Scaling the Convex Barrier with Active Sets

Alessandro De Palma*, Harkirat Singh Behl*, Rudy Bunel,
Philip H.S. Torr, M. Pawan Kumar
University of Oxford

Neural Network Verification

Neural networks lack robustness (adversarial examples).

[Bunel et al., 2018]

Neural Network Verification

Neural networks lack robustness (adversarial examples).

Verification as optimisation:

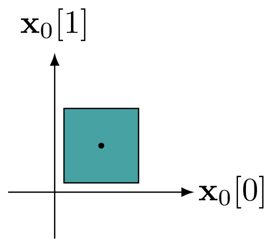
[Bunel et al., 2018]

Neural Network Verification

Neural networks lack robustness (adversarial examples).

Verification as optimisation:

$$\mathbf{x}_0 \in \mathcal{C}$$



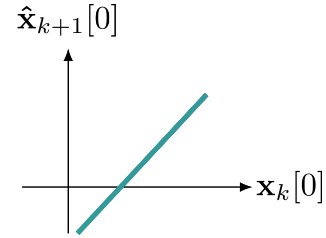
[Bunel et al., 2018]

Neural Network Verification

Neural networks lack robustness (adversarial examples).

Verification as optimisation:

$$\hat{\mathbf{x}}_{k+1} = W_{k+1}\mathbf{x}_k + \mathbf{b}_{k+1}$$



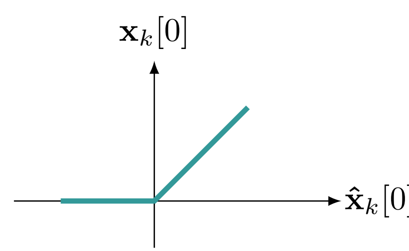
[Bunel et al., 2018]

Neural Network Verification

Neural networks lack robustness (adversarial examples).

Verification as optimisation:

$$\mathbf{x}_k = \sigma\left(\hat{\mathbf{x}}_k\right)$$



[Bunel et al., 2018]

Neural Network Verification

Neural networks lack robustness (adversarial examples).

Check the sign of:

$$
\begin{aligned}
\min_{\mathbf{x},\hat{\mathbf{x}}} \quad & \hat{x}_n \\
\text{s.t.} \quad & \mathbf{x}_0 \in \mathcal{C}, \\
& \hat{\mathbf{x}}_{k+1} = W_{k+1}\mathbf{x}_k + \mathbf{b}_{k+1} \quad k \in [\![0, n-1]\!], \\
& \mathbf{x}_k = \sigma\left(\hat{\mathbf{x}}_k\right) \qquad\qquad\qquad\quad k \in [\![1, n-1]\!].
\end{aligned}
$$

[Bunel et al., 2018]

Neural Network Verification

Neural networks lack robustness (adversarial examples).

Check the sign of:

$$
\begin{aligned}
\min_{\mathbf{x},\hat{\mathbf{x}}} \quad & \hat{x}_n \\
\text{s.t.} \quad & \mathbf{x}_0 \in \mathcal{C}, \\
& \hat{\mathbf{x}}_{k+1} = W_{k+1}\mathbf{x}_k + \mathbf{b}_{k+1} \quad k \in [\![0, n-1]\!], \\
& \mathbf{x}_k = \sigma\left(\hat{\mathbf{x}}_k\right) \quad\quad\quad\quad\quad\quad\ \ k \in [\![1, n-1]\!].
\end{aligned}
$$

NP-HARD

[Bunel et al., 2018]

Neural Network Bounding: Planet

Approximate $\min \hat{x}_n$ via a lower bound:

[Wong and Kolter, 2018; Zhang et al., 2018; Dvijotham et al. 2018; Singh et al. 2018; Bunel et al., 2020.]

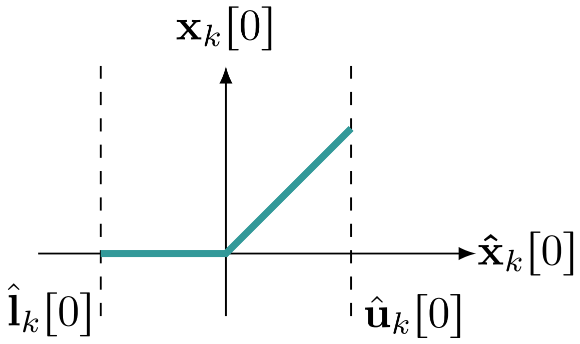
[Ehlers 2017]

Neural Network Bounding: Planet

Approximate $\min \hat{x}_n$ via a lower bound:

[Wong and Kolter, 2018; Zhang et al., 2018; Dvijotham et al. 2018; Singh et al. 2018; Bunel et al., 2020.]

$$\sigma\left(\hat{\mathbf{x}}_k\right)$$



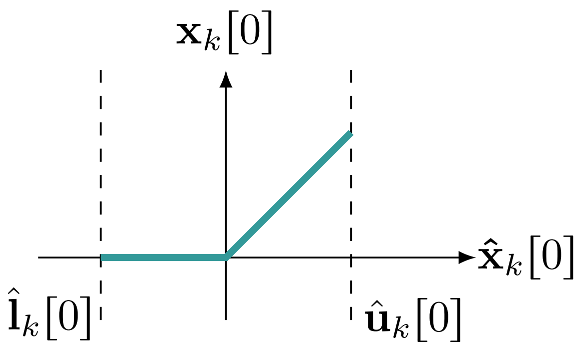
[Ehlers 2017]

Neural Network Bounding: Planet

Approximate $\min \hat{x}_n$ via a lower bound:

[Wong and Kolter, 2018; Zhang et al., 2018; Dvijotham et al. 2018; Singh et al. 2018; Bunel et al., 2020.]

$$\sigma\left(\hat{\mathbf{x}}_k\right)$$



$\longrightarrow$

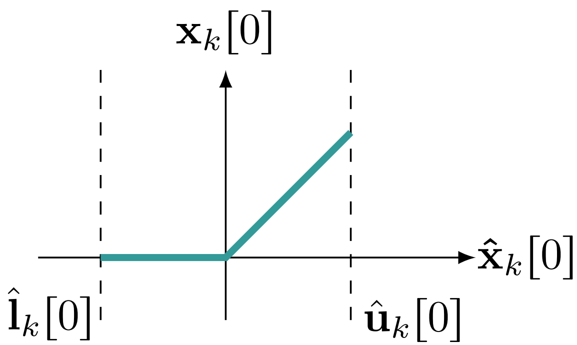
[Ehlers 2017]

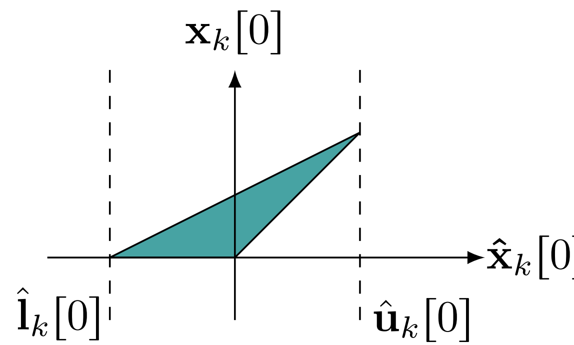
Neural Network Bounding: Planet

Approximate $\min \hat{x}_n$ via a lower bound:

[Wong and Kolter, 2018; Zhang et al., 2018; Dvijotham et al. 2018; Singh et al. 2018; Bunel et al., 2020.]

$$\sigma(\hat{\mathbf{x}}_k) \quad \longrightarrow \quad \mathrm{Conv}(\sigma(\hat{\mathbf{x}}_k), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k)$$

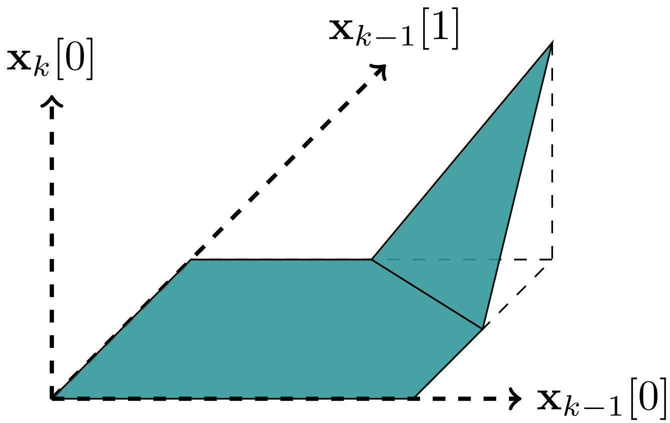




[Ehlers 2017]

Convex Barrier

$$\sigma\left(\hat{\mathbf{x}}_k\right)$$

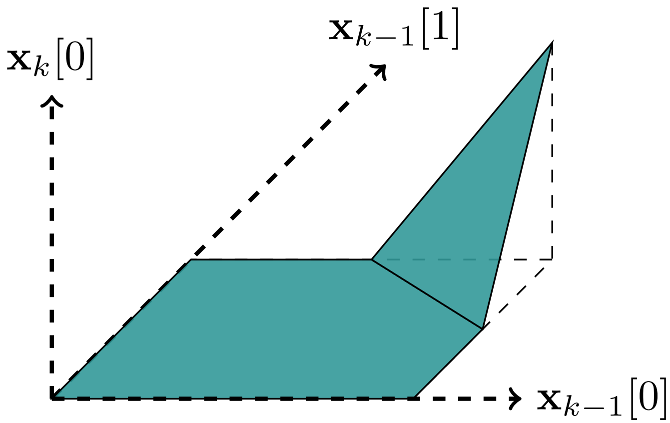


[Salman et al., 2019]

Figures modified from [Anderson et al., 2020]

Convex Barrier

$\sigma(\hat{\mathbf{x}}_k)$



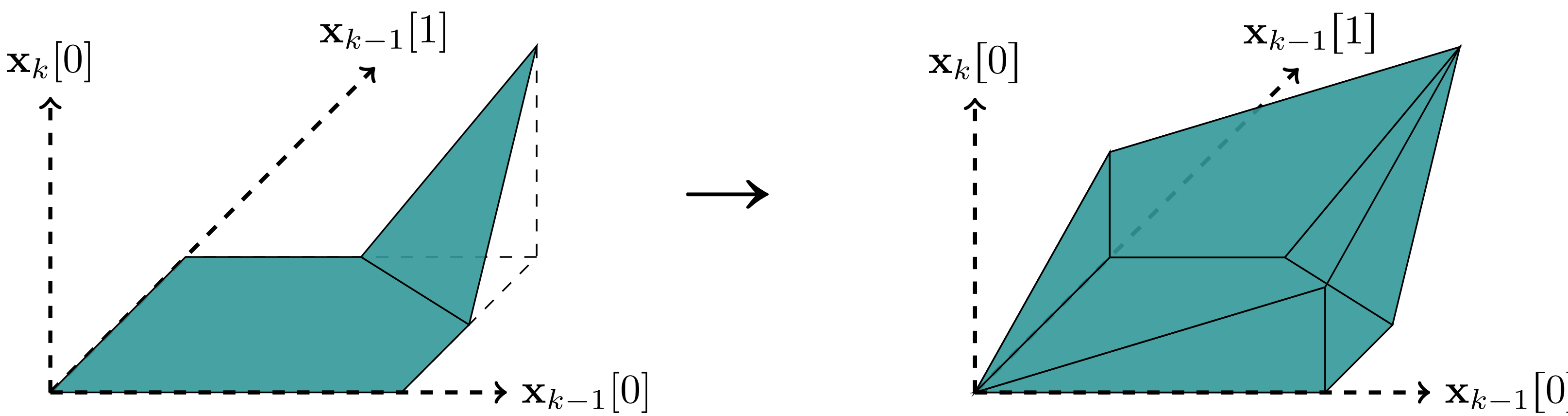
$\longrightarrow$

[Salman et al., 2019]

Figures modified from [Anderson et al., 2020]

Convex Barrier

$$\sigma(\hat{\mathbf{x}}_k) \qquad \mathrm{Conv}(\sigma(\hat{\mathbf{x}}_k), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k)$$

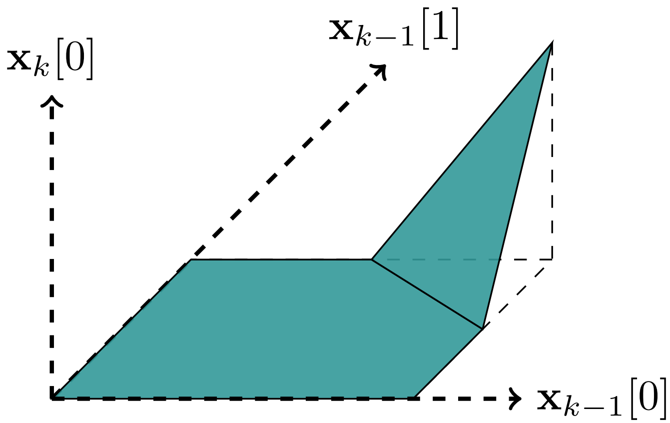


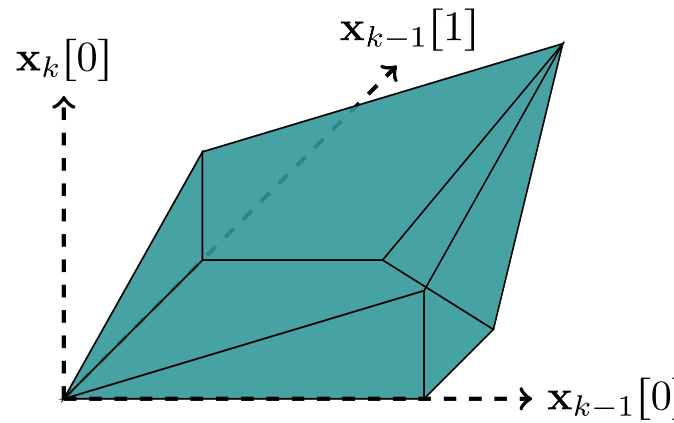
[Salman et al., 2019]

Figures modified from [Anderson et al., 2020]

Convex Barrier

$$\sigma(\hat{\mathbf{x}}_k) \longrightarrow \mathrm{Conv}(\sigma(\hat{\mathbf{x}}_k), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k)$$





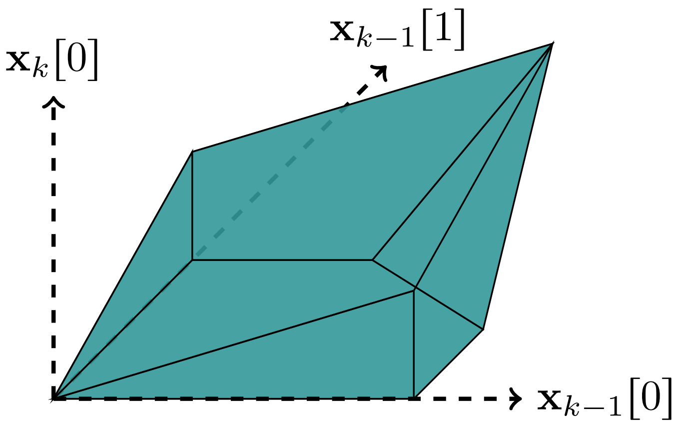
Loose.

[Salman et al., 2019]

Figures modified from [Anderson et al., 2020]

A Tighter Relaxation

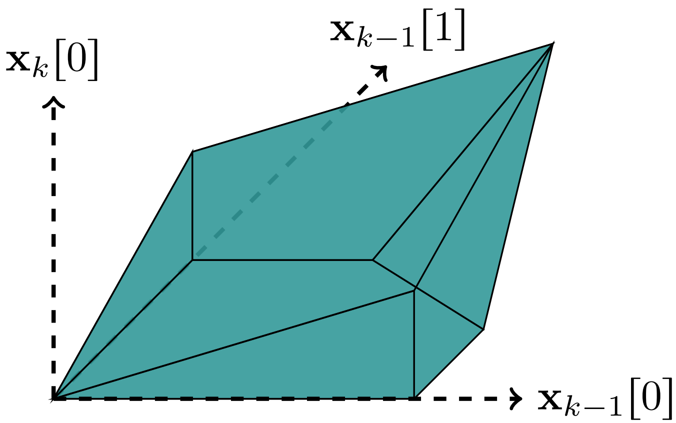
$$\mathrm{Conv}(\sigma(\hat{\mathbf{x}}_k), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k)$$



Figures modified from [Anderson et al., 2020]

A Tighter Relaxation

$$\mathrm{Conv}(\sigma(\hat{\mathbf{x}}_k), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k)$$

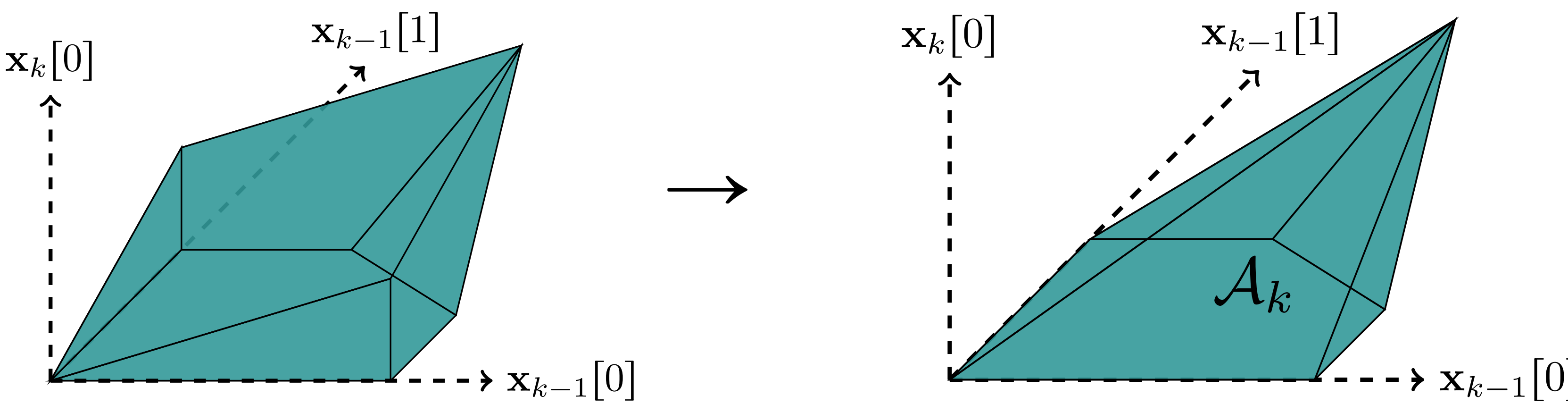


$\longrightarrow$

Figures modified from [Anderson et al., 2020]

A Tighter Relaxation

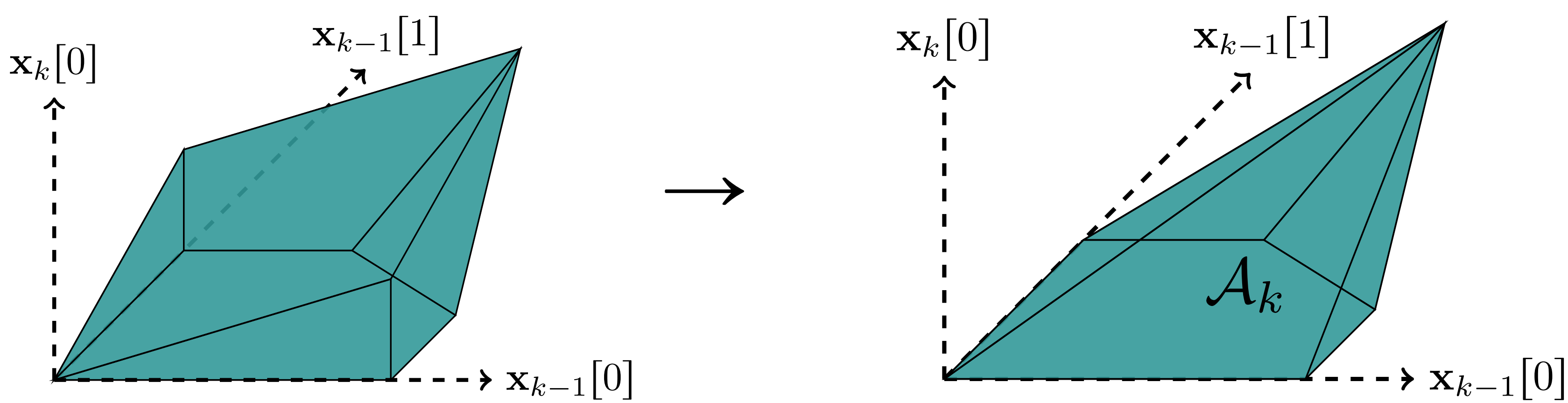
$$\mathrm{Conv}(\sigma(\hat{\mathbf{x}}_k), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k) \qquad \mathrm{Conv}(\sigma\left(W_k \mathbf{x}_{k-1} + \mathbf{b}_k\right), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k)$$



Figures modified from [Anderson et al., 2020]

A Tighter Relaxation

$$\mathrm{Conv}(\sigma(\hat{\mathbf{x}}_k), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k) \qquad \longrightarrow \qquad \mathrm{Conv}(\sigma\left(W_k \mathbf{x}_{k-1} + \mathbf{b}_k\right), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k)$$

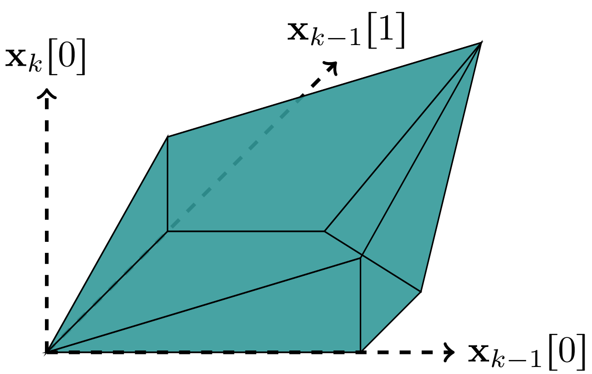


Exponentially many constraints.

Figures modified from [Anderson et al., 2020]

Cutting Plane Algorithm

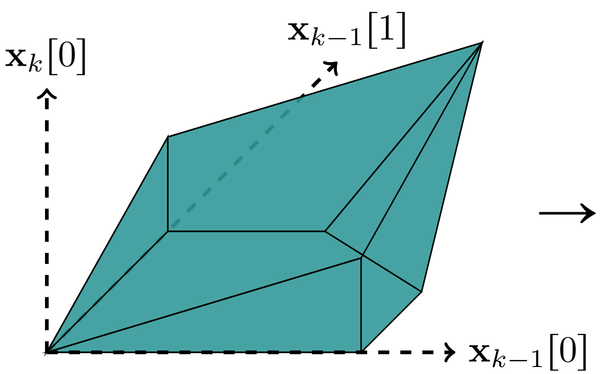
Most violated constraint from $\mathrm{Conv}(\sigma\left(W_k\mathbf{x}_{k-1}+\mathbf{b}_k\right), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k)$ at any point found in *linear-time.*



[Anderson et al., 2020]

Cutting Plane Algorithm

Most violated constraint from $\mathrm{Conv}(\sigma\left(W_k \mathbf{x}_{k-1} + \mathbf{b}_k\right), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k)$ at any point found in *linear-time.*



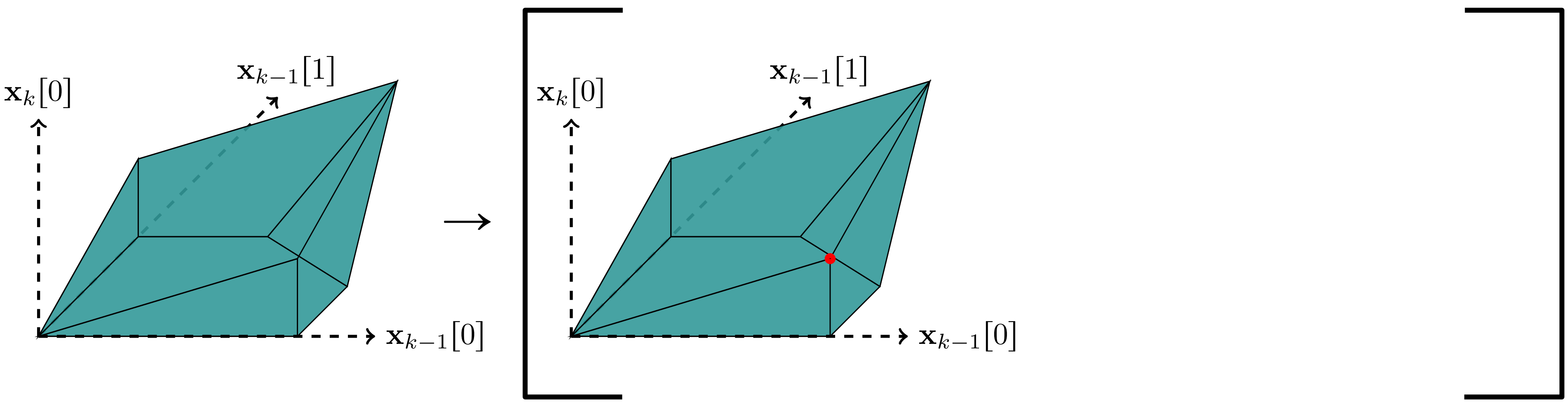
$\rightarrow \left[\qquad\qquad\qquad\qquad \right]$

iteratively

[Anderson et al., 2020]

Cutting Plane Algorithm

Most violated constraint from $\mathrm{Conv}(\sigma\left(W_k \mathbf{x}_{k-1} + \mathbf{b}_k\right), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k)$ at any point found in *linear-time.*

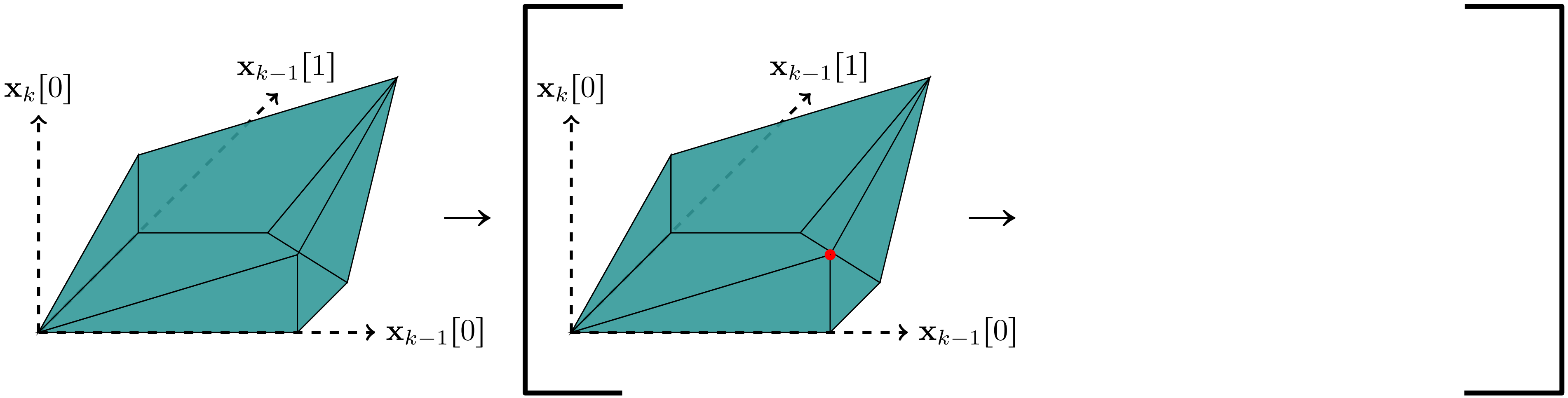


iteratively

[Anderson et al., 2020]

Cutting Plane Algorithm

Most violated constraint from $\mathrm{Conv}(\sigma\left(W_k \mathbf{x}_{k-1} + \mathbf{b}_k\right), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k)$ at any point found in *linear-time.*

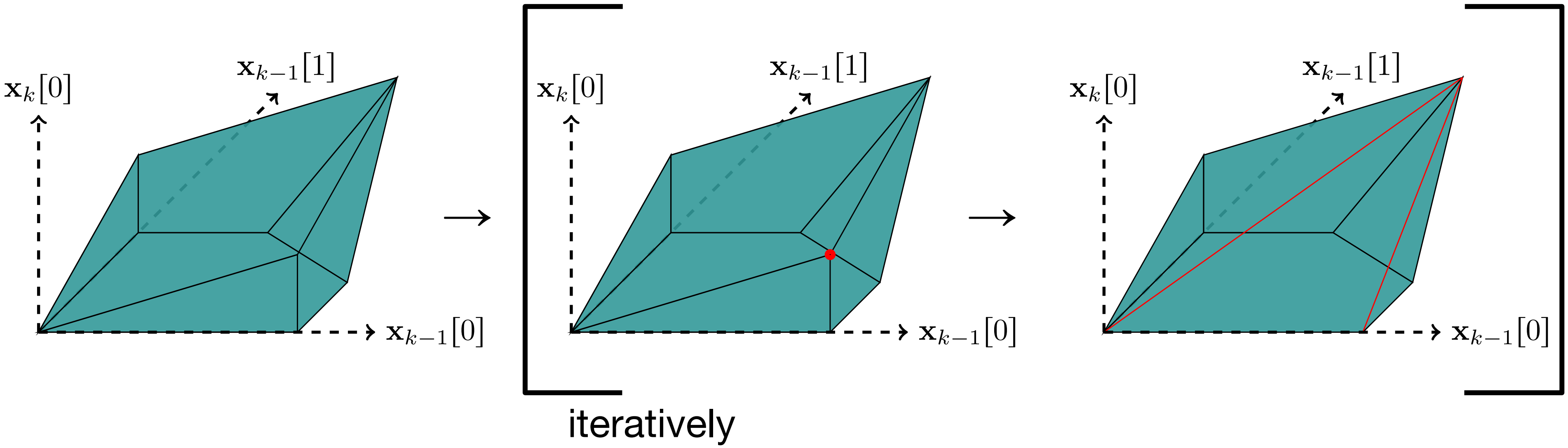


iteratively

[Anderson et al., 2020]

Cutting Plane Algorithm

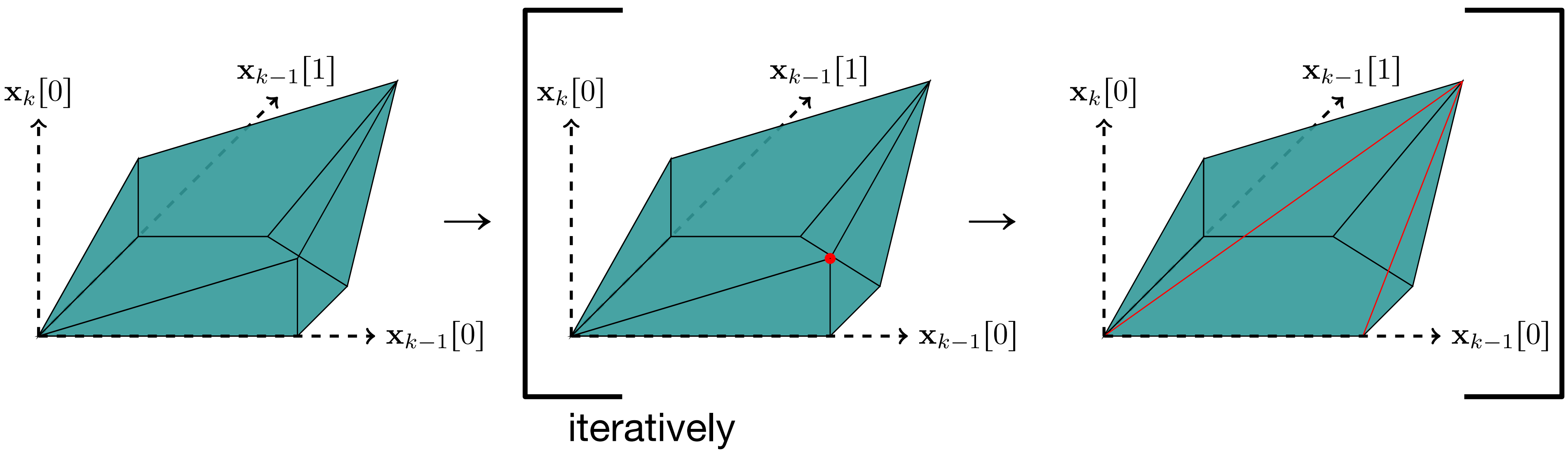
Most violated constraint from $\mathrm{Conv}(\sigma\left(W_k \mathbf{x}_{k-1} + \mathbf{b}_k\right), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k)$ at any point found in *linear-time.*



[Anderson et al., 2020]

Cutting Plane Algorithm

Most violated constraint from $\mathrm{Conv}(\sigma\left(W_k \mathbf{x}_{k-1} + \mathbf{b}_k\right), \hat{\mathbf{l}}_k, \hat{\mathbf{u}}_k)$ at any point found in *linear-time.*



Off-the-shelf LP solvers scale poorly with NN size.

[Bunel et al., 2020]

[Anderson et al., 2020]

An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

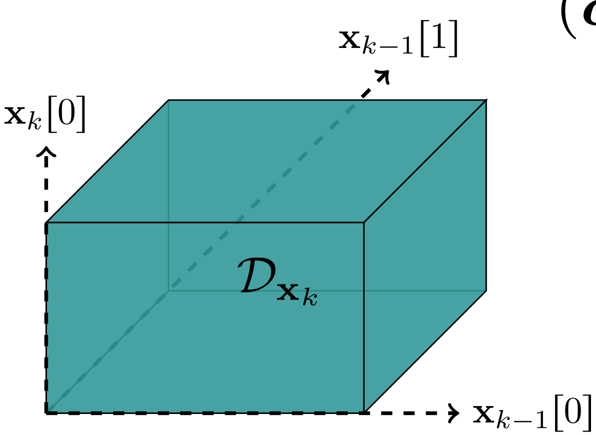
1. *anytime property*

An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

1. <u>anytime property</u>

$$\max_{(\boldsymbol{\alpha},\boldsymbol{\beta}_{\mathcal{A}})\geq 0} \left\{ \min_{(\mathbf{x},\mathbf{z})\in\mathcal{D}} \quad \mathcal{L}(\mathbf{x},\mathbf{z},\boldsymbol{\alpha},\boldsymbol{\beta}_{\mathcal{A}}) \right\}$$

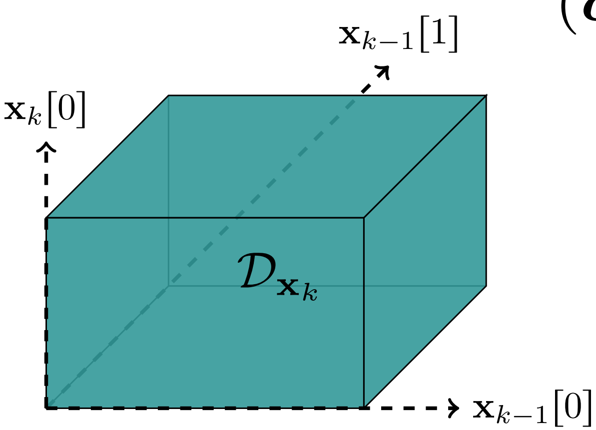


An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

1. *anytime property*

$$\max_{(\boldsymbol{\alpha},\boldsymbol{\beta}_{\mathcal{A}})\geq 0} \left\{ \min_{(\mathbf{x},\mathbf{z})\in\mathcal{D}} \quad \mathcal{L}(\mathbf{x},\mathbf{z},\boldsymbol{\alpha},\boxed{\boldsymbol{\beta}_{\mathcal{A}}}) \right\}$$



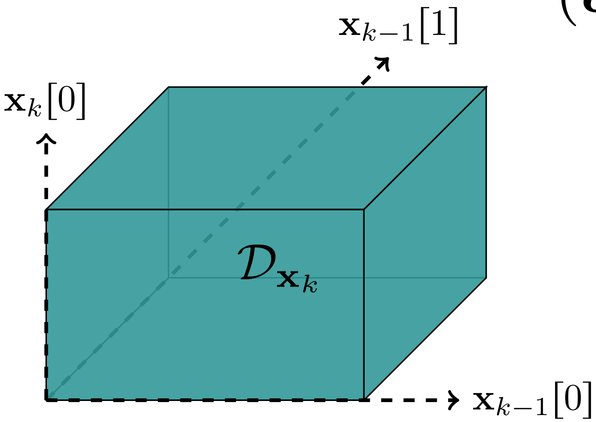
An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

1. <u>anytime property</u>

Exponentially many variables.

$$\max_{(\boldsymbol{\alpha},\boldsymbol{\beta}_{\mathcal{A}})\geq 0}\left\{\min_{(\mathbf{x},\mathbf{z})\in\mathcal{D}} \quad \mathcal{L}(\mathbf{x},\mathbf{z},\boldsymbol{\alpha},\boxed{\boldsymbol{\beta}_{\mathcal{A}}})\right\}$$



An Efficient Customised Solver

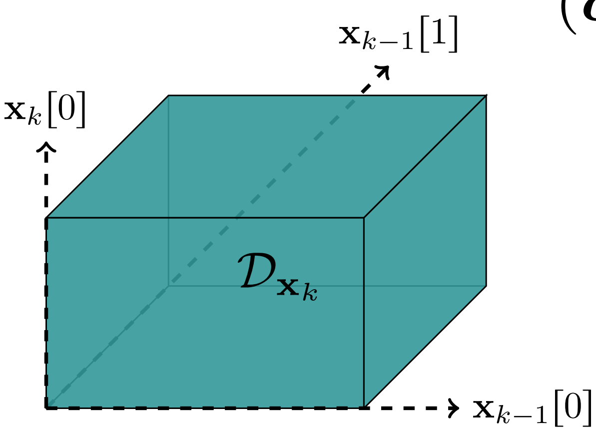
An efficient solver for the tight relaxation needs:

2. *sparsity*

An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

2. sparsity

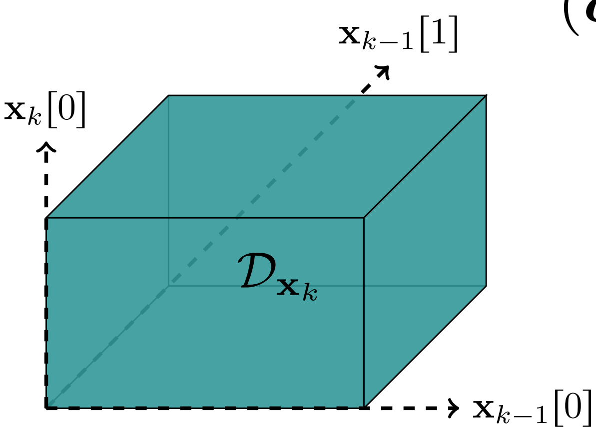


$$\max_{(\boldsymbol{\alpha},\boldsymbol{\beta}_{\mathcal{A}})\geq 0} \left\{ \min_{(\mathbf{x},\mathbf{z})\in\mathcal{D}} \quad \mathcal{L}(\mathbf{x},\mathbf{z},\boldsymbol{\alpha},\boldsymbol{\beta}_{\mathcal{A}}) \right\}$$

An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

2. *sparsity*

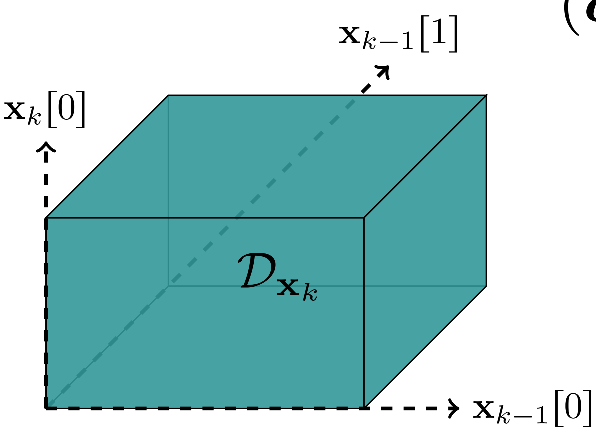


$$\max_{(\boldsymbol{\alpha},\boldsymbol{\beta}_{\mathcal{B}})\geqslant 0} \left\{ \min_{(\mathbf{x},\mathbf{z})\in\mathcal{D}} \quad \mathcal{L}_{\mathcal{B}}(\mathbf{x},\mathbf{z},\boldsymbol{\alpha},\boldsymbol{\beta}_{\mathcal{B}}) \right\}$$

An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

2. *sparsity*



$$\max_{(\boldsymbol{\alpha},\boldsymbol{\beta}_{\mathcal{B}})\geqslant 0}\left\{\min_{(\mathbf{x},\mathbf{z})\in\mathcal{D}} \quad \mathcal{L}_{\mathcal{B}}(\mathbf{x},\mathbf{z},\boldsymbol{\alpha},\boxed{\boldsymbol{\beta}_{\mathcal{B}}})\right\}$$

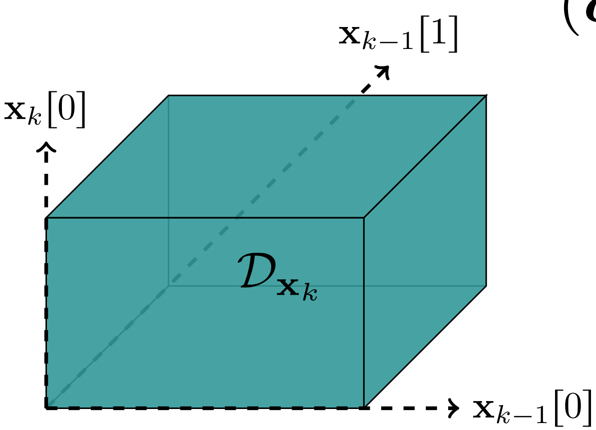
An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

2. *sparsity*

Selection criterion for active set $\mathcal{B}$?

$$\max_{(\boldsymbol{\alpha},\boldsymbol{\beta}_{\mathcal{B}})\geqslant 0}\left\{\min_{(\mathbf{x},\mathbf{z})\in\mathcal{D}} \quad \mathcal{L}_{\mathcal{B}}(\mathbf{x},\mathbf{z},\boldsymbol{\alpha},\boxed{\boldsymbol{\beta}_{\mathcal{B}}})\right\}$$



An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

3. tightness

An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

3. tightness

- Start from Planet dual;

An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

3. tightness

- Start from Planet dual;
- Take a fixed number of supergradient steps;

An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

3. tightness

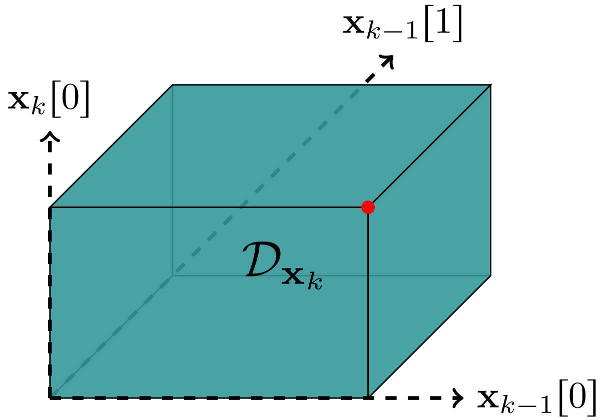
- Start from Planet dual;
- Take a fixed number of supergradient steps;
- Add most violated constraint at primal minimiser.

An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

3. tightness

- Start from Planet dual;
- Take a fixed number of supergradient steps;
- Add most violated constraint at primal minimiser.

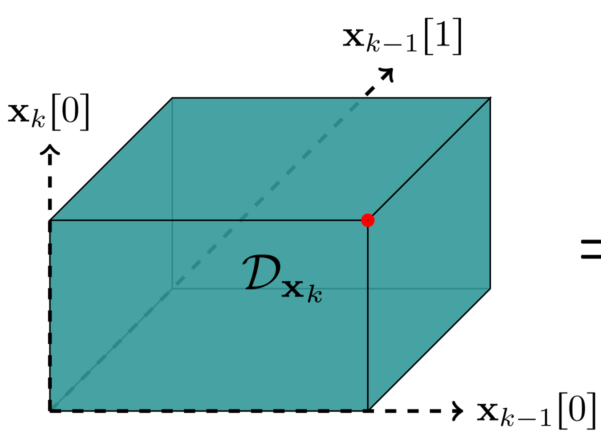


An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

3. <u>tightness</u>

- Start from Planet dual;
- Take a fixed number of supergradient steps;
- Add most violated constraint at primal minimiser.



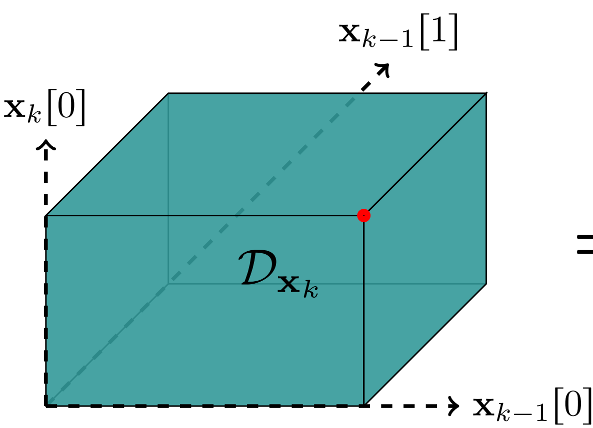
$\Longrightarrow$

An Efficient Customised Solver

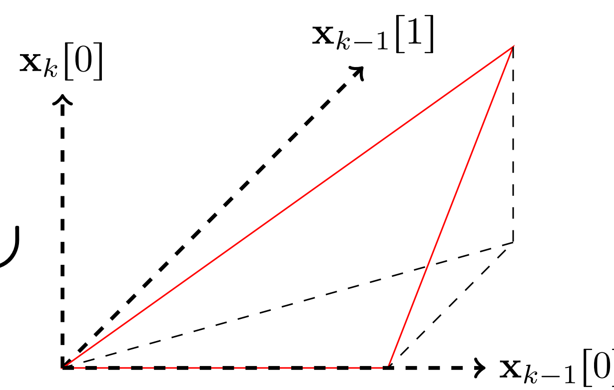
An efficient solver for the tight relaxation needs:

3. tightness

- Start from Planet dual;
- Take a fixed number of supergradient steps;
- Add most violated constraint at primal minimiser.



$$\Longrightarrow \qquad \mathcal{B} \leftarrow \mathcal{B} \cup$$



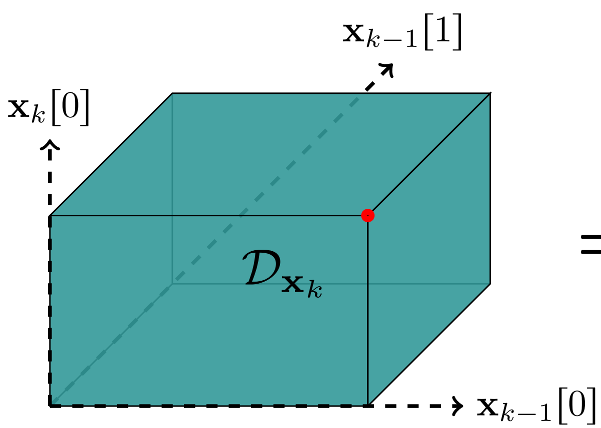
An Efficient Customised Solver

An efficient solver for the tight relaxation needs:

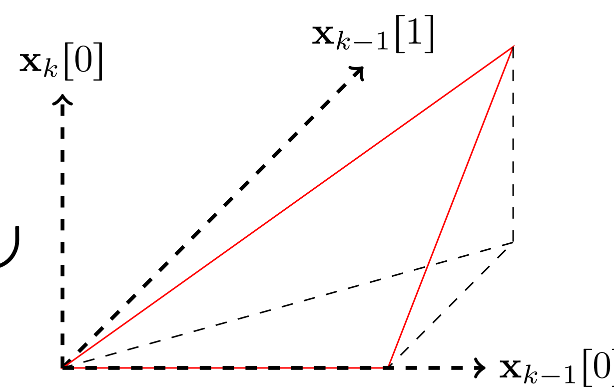
Active Set

3. tightness

- Start from Planet dual;
- Take a fixed number of supergradient steps;
- Add most violated constraint at primal minimiser.



$$\Longrightarrow \quad \mathcal{B} \leftarrow \mathcal{B} \cup$$



Experiments: Baselines

- **BDD+**: Efficient dual solver for Planet relaxation from previous work.

[Bunel et al., 2020, Anderson et al., 2020]

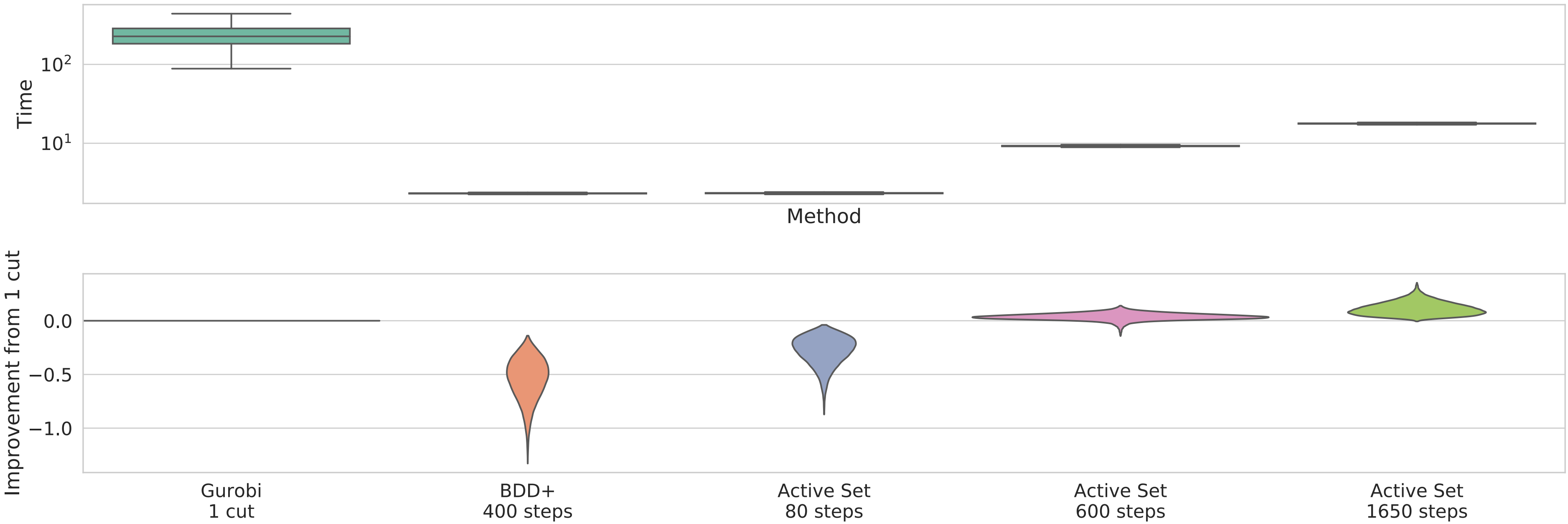
Experiments: Baselines

- **BDD+**: Efficient dual solver for Planet relaxation from previous work.
- **Gurobi**: Primal cutting plane algorithm with one cut.

[Bunel et al., 2020, Anderson et al., 2020]

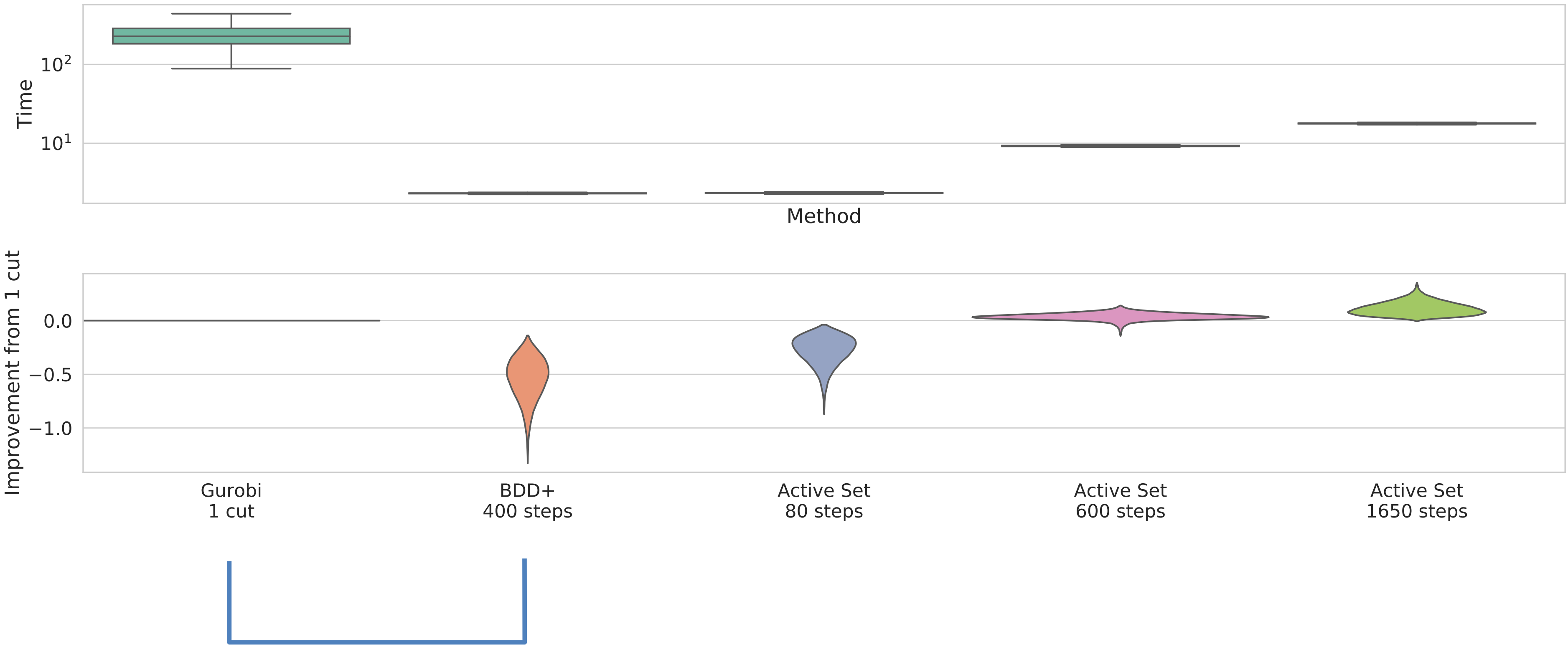
Incomplete Verification

Robustness margin for CIFAR-10, SGD-trained convolutional network with ~6k neurons.



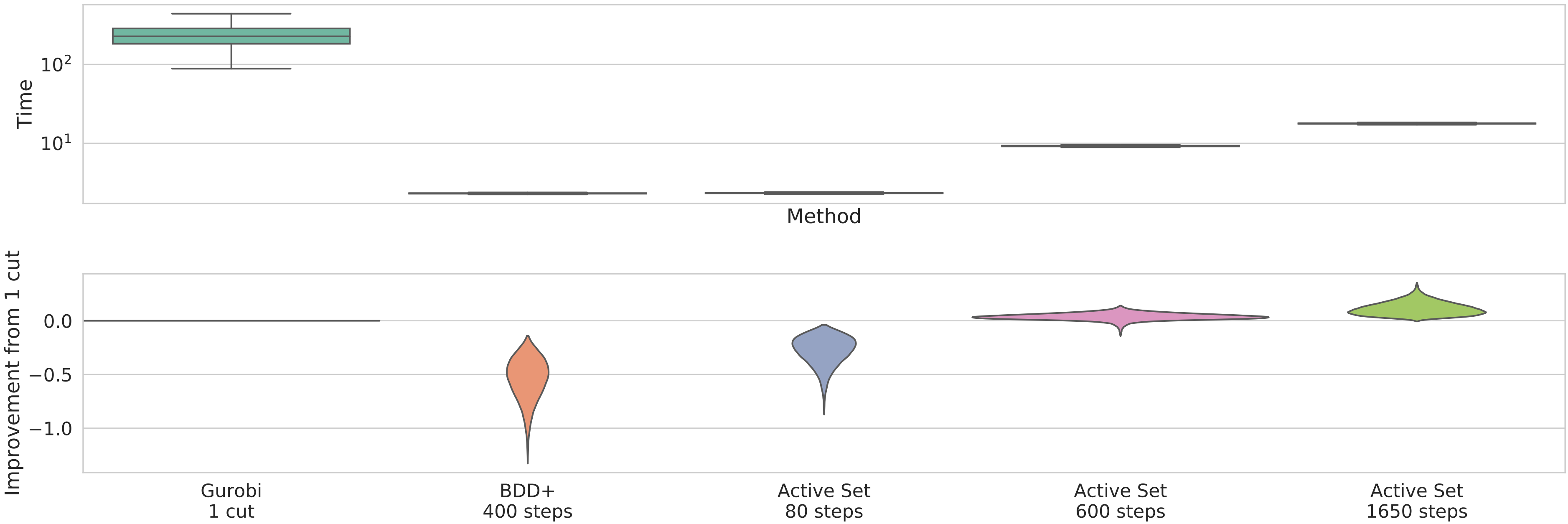
Incomplete Verification

Robustness margin for CIFAR-10, SGD-trained convolutional network with ~6k neurons.



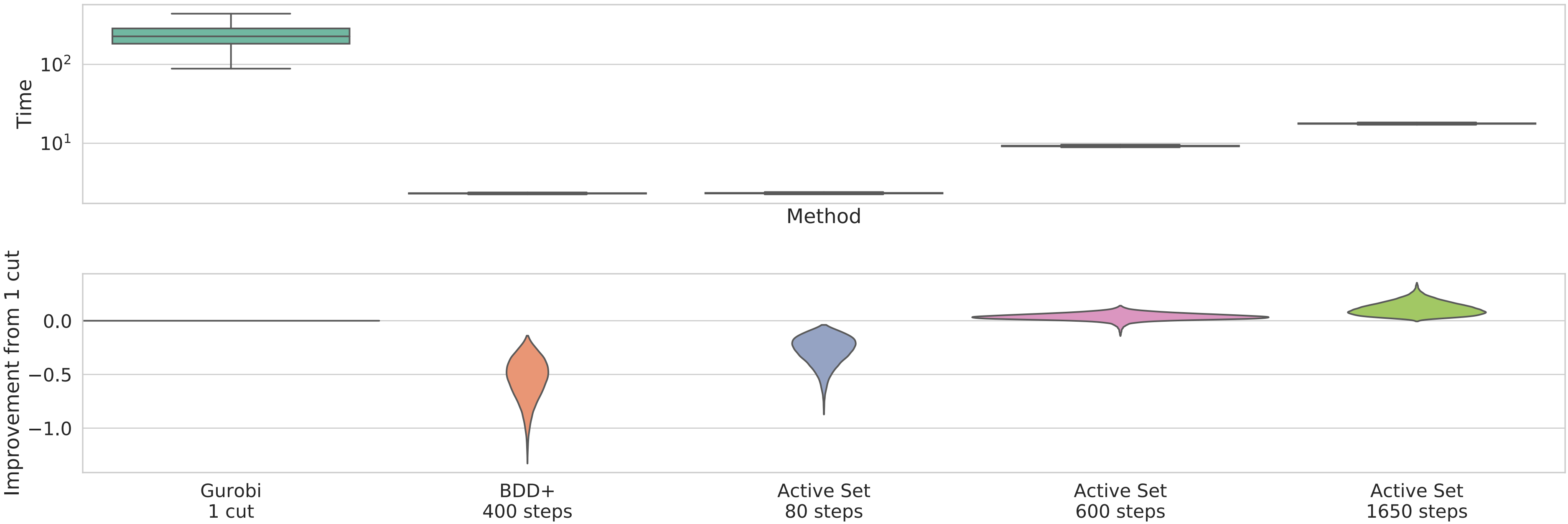
Incomplete Verification

Robustness margin for CIFAR-10, SGD-trained convolutional network with ~6k neurons.



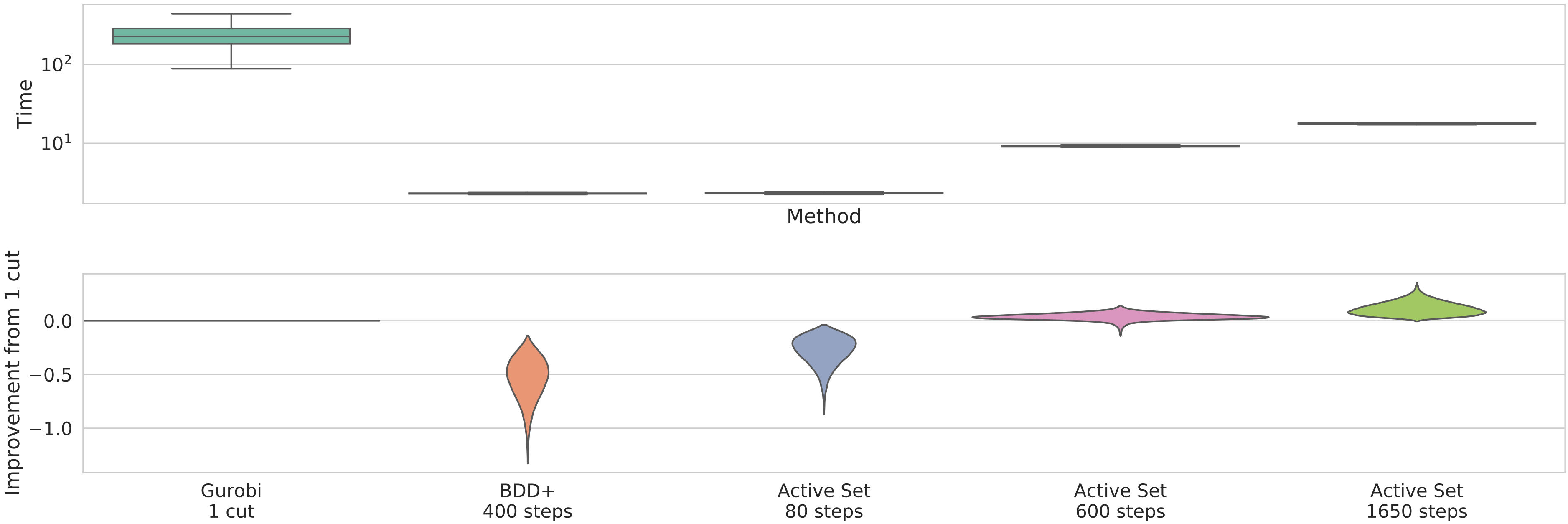
Incomplete Verification

Robustness margin for CIFAR-10, SGD-trained convolutional network with ~6k neurons.



Incomplete Verification

Robustness margin for CIFAR-10, SGD-trained convolutional network with ~6k neurons.



Conclusions

Customised dual solver for tight ReLU relaxation: *a new convex barrier*?

Conclusions

Customised dual solver for tight ReLU relaxation: *a new convex barrier*?

Better bounds in the same time compared to off-the-shelf solvers and previous dual algorithms.

Conclusions

Customised dual solver for tight ReLU relaxation: *a new convex barrier*?

Better bounds in the same time compared to off-the-shelf solvers and previous dual algorithms.

Follow-up work: more memory efficient solver.

[arXiv: Scaling the Convex Barrier with Sparse Dual Algorithms]