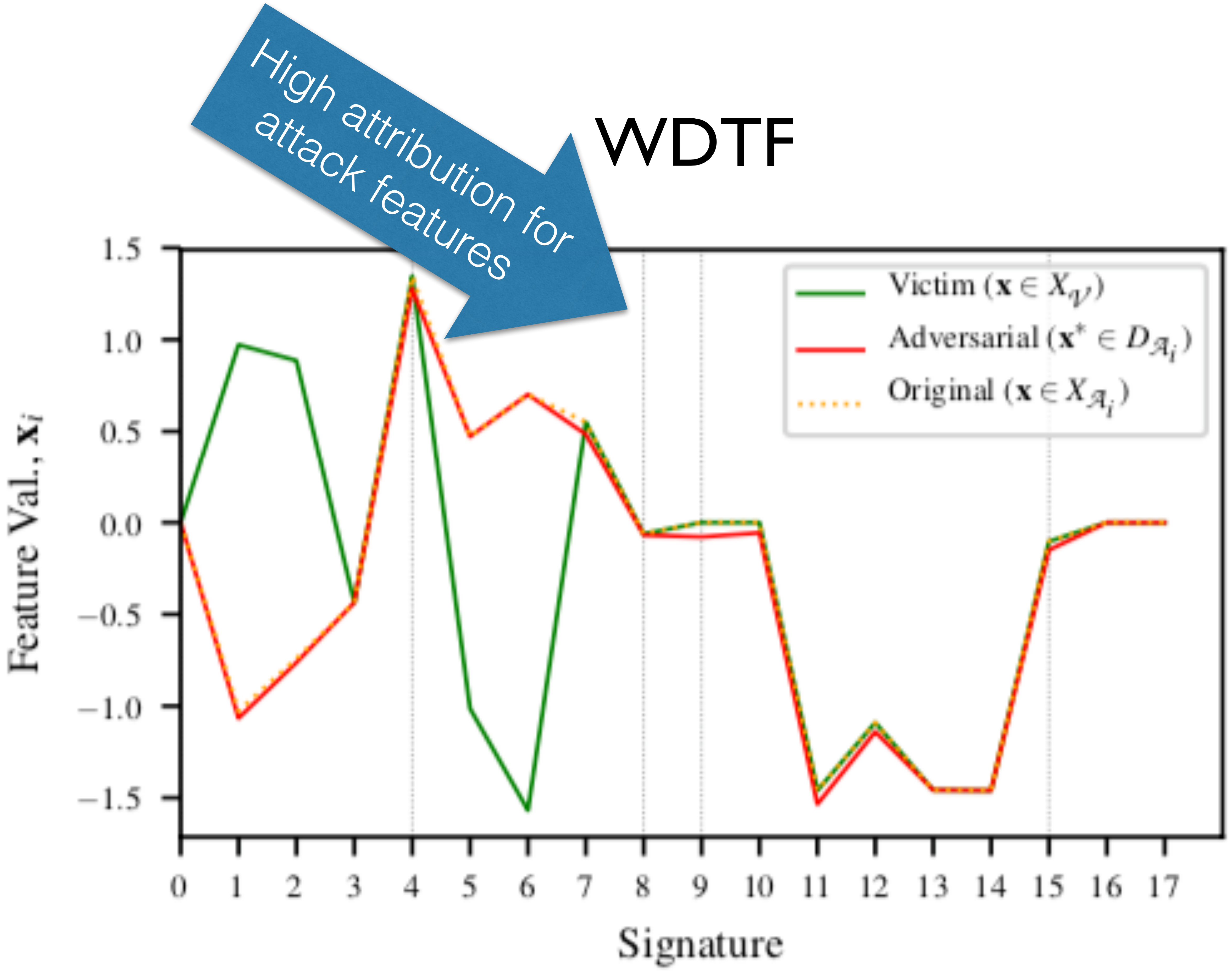






Feature Exploration

Research Question 4: What do attack data distributions look like?



Takeaways

Let us revisit our four high-level research questions:

1. Does a random attack work between different authentication domains?

   Yes, up to 21% chance of masquerade in worst case of GTID system.

2. How many queries are needed to affect integrity of resources?

   In most cases, less than 100 queries are needed for substantial FPR.

3. Do certain types of features contribute to brittle performance?

   Features tend to be sensitive to device properties, but generally unintuitive.

4. How do sample data distributions change between legitimate and attack scenarios?

   Attack distributions tend to appear as noise, difficult to distinguish.

Zeroth Order Extension

Hard-label decision adversaries: Only label is returned from classifier.

QuickFuzz performs random walk through input space to find victims.

- Ideally, inform the movement with gradient estimate.

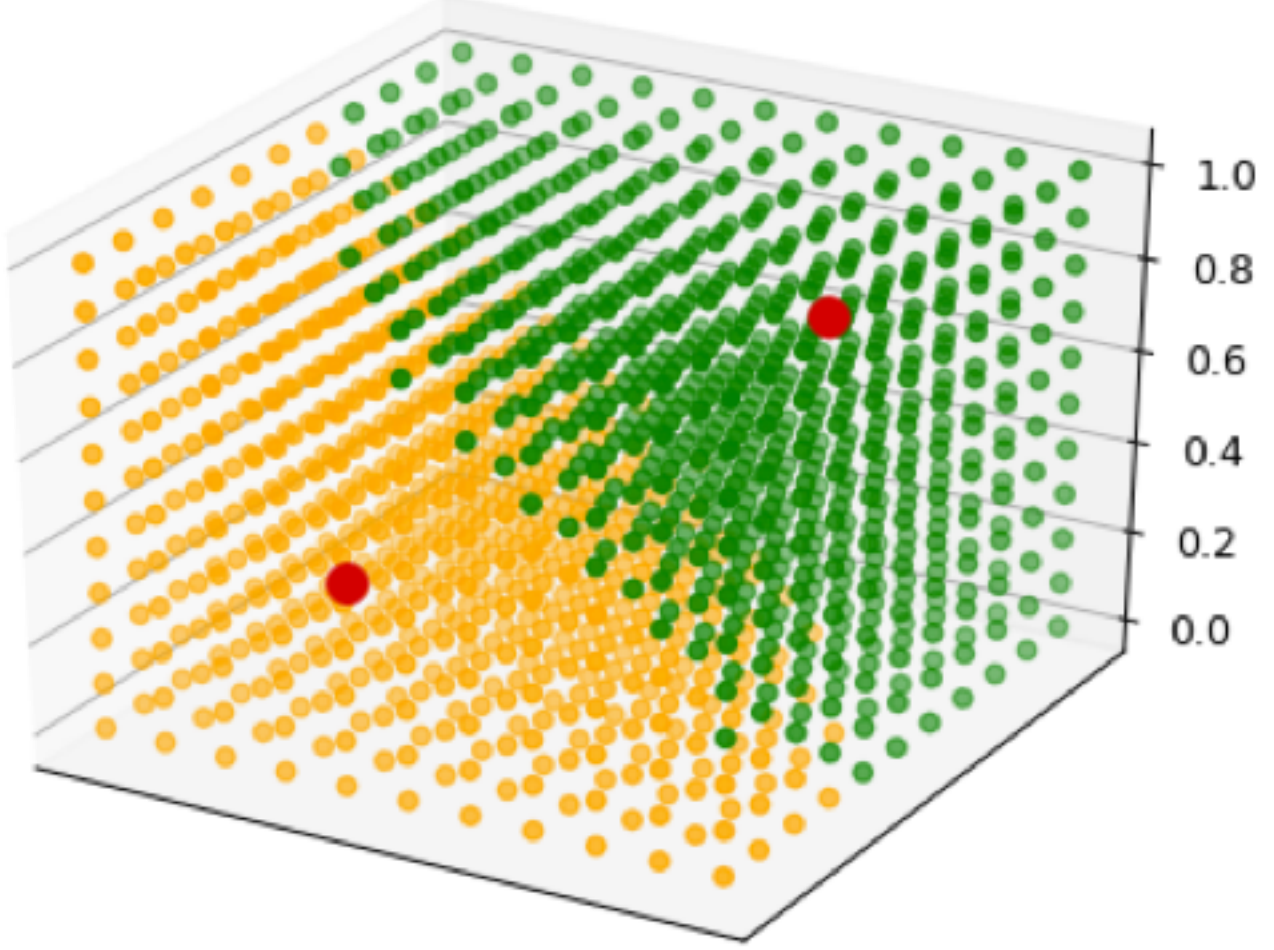
Zeroth-Order Optimization (ZOO) attack:

- Approach decision boundary, estimate gradient at a classifier's decision boundary, repeat, until $\mathbf{x}^*$ is found. [Chen AISec'17, Chen CoRR'19]

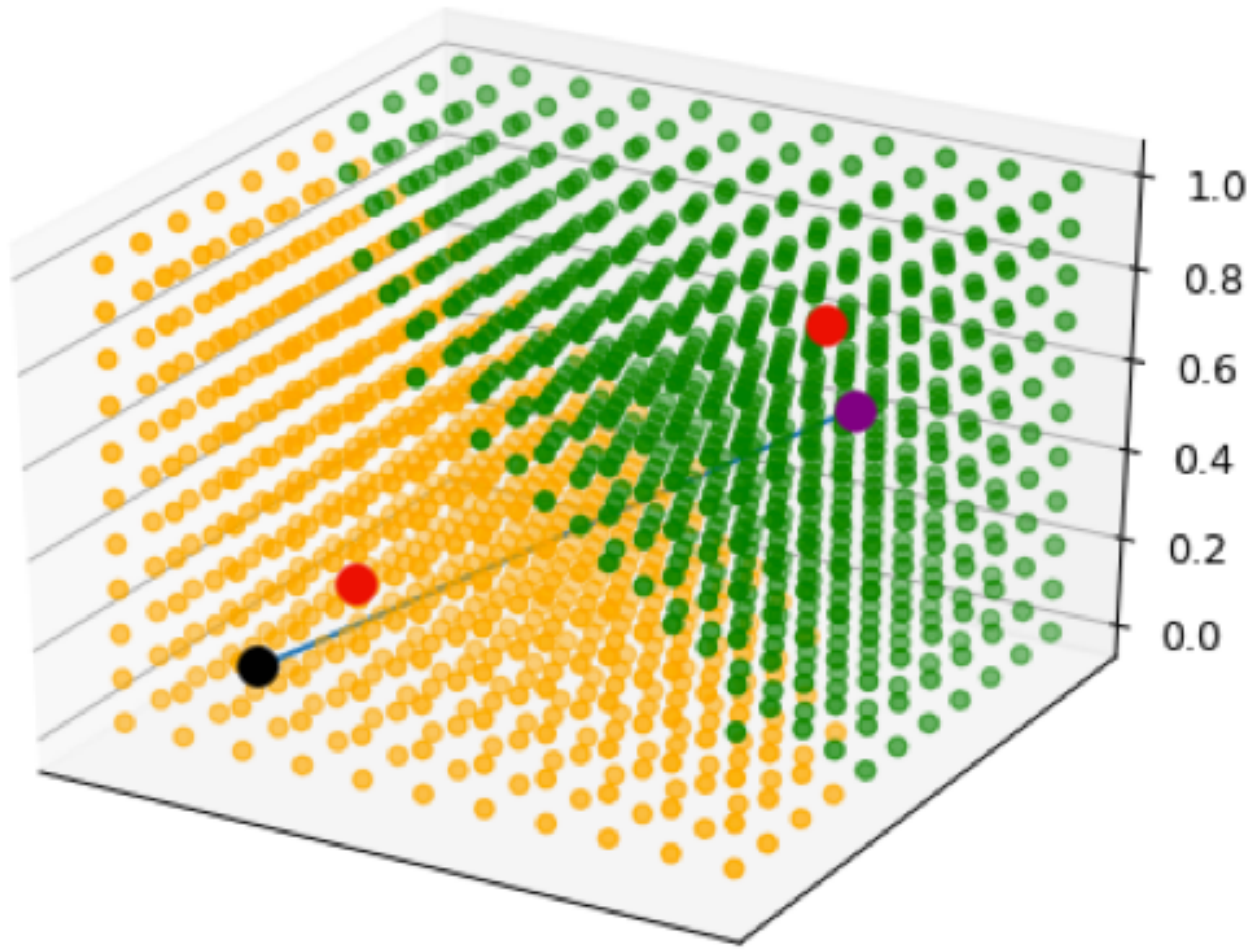
Zeroth Order Extension

Toy Problem:

LDA Decision Boundary

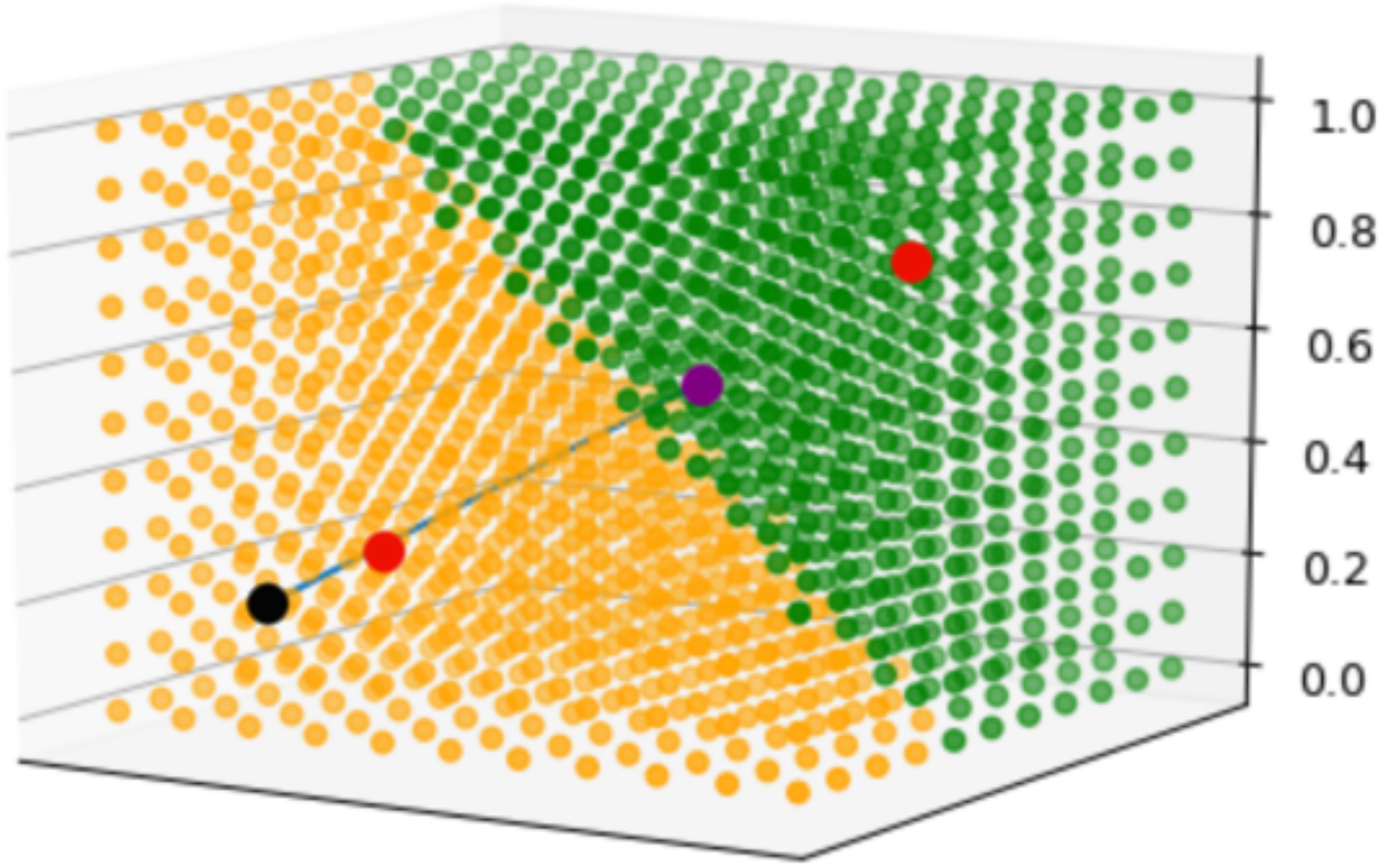


Projection from $\mathbf{x}$ to $\mathbf{x}^{\mathbf{target}}$



Projection from $\mathbf{x}^*$ to $\mathbf{x}^{\mathbf{target}}$

Successful Attack



Takeaway: Extend concept of gradient estimation to authentication setting.

Feature Engineering Redux

- XAI - explaining opaque (black-box) models at instance level (often with other opaque models)
- Interpretable-ML - feature extractor design guided by interpretable primitives
- Can XAI alone reliably inform us? Ongoing work

[Dombrowski CoRR'19]

Thank You

Washington Garcia
w.garcia@ufl.edu