Roadmap

1. Introduction
2. Construction of adversarial examples
3. Improving robustness
4. Certifiable robustness
   - Exact certification
   - **Convex relaxations**
   - Lipschitz-continuity
   - Randomized smoothing

Certification via Convex Relaxation

Recall our **goal**: develop an algorithm that answers the question:

"Is the classifier f_θ around the sample $\mathbf{x}$ adversarial-free (within an ϵ-ball measured by some norm)?"

Exact certification returns **`YES`** if and only if there is no adversarial example within an ϵ ball around the input sample (**`NO`** otherwise)

Now we allow the following answers:

- **`YES`**: We must have that for all $\tilde{\mathbf{x}} \in \mathcal{P}(\mathbf{x})$: $\arg\max F(\tilde{\mathbf{x}}) = \arg\max F(\mathbf{x})$
- **`POTENTIALLY NOT / MAYBE`**: In this case we have no guarantees.
- [**`NO`**: There must exist a $\tilde{\mathbf{x}} \in \mathcal{P}(\mathbf{x})$ such that $\arg\max F(\tilde{\mathbf{x}}) \neq \arg\max F(\mathbf{x})$]

$F(\tilde{\mathbf{x}}) = \hat{\mathbf{x}}^{(L)}$

Recall: Exact Certification

- We call $\mathrm{m_t} = F(\mathbf{x})_{c^*} - F(\mathbf{x})_t$ the classification margin of classes c^* and t.
- Worst-case margin (given $\mathcal{P}(\mathbf{x})$):

$$
\begin{aligned}
m_t^* = \min_{\tilde{\mathbf{x}}} \; & F(\tilde{\mathbf{x}})_{c^*} - F(\tilde{\mathbf{x}})_t \\
\textit{subject to} \quad & \|\tilde{\mathbf{x}} - \mathbf{x}\|_p \leq \epsilon \\
& \boldsymbol{y}^{(0)} = \tilde{\mathbf{x}} \\
& \hat{\mathbf{x}}^{(l)} = \boldsymbol{W}_l \boldsymbol{y}^{(l-1)} + \boldsymbol{b}_l \qquad \forall l = 1 \dots L \\
& \boldsymbol{y}^{(l)} = \mathrm{ReLU}\left(\hat{\mathbf{x}}^{(l)}\right) \qquad \forall l = 1 \dots L-1
\end{aligned}
$$

- $m_t^* > 0$: the classifier's prediction **cannot** be changed from class c^* to t

- As seen previously, solving for m_t^* is NP-hard.
- The (only) problem was the ReLU constraint

Idea: Relaxed Classification Margin

- Instead of solving the exact optimization problem

$$
\begin{aligned}
m_t^* = \min_{\tilde{\mathbf{x}}} \quad & F(\tilde{\mathbf{x}})_{c^*} - F(\tilde{\mathbf{x}})_t \\
\textit{subject to} \quad & \|\tilde{\mathbf{x}} - \mathbf{x}\|_p \le \epsilon \\
& \boldsymbol{y}^{(0)} = \tilde{\mathbf{x}} \\
& \hat{\mathbf{x}}^{(l)} = \boldsymbol{W}_l \boldsymbol{y}^{(l-1)} + \boldsymbol{b}_l \qquad \forall l = 1 \dots L \\
& \boldsymbol{y}^{(l)} = \mathrm{ReLU}\left(\hat{\mathbf{x}}^{(l)}\right) \qquad \forall l = 1 \dots L-1
\end{aligned}
$$

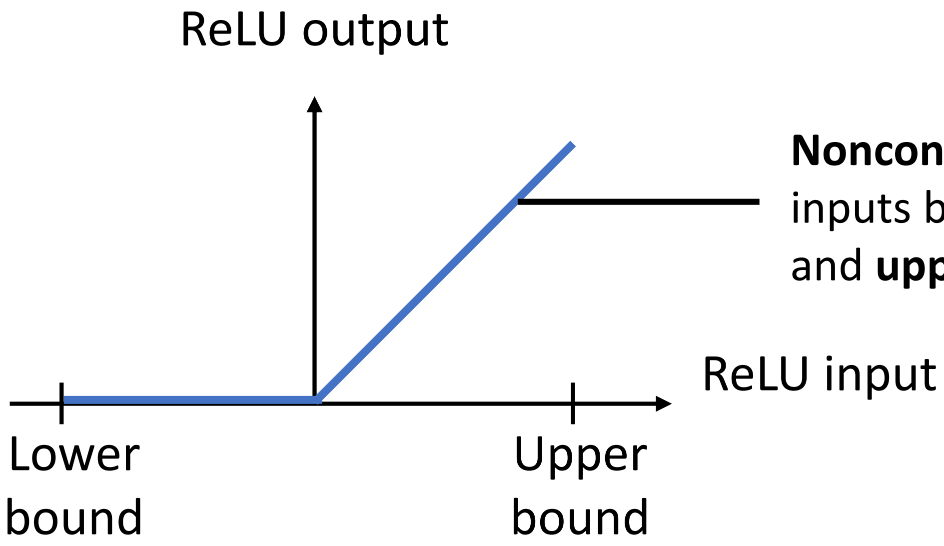
  we solve a **relaxed** optimization problem
  - E.g. with some constraints relaxed or removed

➢ Results in a **lower bound** $\underline{m}_t^*$ on the true minimum m_t^*.
  - If $\underline{m}_t^* > 0$: the classifier's prediction **cannot** be changed from class c^* to t
  - Can make the optimization possible in **polynomial time**.
  - The **price** we pay is that we cannot make a **`YES`** or **`NO`** statement in some cases.

ReLU: Nonconvex Output Set

Original constraint

$$y^{(l)} = \mathrm{ReLU}\left(\hat{\mathbf{x}}^{(l)}\right)$$

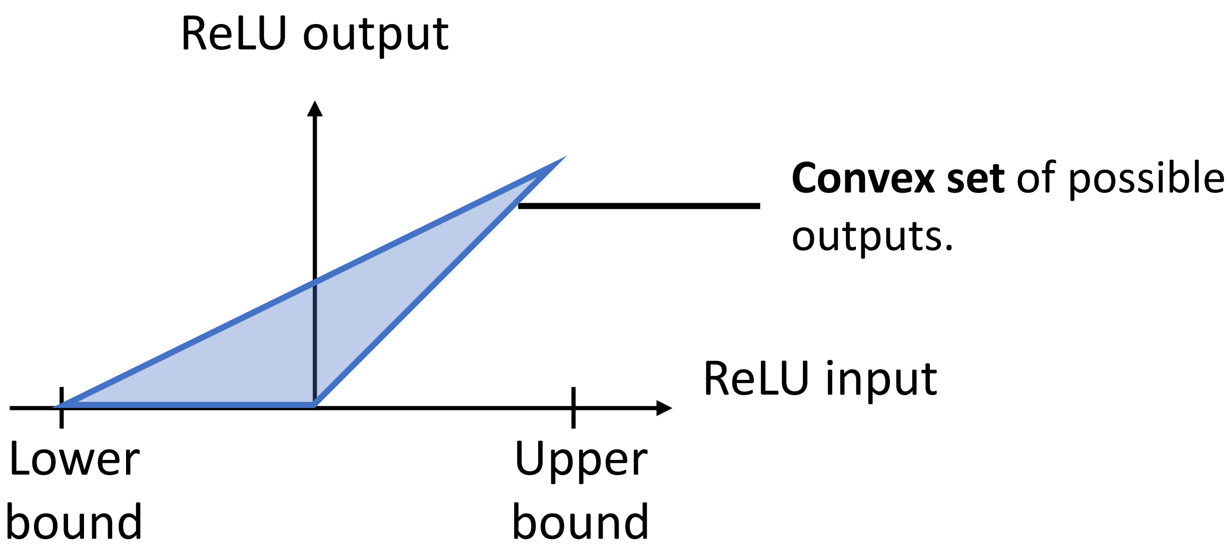


Nonconvex output set for inputs between the **lower** and **upper bound**

Convex ReLU Relaxation: Illustration

Relaxed constraint

$$\boldsymbol{y}^{(l)} \in \text{Triangle}\left(\hat{\mathbf{x}}^{(l)}\right)$$



Note: The output of the ReLU activation is **no longer deterministic** but a **variable** to optimize over (like the input)!

Convex ReLU Relaxation: Formal Definition

We replace the $\boldsymbol{y}^{(l)} = \mathrm{ReLU}(\hat{\mathbf{x}}^{(l)})$ constraint by (see [Wong and Kolter, 2018]):

- $\boldsymbol{y}_i^{(l)} \geq 0$
- $\boldsymbol{y}_i^{(l)} \geq \hat{\mathbf{x}}_i^{(l)}$
- $\left(\boldsymbol{u}_i^{(l)} - \boldsymbol{l}_i^{(l)}\right) \boldsymbol{y}_i^{(l)} - \boldsymbol{u}_i^{(l)} \hat{\mathbf{x}}_i^{(l)} \leq -\boldsymbol{u}_i^{(l)} \boldsymbol{l}_i^{(l)}$

Where $[\boldsymbol{l}^{(l)}, \boldsymbol{u}^{(l)}]$ denote the **element-wise lower** and **upper bounds** on the ReLU input at layer l, which we have encountered in the exact certification session.

These are **linear** constraints only!

Note: This relaxation is equivalent to relaxing the integer constraint $\boldsymbol{a}_i \in \{0, 1\}$ we have seen in the MILP to $\boldsymbol{a}_i \in [0,1]$.

Overall LP

- Since now all constraints are linear, we obtain a linear program (LP):

$$\underline{m}_t^* = \min_{\tilde{\mathbf{x}}, \boldsymbol{y}^{(l)}, \hat{\mathbf{x}}^{(l)}} [\hat{\mathbf{x}}^{(L)}]_{c^*} - [\hat{\mathbf{x}}^{(L)}]_t$$

$$\textit{subject to} \quad \mathbf{x}_i - \tilde{\mathbf{x}}_i \leq \epsilon \quad \forall i$$

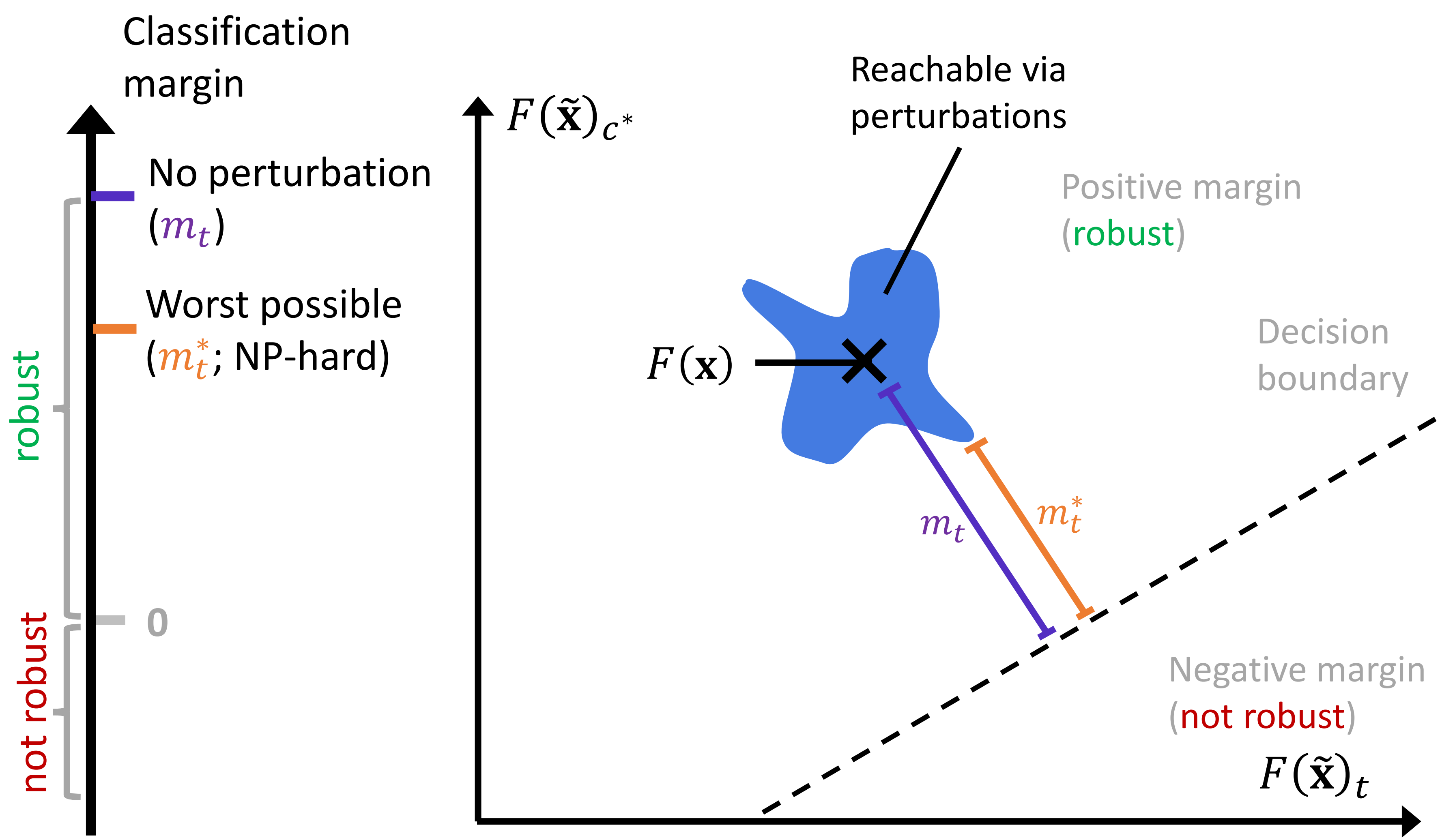
$$\tilde{\mathbf{x}}_i - \mathbf{x}_i \leq \epsilon \quad \forall i$$

$$\boldsymbol{y}^{(0)} = \tilde{\mathbf{x}}$$

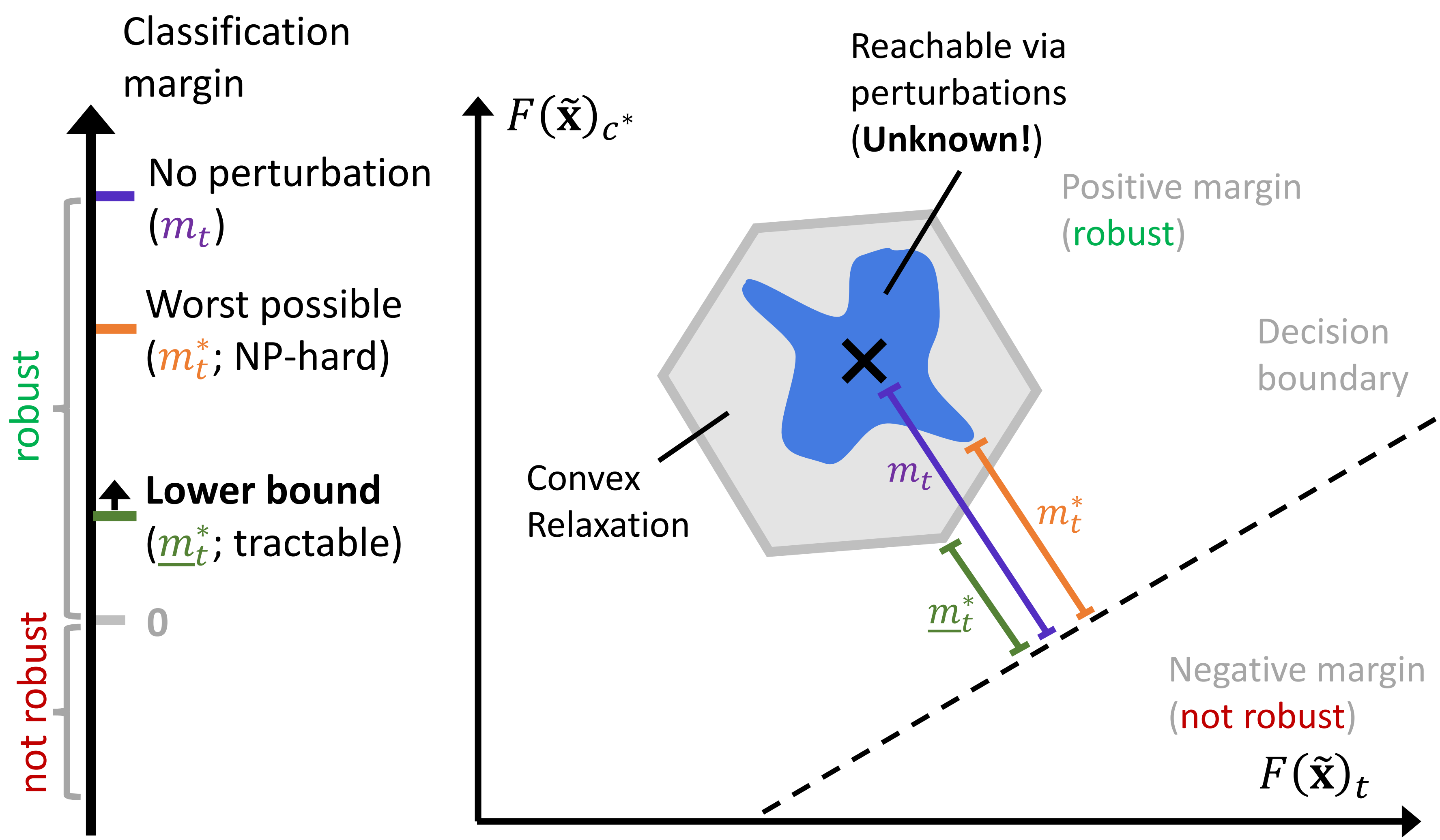
$$\hat{\mathbf{x}}^{(l)} = \boldsymbol{W}_l \boldsymbol{y}^{(l-1)} + \boldsymbol{b}_l \qquad \forall l = 1 \ldots L$$

$$\left.\begin{aligned} &\boldsymbol{y}_i^{(l)} \geq \hat{\mathbf{x}}_i^{(l)} \\ &\boldsymbol{y}_i^{(l)} \geq 0 \\ &\left(\boldsymbol{u}_i^{(l)} - \boldsymbol{l}_i^{(l)}\right) \boldsymbol{y}_i^{(l)} - \boldsymbol{u}_i^{(l)} \hat{\mathbf{x}}_i^{(l)} \leq -\boldsymbol{u}_i^{(l)} \boldsymbol{l}_i^{(l)} \end{aligned}\right. \qquad \begin{gathered} \forall l = 1 \ldots L-1 \\ \forall i \end{gathered}$$

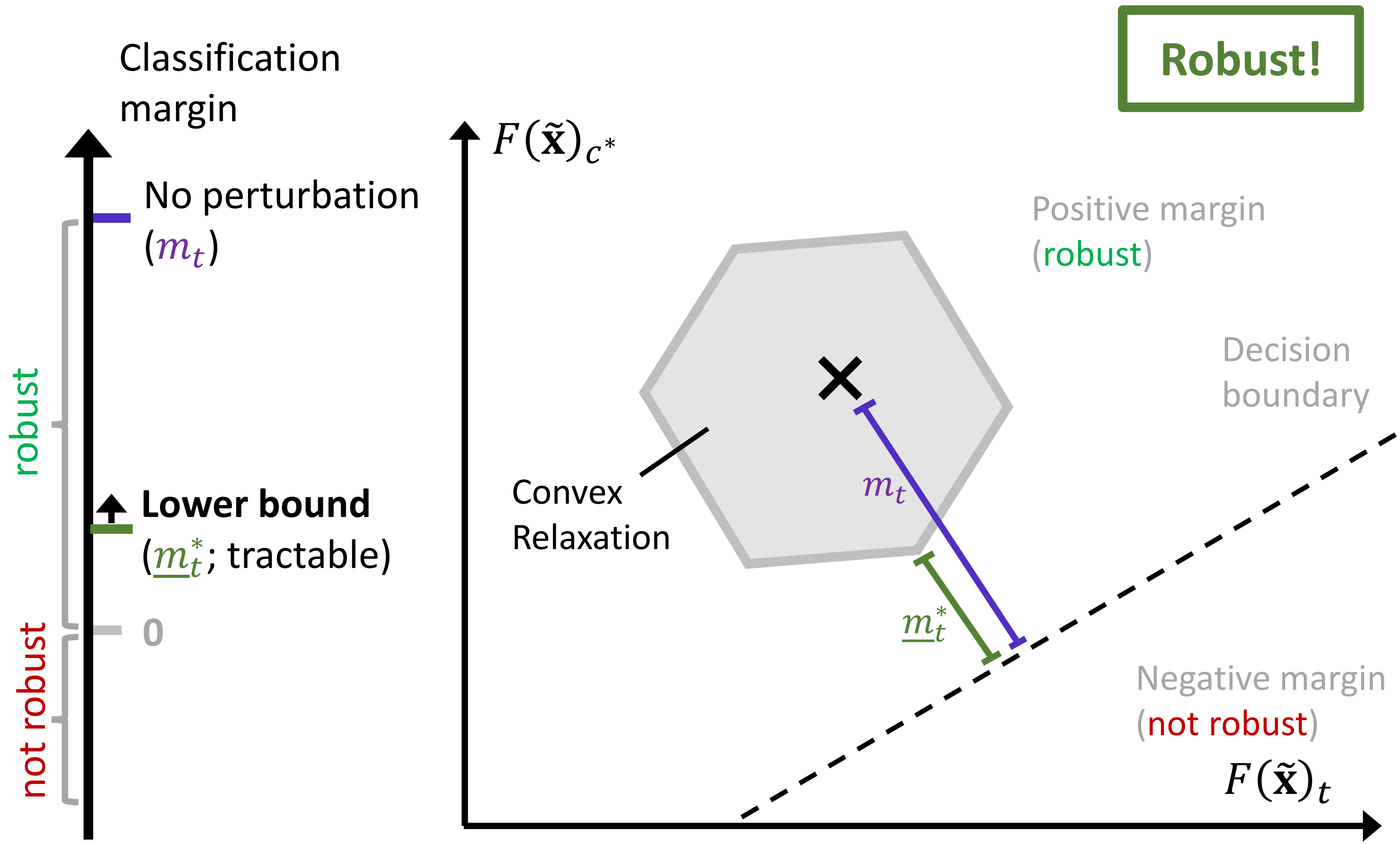
Recap: Exact Robustness Certification Illustration



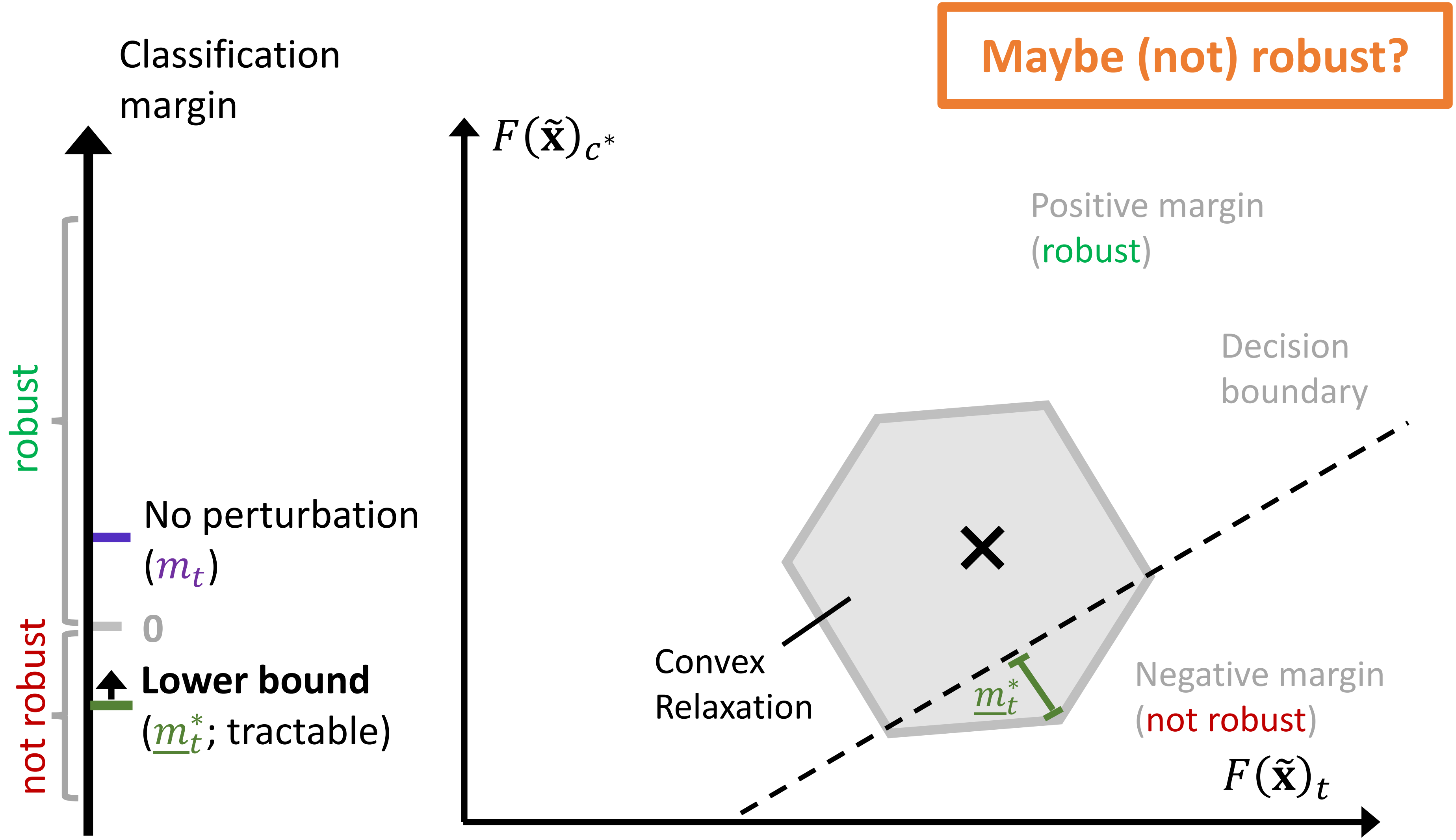
Robustness Certification via Convex Relaxation



Robustness Certification via Convex Relaxation



Robustness Certification via Convex Relaxation



Intermediate Summary

- By relaxing the ReLU activation function we have turned the problem of robustness certification into a **linear program** → tractable to compute
- The price we paid is that there are instances for which we **cannot decide**.

- The $\arg\min$ of the optimization is a **perturbed instance** $\tilde{\mathbf{x}}$.
  - We can feed it into the original neural network and observe whether the classification changes.
  - If yes: we have an **adversarial example** and therefore proven non-robustness → our algorithm can report "**NO**", i.e. not adversarial-free

- In contrast to exact certification, the **tightness of the lower and upper bounds** influence the quality of the relaxation, i.e. how often we return **`MAYBE`**.

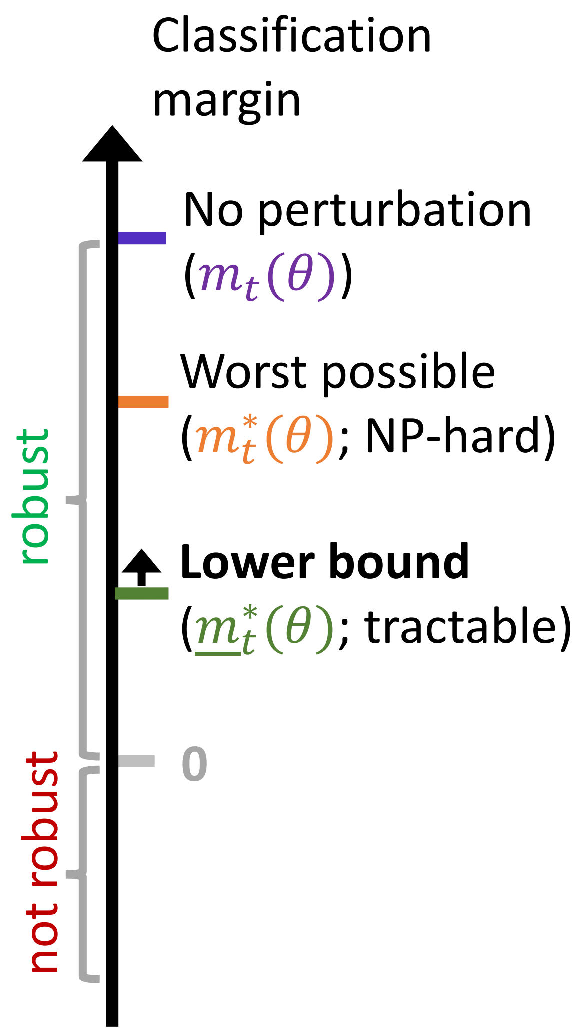
Questions – Rob2

1. When the optimal value from our **convex relaxation**, $\underline{m}_t^*$, is negative, this means that …
   a) The classifier is not robust (w.r.t. the current sample $\boldsymbol{x}$)
   b) The classifier is robust (w.r.t. the current sample $\boldsymbol{x}$)
   c) We cannot make a statement

2. Same question but now the **exact certification**, m_t^*, is negative

3. Can you think about scenarios where $m_t^* = \underline{m}_t^*$?

Notation: Margins w.r.t. θ



- Previously we assumed a fixed neural network with given weights/biases per layer
  - let's indicate all these parameters by θ

- Important: The optimization problems (objective + constraints) depend on θ; thus, also the optimal solutions (i.e. the margins!) of these problems depend on θ

➢ different weights θ lead to different (worst-case) margins

- During (robust) training we aim to find a good θ, e.g., via gradient descent

- To make this dependency explicit we write $m_t^*(\theta)$

Recall: Robust Training

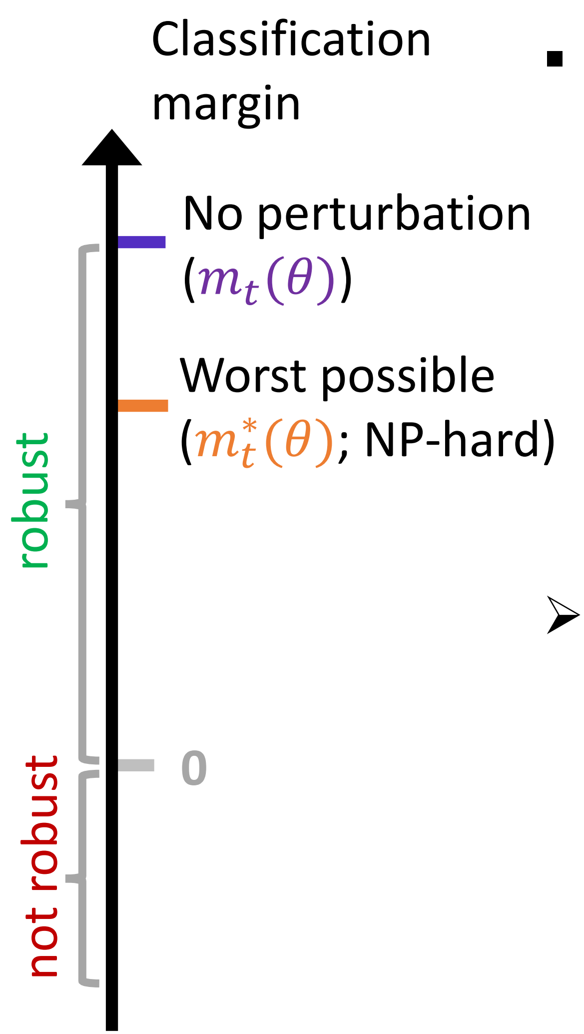
- In robust training we aim to optimize the robust loss w.r.t. θ:

$$\min_{\theta} \mathop{\mathbb{E}}_{(\mathbf{x},y)\in\mathbb{P}_{\text{data}}} \left[\sup_{\tilde{\mathbf{x}}\in\mathcal{P}(\mathbf{x})} \ell\left(f_\theta(\tilde{\mathbf{x}}), y\right) \right]$$

  - The challenge is how to compute $\nabla_\theta \left(\sup_{\tilde{\mathbf{x}}\in\mathcal{P}(\mathbf{x})} \ell\left(f_\theta(\tilde{\mathbf{x}}), y\right)\right)$

- The ReLU relaxation via an LP lends itself nicely to efficiently **optimize a robust loss** based on the certification
  - Note: here we focus on the general idea only

Robust Training



- To keep the discussion simple, let's assume the loss is the following margin loss:
  - if instance is correctly classified → loss = 0
  - if misclassified → loss = margin to the decision boundary (in logit space)

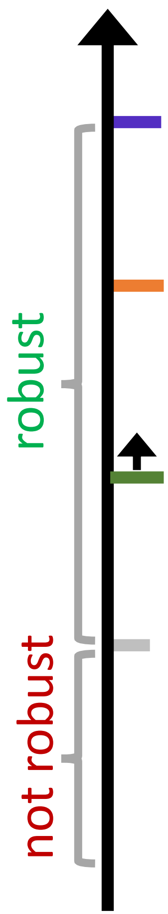
$$\ell\left(f_\theta(\tilde{\mathbf{x}}), y\right) = \max_{t \neq y} \max\left(F_\theta(\tilde{\mathbf{x}})_t - F_\theta(\tilde{\mathbf{x}})_y, 0\right)$$

➢ Thus, the supremum evaluates to

$$\sup_{\tilde{\mathbf{x}} \in \mathcal{P}(\mathbf{x})} \ell\left(f_\theta(\tilde{\mathbf{x}}), y\right) = \max_{t \neq y} \max\left(-m_t^*(\theta), 0\right)$$

Robust Training with Lower Bounds

Classification margin



No perturbation ($m_t(\theta)$)

Worst possible ($m_t^*(\theta)$; NP-hard)

Lower bound ($\underline{m}_t^*(\theta)$; tractable but still an optimization problem)

0

- To make it tractable, we instead optimize via the lower bound
  → i.e. we optimize a more "pessimistic" loss

$$\sup_{\tilde{\mathbf{x}} \in \mathcal{P}(\mathbf{x})} \ell\left(f_\theta(\tilde{\mathbf{x}}), y\right) = \max_{t \neq y} \max(-m_t^*(\theta), 0)$$

$$\leq \max_{t \neq y} \max(-\underline{m}_t^*(\theta), 0)$$

- Challenge: $\underline{m}_t^*(\theta)$ is obtained via an LP. Difficult/expensive to get the gradient $\nabla_\theta \underline{m}_t^*(\theta)$.

- Can we find another lower bound which does not require to solve an optimization problem?

Recap: Strong Duality

primal

$$\min_{\boldsymbol{x}} h_0(\boldsymbol{x})$$
$$\text{s.t. } h_i(\boldsymbol{x}) \leq 0 \quad i = 1 \dots M$$

dual

$$\max_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha})$$
$$\text{s.t. } \boldsymbol{\alpha}_i \geq 0 \quad i = 1 \dots M$$

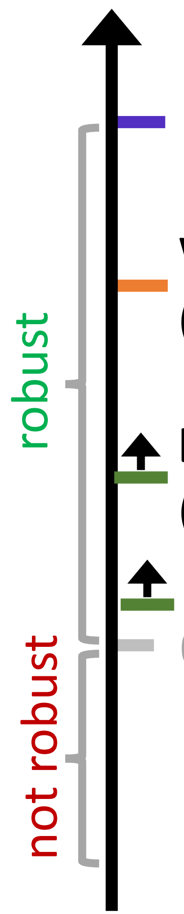
In our case, the primal is the LP from the slide „Overall LP"

In our case, both primal and dual depend additionally on θ. Thus, it would be more accurate to write $g_\theta(\boldsymbol{\alpha})$

$$h_0(\boldsymbol{x}') \geq h(\boldsymbol{x}^*) = g(\boldsymbol{\alpha}^*) \geq g(\boldsymbol{\alpha}')$$

any feasible $\boldsymbol{x}'$ optimal $\boldsymbol{x}^*$ optimal $\boldsymbol{\alpha}^*$ any feasible $\boldsymbol{\alpha}'$

Robust Training via Duality



Classification margin

No perturbation ($m_t(\theta)$)

Worst possible ($m_t^*(\theta)$; NP-hard)

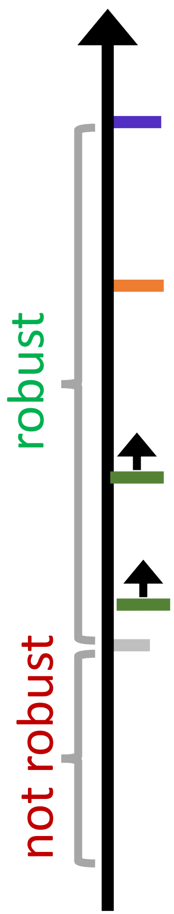
Lower bound = minimum of primal LP = maximum of dual LP
($\underline{m}_t^*(\theta)$; tractable but still an optimization problem)

2. Lower bound = value obtained for a feasible point of the dual LP
($\underline{m}_t^{dual}(\theta)$; tractable since just some analytical function)

- We do not need to perform optimization to get a lower bound
- Just plug in some feasible point $\boldsymbol{\alpha}'$ into the objective function of the dual LP
- $\underline{m}_t^{dual}(\theta) = g_\theta(\boldsymbol{\alpha}')$

Robust Training via Duality

Classification margin



No perturbation ($m_t(\theta)$)

Worst possible ($m_t^*(\theta)$; NP-hard)

$$\sup_{\tilde{\mathbf{x}} \in \mathcal{P}(\mathbf{x})} \ell\left(f_\theta(\tilde{\mathbf{x}}), y\right) = \max_{t \neq y} \max(-m_t^*(\theta), 0)$$

$$\leq \max_{t \neq y} \max(-\underline{m}_t^*(\theta), 0)$$

$$\leq \max_{t \neq y} \max(-\underline{m}_t^{dual}(\theta), 0)$$

$$= \max_{t \neq y} \max(-g_\theta(\boldsymbol{\alpha}'), 0)$$

Lower bound = minimum of primal LP = maximum of dual LP
($\underline{m}_t^*(\theta)$; tractable but still an optimization problem)

2. Lower bound = value obtained for a feasible point of the dual LP
($\underline{m}_t^{dual}(\theta)$; tractable since just some analytical function)

We reached our goal: $\nabla_\theta g_\theta(\boldsymbol{\alpha}')$ can be easily computed!

Summary

- In robust training we aim to optimize the robust loss:

$$\min_{\theta} \underset{(\mathbf{x}, y) \in \mathbb{P}_{\text{data}}}{\mathbb{E}} \left[\sup_{\tilde{\mathbf{x}} \in \mathcal{P}(\mathbf{x})} \ell\left(f_\theta(\tilde{\mathbf{x}}), y\right) \right]$$

- 1. We replaced the supremum by an even larger value (i.e. by a more tractable bound)
- 2. Instead of deriving the bound via an optimization problem, we used the concept of duality (any feasible point of the dual leads to a valid bound)

- Comparison to adversarial training: The supremum was replaced by a simple surrogate (loss evaluated at an adversarial point)

- In both cases
  - Computing the gradient ∇_θ of the bound/surrogate is (relatively) easy
  - You obtain ML models which are more robust

Recommended Reading

- Lecture 13: Certified Defenses II: Convex Relaxations of Jerry Li‘s course on Robustness in Machine Learning (CSE 599-M), https://jerryzli.github.io/robust-ml-fall19.html

References

- Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In International Conference on Machine Learning, pages 5283–5292, 2018.