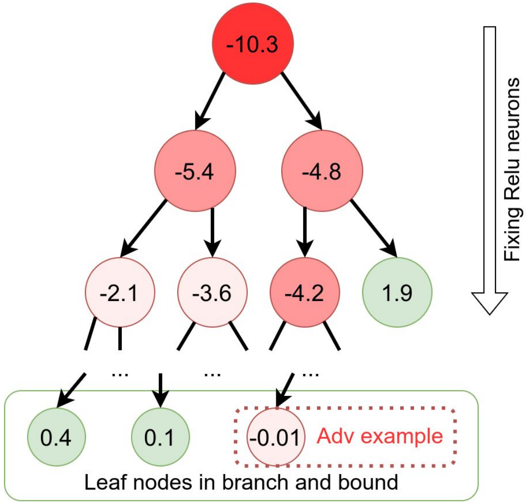
A **Branch and Bound** Framework for **Stronger Adversarial Attacks** of ReLU Networks



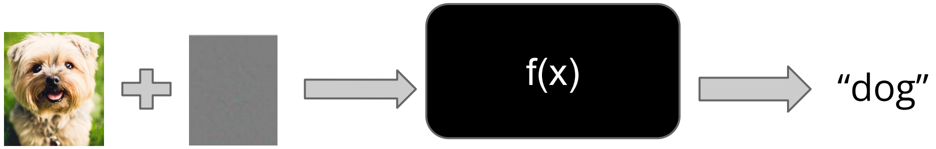
Huan Zhang* (CMU), Shiqi Wang* (Columbia), Kaidi Xu (Drexel University), Yihan Wang (UCLA), Suman Jana (Columbia), Cho-Jui Hsieh (UCLA), Zico Kolter (CMU/Bosch) (*co-first authors)

ICML International Conference On Machine Learning

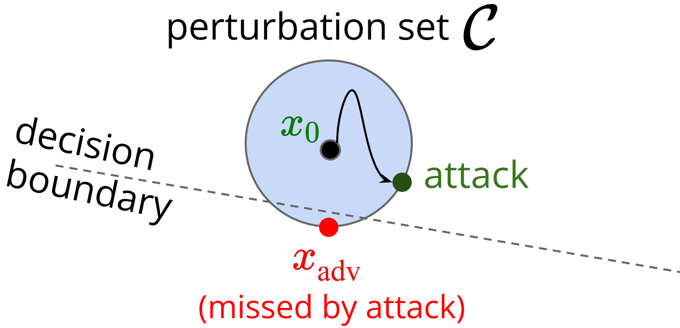
CROWN
Winner of International Verification of Neural Networks Competition (VNN-COMP'21)

BaB-attack has been integrated as part of our ***α,β*-CROWN Neural Network Verification Tool**: abCROWN.org

Revisit Adversarial Attacks



Adversarial noise in perturbation set



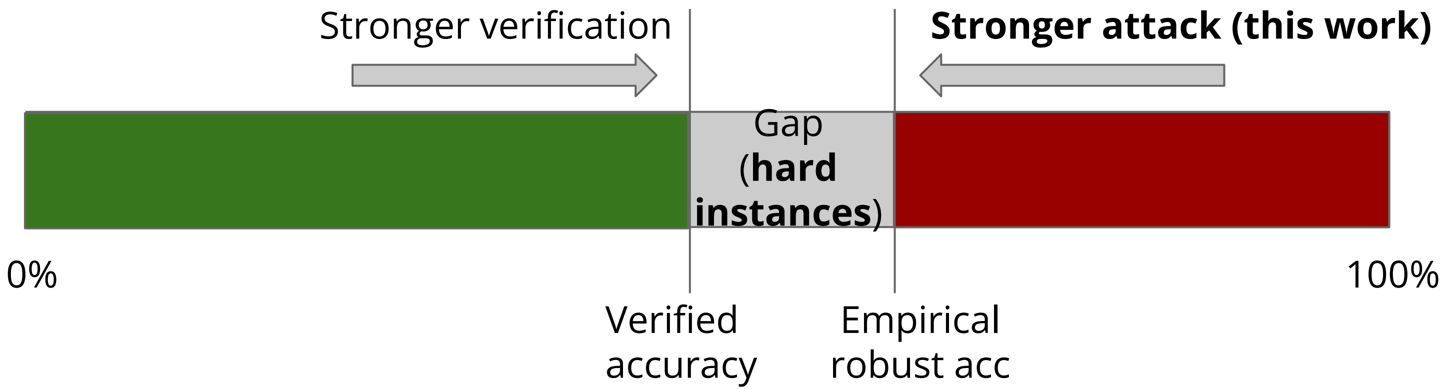
$$f^* = \min_{x \in \mathcal{C}} f(x)$$

(assuming a binary classifier with +1 label)

- Most existing attacks search adversarial examples in the **input space** (e.g., via gradient ascent)
- Cannot generally converge to the global optimal; need good initialization; we **cannot systematically enumerate** the **continuous input space**

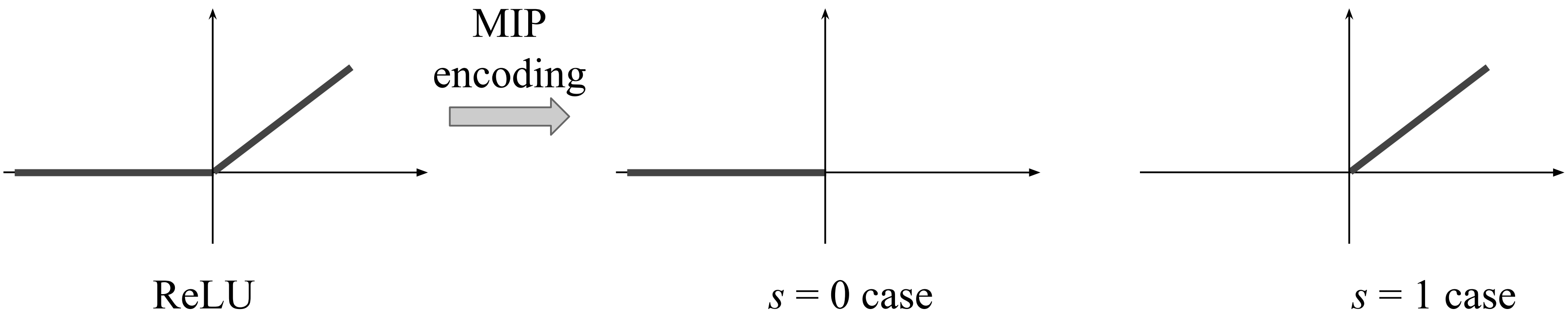
Do We Have Strong Enough Attacks?

- Can try to either verify (guaranteed robust) or attack each example (guaranteed vulnerable)?
- We often cannot **precisely characterize** the robustness of a model (*even for small models*): there exists a **gap** between verification and attacks
- SOTA verifiers have made a good progress recently (VNN-COMP 2021)



MIP Formulation in *Activation Space* for Attacks

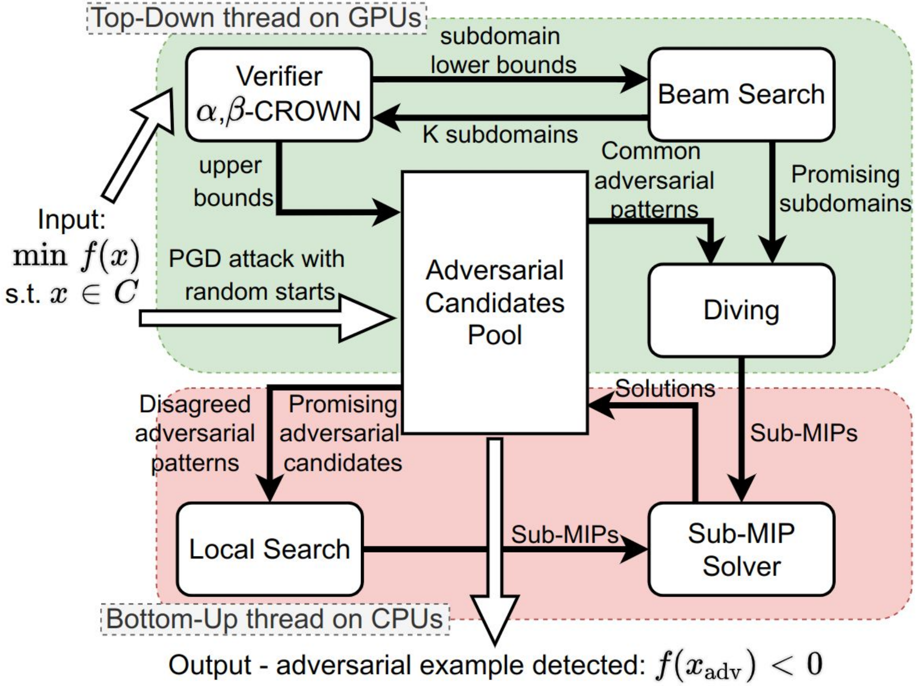
- For ReLU networks, it is possible to encode an adversarial attack as a **Mixed Integer Programming** problem (Tjeng et al., 2018)
- An adversarial example can be represented in activation space (set of binary variables representing ReLUs), which is **discrete** and can be **systematically enumerated**
- A MIP solver can search in activation space, but is often slow



Our Branch and Bound Attack

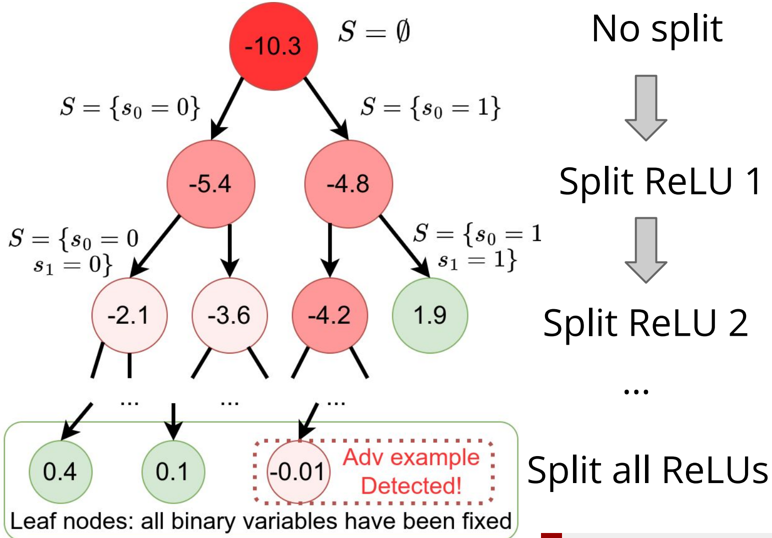
Goal:

- An adversarial attack based a **systematic search in activation space**, stronger than input space search
- Further **close the gap** between verification and attacks, and give a more precise characterization of NN
- **Efficient** and GPU accelerated, much faster than using MIP solvers directly



Searching Attacks in *Activation Space*

- **Systematically searching** in activation space using **branch and bound**
- Each ReLU neuron can be split into the s=0 and s=1 cases
- NN output can be lower bounded after each split

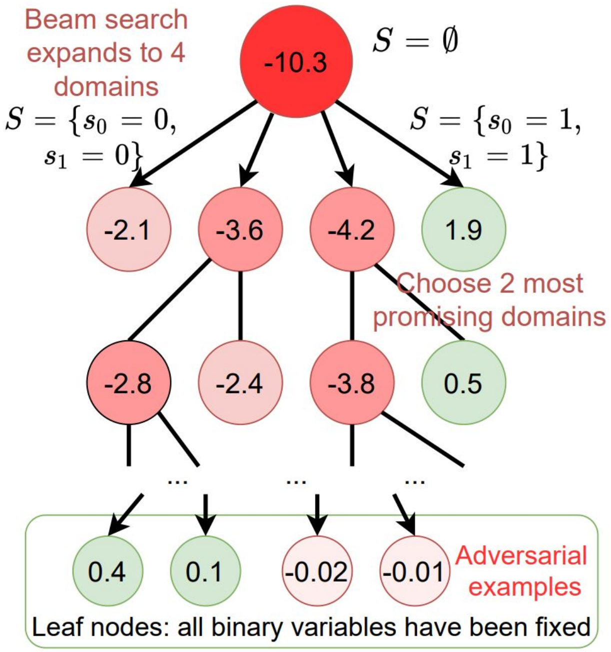


Adversarial examples located at **leaves** with bounds <= 0

Challenge: searching in activation space can be slow with many ReLU neurons

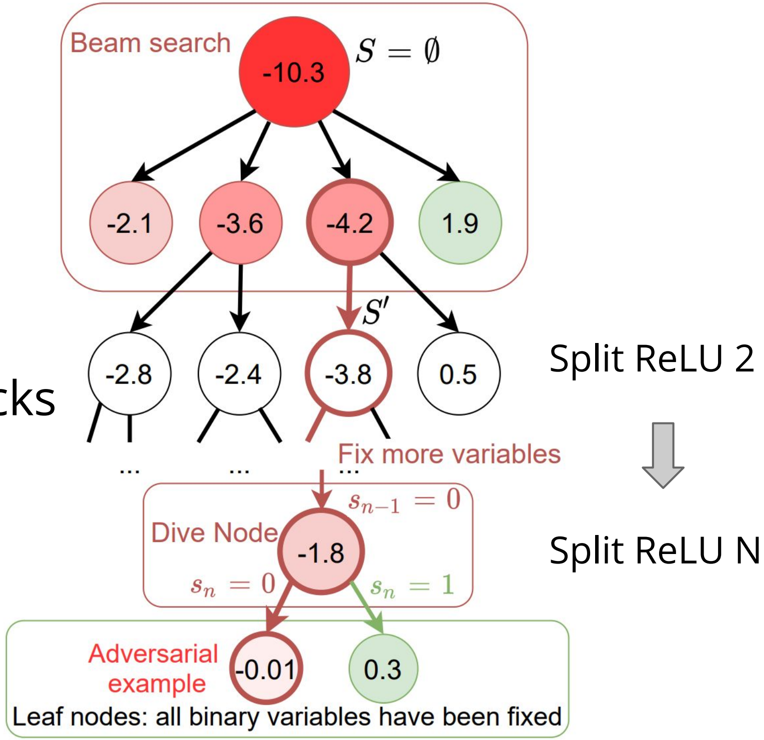
Our Strategies: Beam Search with NN Verifiers

- Challenge: how to reach leaf nodes quickly to locate adv. examples?
- Strategy 1: use **beam search** guided by neural network verifiers
- Benefits:
  - Prioritize most promising subdomains, reducing search space
  - **GPU acceleration** with bound propagation based NN verifiers (e.g., *α*, *β*-CROWN)



Our Strategies: Diving

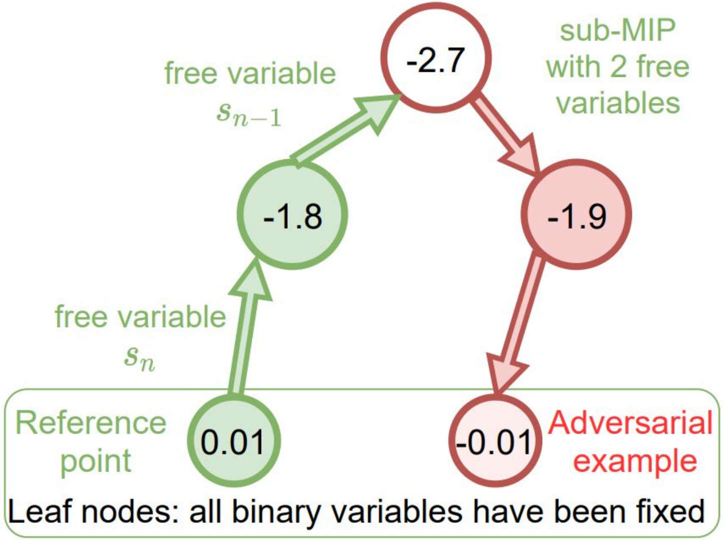
- Strategy 2: Go deeper in the search train by **fixing more variables** at once, based on statistics on adversarial candidates (e.g., common activation patterns)
- Benefits:
  - Utilize information from adversarial candidates generated from cheap attacks (e.g., PGD)
  - Reach leaf nodes faster



Our Strategies: Local Search

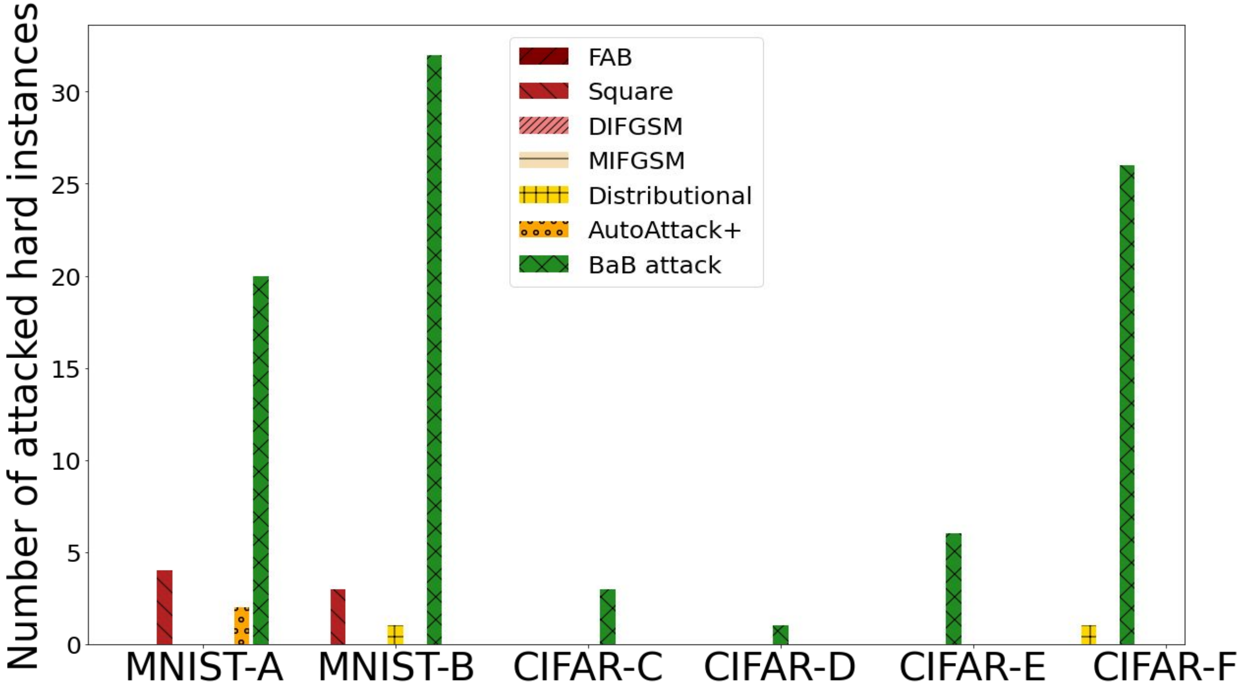
- Strategy 2: Go deeper in the search train by **fixing more variables** at once, based on statistics on adversarial candidates (e.g., common activation patterns)

- Strategy 3: **Local search** in activation spa around an adversarial candidate



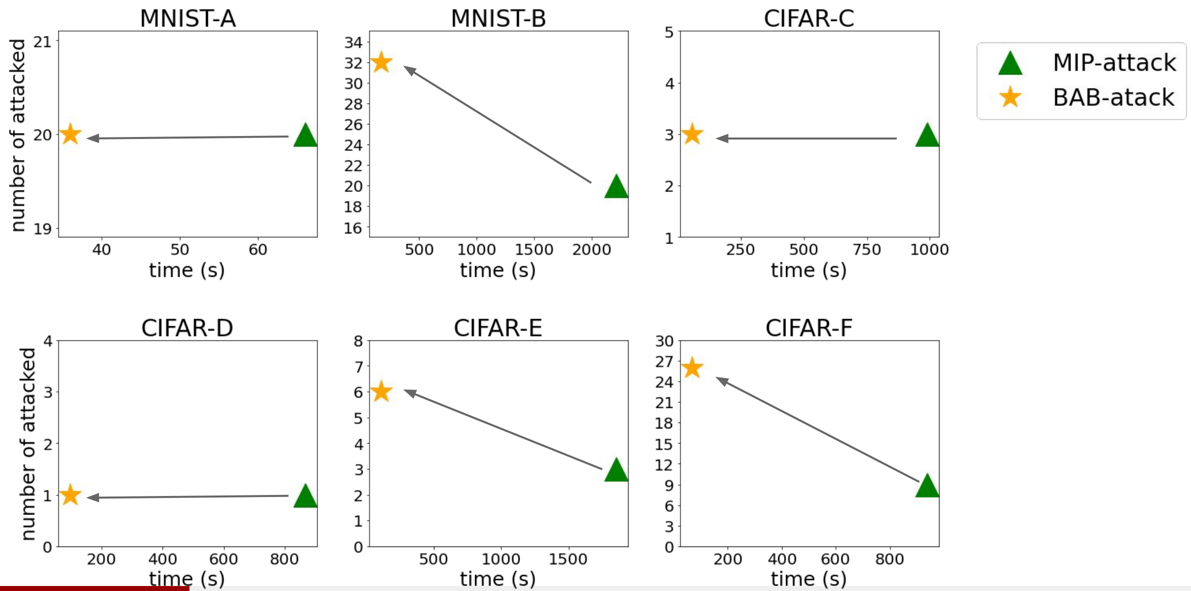
Results (BaB Attack vs. Input Space Attacks)

- Results on **hard instances** (*cannot* be verified by any NN verifiers, and *cannot* be attacked by **1000-step PGD with 500 restarts + AutoAttack**)



Results (BaB Attack vs. MIP-based Attack)

- Solve the MIP formulation for attack directly can be quite slow (no GPU acceleration, no information from cheap attacks)
- Faster and often can find more adversarial examples



Thank you!

Email: **huan@huan-zhang.com**

BaB-attack has been integrated as part of our α, β-CROWN Verification Tool:

abCROWN.org

Interested in NN verification? Attend the **ICML Workshop on Formal Verification of Machine Learning** on July 22 (Friday)