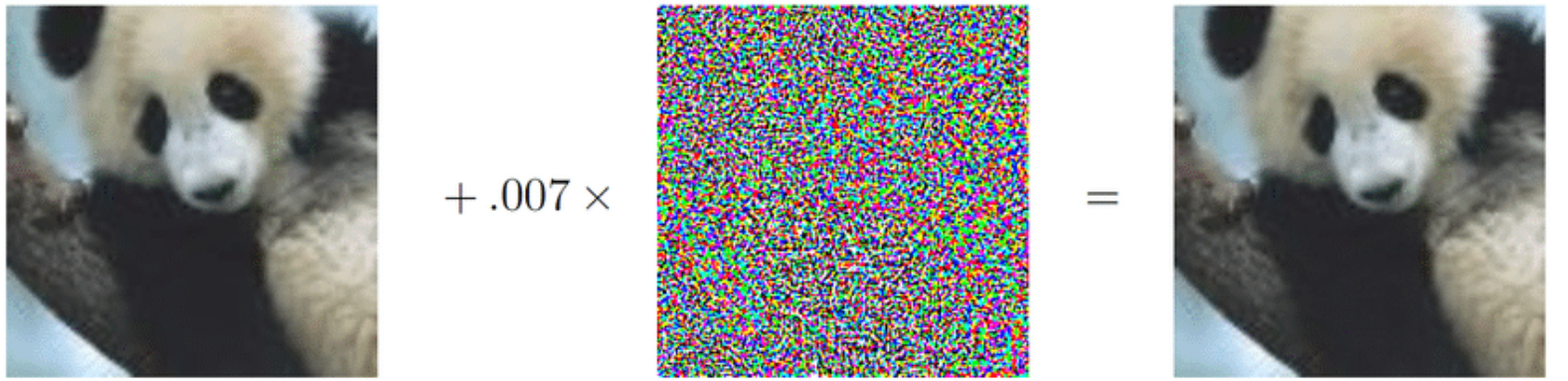
IJCAI-ECAI 2022

# A Unified Framework for Adversarial Attack and Defense in Constrained Feature Space

Thibault Simonetto, Salijona Dyrmishi, Salah Ghamizi, Maxime Cordy, Yves Le Traon

University of Luxembourg

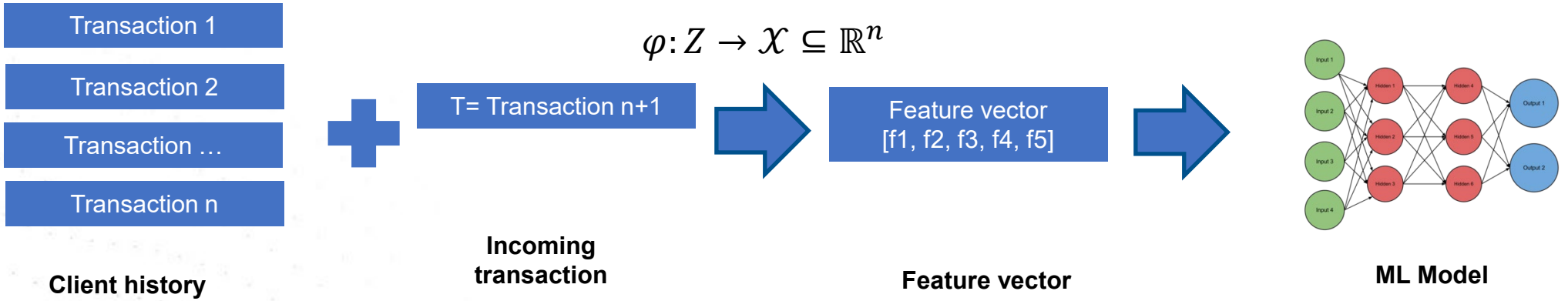
# Evaluating the robustness with adversarial example




J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014

# Motivation

ML model is integrated in a larger software system that takes as **input domain objects.**




Domain object Space $Z$
respects some **natural condition**

Feature space $\mathcal{X}_\Omega$
respects a **set of constraints** $\Omega$

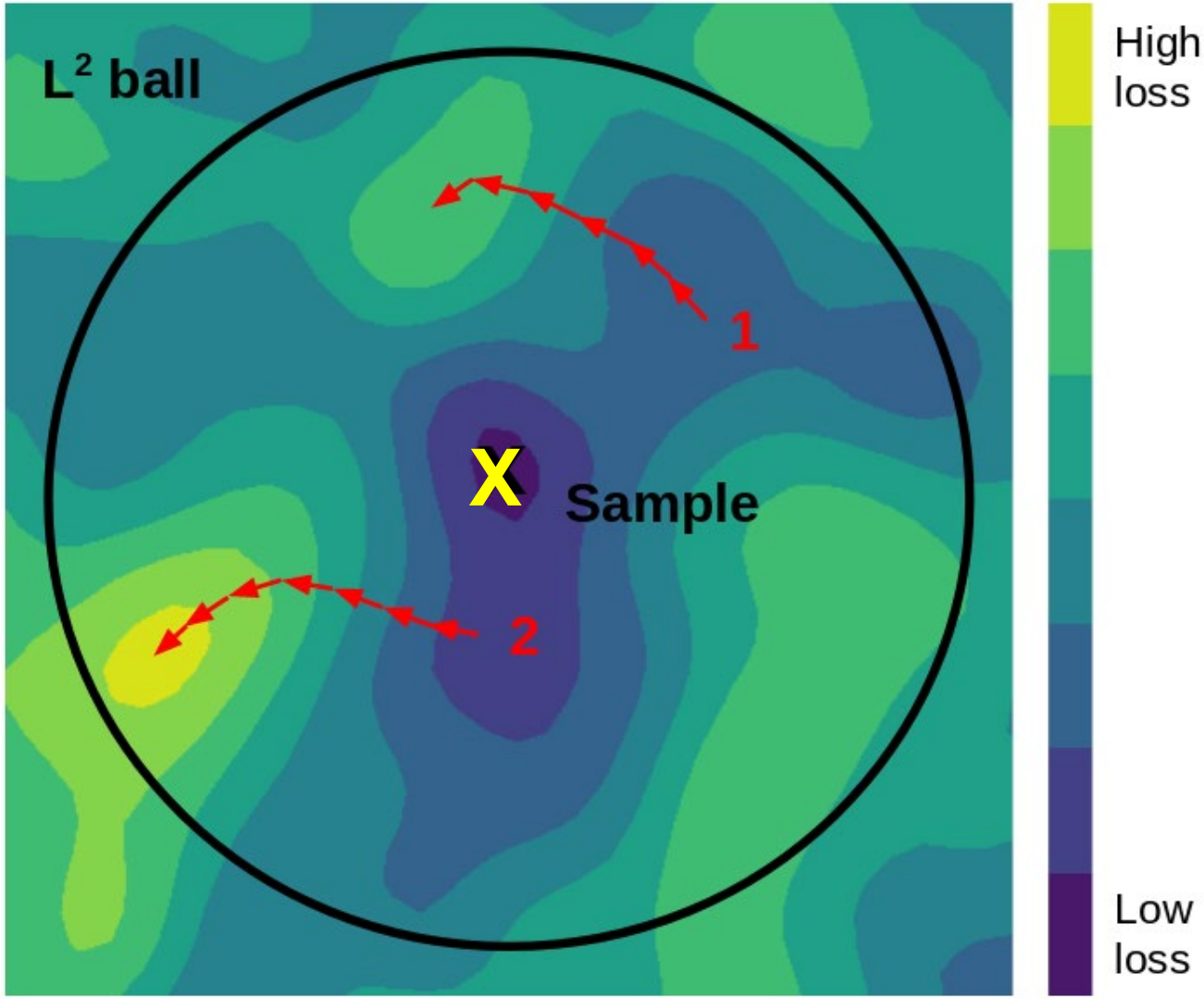
# Constraints on feature space

Related features linked by a constraint:

$$open\_acc \leq total\_acc$$

$$installment = loam\_amount \times \frac{int\_rate \times (1 + int\_rate)^{term}}{(1 + int\_rate)^{term} - 1}$$

# Existing attacks fails to generate constrained examples



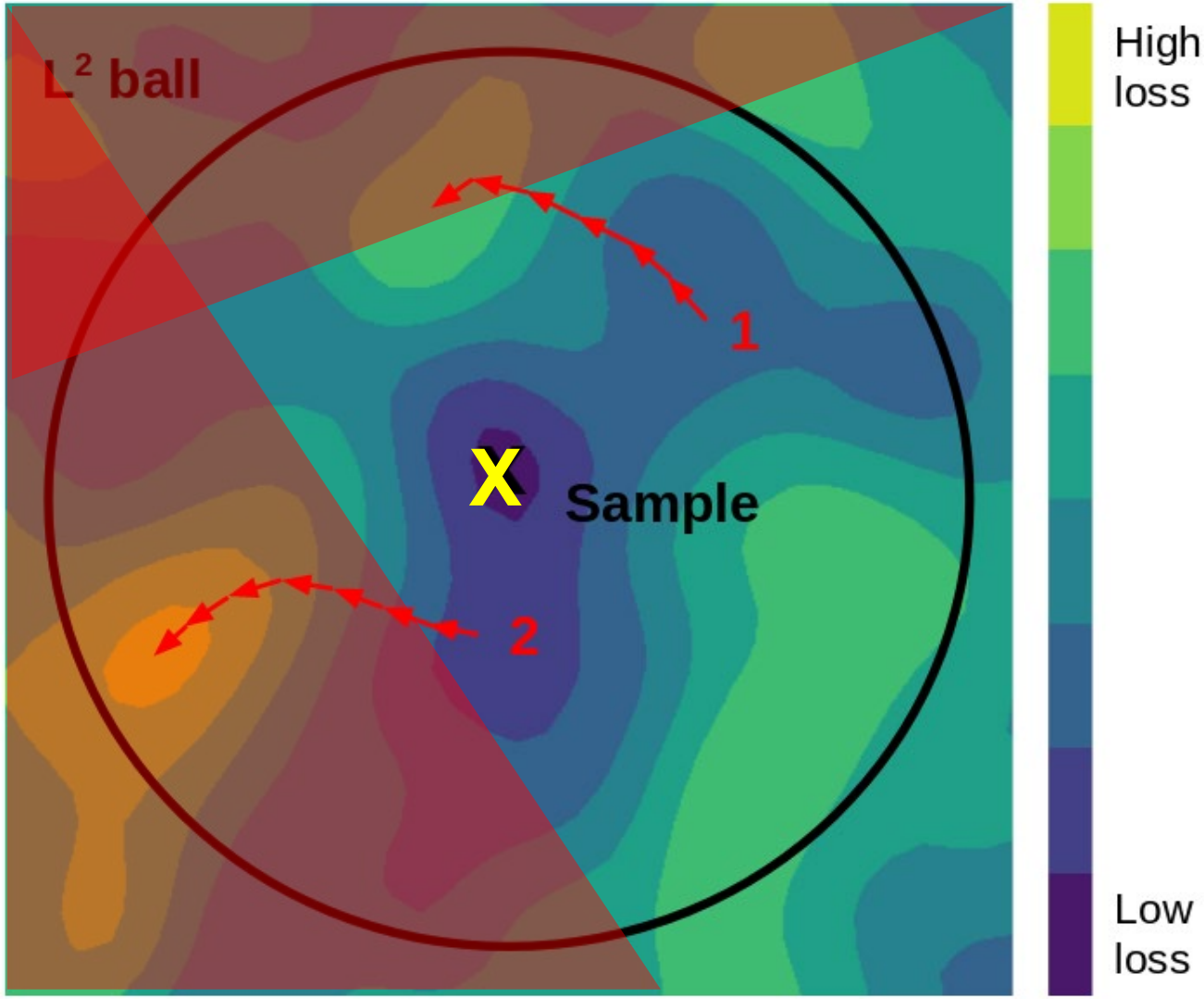

Objective: for $x$ find $\delta$

✓ With $H(x) \neq H(x+\delta)$

✓ With $L_p(x, x+\delta) < \epsilon$

Image modified from https://towardsdatascience.com/know-your-enemy-7f7c5038bdf3

# Existing attacks fails to generate constrained examples

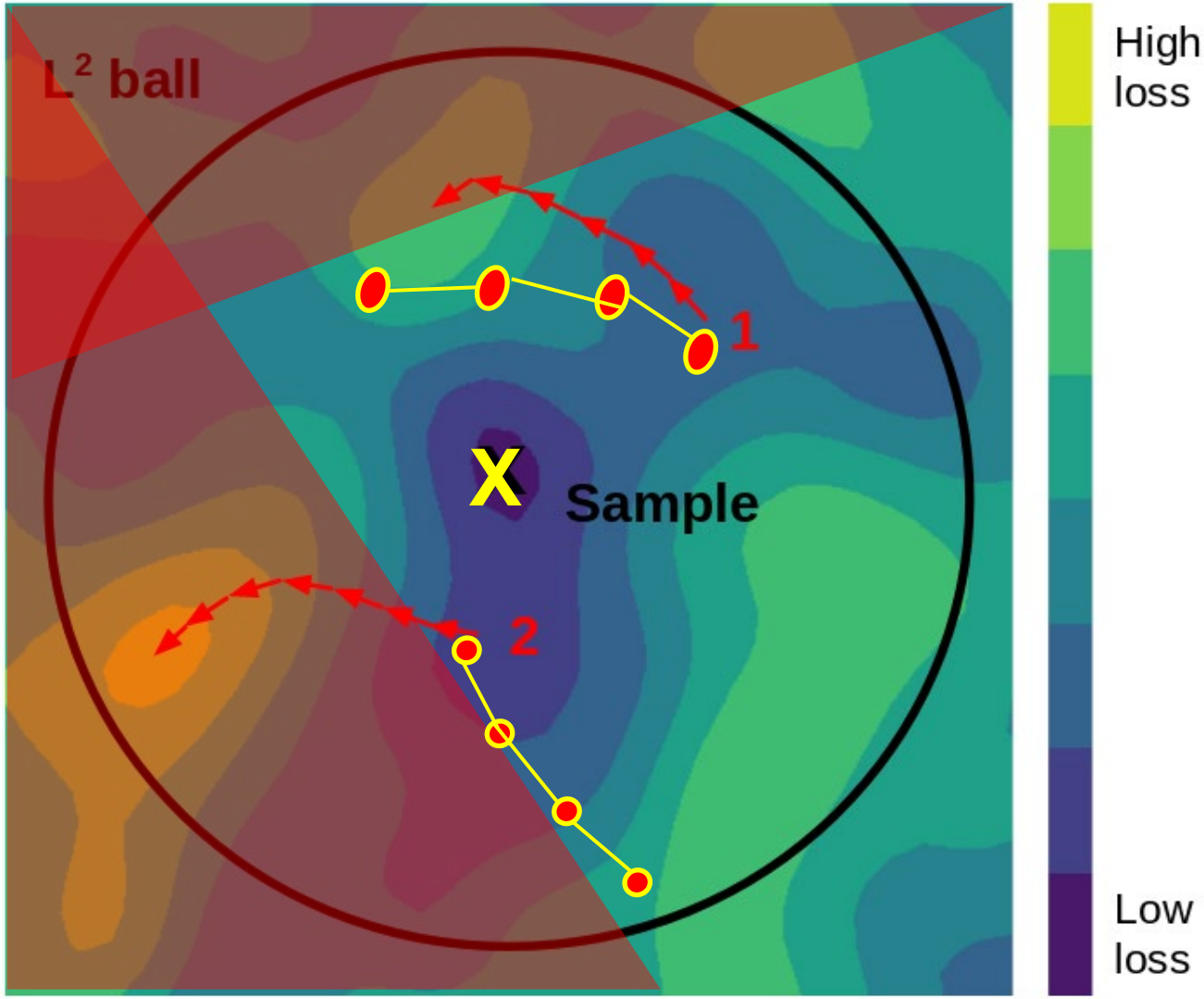



Objective: for $x$ find $\delta$

✓ With $H(x) \neq H(x+\delta)$

✓ With $L_p(x, x+\delta) < \epsilon$

# Existing attacks fails to generate constrained examples




Objective: for $x$ find $\delta$

- ✓ With $H(x) \neq H(x+\delta)$
- ✓ With $L_p(x, x+\delta) < \epsilon$
- ✓ $x + \delta \in \mathcal{X}_\Omega$

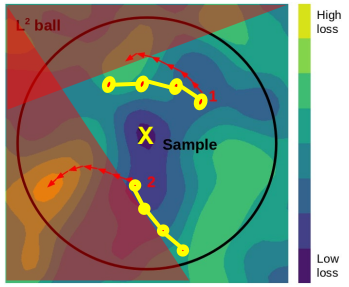
# Contributions

First **generic** framework for adversarial attacks under **domain constraints**

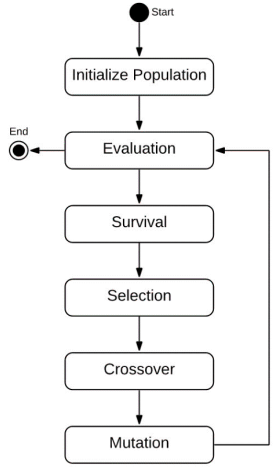
$$\omega := \omega_1 \wedge \omega_2 \mid \omega_1 \vee \omega_2 \mid \psi_1 \succeq \psi_2 \mid f \in \{\psi_1 \dots \psi_k\}$$
$$\psi := c \mid f \mid \psi_1 \oplus \psi_2 \mid x_i$$

| Constraints formulae | Penalty function |
|---|---|
| $\omega_1 \wedge \omega_2$ | $\omega_1 + \omega_2$ |
| $\omega_1 \vee \omega_2$ | $\min(\omega_1, \omega_2)$ |
| $\psi \in \Psi = \{\psi_1, \dots \psi_k\}$ | $\min(\{\psi_i \in \Psi :\mid \psi - \psi_i \mid\})$ |
| $\psi_1 \leq \psi_2$ | $max(0, \psi_1 - \psi_2)$ |
| $\psi_1 < \psi_2$ | $max(0, \psi_1 - \psi_2 + \tau)$ |
| $\psi_1 = \psi_2$ | $\mid \psi_1 - \psi_2 \mid$ |

**Generic constraints language**




C-PGD




MoEvA2

**2 Attacks**

# Constraints as penalty function

Constraint grammar

$$\omega := \omega_1 \wedge \omega_2 \mid \omega_1 \vee \omega_2 \mid \psi_1 \succeq \psi_2 \mid f \in \{\psi_1 \dots \psi_k\}$$
$$\psi := c \mid f \mid \psi_1 \oplus \psi_2 \mid x_i$$

$\boldsymbol{f} \in \boldsymbol{F}$ is the value of feature $f$ for a given input $x'$,
$\boldsymbol{c}$ is a constant real value,
$\boldsymbol{\omega}, \boldsymbol{\omega_1}, \boldsymbol{\omega_2}$ are constraint formulae,
$\succcurlyeq \in \{<, \leq, =, \neq, \geq, >\}$,
$\boldsymbol{\psi}, \boldsymbol{\psi_1}, \dots, \boldsymbol{\psi_k}$ are numeric expressions,
$\oplus \in \{+, -, *, \backslash\}$, and
$\boldsymbol{x_i}$ is the value of the $i^{\text{th}}$ feature of the clean input $x$

# Constraints as penalty function

Constraint grammar

$$\omega := \omega_1 \wedge \omega_2 \mid \omega_1 \vee \omega_2 \mid \psi_1 \succeq \psi_2 \mid f \in \{\psi_1 \dots \psi_k\}$$
$$\psi := c \mid f \mid \psi_1 \oplus \psi_2 \mid x_i$$

$\boldsymbol{f} \in \boldsymbol{F}$ is the value of feature $f$ for a given input $x'$,
$\boldsymbol{c}$ is a constant real value,
$\boldsymbol{\omega}, \boldsymbol{\omega_1}, \boldsymbol{\omega_2}$ are constraint formulae,
$\succcurlyeq \in \{<, \leq, =, \neq, \geq, >\}$,
$\boldsymbol{\psi}, \boldsymbol{\psi_1}, \dots, \boldsymbol{\psi_k}$ are numeric expressions,
$\oplus \in \{+, -, *, \backslash\}$, and
$\boldsymbol{x_i}$ is the value of the $i^{\text{th}}$ feature of the clean input $x$

# Constraints as penalty function

Constraint grammar

$$\omega := \omega_1 \wedge \omega_2 \mid \omega_1 \vee \omega_2 \mid \psi_1 \succeq \psi_2 \mid f \in \{\psi_1 \dots \psi_k\}$$
$$\psi := c \mid f \mid \psi_1 \oplus \psi_2 \mid x_i$$

$\boldsymbol{f} \in \boldsymbol{F}$ is the value of feature $f$ for a given input $x'$,
$\boldsymbol{c}$ is a constant real value,
$\boldsymbol{\omega}, \boldsymbol{\omega_1}, \boldsymbol{\omega_2}$ are constraint formulae,
$\succcurlyeq \in \{<, \leq, =, \neq, \geq, >\}$,
$\boldsymbol{\psi}, \boldsymbol{\psi_1}, \dots, \boldsymbol{\psi_k}$ are numeric expressions,
$\oplus \in \{+, -, *, \setminus\}$, and
$\boldsymbol{x_i}$ is the value of the $i^{\text{th}}$ feature of the clean input $x$

Mapping to penalty functions

| Constraints formulae | Penalty function |
|---|---|
| $\omega_1 \wedge \omega_2$ | $\omega_1 + \omega_2$ |
| $\omega_1 \vee \omega_2$ | $\min(\omega_1, \omega_2)$ |
| $\psi \in \Psi = \{\psi_1, \dots \psi_k\}$ | $\min(\{\psi_i \in \Psi :\mid \psi - \psi_i \mid\})$ |
| $\psi_1 \leq \psi_2$ | $max(0, \psi_1 - \psi_2)$ |
| $\psi_1 < \psi_2$ | $max(0, \psi_1 - \psi_2 + \tau)$ |
| $\psi_1 = \psi_2$ | $\mid \psi_1 - \psi_2 \mid$ |

Table 1: From constraint formulae to penalty functions. $\tau$ is an infinitesimal value.

**Constraint is satisfied if and only if $g(\boldsymbol{\omega}, \boldsymbol{x}) = 0$**

# Constraints as penalty function

Constraint grammar

$$\omega := \omega_1 \wedge \omega_2 \mid \omega_1 \vee \omega_2 \mid \psi_1 \succeq \psi_2 \mid f \in \{\psi_1 \dots \psi_k\}$$
$$\psi := c \mid f \mid \psi_1 \oplus \psi_2 \mid x_i$$

$\boldsymbol{f} \in \boldsymbol{F}$ is the value of feature $f$ for a given input $x'$,
$\boldsymbol{c}$ is a constant real value,
$\boldsymbol{\omega}, \boldsymbol{\omega_1}, \boldsymbol{\omega_2}$ are constraint formulae,
$\succcurlyeq \in \{<, \leq, =, \neq, \geq, >\}$,
$\boldsymbol{\psi}, \boldsymbol{\psi_1}, \dots, \boldsymbol{\psi_k}$ are numeric expressions,
$\oplus \in \{+, -, *, \setminus\}$, and
$\boldsymbol{x_i}$ is the value of the $i^{\text{th}}$ feature of the clean input $x$

Mapping to penalty functions

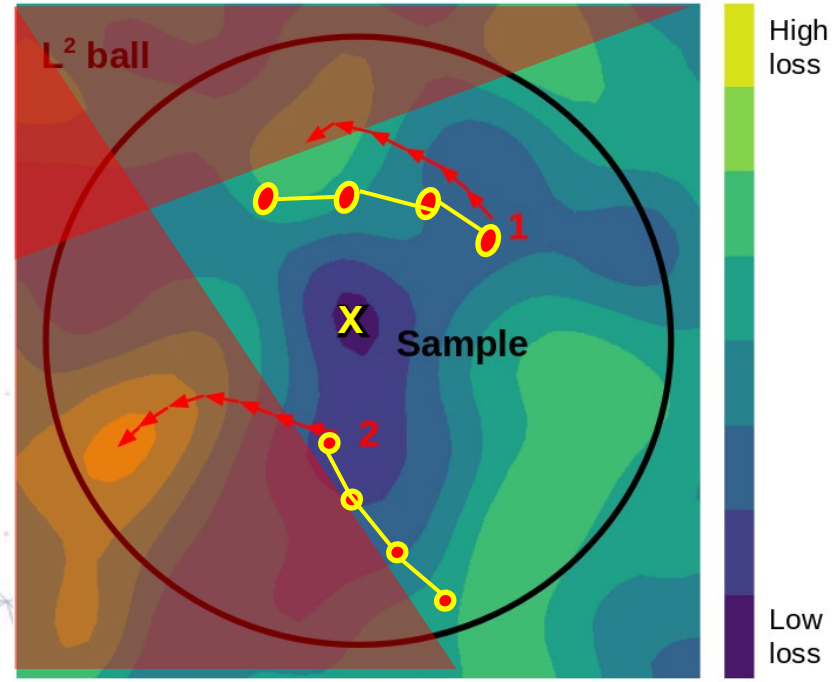
| Constraints formulae | Penalty function |
|---|---|
| $\omega_1 \wedge \omega_2$ | $\omega_1 + \omega_2$ |
| $\omega_1 \vee \omega_2$ | $\min(\omega_1, \omega_2)$ |
| $\psi \in \Psi = \{\psi_1, \dots \psi_k\}$ | $\min(\{\psi_i \in \Psi :\mid \psi - \psi_i \mid\})$ |
| $\psi_1 \leq \psi_2$ | $max(0, \psi_1 - \psi_2)$ |
| $\psi_1 < \psi_2$ | $max(0, \psi_1 - \psi_2 + \tau)$ |
| $\psi_1 = \psi_2$ | $\mid \psi_1 - \psi_2 \mid$ |

Table 1: From constraint formulae to penalty functions. $\tau$ is an infinitesimal value.

**Constraint is satisfied if and only if $g(\boldsymbol{\omega}, \boldsymbol{x}) = 0$**

**Sufficient expressiveness to instantiate constraints in different domains**

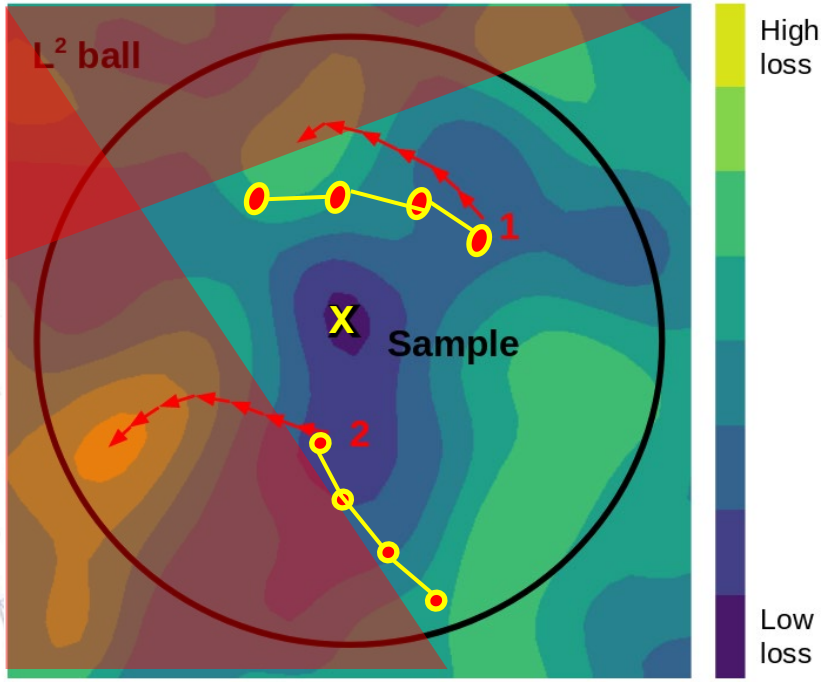
# Approach 1: C-PGD: Gradient evaluation of the constraints




**Projected Gradient Descent (PGD)**

$$x^{t+1} = x^t + \alpha\, sign(\nabla_{x^t} loss(h(x^t), y))$$

# Approach 1: C-PGD: Gradient evaluation of the constraints




**Projected Gradient Descent (PGD)**

$$x^{t+1} = x^t + \alpha\ sign(\nabla_{x^t} loss(h(x^t), y))$$

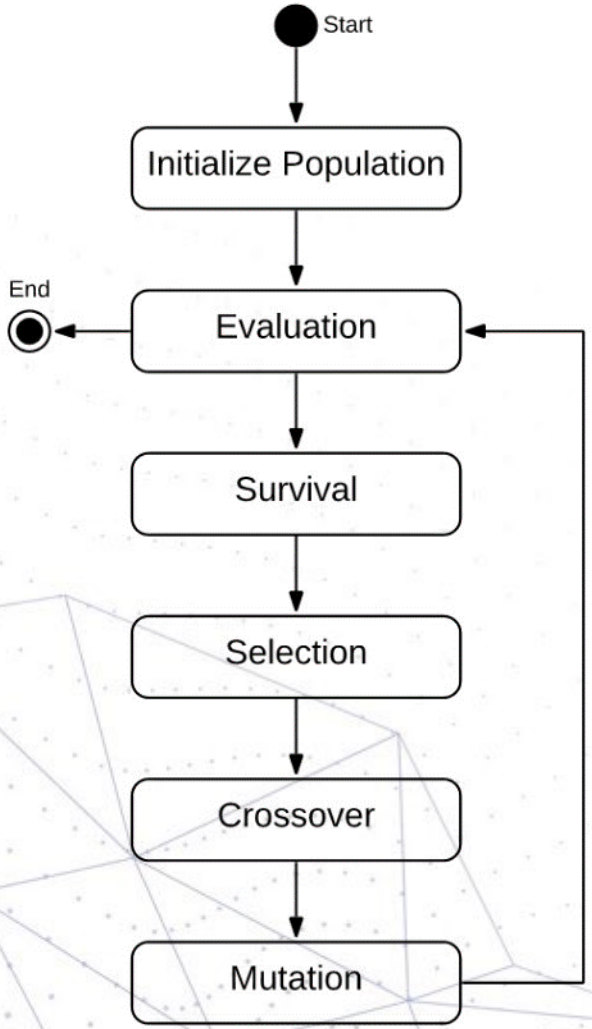
**Constraints regularization**

$$\nabla_{x^t} loss(h(x^t), y) - \nabla_{x^t} penalty(x^t)$$

**Constrained Projected Gradient Descent (C-PGD)**

$$x^{t+1} = x^t + \alpha\ sign\left(\nabla_{x^t} loss\left(h(x^t), y\right) - \nabla_{x^t} penalty(x^t)\right)$$

# Approach 2: MoEvA2: Evolutionary approach




**Multi-objective** genetic algorithm (NSGA-III)

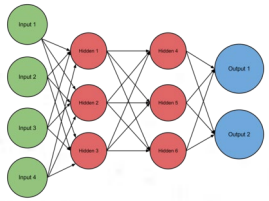
$$minimise\ g_1(x) \equiv h(x)$$

$$minimise\ g_2(x) \equiv\ L_p(x - x_0)$$

$$minimise\ g_3(x) \equiv \sum_{\omega_i \in \Omega} penalty(x, \omega_i)$$

# How effective are our approaches at generating adversarial examples?

# How effective are our approaches at generating adversarial examples?



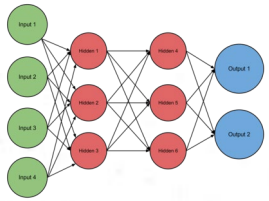
NN

Random Forest

| | Dataset | Attack |
|---|---|---|
|  | LCLD | PGD<br>PGD + SAT<br>C-PGD<br>MoEvA2 |
|  | CTU-13 | PGD<br>PGD + SAT<br>C-PGD<br>MoEvA2 |
|  | LCLD | Papernot *<br>MoEvA2 |
|  | CTU-13 | Papernot *<br>MoEvA2 |
|  | Malware | Papernot *<br>MoEvA2 |
|  | URL | Papernot *<br>MoEvA2 |

* Extended to random forest

# How effective are our approaches at generating adversarial examples?



NN

Random Forest

| | Dataset | Attack | C&M |
|---|---|---|---|
|  | LCLD | PGD | 0.00 |
| | | PGD + SAT | 0.00 |
| | | C-PGD | 9.85 |
| | | MoEvA2 | 97.48 |
|  | CTU-13 | PGD | 0.00 |
| | | PGD + SAT | 0.00 |
| | | C-PGD | 0.00 |
| | | MoEvA2 | 100.00 |
|  | LCLD | Papernot * | 0.00 |
| | | MoEvA2 | 41.51 |
|  | CTU-13 | Papernot * | 0.0 |
| | | MoEvA2 | 5.41 |
|  | Malware | Papernot * | 0.00 |
| | | MoEvA2 | 39.30 |
|  | URL | Papernot * | 8.50 |
| | | MoEvA2 | 31.89 |

**Attacks unaware of domain constraints often fail.**

**C-PGD is effective on a single dataset.**

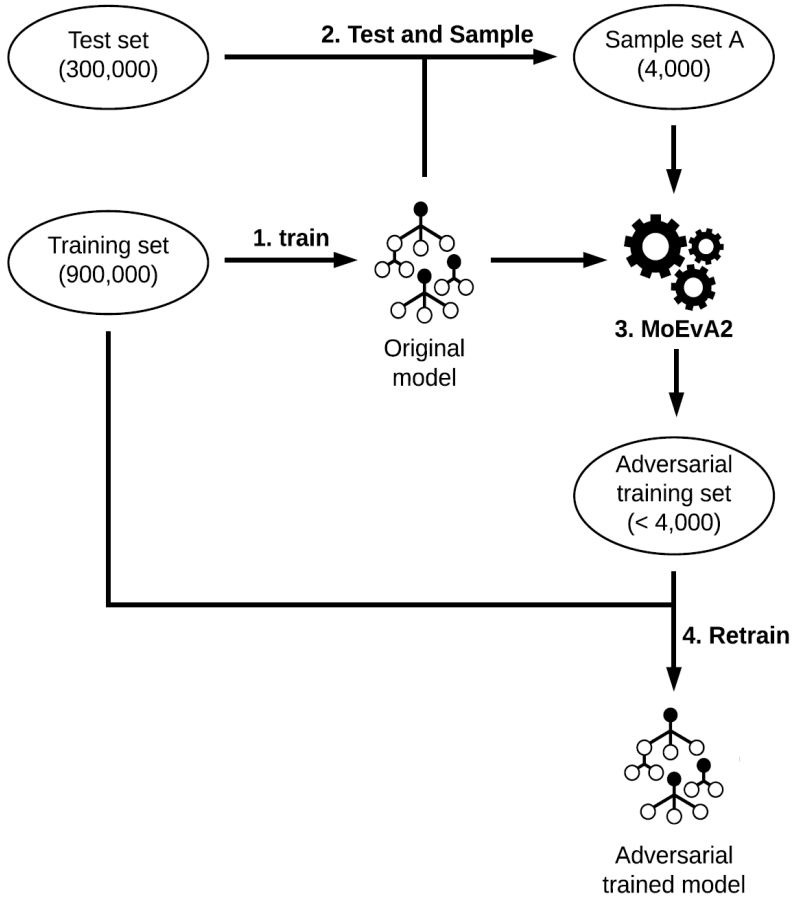
**MoEvA2 is always successful.**

* Extended to random forest

# How can we improve the robustness of our machine learning models?

A Unified Framework for Adversarial Attack and Defense in Constrained Feature Space. T. Simonetto, S. Dyrmishi, S. Ghamizi, M. Cordy, Y. Le Traon. 2022

# How can we improve the robustness of our machine learning models?




**Adversarial retraining**

# How can we improve the robustness of our machine learning models?

We hypothesize that **augmenting $\Omega$** with a set of engineered constraints can **robustify a model**.

# How can we improve the robustness of our machine learning models?

We hypothesize that **augmenting $\Omega$** with a set of engineered constraints can **robustify a model**.

We engineer a **new feature**

$$f_e = f_1 \oplus f_2$$

# How can we improve the robustness of our machine learning models?

We hypothesize that **augmenting $\Omega$** with a set of engineered constraints can **robustify a model**.

We engineer a **new feature**

$$f_e = f_1 \oplus f_2$$

We have the **new constraint**

$$\omega_e \vDash (f_e = f_1 \oplus f_2)$$

# How effective are defense techniques ?

| Defense | Attack | LCLD | CTU-13 |
|---|---|---|---|
| None | C-PGD | 9.85 | 0.00 |
| None | MoEvA2 | 97.48 | 100.00 |
| C-PGD Adv. retraining * | C-PGD | 8.78 | NA |
| C-PGD Adv. retraining * | MoEvA2 | 94.90 | NA |
| MoEvA2 Adv. retraining * | C-PGD | 2.70 | NA |
| MoEvA2 Adv. retraining * | MoEvA2 | 85.20 | 0.8 |
| Constraints augment. | C-PGD | 0.00 | NA |
| Constraints augment. | MoEvA2 | 80.43 | 0.00 |
| MoEvA2 Adv. retrain. † | MoEvA2 | 82.00 | NA |
| Combined defenses † | MoEvA2 | 77.43 | NA |

Table 3: Success rate of C-PGD and MoEvA2 after adversarial retraining and constraint augmentation (on neural networks). For a fair comparison, the model denoted by the same symbols (* or †) are trained with the same number of adversarial examples, generated from the same original samples.

**Constraint augmentation is an effective alternative defense** to adversarial retraining.

**Constraint augmentation and adversarial retraining have complementary effects.**

# Conclusion

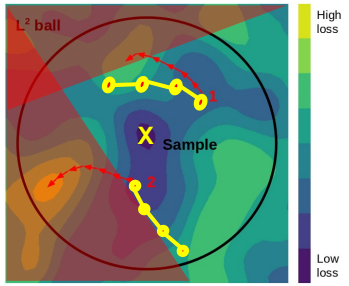
| Constraints formulae | Penalty function |
|---|---|
| $\omega_1 \wedge \omega_2$ | $\omega_1 + \omega_2$ |
| $\omega_1 \vee \omega_2$ | $\min(\omega_1, \omega_2)$ |
| $\psi \in \Psi = \{\psi_1, \dots \psi_k\}$ | $\min(\{\psi_i \in \Psi :\mid \psi - \psi_i \mid\})$ |
| $\psi_1 \leq \psi_2$ | $max(0, \psi_1 - \psi_2)$ |
| $\psi_1 < \psi_2$ | $max(0, \psi_1 - \psi_2 + \tau)$ |
| $\psi_1 = \psi_2$ | $\mid \psi_1 - \psi_2 \mid$ |

Generic constraints language

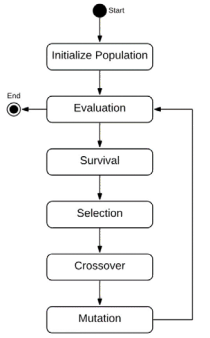
# Conclusion

| Constraints formulae | Penalty function |
|---|---|
| $\omega_1 \wedge \omega_2$ | $\omega_1 + \omega_2$ |
| $\omega_1 \vee \omega_2$ | $\min(\omega_1, \omega_2)$ |
| $\psi \in \Psi = \{\psi_1, \dots \psi_k\}$ | $\min(\{\psi_i \in \Psi :\mid \psi - \psi_i \mid\})$ |
| $\psi_1 \leq \psi_2$ | $max(0, \psi_1 - \psi_2)$ |
| $\psi_1 < \psi_2$ | $max(0, \psi_1 - \psi_2 + \tau)$ |
| $\psi_1 = \psi_2$ | $\mid \psi_1 - \psi_2 \mid$ |

Generic constraints language




C-PGD



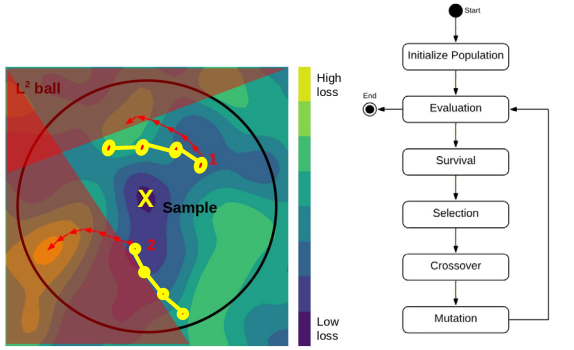

MoEvA2

2 Constrained Attacks

# Conclusion

| Constraints formulae | Penalty function |
|---|---|
| $\omega_1 \wedge \omega_2$ | $\omega_1 + \omega_2$ |
| $\omega_1 \vee \omega_2$ | $\min(\omega_1, \omega_2)$ |
| $\psi \in \Psi = \{\psi_1, \dots \psi_k\}$ | $\min(\{\psi_i \in \Psi :\mid \psi - \psi_i \mid\})$ |
| $\psi_1 \leq \psi_2$ | $max(0, \psi_1 - \psi_2)$ |
| $\psi_1 < \psi_2$ | $max(0, \psi_1 - \psi_2 + \tau)$ |
| $\psi_1 = \psi_2$ | $\mid \psi_1 - \psi_2 \mid$ |

Generic constraints language




C-PGD MoEvA2

2 Constrained Attacks

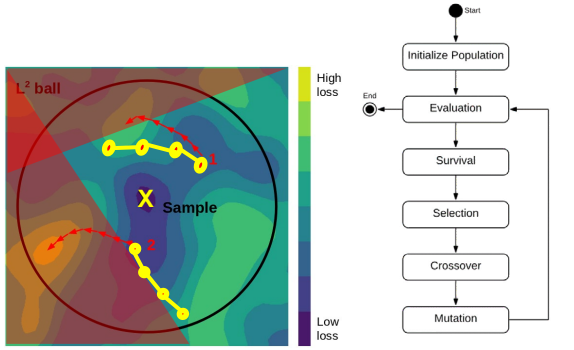
**Attacks unaware of domain constraints often fail.**

**MoEvA2 is always successful.**

# Conclusion

| Constraints formulae | Penalty function |
|---|---|
| $\omega_1 \wedge \omega_2$ | $\omega_1 + \omega_2$ |
| $\omega_1 \vee \omega_2$ | $\min(\omega_1, \omega_2)$ |
| $\psi \in \Psi = \{\psi_1, \ldots \psi_k\}$ | $\min(\{\psi_i \in \Psi :\mid \psi - \psi_i \mid\})$ |
| $\psi_1 \leq \psi_2$ | $max(0, \psi_1 - \psi_2)$ |
| $\psi_1 < \psi_2$ | $max(0, \psi_1 - \psi_2 + \tau)$ |
| $\psi_1 = \psi_2$ | $\mid \psi_1 - \psi_2 \mid$ |

Generic constraints language




C-PGD MoEvA2

2 Constrained Attacks

**Attacks unaware of domain constraints often fail.**

**MoEvA2 is always successful.**

$$\omega_e \vDash (f_e = f_1 \oplus f_2)$$

Constrained Augmentation

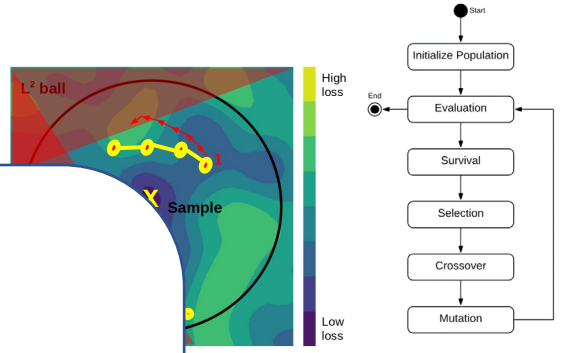
New defense method as effective as adversarial retraining

A Unified Framework for Adversarial Attack and Defense in Constrained Feature Space. T. Simonetto, S. Dyrmishi, S. Ghamizi, M. Cordy, Y. Le Traon. 2022

# Conclusion

| Constraints formulae | Penalty function |
|---|---|
| $\omega_1 \wedge \omega_2$ | $\omega_1 + \omega_2$ |
| $\omega_1 \vee \omega_2$ | $\min(\omega_1, \omega_2)$ |
| $\psi \in \Psi = \{\psi_1, \dots \psi_k\}$ | $\min(\{\psi_i \in \Psi$ |
| $\psi_1 \leq \psi_2$ | $max(0, \psi_1 -$ |
| $\psi_1 < \psi_2$ | $max(0, \psi_1 -$ |
| $\psi_1 = \psi_2$ | $\mid \psi_1 - \psi_2 \mid$ |

Generic constraints lang




PGD




MoEvA2

Constrained Attacks

**Checkout our framework**

**Stand 149 Row 5**



**Attacks unaware of do constraints often fa**

**MoEvA2 is always successful.**

$\models (f_e = f_1 \oplus f_2)$

trained Augmentation

New defense method as effective as adversarial retraining