BICCOS: Scalable Neural Network Verification with Branch-and-bound Inferred Cutting Planes

Duo Zhou[1] Christopher Brix[2] Grani A. Hanasusanto[1] Huan Zhang[1]

[1]University of Illinois Urbana-Champaign

[2]RWTH Aachen University

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

RWTH AACHEN UNIVERSITY

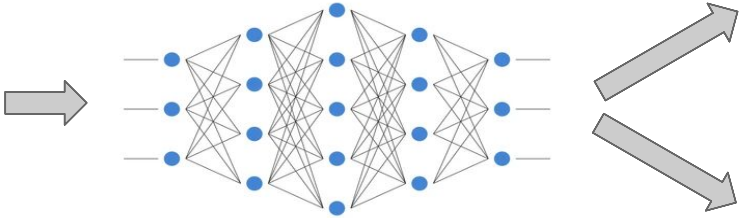
NEURAL INFORMATION PROCESSING SYSTEMS

Motivation: Scalable Neural Network Verification

$\forall x \in \mathcal{S}$

Eykholt et al. 2018
(Real world attack to NN)

$f(x)$



STOP 😀

$f(x) > 0$

Speed Limit 😈

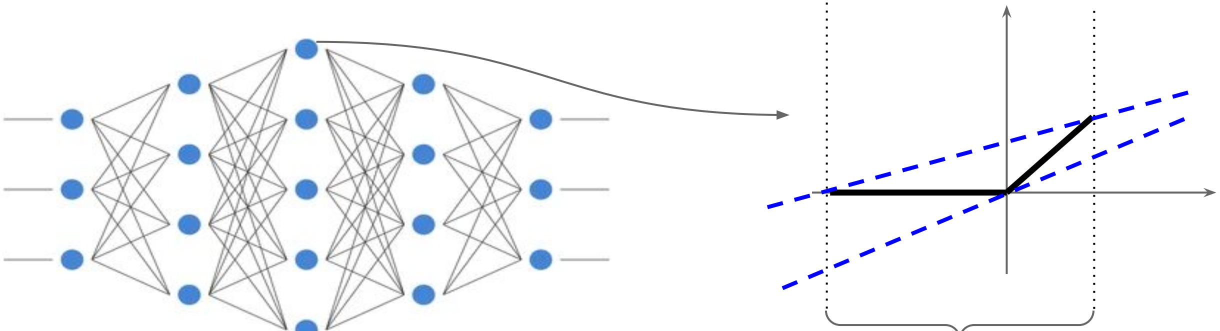
Otherwise

Problem Statement. Prove: $\forall x \in \mathcal{S}, f(x) > 0 \Leftrightarrow \min_{x \in \mathcal{S}} f(x) > 0$

SOTA: GCP-CROWN[1], Bound Propagation + *General* Cutting Planes.
Challenges: Cutting Planes are from external MIP solvers, which **can not scale**
Our Goal: *specialized* and efficient cutting planes for NN verification

[1] Zhang, Huan, et al. "General cutting planes for bound-propagation-based neural network verification." *Advances in neural information processing systems* 35 (2022): 1656-1670.

Preliminary: Math Programming & Bound Propagation

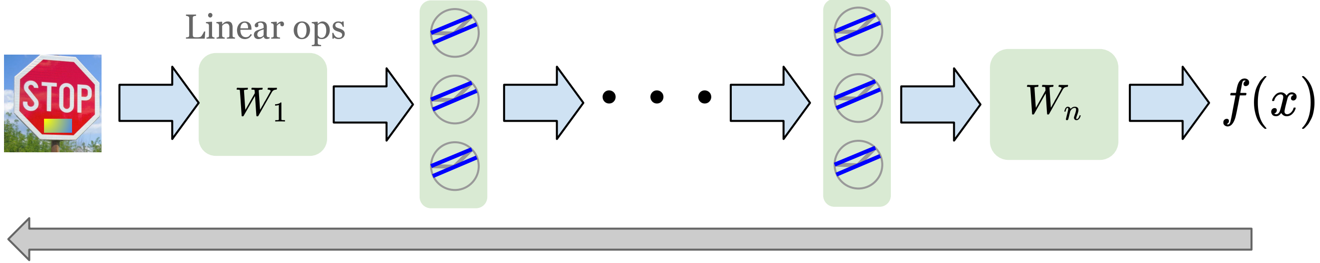


Pre-activation bounds

(can be pre-computed using CROWN)

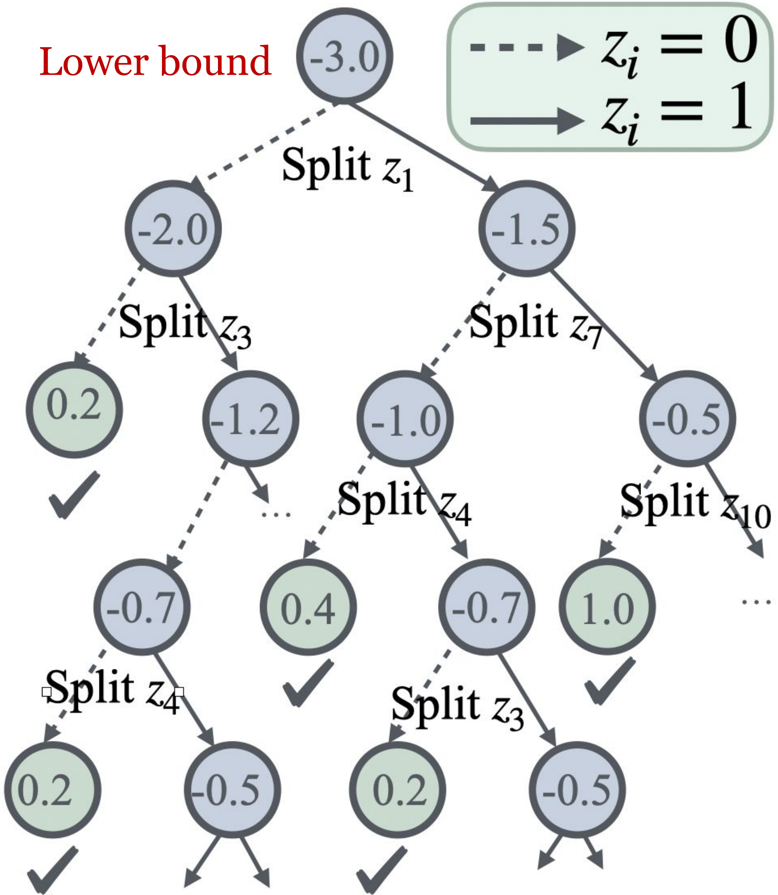
$$\hat{x}_j^{(i)} \geq 0;$$
$$\hat{x}_j^{(i)} \geq x_j^{(i)};$$
$$\hat{x}_j^{(i)} \leq u_j^{(i)} z_j^{(i)};$$
$$\hat{x}_j^{(i)} \leq x_j^{(i)} - l_j^{(i)}(1 - z_j^{(i)});$$
$$z_j^{(i)} \in \{0, 1\} \xrightarrow{\text{Relax}} [0, 1];$$

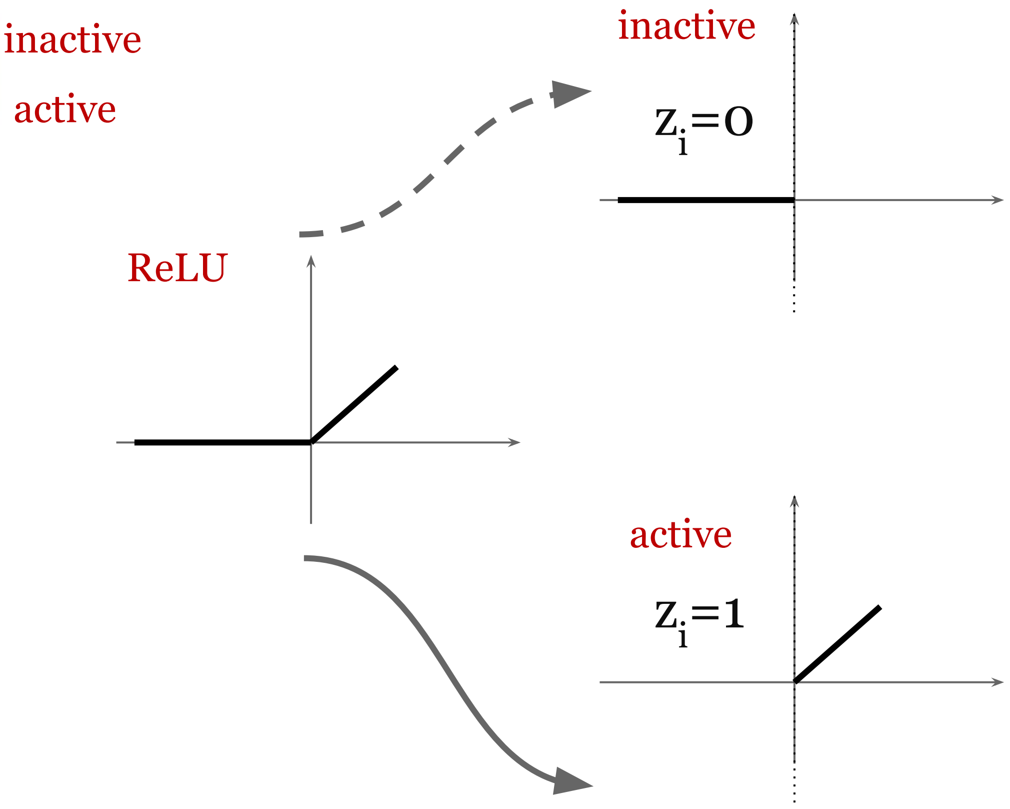
$x_0 := x$: Input; $\quad x_j^{(i)}$: Pre-ReLU(i, j); $\quad \hat{x}_j^{(i)}$: Post-ReLU(i, j); $\quad z_j^{(i)}$: ReLU Indicator(i, j)



Backward propagate lower bounds

Preliminary: Branch-and-bound (BaB)

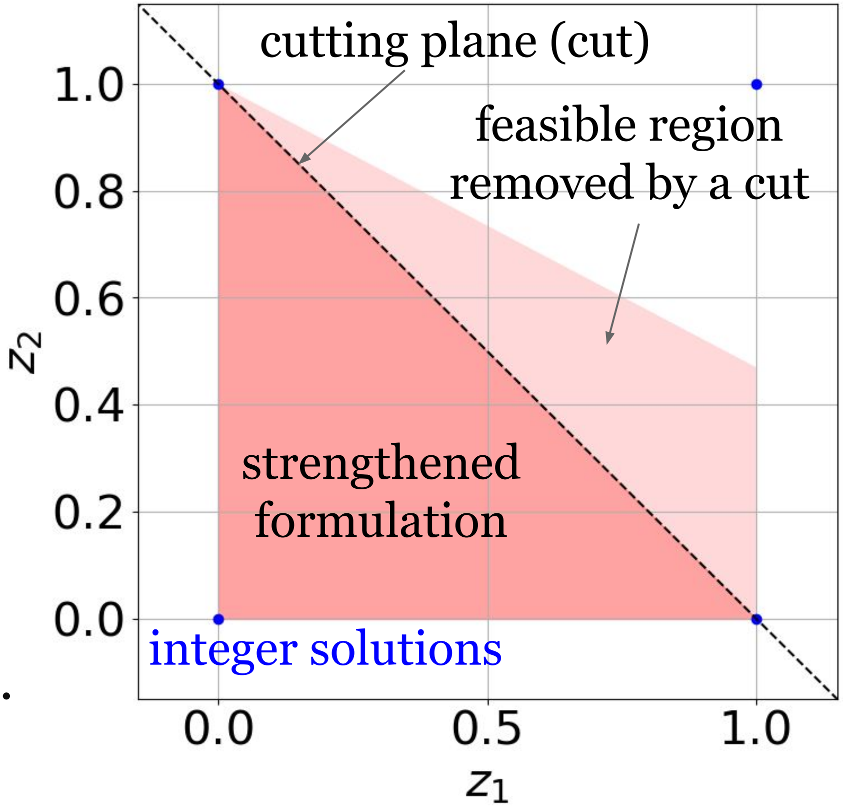




Goal of Our Paper: Find Cutting Planes

A cutting plane reduces the solution space by excluding infeasible regions without impacting feasible integer solutions.

Our objective: develop **specialized** cutting planes **from NN verification** procedure that strengthen the formulation while ensuring **scalability**.

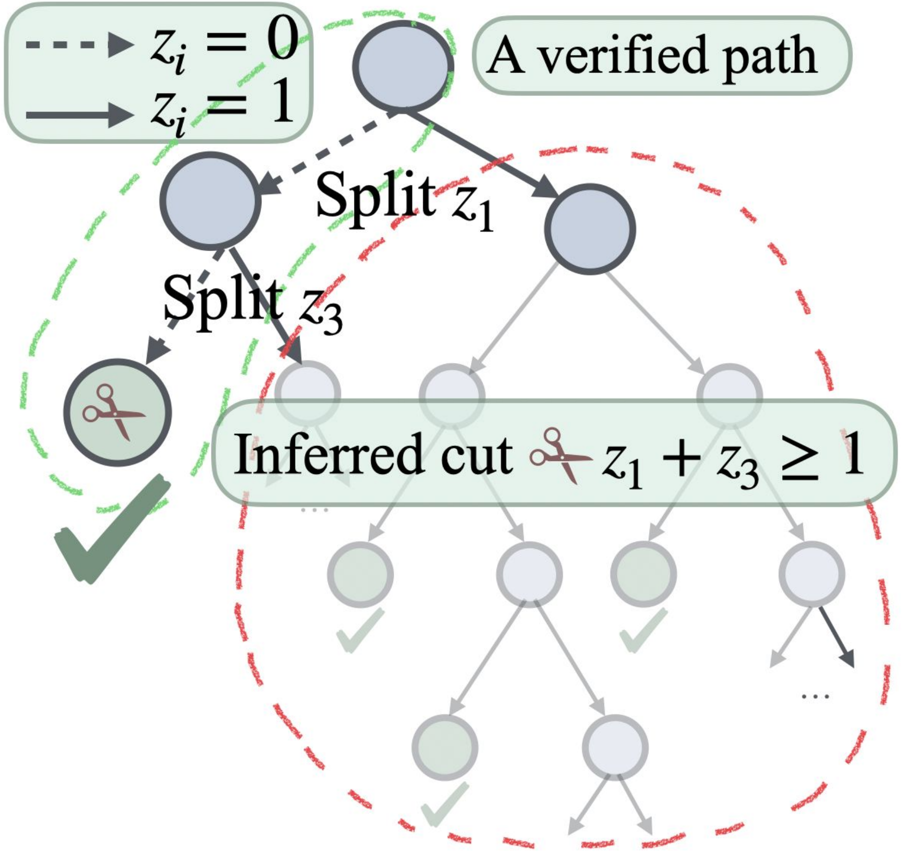


Main idea: BaB inferred cuts from verified subproblems

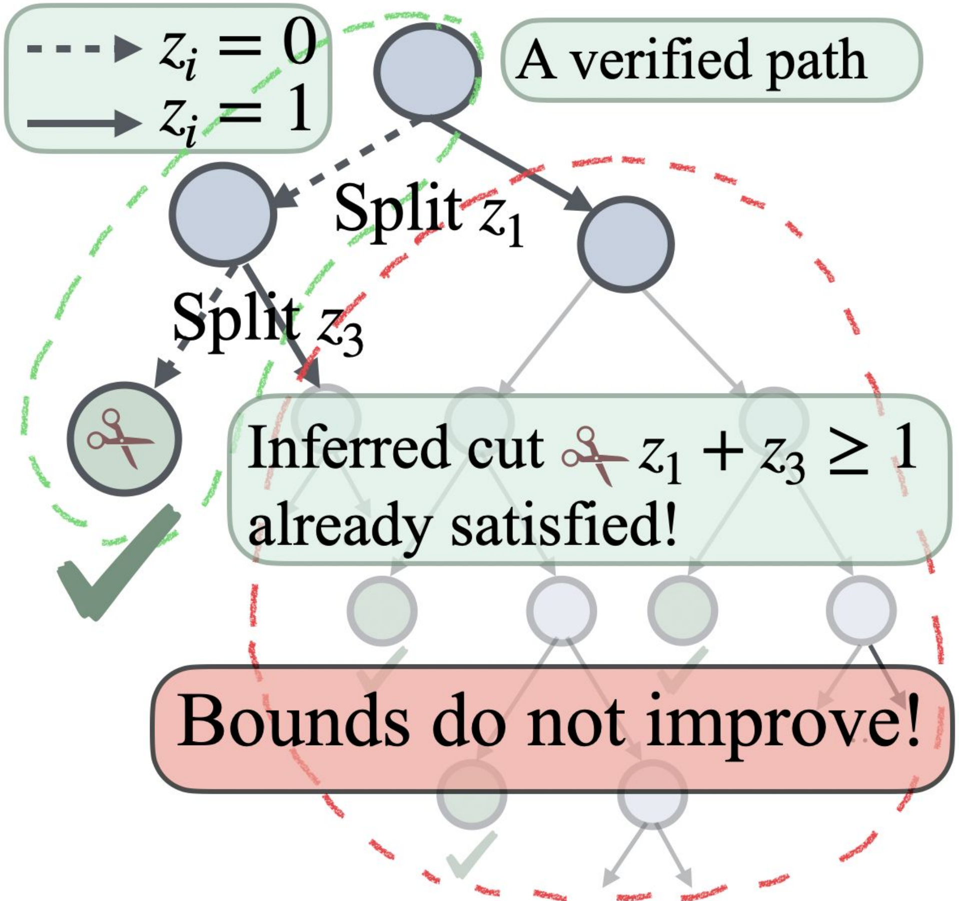
- After splitting z_1 and z_3 both to inactive cases, subproblem verified
- Neuron z_1 and z_3 cannot simultaneously be inactive. To exclude this situation, we create a new constraint (cut) on the relaxed z:

$$z_1 + z_3 \geq 1$$

- See the general form in our paper



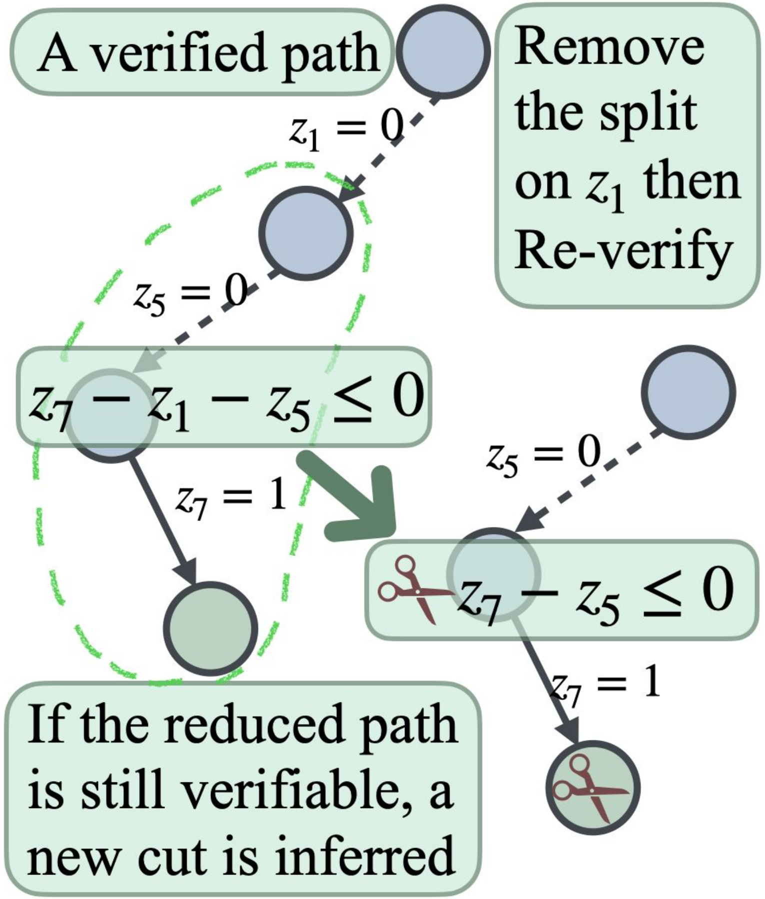
However, just adding this cut does not improve verification!



The rest of subproblems all have z_1=1 or z_3=1

None of the other subproblems on the search tree violate the cut $z_1 + z_3 \geq 1$, so their bounds do not improve.

Solution: Constraint Strengthening - shorter cuts are stronger!


A verified path
$z_1 = 0$
Remove the split on z_1 then Re-verify
$z_5 = 0$
$z_7 - z_1 - z_5 \leq 0$
$z_7 = 1$
$z_5 = 0$
$z_7 - z_5 \leq 0$
$z_7 = 1$
If the reduced path is still verifiable, a new cut is inferred

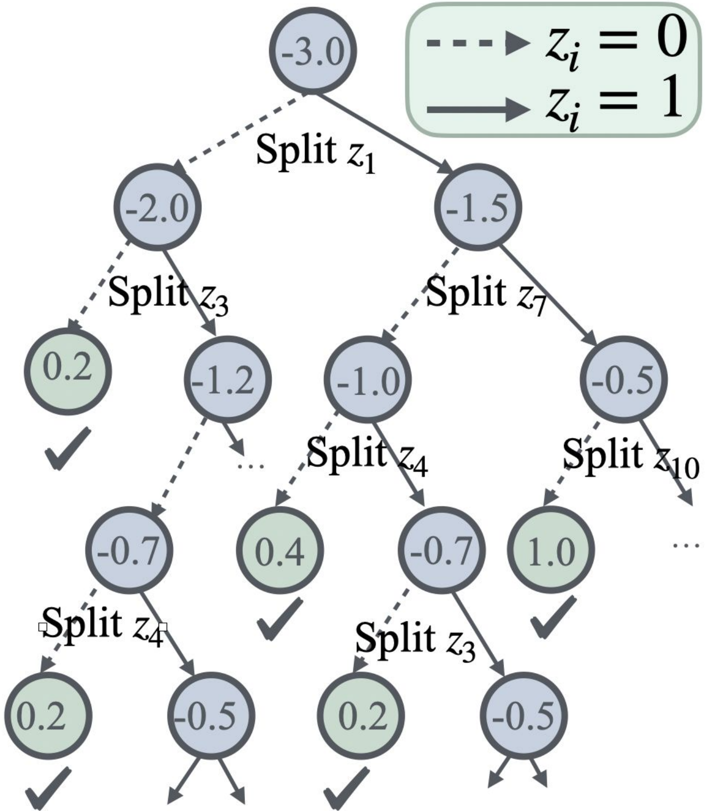
Starting from the verified path left,

If $z_1 = 0$ is not needed to make the path verifiable, we can get a strengthened cut.

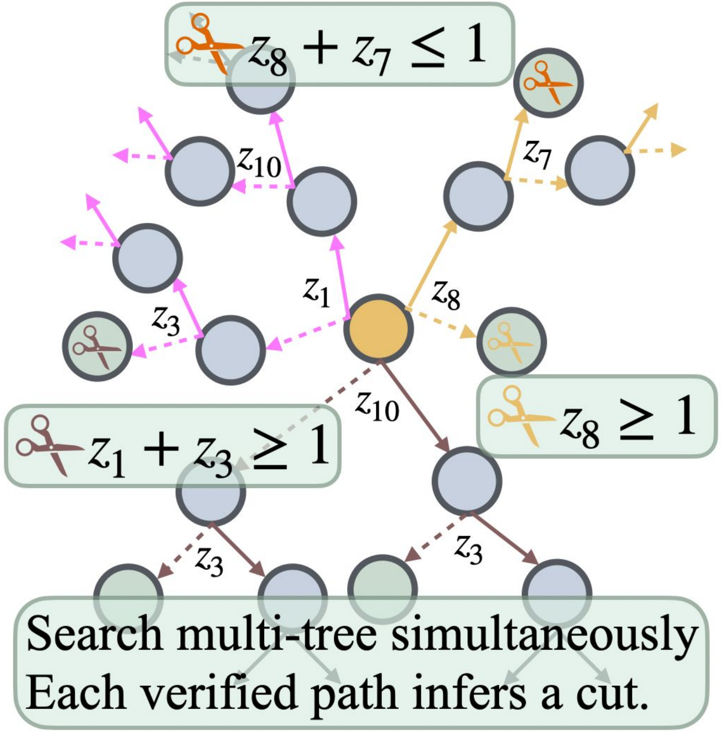
Simplifying the cut by involving fewer z variables reduces the dimensionality of the hyperplane

Even more short cuts via Multi-Tree Searching (MTS)

Regular BaB Search Tree

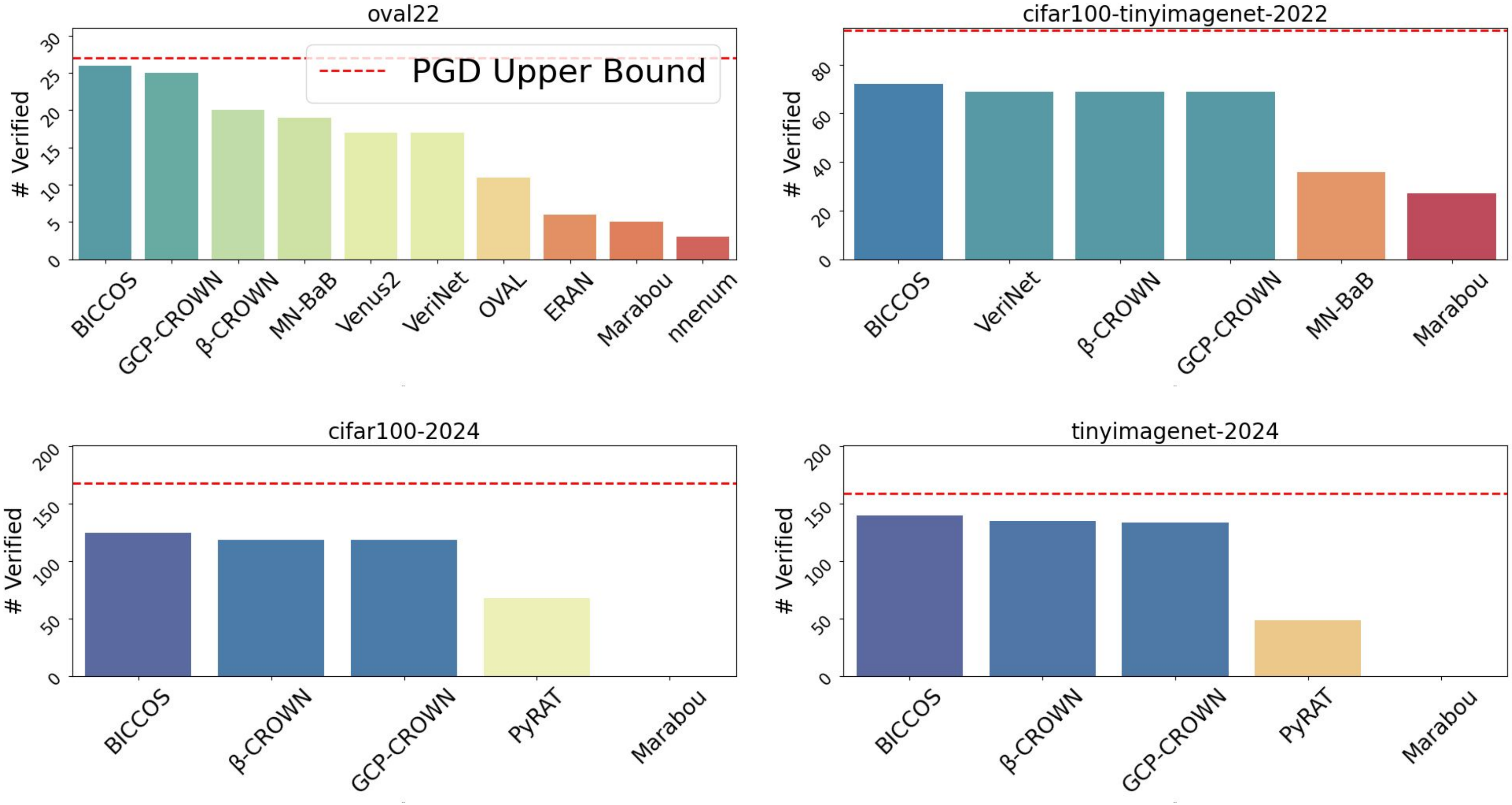


Multiple trees are searched in parallel, with cuts from one tree applied across others

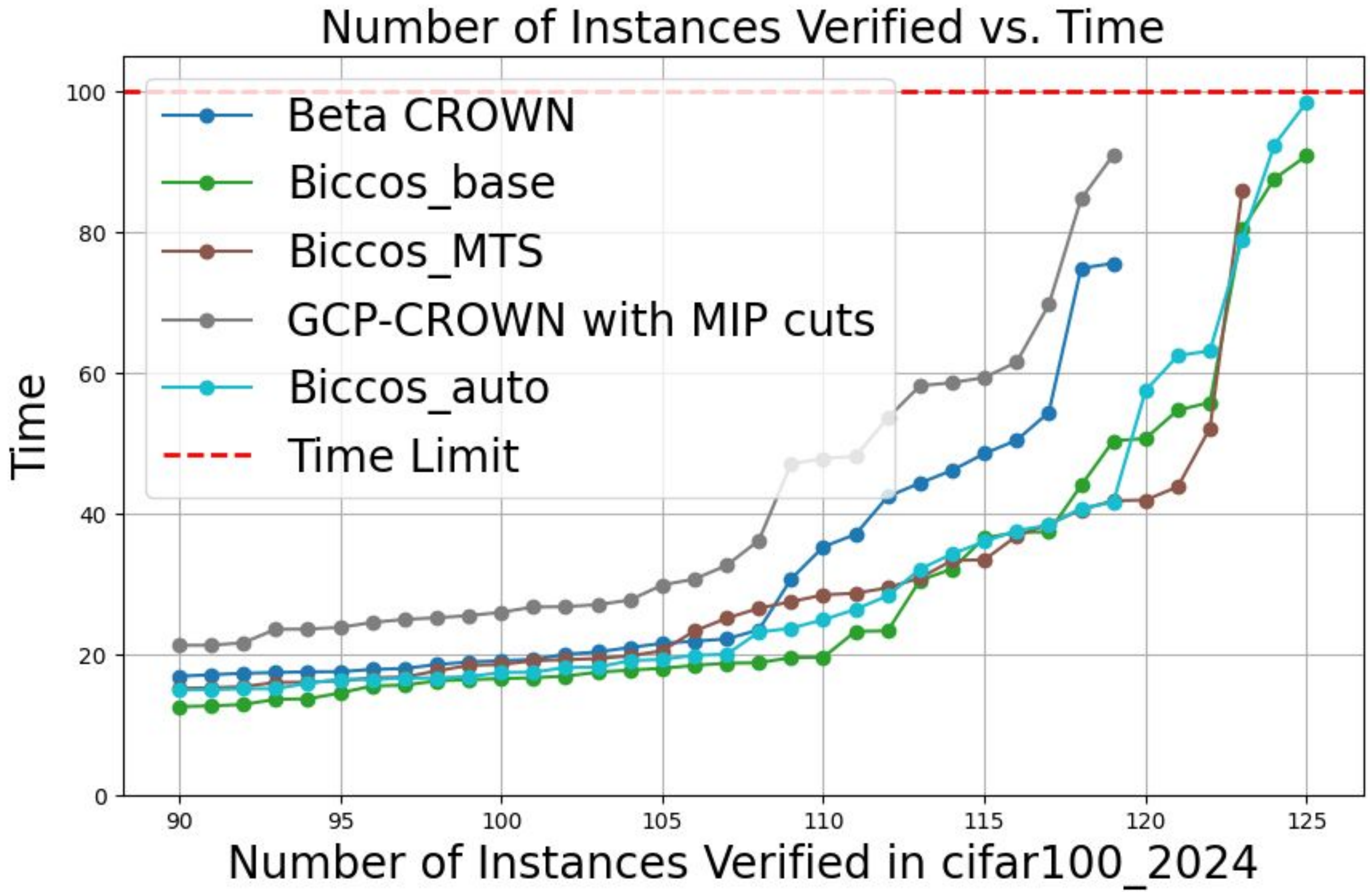


Search multi-tree simultaneously
Each verified path infers a cut.

Results on VNN-COMP benchmarks



Ablation study on components of BICCOS



Conclusion

- Specialized and scalable cutting plane generation for NN verification
- Cut improvements with constraint strengthening and multi-tree search
- SOTA performance on multiple NN verification benchmarks
- Integrated to the SOTA verifier α,β-CROWN: https://**abcrown.org**.

αβ
CROWN

Winner of International Verification
of Neural Networks Competitions
VNN-COMP 2021-2024