Most slides are adapted from AAAI 2022 tutorial and from internet

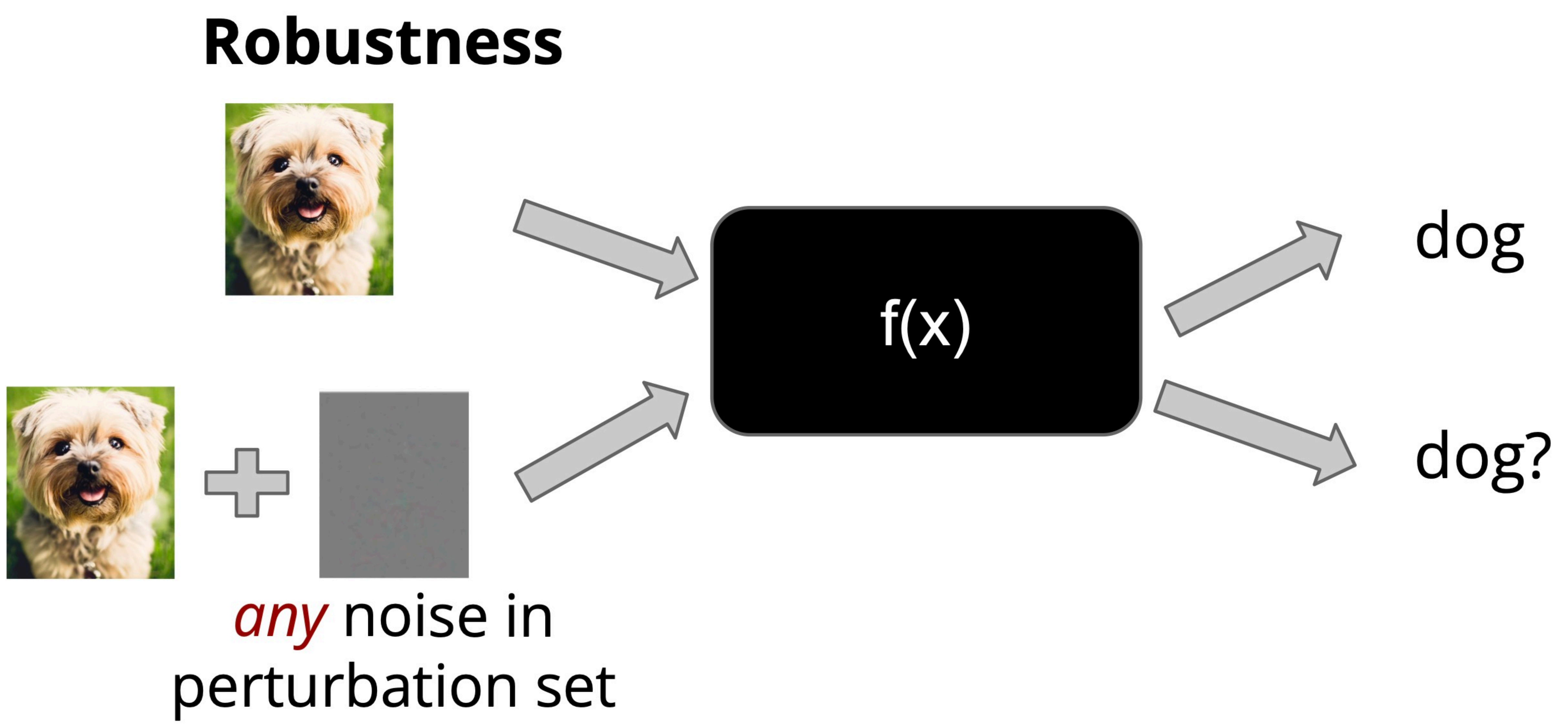
COMP6211I: Trustworthy Machine Learning

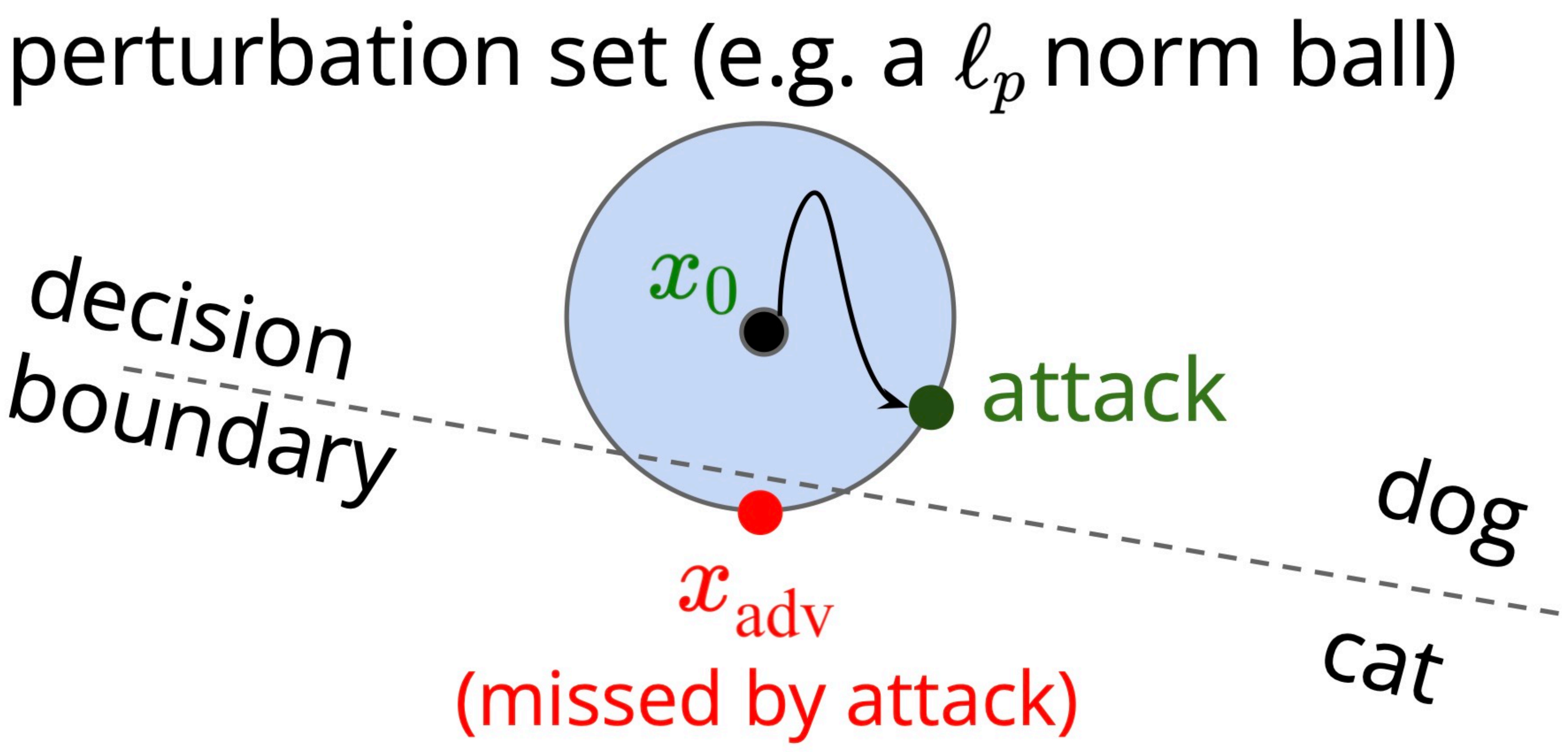
Test-time Integrity (verification) part 2

Minhao CHENG

THE DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

What is neural network verification?



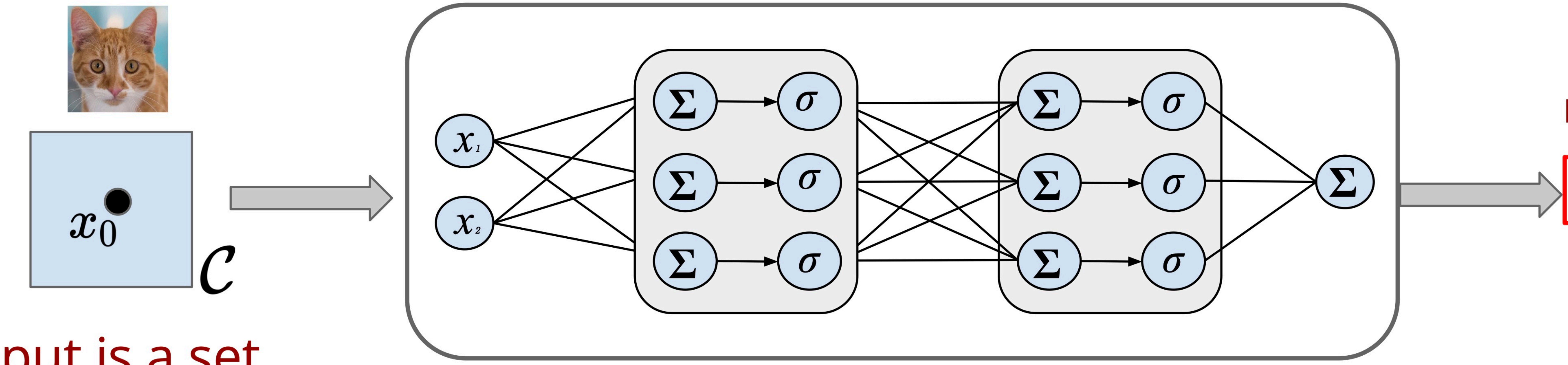


- Verification requires a formal proof to show the property holds
- In the robustness verification setting, a model can't be attack $\neq$ Verified
- Many heuristic defense was broken under stronger attacks
- A verified model cannot be attacked by any attacks (including unforeseen ones)

The Basic Formulation of Robustness Verification

Suppose $f(x_0) > 0$. Can we verify this property:

$$f(x) > 0, \forall x \in \mathcal{C}$$



Input is a <u>set</u>

Neural Network

Output is a range/set of scores

$$\boxed{0.2} \leq f(x) \leq 2.2$$

"cat" even in the worst case

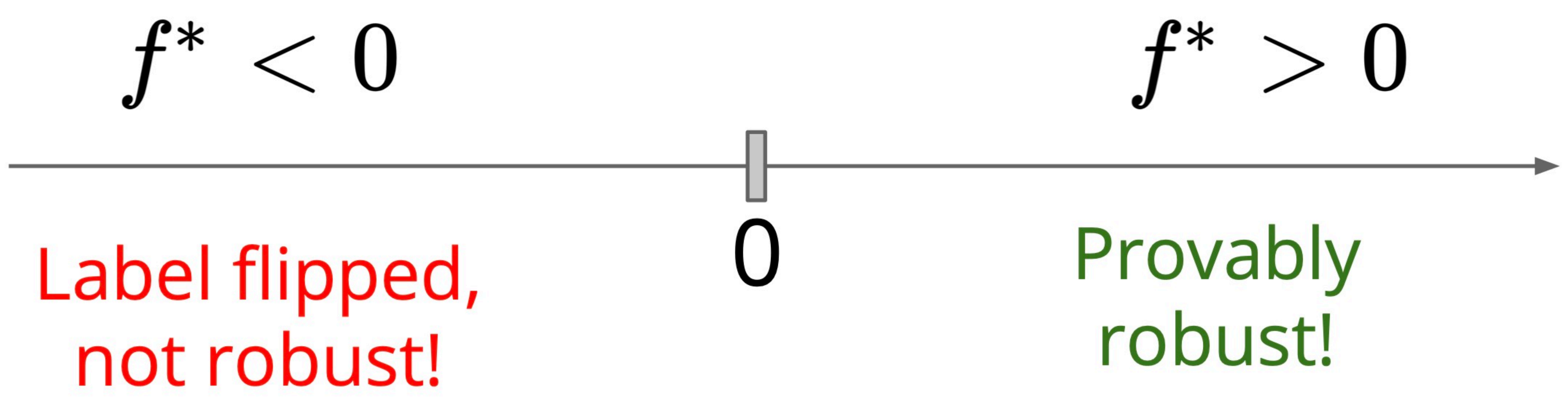
Must consider a set of infinite points as the input of the NN.

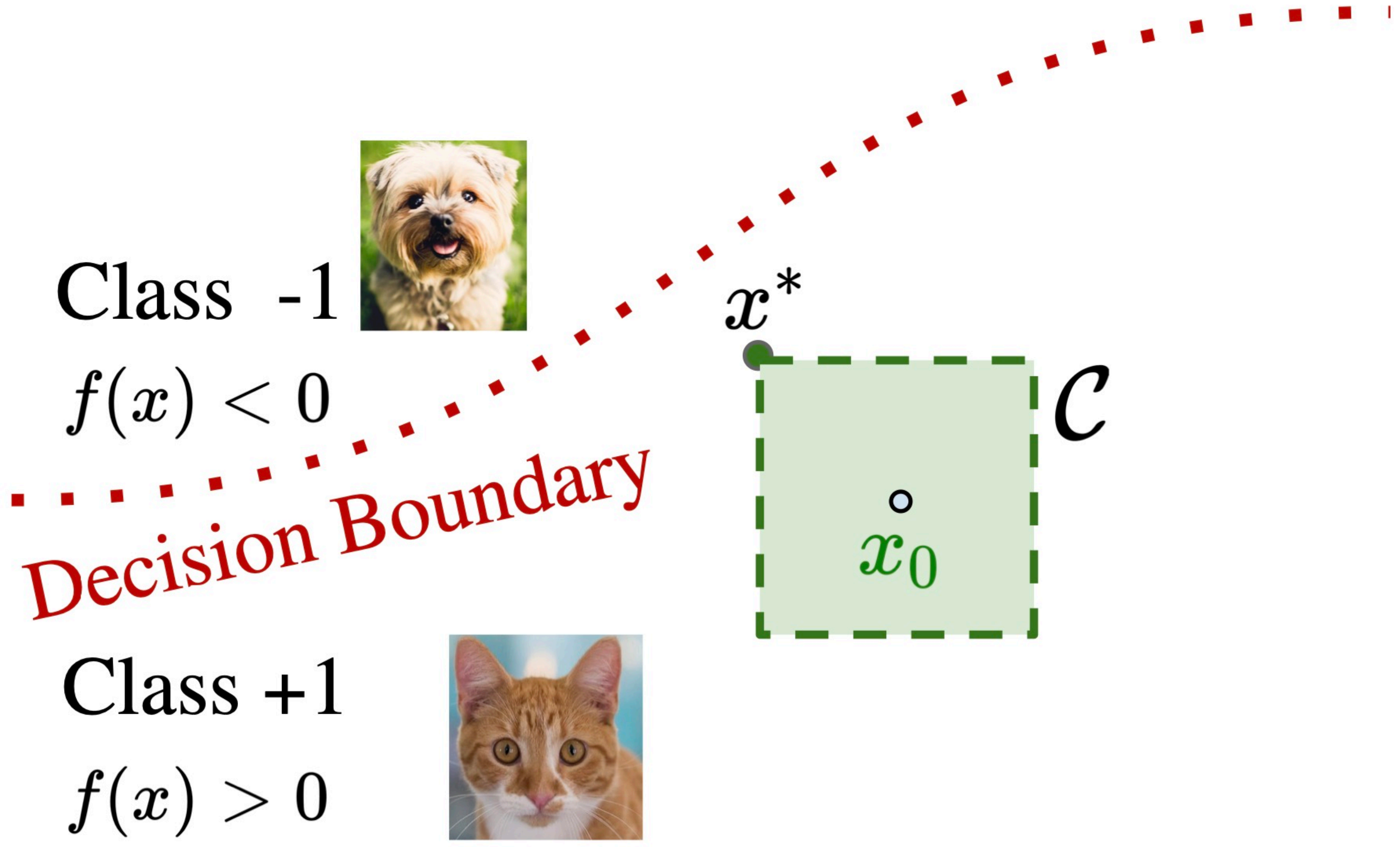
The Basic Formulation of Robustness Verification

Assuming $f(x_0) > 0$, we solve the optimization problem to find the worst case:

$$f^* = \min_{x \in \mathcal{C}} f(x)$$

$\mathcal{C}$ is usually a perturbation set "around" x_0, e.g., $\mathcal{C} := \{x \| \|x - x_0\|_p \leq \epsilon\}$

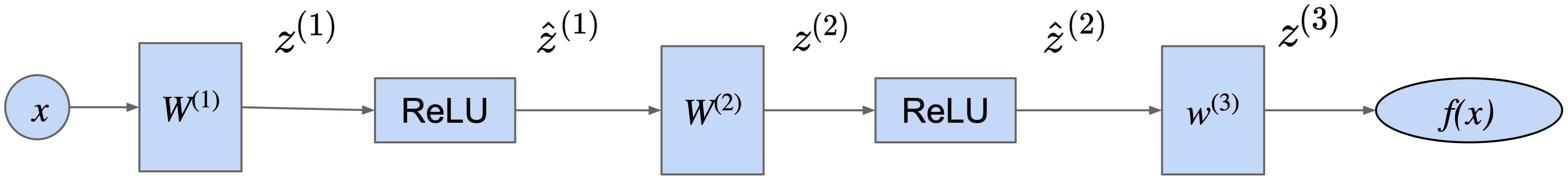




Is it a hard problem?

CROWN backward bound propagation

- Goal: find linear relationships between output and every hidden neuron, until we reach the input!

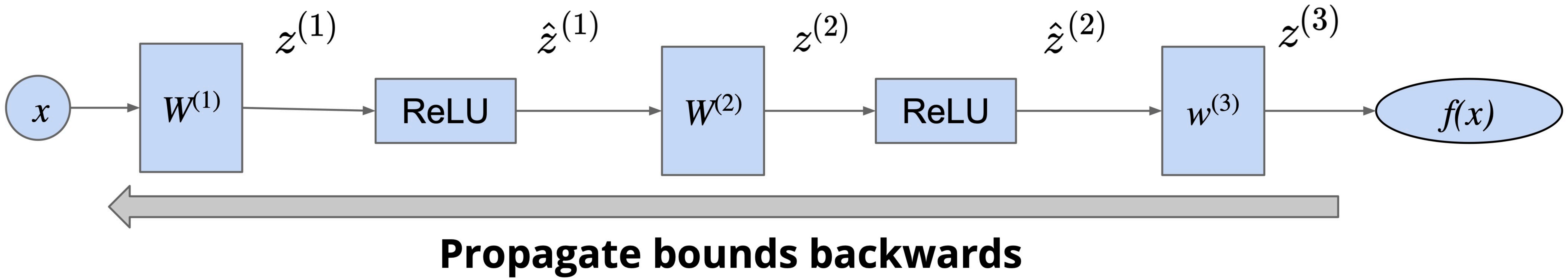


$$f(x) \geq \boxed{w^{(3)\top} D^{(2)} W^{(2)} D^{(1)} W^{(1)}} x + \text{const.}$$

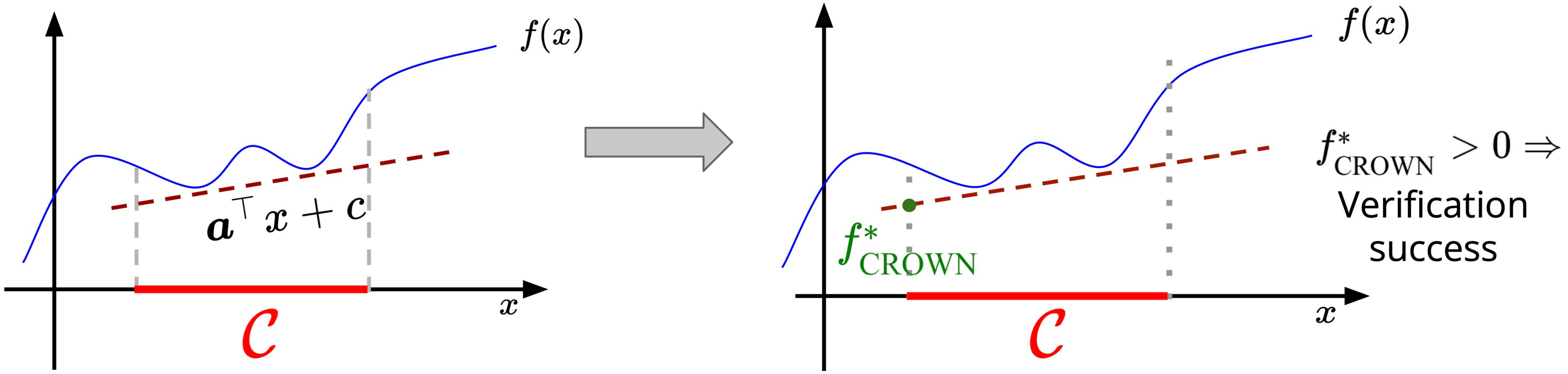
CROWN linear bound: $\min_{x \in \mathcal{C}} f(x) \geq \min_{x \in \mathcal{C}} \boldsymbol{a}_{\text{CROWN}}^{\top} x + c_{\text{CROWN}} := \min_{x \in \mathcal{C}} f_{\text{CROWN}}(x)$

Where $\boldsymbol{a}_{\text{CROWN}}$ and $\boldsymbol{c}_{\text{CROWN}}$ are functions of NN weights, and can be computed efficiently on GPUs in a backward manner

The CROWN lower bound

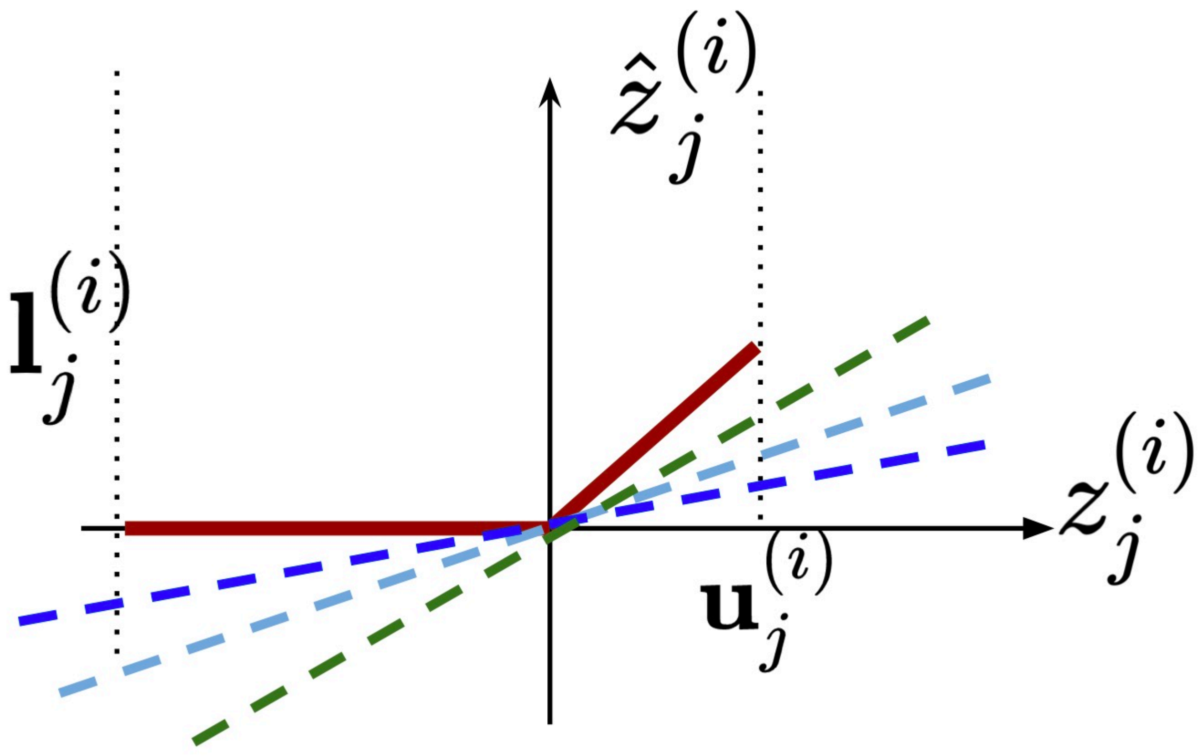


$$\min_{x\in\mathcal{C}} f(x) \geq \min_{x\in\mathcal{C}} \boldsymbol{a}_{\text{CROWN}}^{\top} x + c_{\text{CROWN}} := \min_{x\in\mathcal{C}} f_{\text{CROWN}}(x)$$



α-CROWN: further tighten the bounds

- ReLU neurons have a flexible lower bound for relaxation
- Try different lower bounds to find tightest bound
- Each unstable ReLU has a lower bound to select, so lots of freedom here



$$\hat{z}_j^{(i)} \geq \boldsymbol{\alpha}_j^{(i)} z_j^{(i)} \boxed{(0 \leq \boldsymbol{\alpha}_j^{(i)} \leq 1)}$$

Adjustable!

$\mathbf{l}_j^{(i)} \leq z_j^{(i)} \leq \mathbf{u}_j^{(i)}$ are pre-activation bounds, also computed using CROWN

α-CROWN: further tighten the bounds

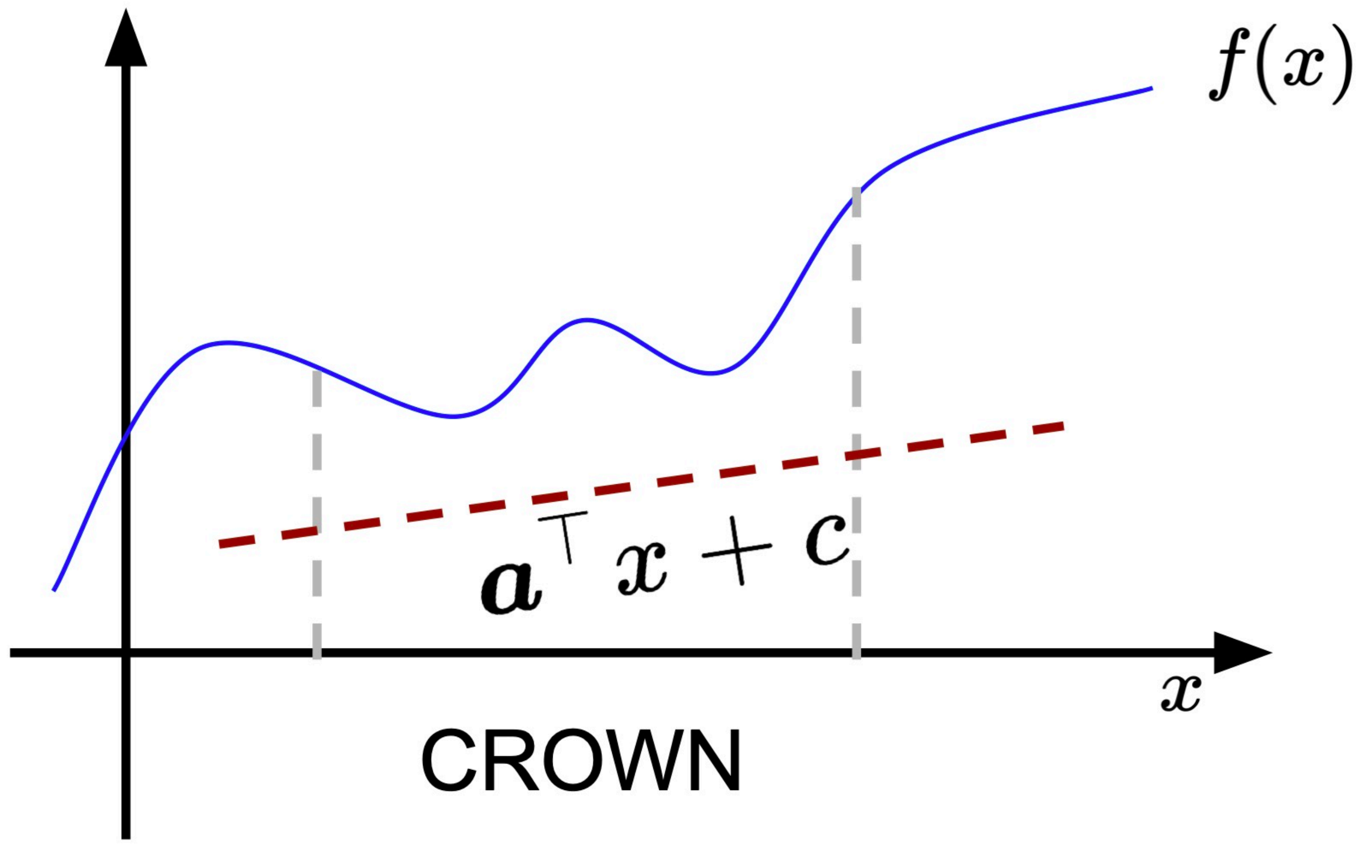
$$f^*_{\text{CROWN}} = \min_{x \in \mathcal{C}} \boldsymbol{a}^\top_{\text{CROWN}} x + c_{\text{CROWN}}$$

Actually a function of $\boldsymbol{\alpha}$. How to effectively optimize $\boldsymbol{\alpha}$ to find the best bound?

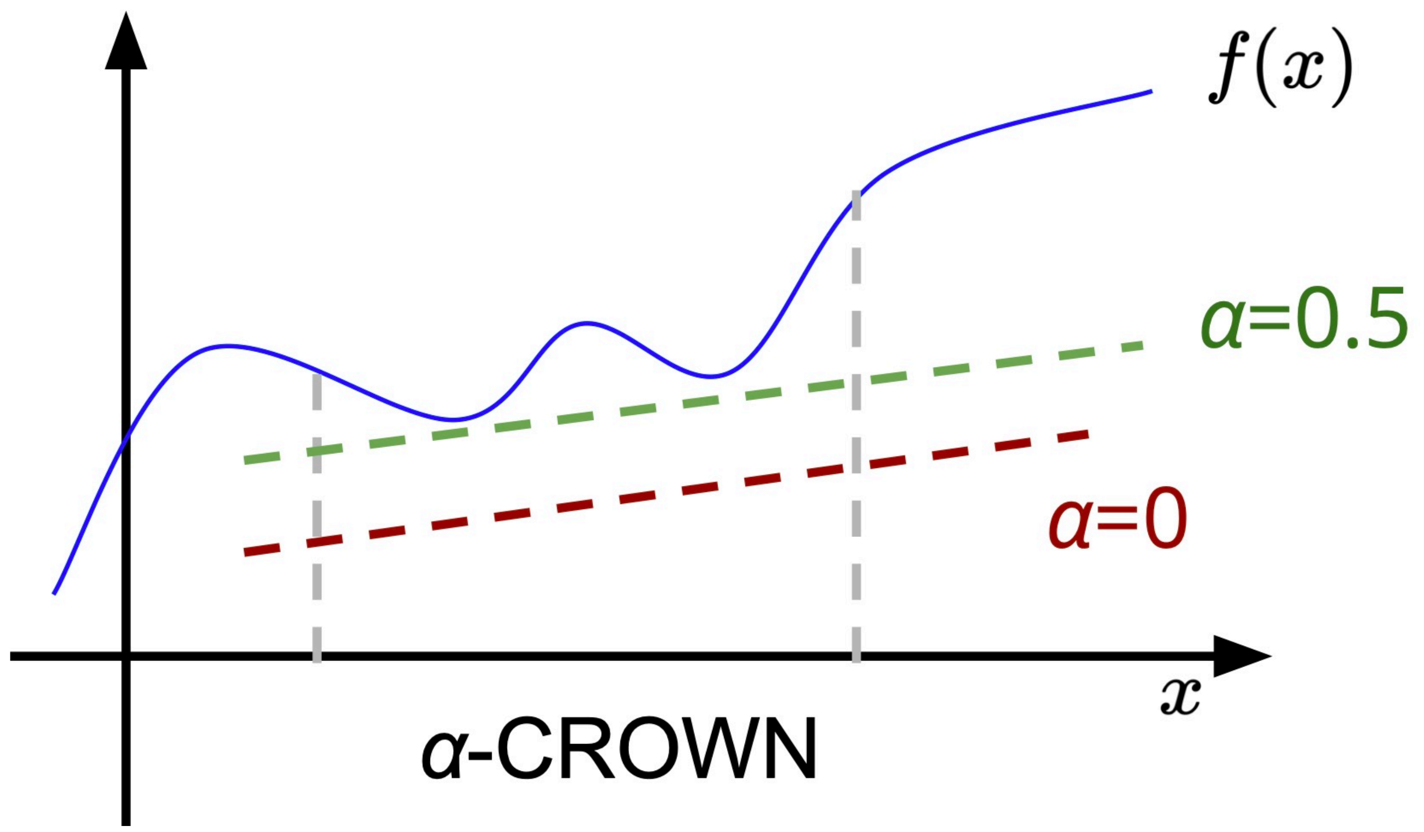
Key idea: tighten bounds using gradients

$$f^* = \min_{x \in \mathcal{C}} f(x) \geq \boxed{\max_{0 \leq \boldsymbol{\alpha} \leq 1}} \min_{x \in \mathcal{C}} f_{\text{CROWN}}(x; \boldsymbol{\alpha})$$

Inner minimization can usually be solved in closed form



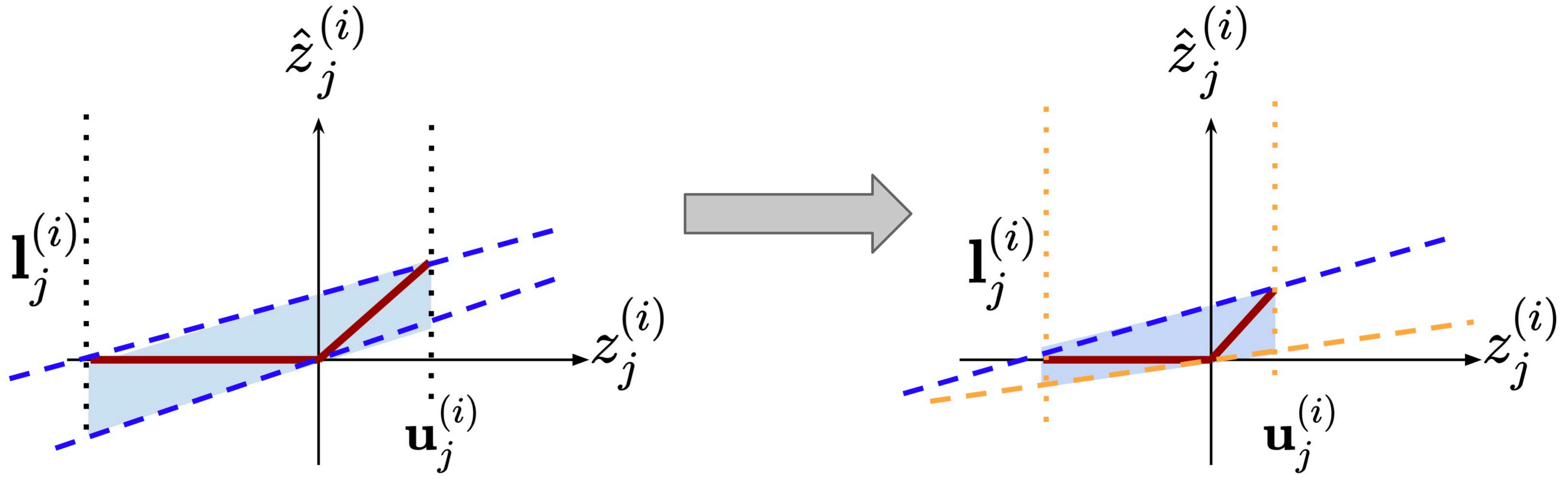
CROWN

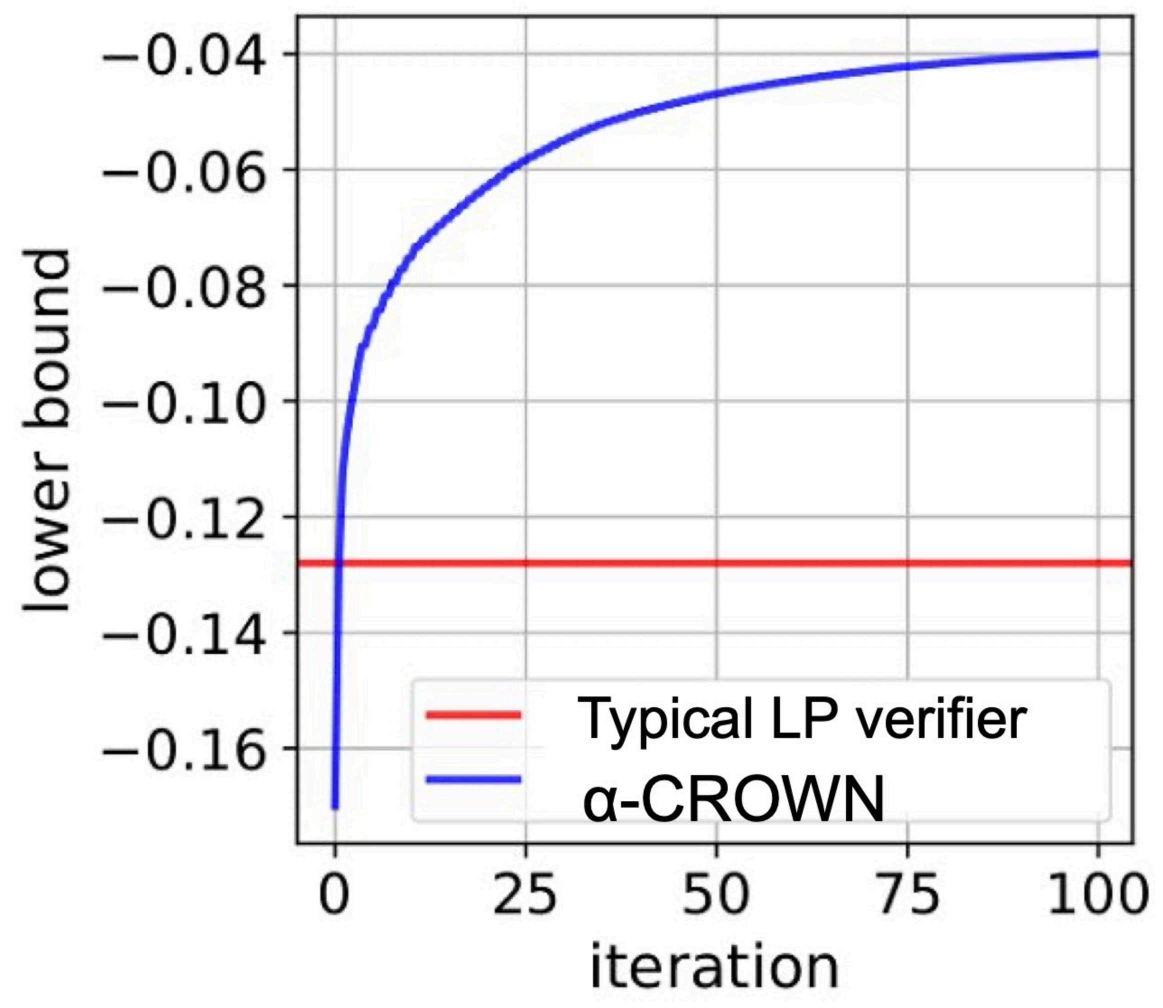


α-CROWN

α-CROWN: further tighten the bounds

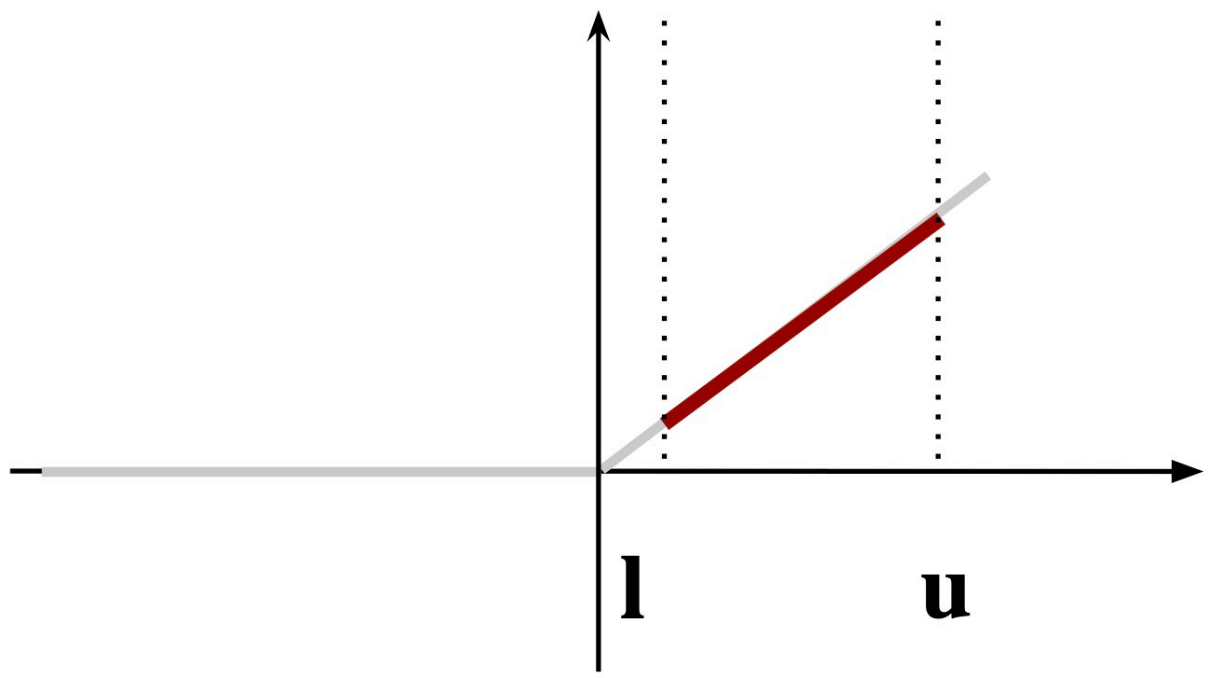
- We can use gradient to optimize the relaxation, to make the bound tighter (tighter bound => stronger incomplete verification)
- We can make the bound **tighter** than the more expensive LP-based verifiers
- Optimization can be done rapidly on **GPUs**



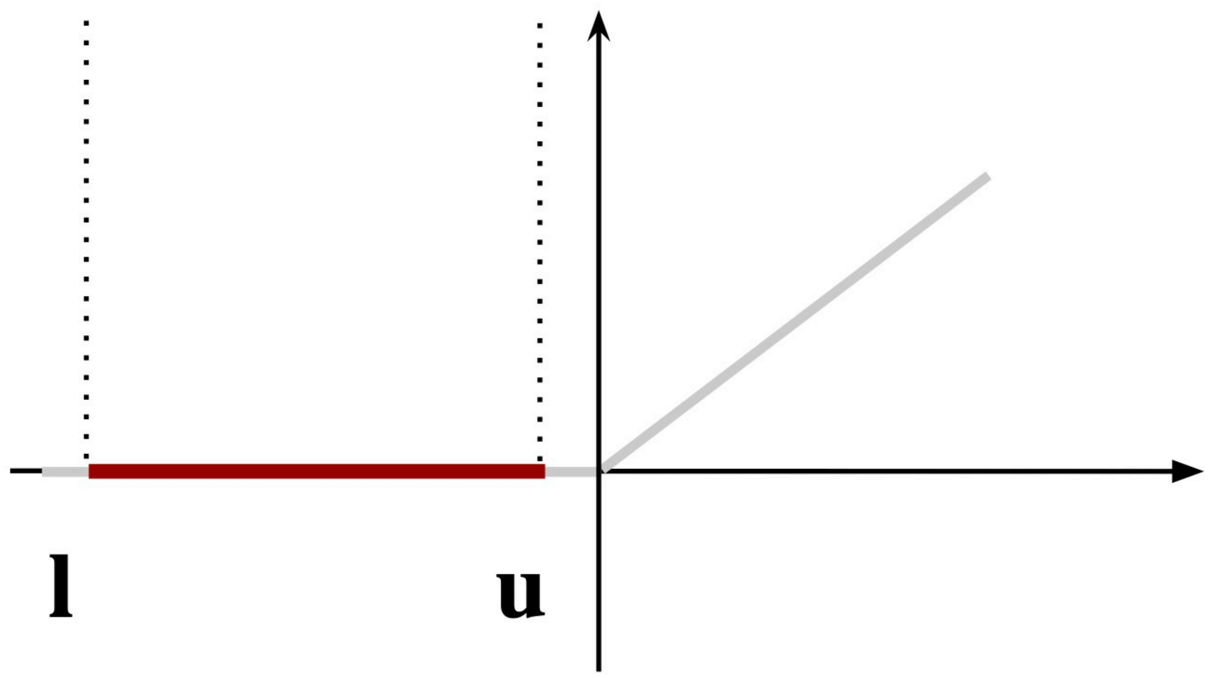


Branch and bound for ReLU Network Verfication

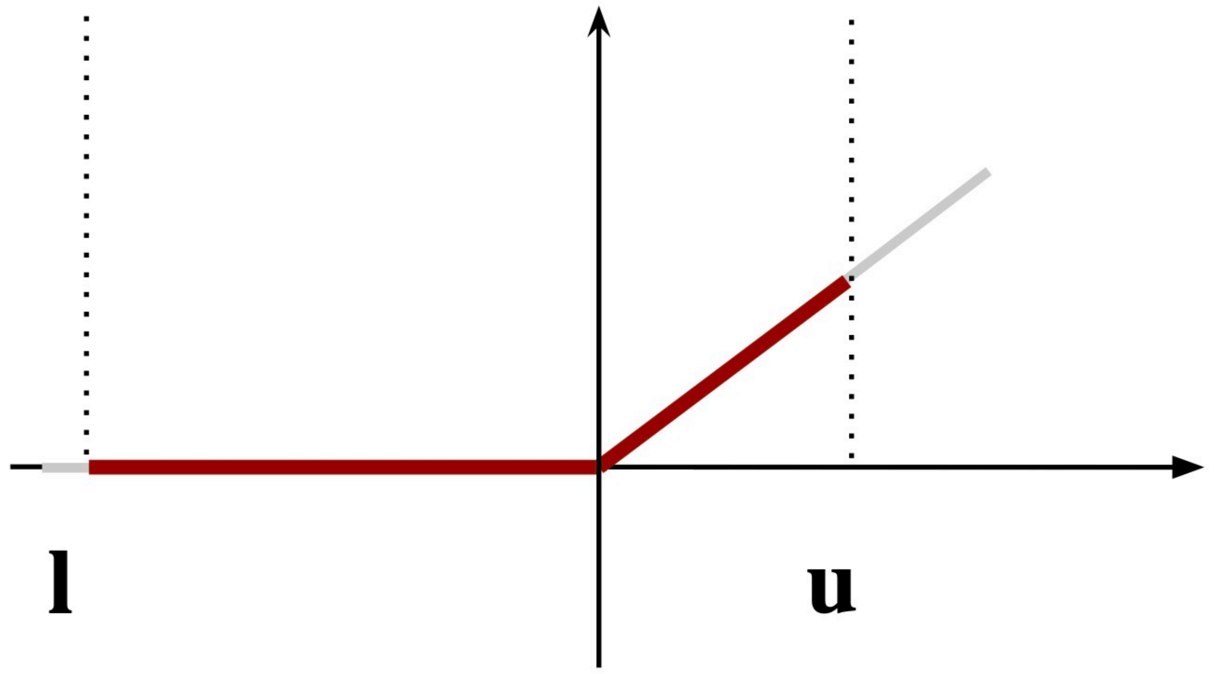
Recall that ReLU neurons have three cases depending on pre-activation bounds:



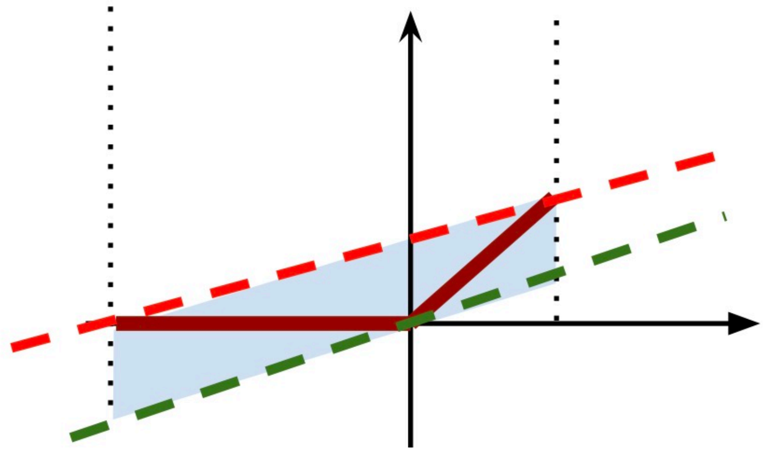
$\mathbf{l} \geq 0$, always active
(linear)



$\mathbf{u} \leq 0$, Always inactive
(zero)

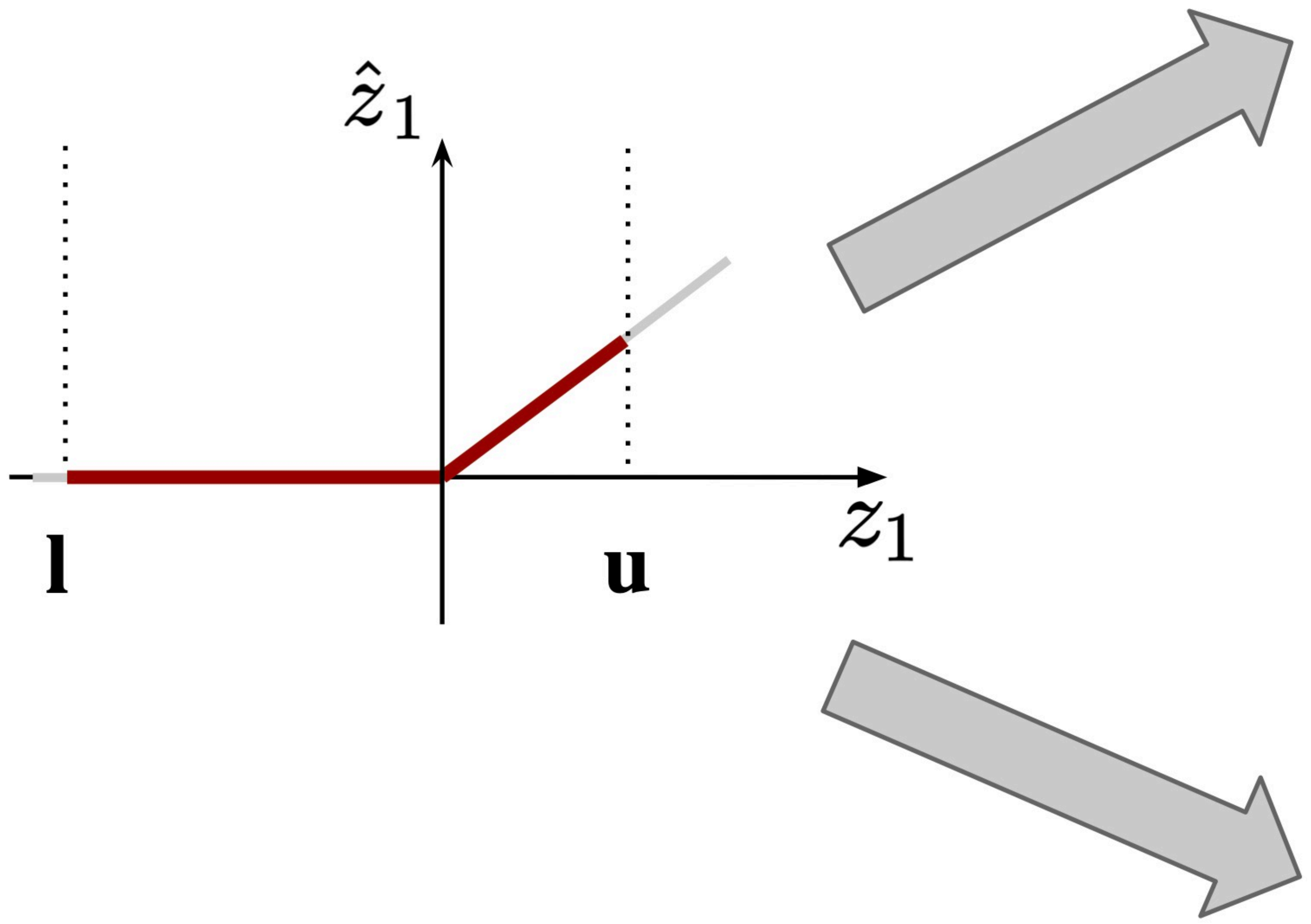


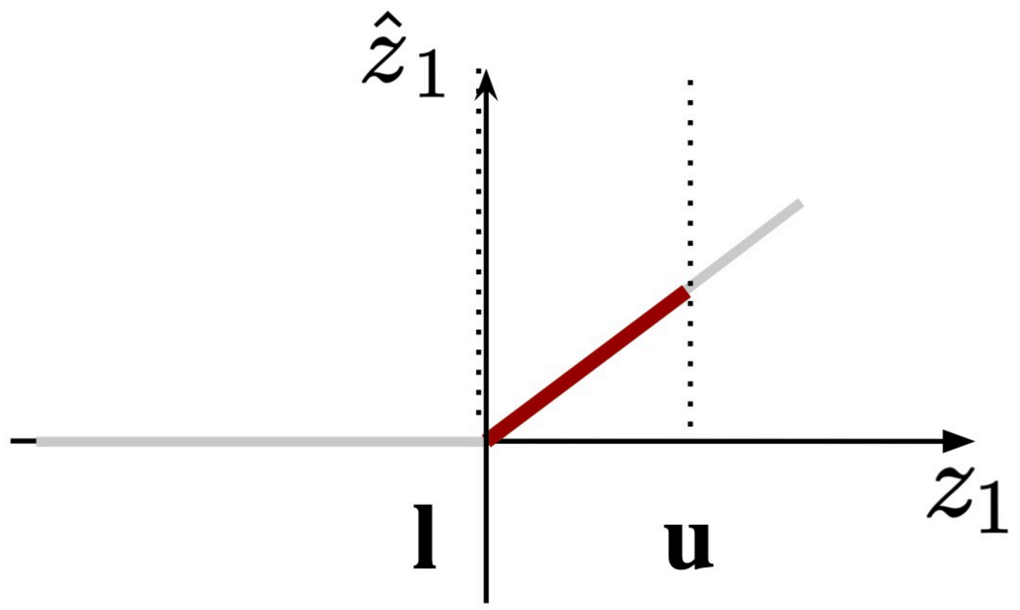
$\mathbf{l} \leq 0 \leq \mathbf{u}$
Unstable (non-linear)
Must be **relaxed**; relaxation makes bounds looser



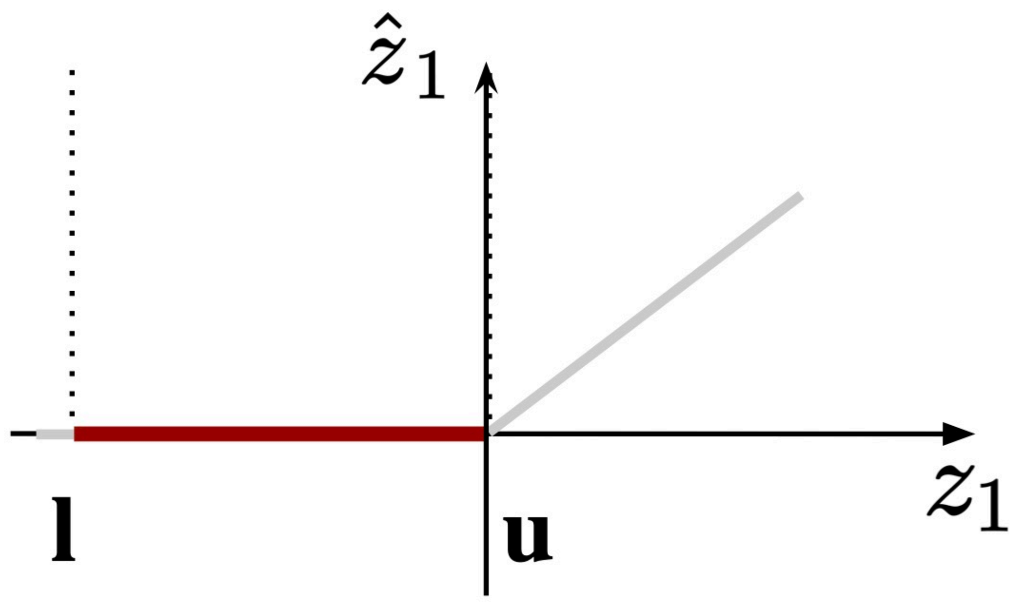
Branch and bound: The branching step

Split each “unstable” ReLU neurons to two subproblems:





Relu is linear, no relaxation needed



Additional linear constraint (“split constraint”):

$$z_1 > 0$$

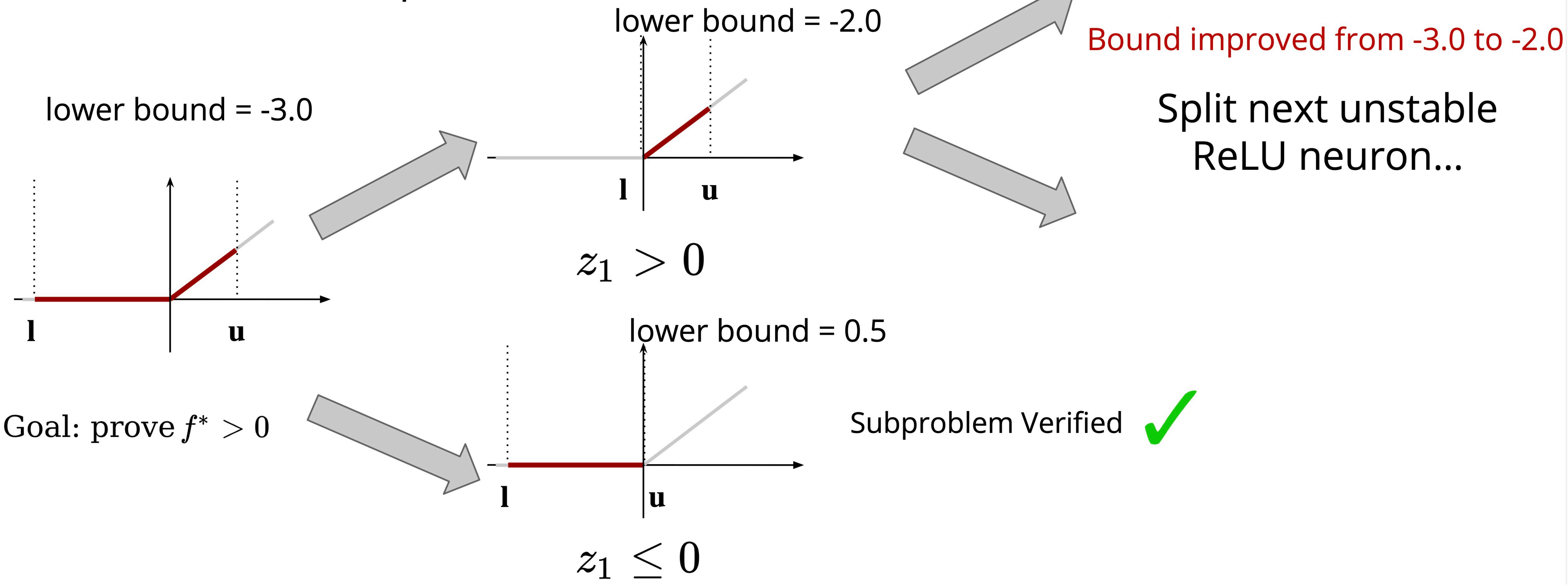
OR

$$z_1 \leq 0$$

The additional constraint can make bounds tighter

Branch and bound: The bounding step

Using an incomplete solver (traditionally, LP-based verifier) to get the lower bound for each subproblem:



Branch and bound search tree

Branch and bound improves the lower bound

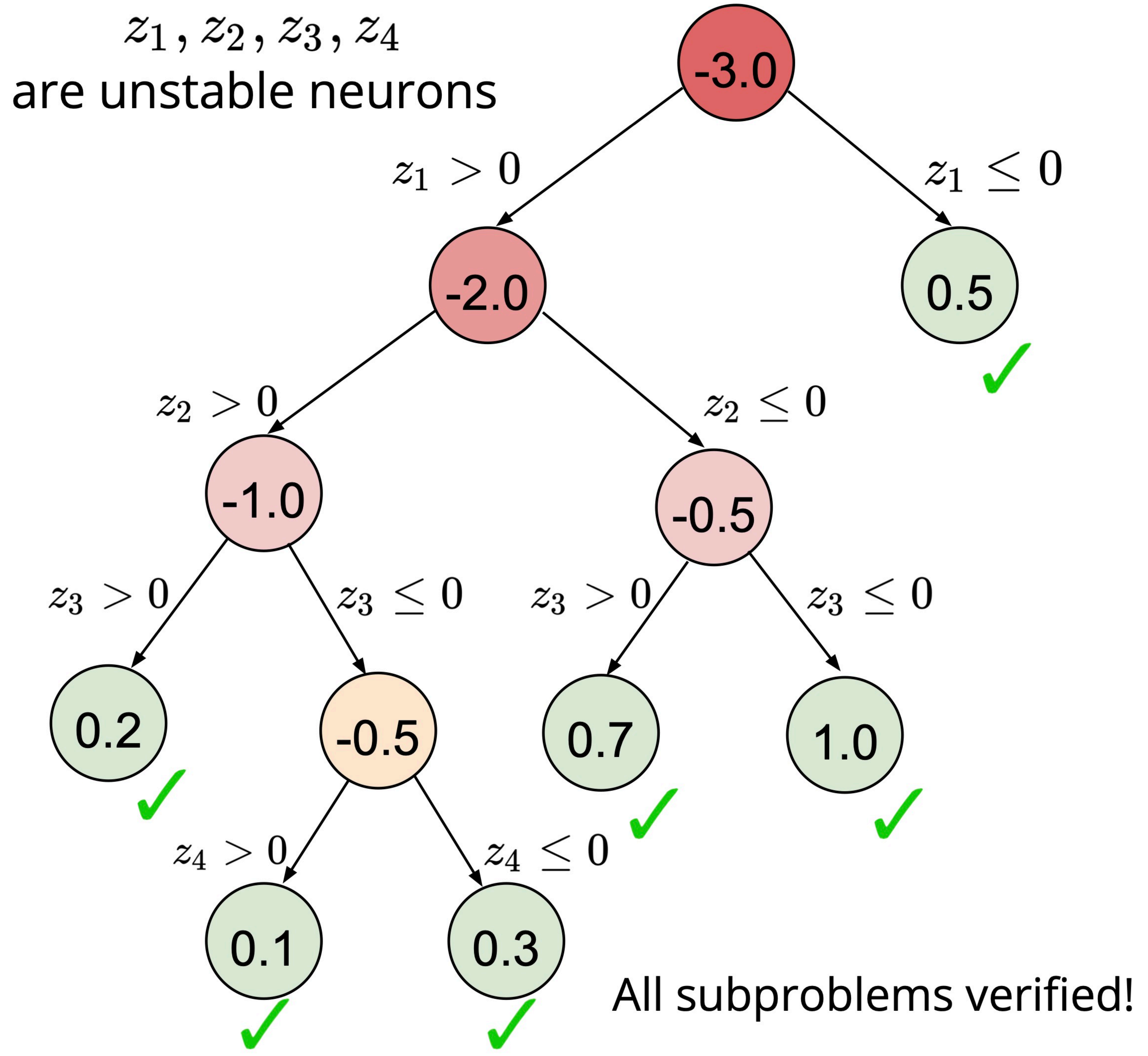
Lower bound = -3.0
(computed by a incomplete verifier)

Lower bound = min(-2.0, 0.5) = -2.0

Lower bound = min(-1.0, -0.5) = -1.0

Lower bound = -0.5

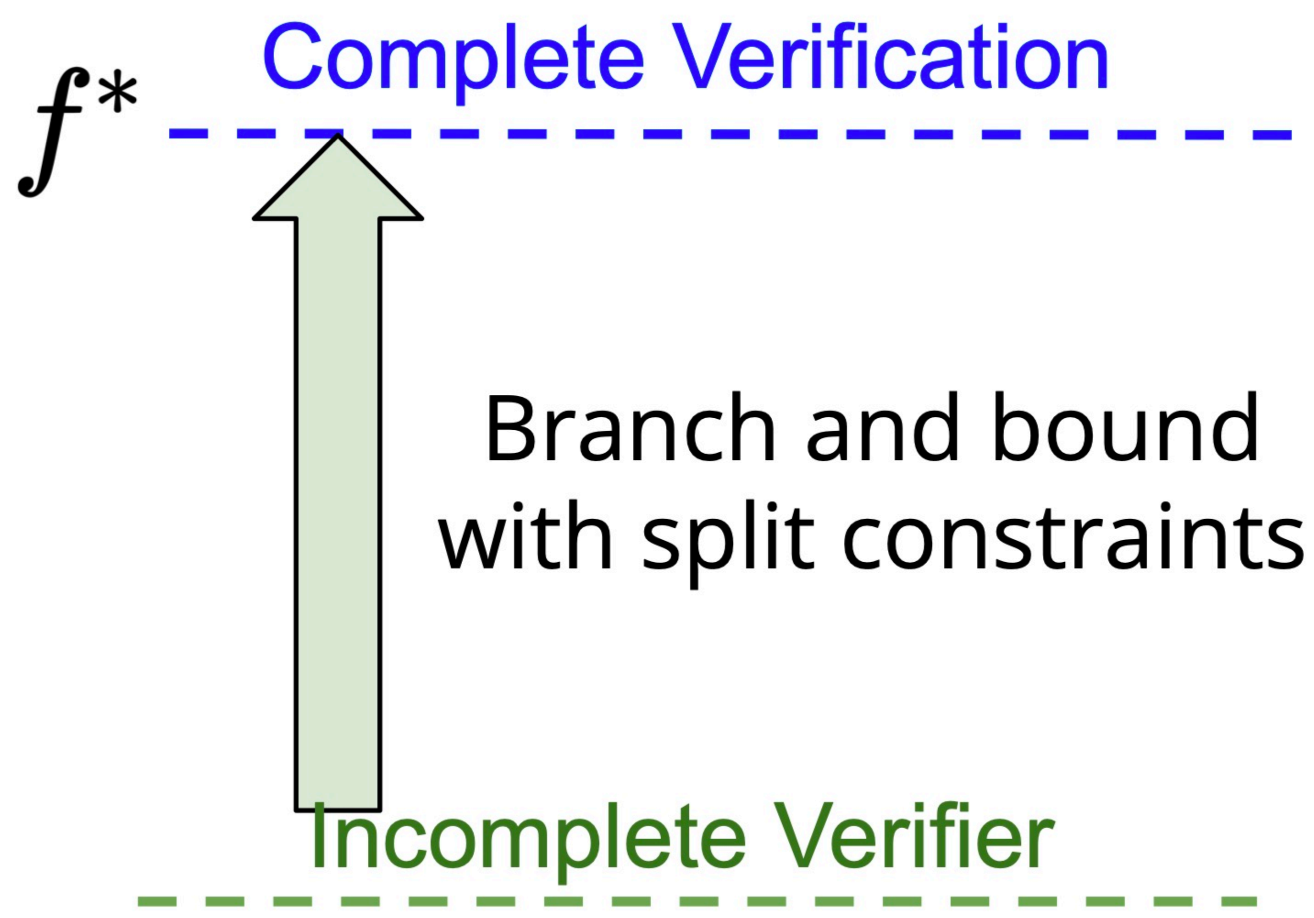
Lower bound = 0.1

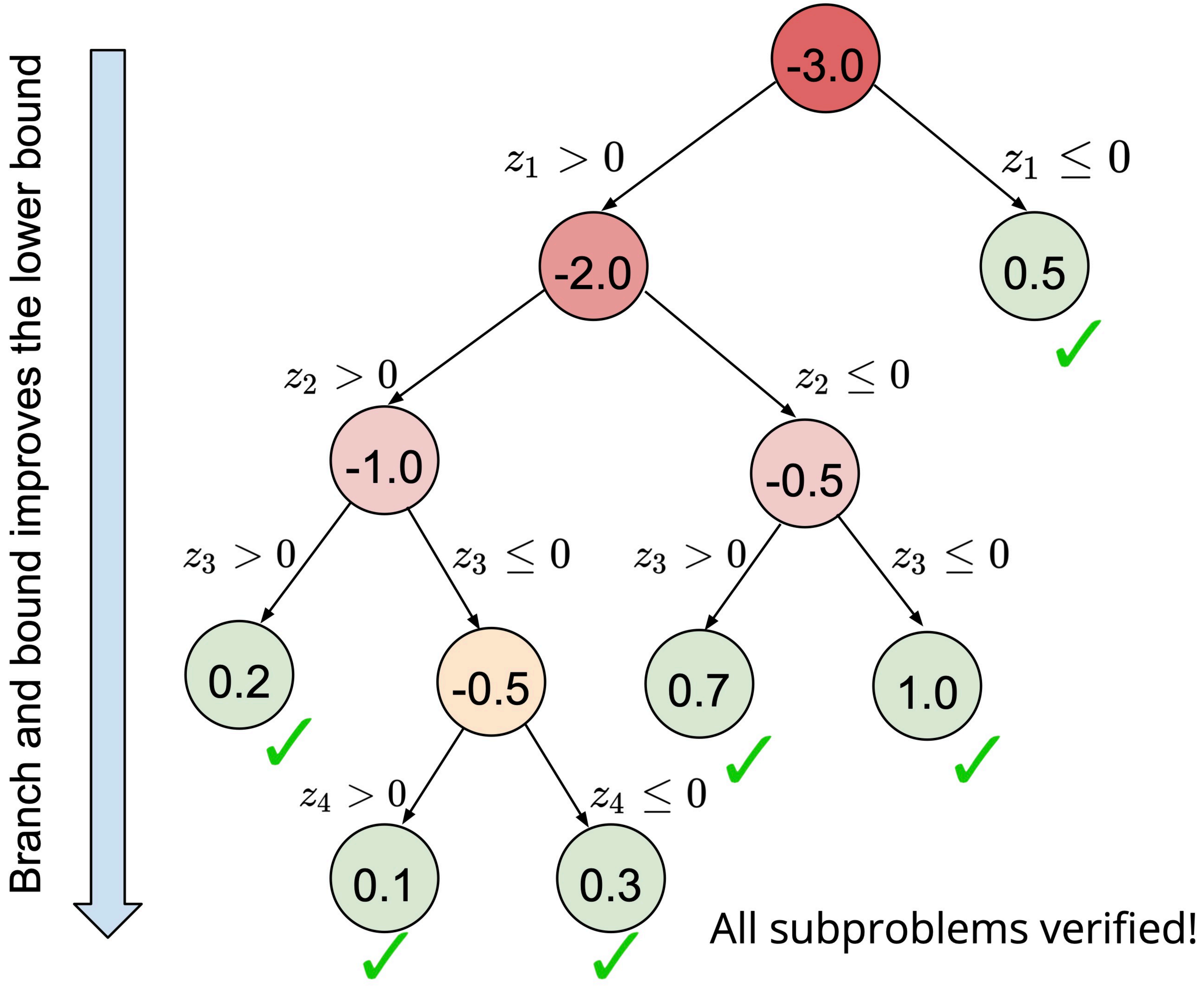


Branch and bound search tree

Branch and bound is complete if each relaxed subproblem (**with split constraint**) can be solved to optimal.

$$f^* = \min_{x \in \mathcal{C}} f(x)$$





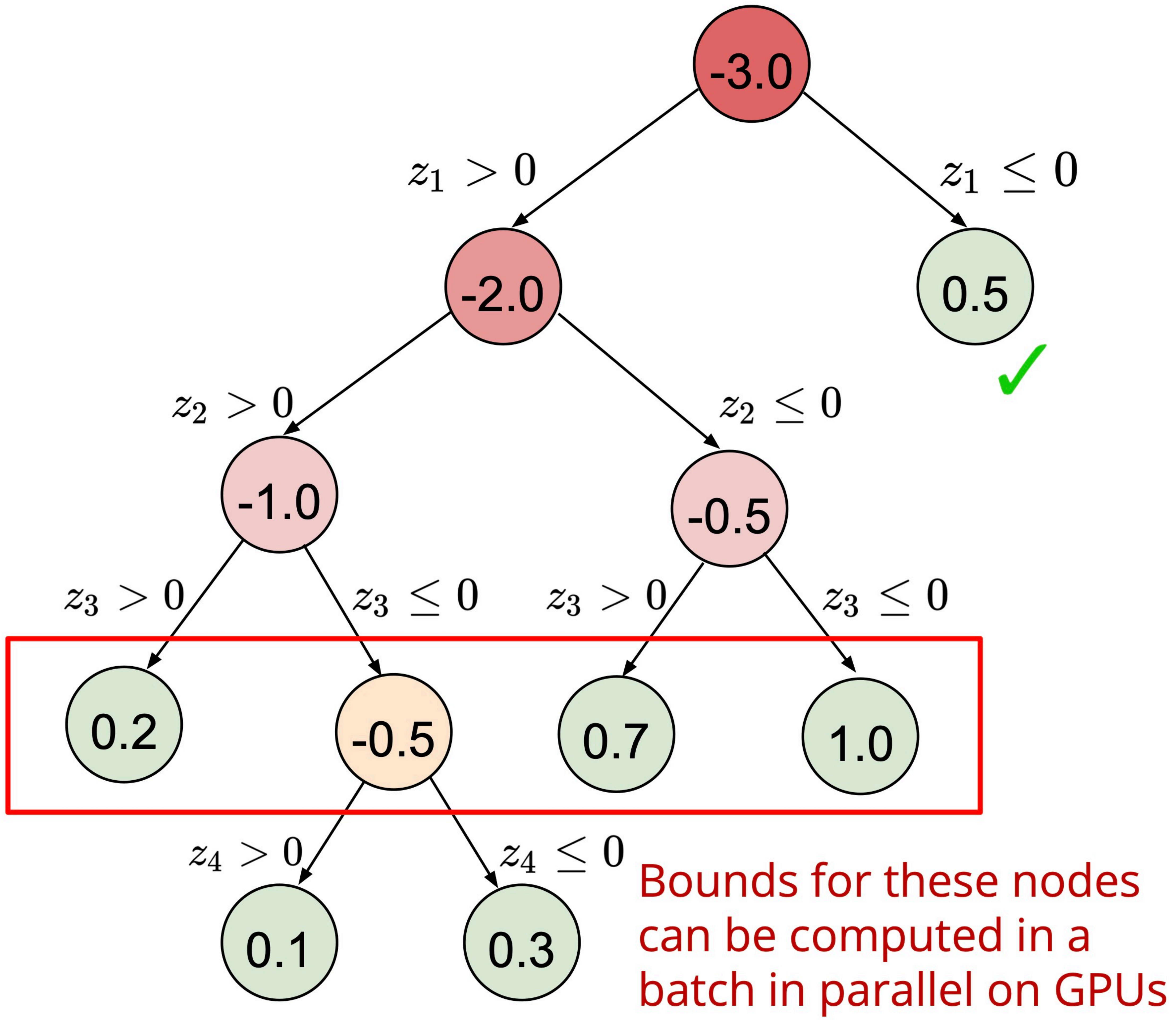
Branch and bound search tree

Idea:

Combine rapid bound propagation based incomplete verifiers on GPUs with branch and bound (BaB) to achieve complete verification

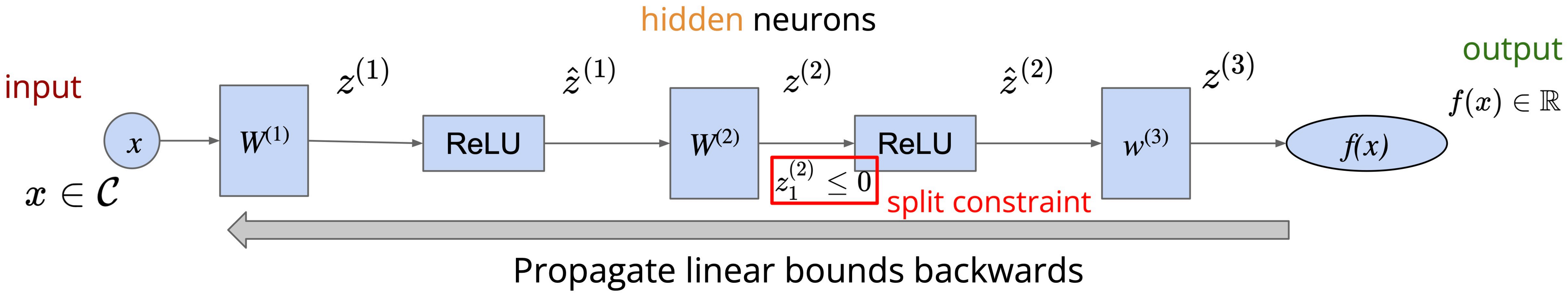
Outcome:

up to 100-1000x faster than MIP based approach, enable us to scale complete verification to larger models



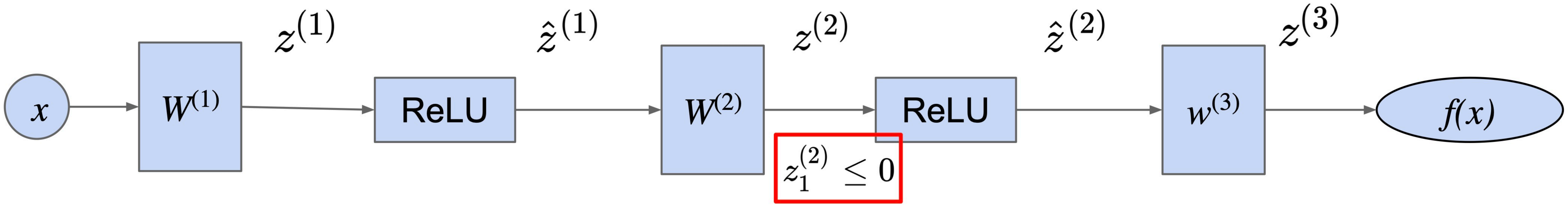
β-CROWN: Bound propagation with split constraints

- To use branch and bound, bound propagation must incorporate the split constraints; CROWN *cannot* handle it



β-CROWN: Bound propagation with split constraints

- Deal with split constraints with Lagrangians



CROWN: $$\min_{x\in\mathcal{C}} f(x) \ge \min_{x\in\mathcal{C}} w^{(3)\top} D^{(2)} z^{(2)} + \text{const.}$$ Cannot handle split constraint

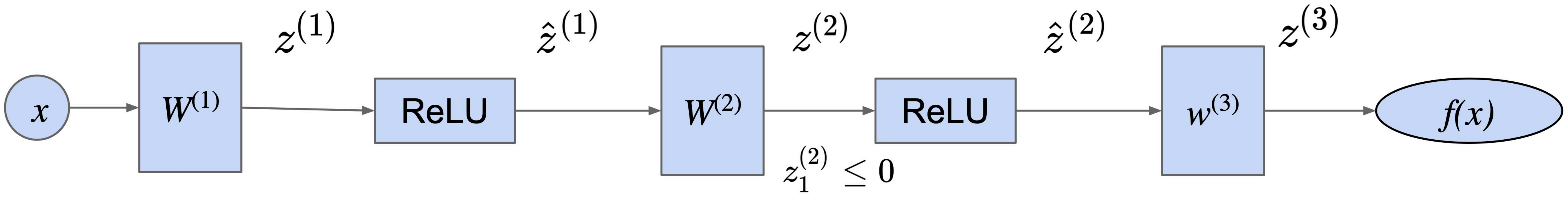
***β*-CROWN**: $$\min_{x\in\mathcal{C}, \boxed{z_1^{(2)}<0}} f(x) \ge \max_{\beta\ge 0} \min_{x\in\mathcal{C}} w^{(3)\top} D^{(2)} z^{(2)} + \boxed{\beta^\top S^{(2)} z^{(2)}} + \text{const.}$$

Lagrangian/KKT multipliers

S is an diagonal matrix with +/-1 and 0

β-CROWN: Bound propagation with split constraints

- Lagrangians are also propagated!



CROWN:

$$\min_{x \in \mathcal{C}} f(x) \geq \min_{x \in \mathcal{C}} \boxed{w^{(3)\top} D^{(2)}} z^{(2)} + \text{const.}$$

***β*-CROWN**:

$$\min_{x \in \mathcal{C}, z_1^{(2)} \in <0} f(x) \geq \max_{\beta \geq 0} \min_{x \in \mathcal{C}} \boxed{\left(w^{(3)\top} D^{(2)} + \beta^\top S^{(2)} \right)} z^{(2)} + \text{const.}$$

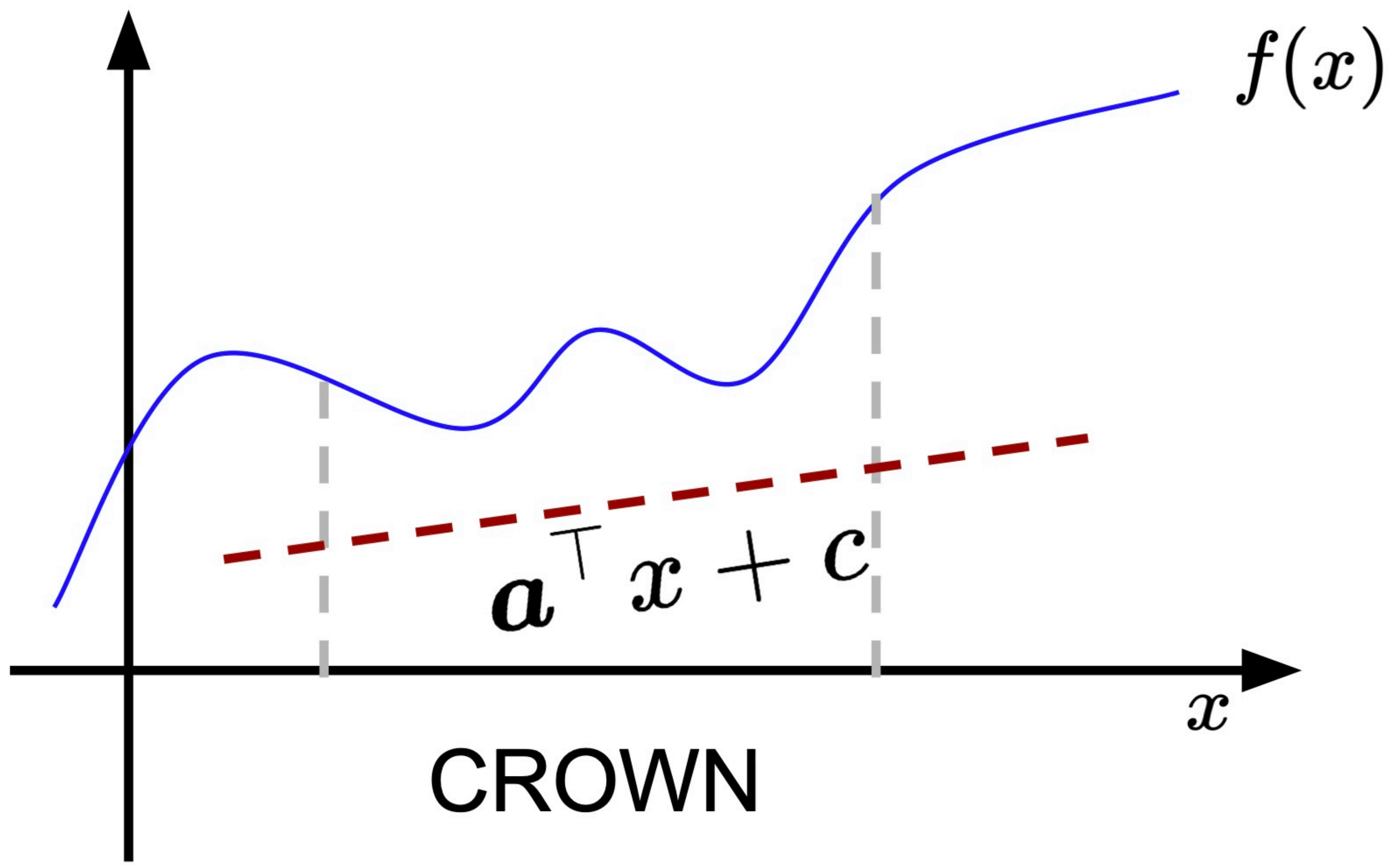
Linear coefficients changed with one additional term during propagation

β-CROWN: Bound propagation with split constraints

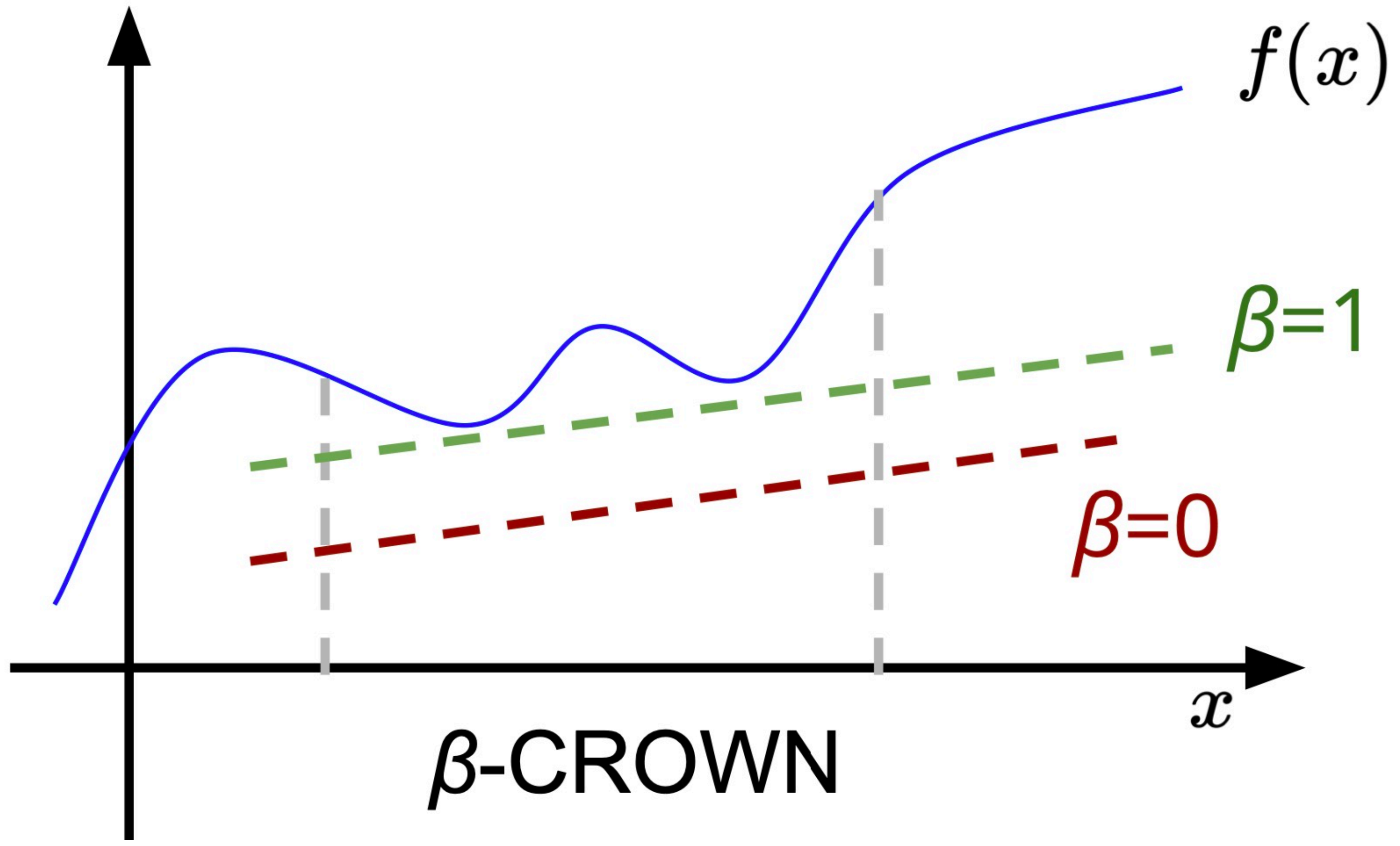
$\boldsymbol{\beta}$-CROWN main theorem: $$\min_{x \in \mathcal{C}, z \in \mathcal{Z}} f(x) \geq \max_{\boldsymbol{\beta} \geq 0} \min_{x \in \mathcal{C}} (\boldsymbol{a} + \mathbf{P}\boldsymbol{\beta})^\top x + \mathbf{q}^\top \boldsymbol{\beta} + c,$$

all split constraints

Compared to (vanilla) CROWN (β=0): $$\min_{x \in \mathcal{C}} f(x) \geq \min_{x \in \mathcal{C}} \boldsymbol{a}^\top x + \boldsymbol{c}$$



Different $\boldsymbol{\beta}$ corresponds to different bounds, and we can choose the tightest one



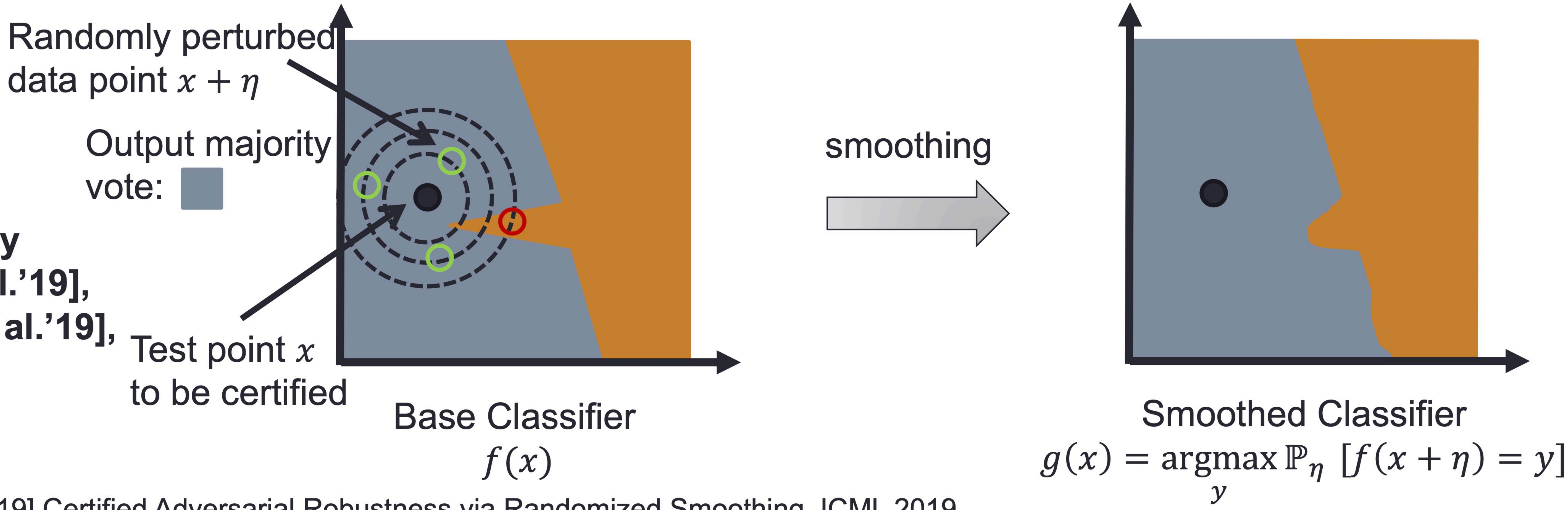
Randomized smoothing

Randomized smoothing

- Assume we have a base classifier *f* that maps inputs *x* to labels *y*, i.e., $f(x) = y$
- The approach creates corrupted versions of the image *x* by applying Gaussian noise with 0 mean and variance σ^2, i.e., $\eta \sim \mathcal{N}(0, \sigma^2 I)$
  - Left figure: input sample x; Right figure: image corrupted with Gaussian noise $x + \eta$
- A smoothed classifier *g* is obtained by outputting the majority vote of the prediction on many Gaussian-corrupted images $x + \eta$
  - The added random noise improves the robustness to adversarial perturbations

Randomized smoothing

Proposed by [Cohen et al.'19], [Lecuyer et al.'19], [Li et al.'19]



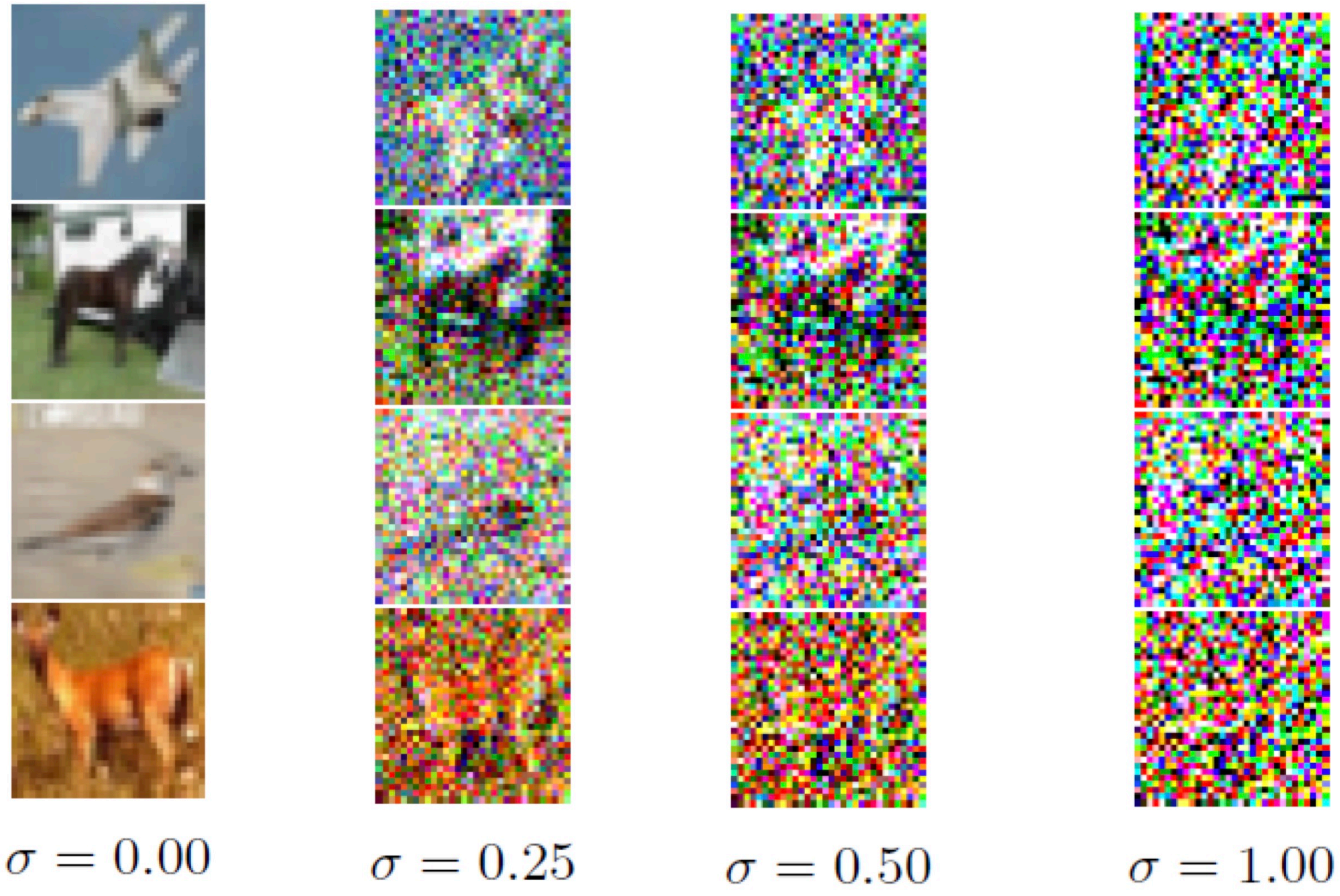
[Cohen et al.'19] Certified Adversarial Robustness via Randomized Smoothing, ICML 2019.
[Lecuyer et al.'19] Certified Robustness to Adversarial Examples with Differential Privacy, S&P 2019.
[Li et al.'19] Certified Adversarial Robustness with Additive Noise, NeurIPS 2019.

Randomized smoothing

- To design a smoothed classifier *g* at the input sample *x* requires to identify the most likely class $\hat{c}_A$ returned by the base classifier *f* on noisy images
  - Step 1: create *n* versions of *x* corrupted with Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2 I)$
  - Step 2: evaluate the predictions by base classifier for all corrupted images, $f(x + \eta)$
  - Step 3: identify the top two classes $\hat{c}_A$ and $\hat{c}_B$ with the highest number of predictions on $f(x + \eta)$
  - Step 4: if n_A (number of predictions by *f* for the top class $\hat{c}_A$) is much greater than n_B (number of predictions for the second highest class $\hat{c}_B$), return $\hat{c}_A$ as the prediction by $g(x)$
    - Otherwise, if $n_A - n_B < \alpha$, abstain from making a prediction

Randomized smoothing

- Examples of noisy images from CIFAR-10 with varying levels of Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ from $\sigma = 0$ to $\sigma = 1$



Randomized smoothing

- Intuitively, the certified radius *R* is large when:
  - The noise level σ is high
  - The probability of the top class $p_A = \mathbb{P}(g(x+\eta) = c_A)$ is high
  - The probability of second top class $p_B = \mathbb{P}(g(x+\eta) = c_B)$ is low
- The authors prove that the certified radius *R* is given by:

$$R = \frac{\sigma}{2}\left(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)\right)$$

  - Φ^{-1} is the inverse of the Gaussian cumulative distribution function
  - For binary classification, $R = \sigma\Phi^{-1}(p_A)$, because $\Phi^{-1}(p_B) = -\Phi^{-1}(p_A)$

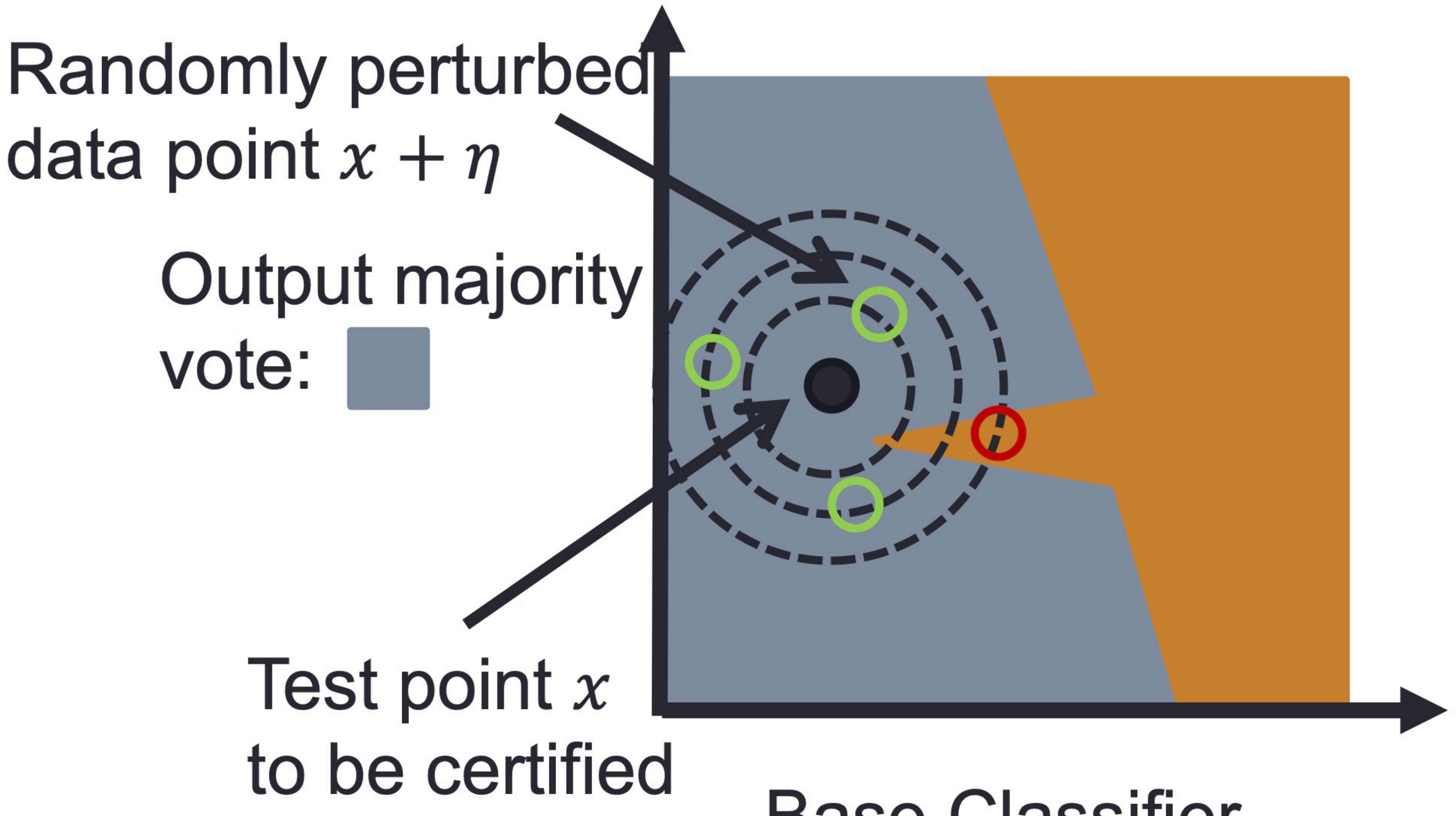
Randomized smoothing

Certified robust radius by [Cohen et al.'19]:

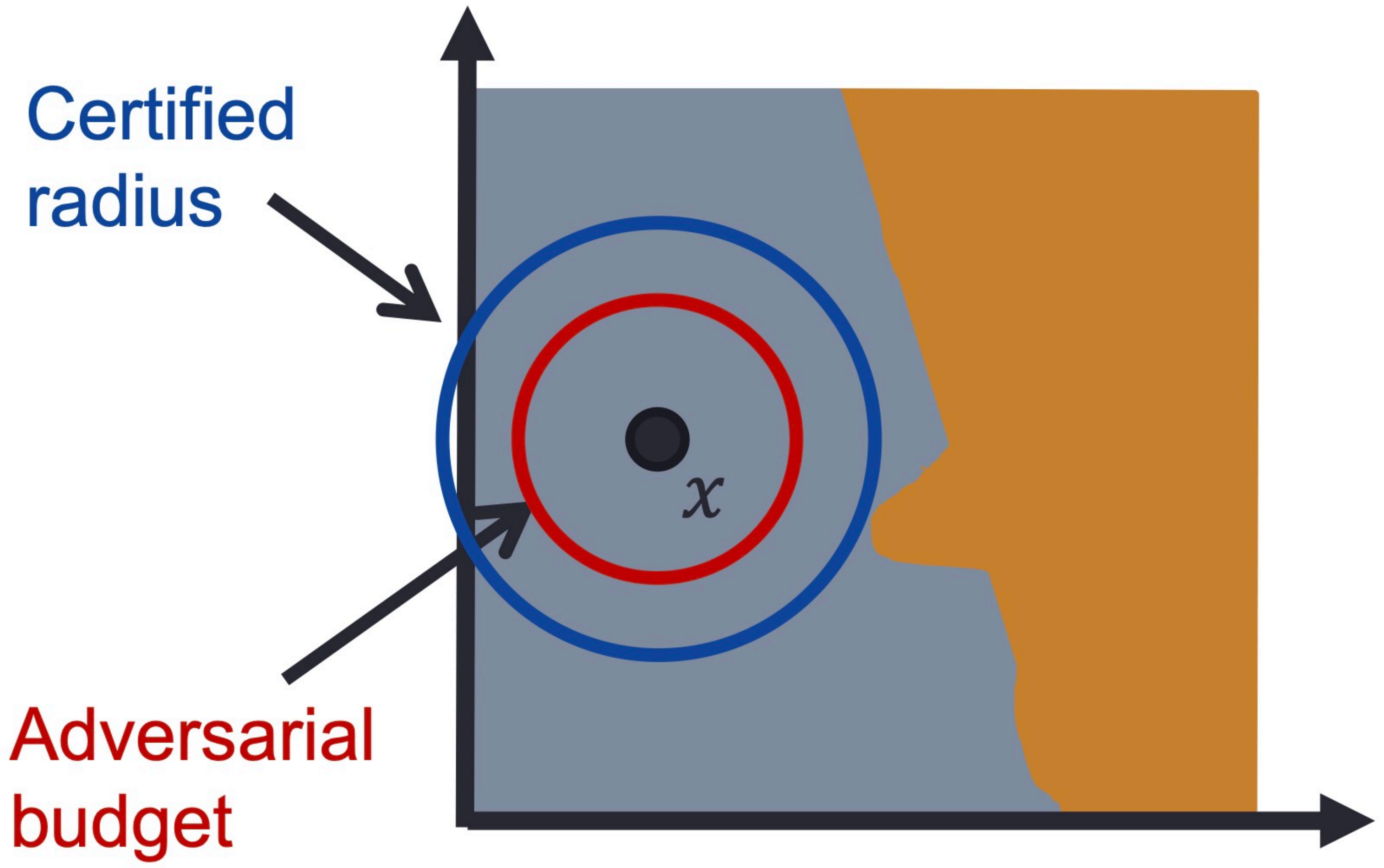
Confidence of majority vote

Given any input $x \in \mathbb{R}^d$, let η be Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ and $p = \max_y \mathbb{P}_\eta[f(x+\eta) = y]$. Then $g(x) = g(x+\delta)$ for any δ such that $||\delta||_2 \leq \Phi^{-1}(p)\sigma$, where Φ is CDF of standard Gaussian.

Computable certified radius for x



Base Classifier
$f(x)$



Smoothed Classifier
$g(x) = \operatorname{argmax}_y \mathbb{P}_\eta \, [f(x+\eta) = y]$

Randomized smoothing

Certified robust radius by [Cohen et al.'19]:

Confidence of majority vote

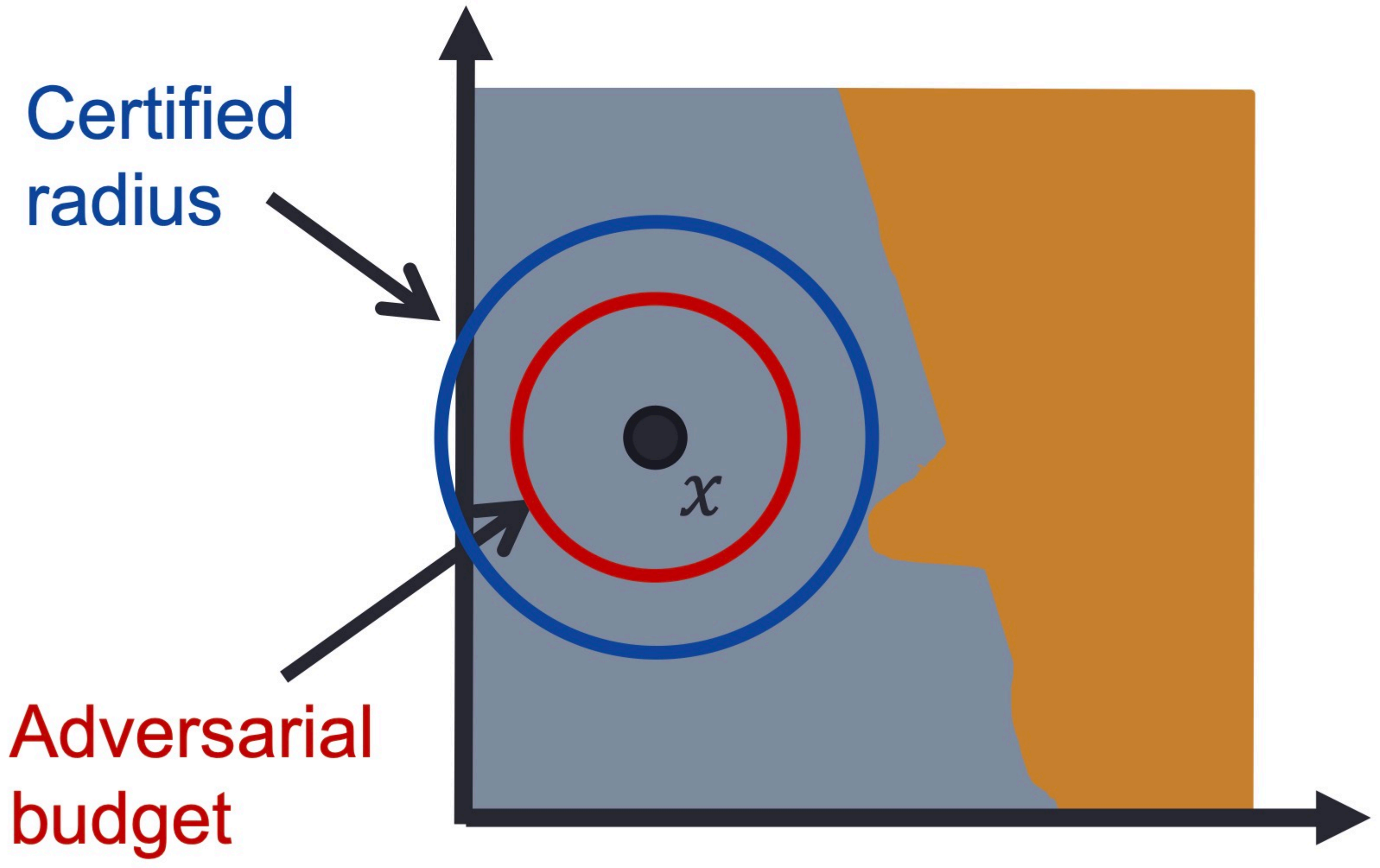
Given any input $x \in \mathbb{R}^d$, let η be Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ and $p = \max_y \mathbb{P}_\eta[f(x+\eta) = y]$. Then $g(x) = g(x+\delta)$ for any δ such that $\|\delta\|_2 \leq \Phi^{-1}(p)\sigma$, where Φ is CDF of standard Gaussian.

Computable certified radius for x

Classifier $g(x)$ is certifiably correct for x, if

1. certified radius > adv budget
2. classifier $g(x)$ is correct for x

Calculate the percentage of certifiably correct x and obtain certified accuracy for a dataset



Smoothed Classifier

$$g(x) = \operatorname{argmax}_y \mathbb{P}_\eta \; [f(x+\eta) = y]$$

Proof sketch

- Neyman-pearson
  - Given a sample from one of two distributions: null X or alternative Y
  - Two errors:
    - say “X” when the true answer is “Y” —> better
    - say “Y” when the true answer is “X” —> limit its probability <= some failure rate α
  - Optimal rule:
    - deterministically on the set $S^* = \{z \in \mathbb{R}^d : \frac{\mu_Y(z)}{\mu_X(z)} \geq t\}$ for whichever t makes $\mathbb{P}(X \in S^*) = \alpha$.

Proof sketch

- *Let* $X \sim \mathcal{N}(x, \sigma^2 I)$ *and* $Y \sim \mathcal{N}(x+\delta, \sigma^2 I)$.

By Lemma 3 it suffices to simply show that for any β, there is some $t > 0$ for which:

$$\{z : \delta^T z \leq \beta\} = \left\{z : \frac{\mu_Y(z)}{\mu_X(z)} \leq t\right\} \quad \text{and} \quad \{z : \delta^T z \geq \beta\} = \left\{z : \frac{\mu_Y(z)}{\mu_X(z)} \geq t\right\}$$

The likelihood ratio for this choice of X and Y turns out to be:

$$\begin{aligned} \frac{\mu_Y(z)}{\mu_X(z)} &= \frac{\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^d (z_i - (x_i + \delta_i))^2\right)}{\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^d (z_i - x_i)^2\right)} \\ &= \exp\left(\frac{1}{2\sigma^2}\sum_{i=1}^d 2z_i\delta_i - \delta_i^2 - 2x_i\delta_i\right) \\ &= \exp(a\delta^T z + b) \end{aligned}$$

where $a > 0$ and b are constants w.r.t z, specifically $a = \frac{1}{\sigma^2}$ and $b = \frac{-(2\delta^T x + \|\delta\|^2)}{2\sigma^2}$.

Therefore, given any β we may take $t = \exp(a\beta + b)$, noticing that

$$\begin{aligned} \delta^T z \leq \beta &\iff \exp(a\delta^T z + b) \leq t \\ \delta^T z \geq \beta &\iff \exp(a\delta^T z + b) \geq t \end{aligned}$$

Proof sketch

Proof. To show that $g(x+\delta) = c_A$, it follows from the definition of g that we need to show that

$$\mathbb{P}(f(x+\delta+\varepsilon) = c_A) > \max_{c_B \neq c_A} \mathbb{P}(f(x+\delta+\varepsilon) = c_B)$$

We will prove that $\mathbb{P}(f(x+\delta+\varepsilon) = c_A) > \mathbb{P}(f(x+\delta+\varepsilon) = c_B)$ for every class $c_B \neq c_A$. Fix one such class c_B without loss of generality.

For brevity, define the random variables

$$\begin{aligned} X &:= x + \varepsilon = \mathcal{N}(x, \sigma^2 I) \\ Y &:= x + \delta + \varepsilon = \mathcal{N}(x+\delta, \sigma^2 I) \end{aligned}$$

In this notation, we know from (6) that

$$\mathbb{P}(f(X) = c_A) \geq \underline{p_A} \quad \text{and} \quad \mathbb{P}(f(X) = c_B) \leq \overline{p_B} \tag{8}$$

Proof sketch

Proof. To show that $g(x+\delta) = c_A$, it follows from the definition of g that we need to show that

$$\mathbb{P}(f(x+\delta+\varepsilon) = c_A) > \max_{c_B \neq c_A} \mathbb{P}(f(x+\delta+\varepsilon) = c_B)$$

We will prove that $\mathbb{P}(f(x+\delta+\varepsilon) = c_A) > \mathbb{P}(f(x+\delta+\varepsilon) = c_B)$ for every class $c_B \neq c_A$. Fix one such class c_B without loss of generality.

For brevity, define the random variables

$$\begin{aligned} X &:= x + \varepsilon = \mathcal{N}(x, \sigma^2 I) \\ Y &:= x + \delta + \varepsilon = \mathcal{N}(x+\delta, \sigma^2 I) \end{aligned}$$

In this notation, we know from (6) that

$$\mathbb{P}(f(X) = c_A) \geq \underline{p_A} \quad \text{and} \quad \mathbb{P}(f(X) = c_B) \leq \overline{p_B} \tag{8}$$

Proof sketch

and our goal is to show that

$$\mathbb{P}(f(Y) = c_A) > \mathbb{P}(f(Y) = c_B) \tag{9}$$

Define the half-spaces:

$$\begin{aligned} A &:= \{z : \delta^T(z - x) \leq \sigma\|\delta\|\Phi^{-1}(\underline{p_A})\} \\ B &:= \{z : \delta^T(z - x) \geq \sigma\|\delta\|\Phi^{-1}(1 - \overline{p_B})\} \end{aligned}$$

Algebra (deferred to the end) shows that $\mathbb{P}(X \in A) = \underline{p_A}$. Therefore, by (8) we know that $\mathbb{P}(f(X) = c_A) \geq \mathbb{P}(X \in A)$. Hence we may apply Lemma 4 with $h(z) := \mathbf{1}[f(z) = c_A]$ to conclude:

$$\mathbb{P}(f(Y) = c_A) \geq \mathbb{P}(Y \in A) \tag{10}$$

Similarly, algebra shows that $\mathbb{P}(X \in B) = \overline{p_B}$. Therefore, by (8) we know that $\mathbb{P}(f(X) = c_B) \leq \mathbb{P}(X \in B)$. Hence we may apply Lemma 4 with $h(z) := \mathbf{1}[f(z) = c_B]$ to conclude:

$$\mathbb{P}(f(Y) = c_B) \leq \mathbb{P}(Y \in B) \tag{11}$$

To guarantee (9), we see from (10, 11) that it suffices to show that $\mathbb{P}(Y \in A) > \mathbb{P}(Y \in B)$, as this step completes the chain of inequalities

$$\mathbb{P}(f(Y) = c_A) \geq \mathbb{P}(Y \in A) > \mathbb{P}(Y \in B) \geq \mathbb{P}(f(Y) = c_B) \tag{12}$$

Proof sketch

We can compute the following:

$$\mathbb{P}(Y \in A) = \Phi\left(\Phi^{-1}(\underline{p_A}) - \frac{\|\delta\|}{\sigma}\right) \tag{13}$$

$$\mathbb{P}(Y \in B) = \Phi\left(\Phi^{-1}(\overline{p_B}) + \frac{\|\delta\|}{\sigma}\right) \tag{14}$$

Finally, algebra shows that $\mathbb{P}(Y \in A) > \mathbb{P}(Y \in B)$ if and only if:

$$\|\delta\| < \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})) \tag{15}$$

Randomized smoothing

- Certified top-1 accuracy by ResNet50 on ImageNet with the random smoothing approach
  - Top row: the certified top-1 accuracy of 49% under adversarial perturbations $\ell_2 < 0.5$
    - This is achieved with noise level $\sigma = 0.25$
    - For any perturbation with radius $\ell_2 < 0.5$, the robust classifier will correctly predict the class
      - Note that perturbation with ℓ_2 norm < 0.5 is fairly small
      - For example, perturbation with $\ell_2 = 1$ can change one pixel by 1 (=255/255), or change 10 pixels by 0.3 (≈80/255), or change 1,000 pixels by 0.03 (≈8/255)
  - Increasing the ℓ_2 radius from 0.5 to 3.0 reduces the certified accuracy
  - For comparison, the standard top-1 accuracy on clean images by the smoothed classifier *g* is 67%

ℓ_2 RADIUS	BEST σ	CERT. ACC (%)	STD. ACC(%)
0.5	0.25	49	67
1.0	0.50	37	57
2.0	0.50	19	57
3.0	1.00	12	44