# Polyhedral Complex Extraction from ReLU Networks using Edge Subdivision

**Arturs Berzins**
Department of Mathematics and Cybernetics
SINTEF

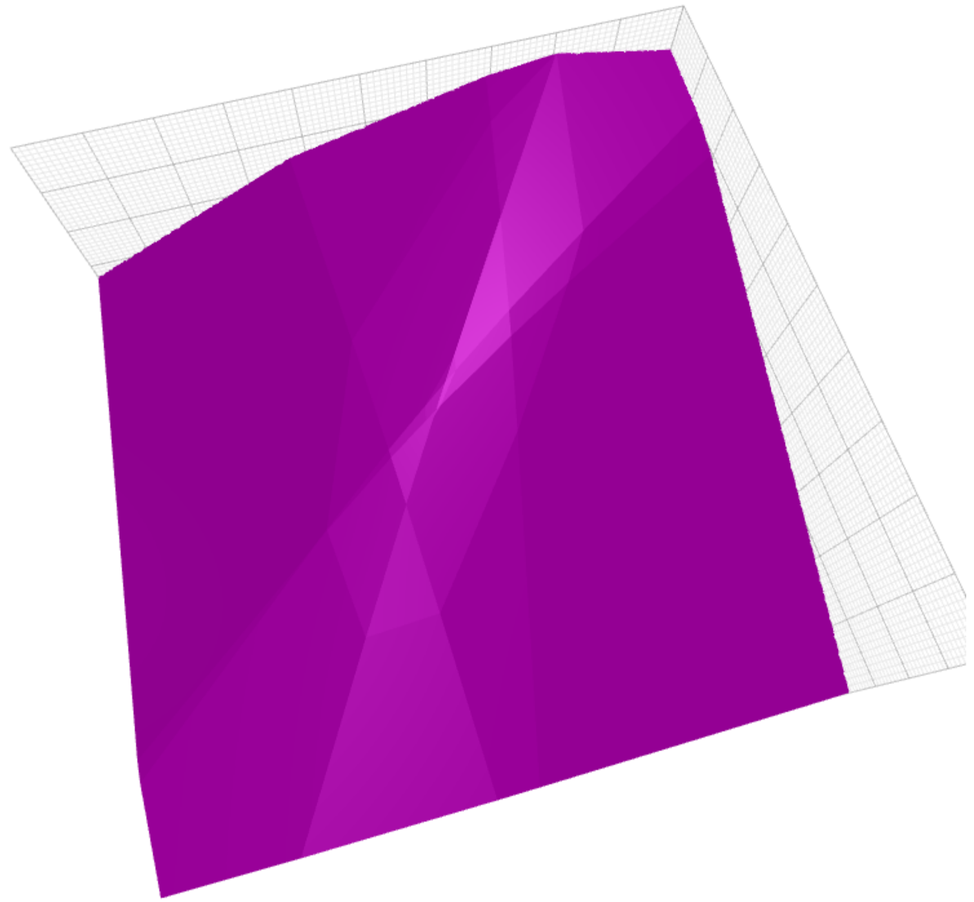
ICML 2023

This project was supported by the EU's Horizon 2020 Research and Innovation Programme under Grant Agreement number 86084 and an extraordinary basic funding grant by the Research Council of Norway.

ReLU NNs are **Continuous Piece-Wise Affine**
(abs, hard hyperbolic tangent, hard sigmoid, max-pooling)

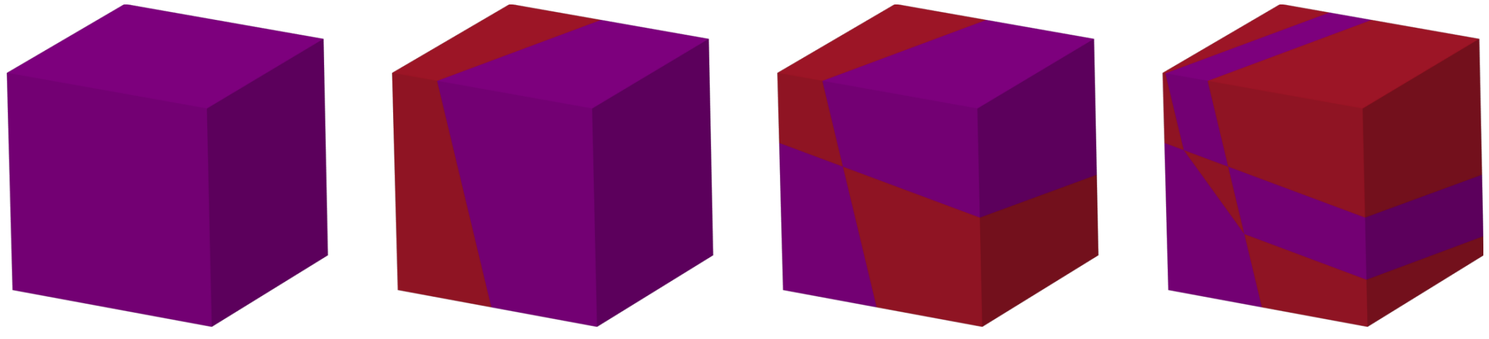
CPWA NNs induce a **polyhedral complex**

How to **extract** this complex?



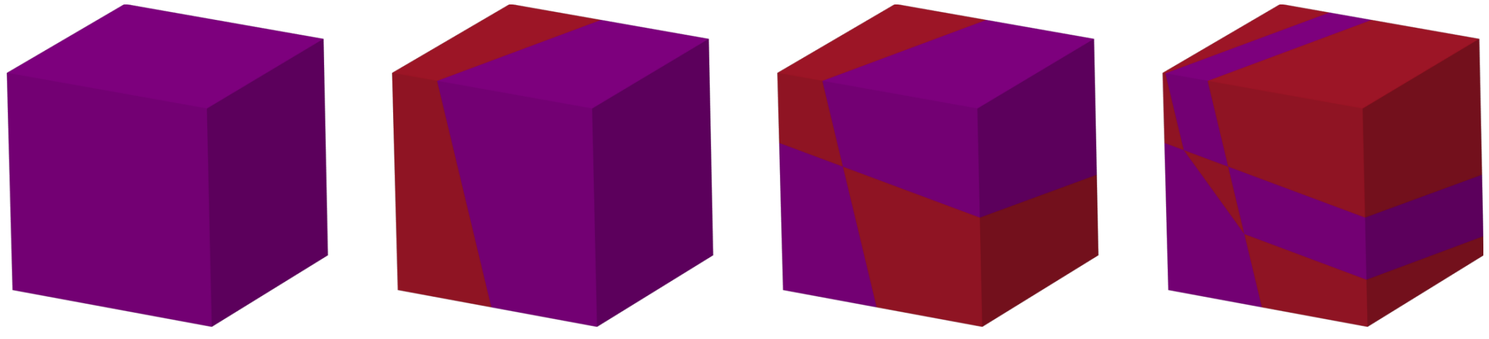
# Region subdivision

# Region subdivision

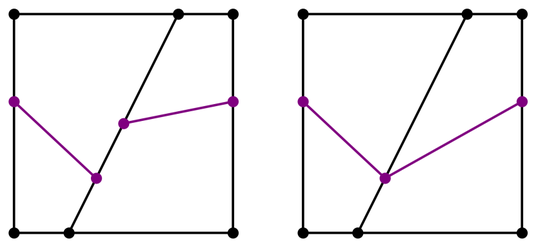


cut each **region** with its affine hyperplane

# Region subdivision

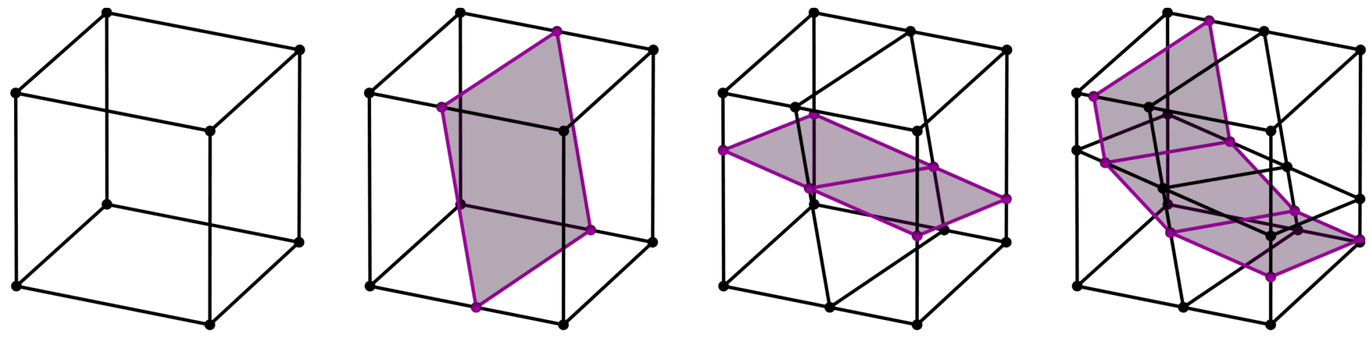


cut each **region** with its affine hyperplane



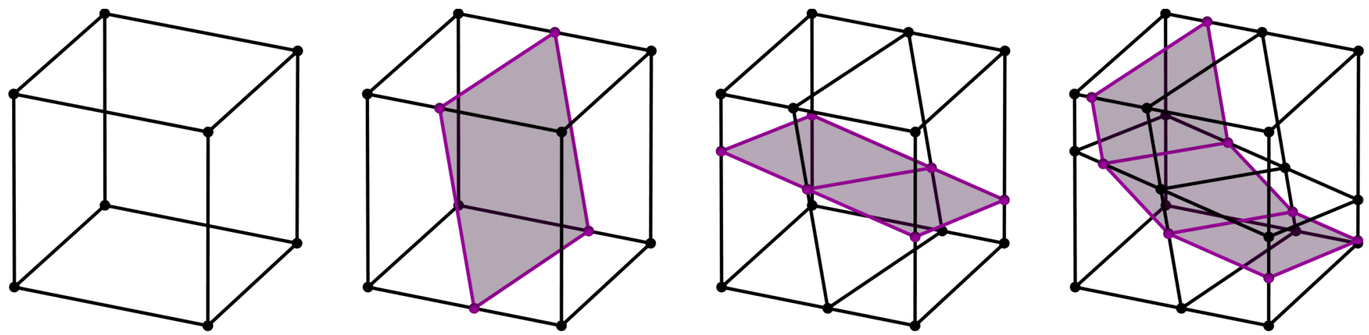
$2^{D-1}$ fold **redundancy** due to continuity

# Edge subdivision

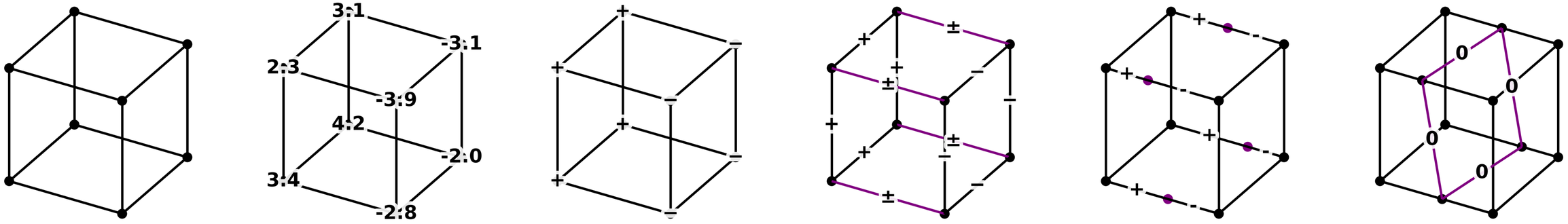


cut **edges** with folded hyperplanes

# Edge subdivision



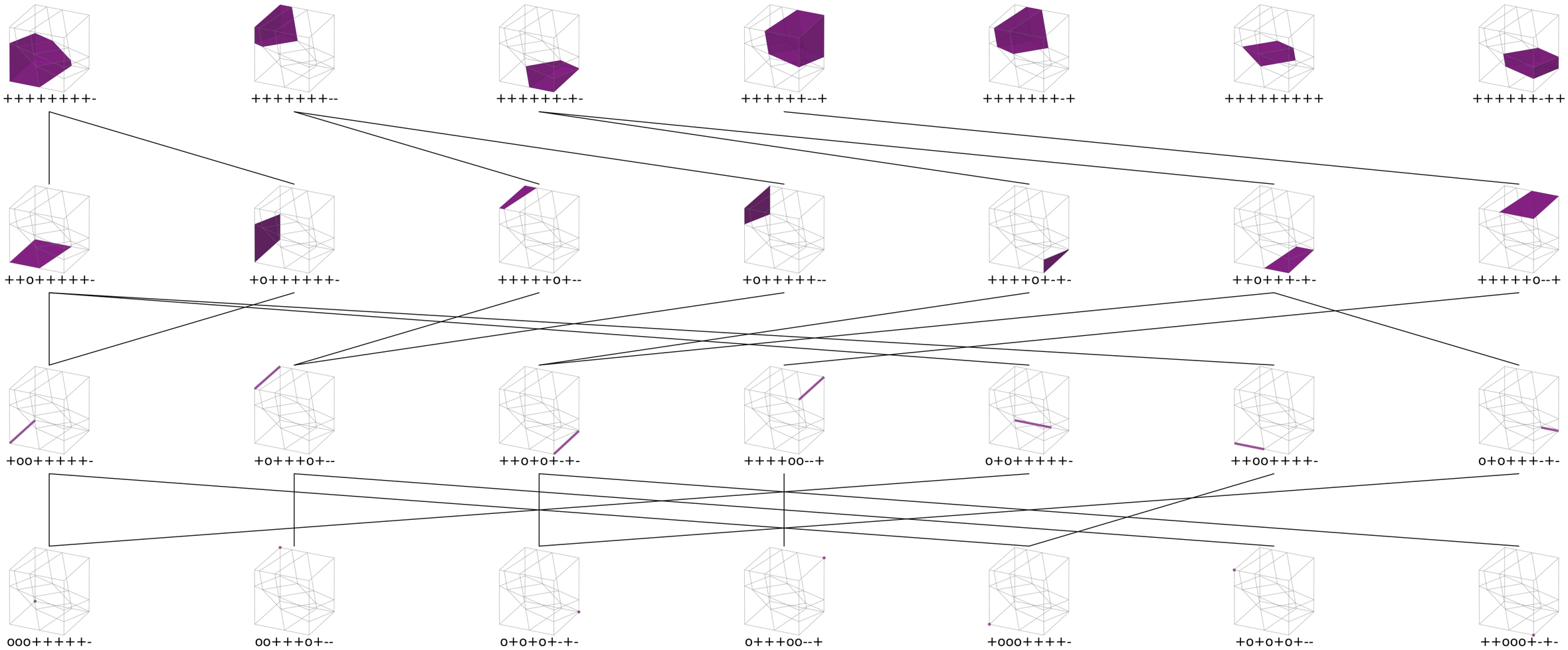
cut **edges** with folded hyperplanes




(0) (1) (2) (3) (4) (5)

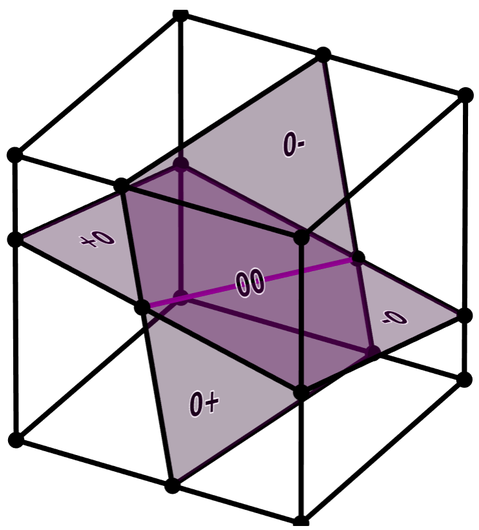
.. in five steps

# Sign-vectors



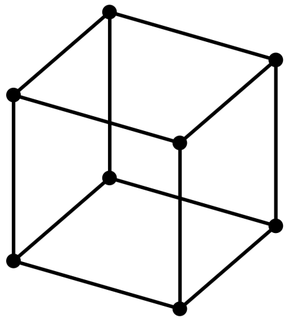

sign-vectors indicate the pre-activation signs of the neurons
they encode the combinatorial structure of the complex

# Perturbation

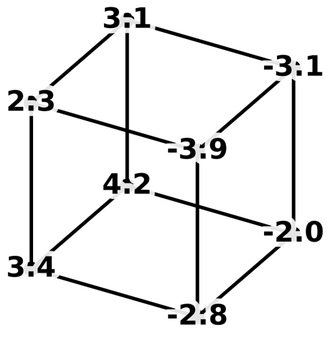



parents of a cell can be built by perturbing the zeros

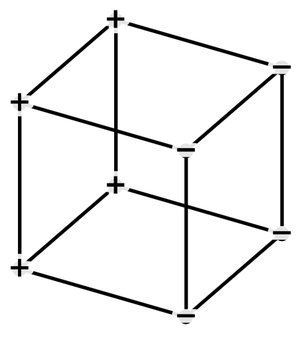
# Edge subdivision



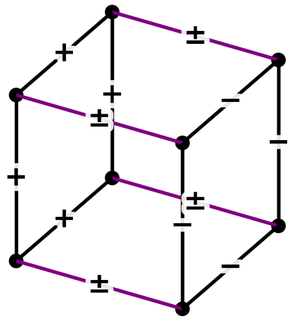
(0)



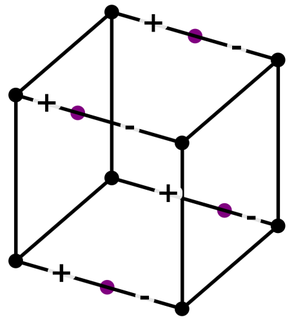

(1)



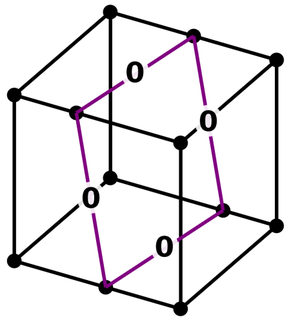
(2)



(3)

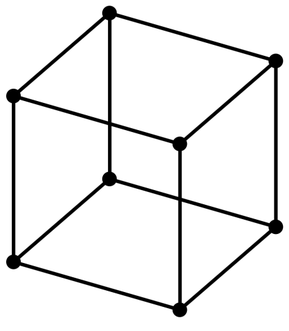


(4)

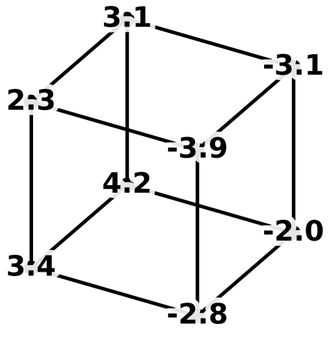



(5)

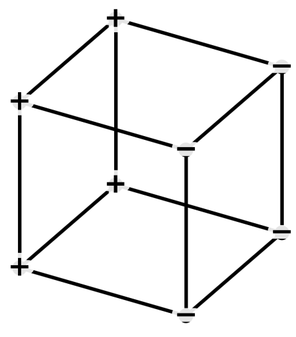
# Edge subdivision



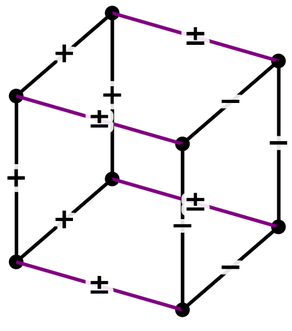
(0)



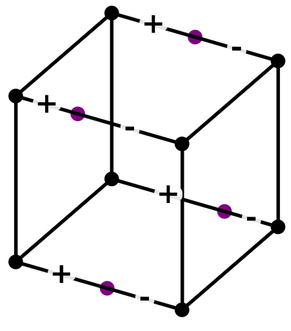

(1)



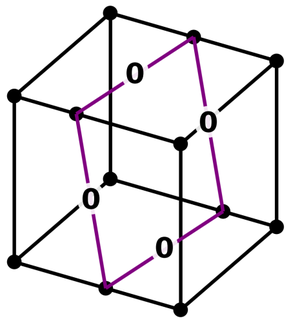
(2)



(3)



(4)




(5)

all steps scale linearly with complex size

# Implementation

PyTorch

github.com/arturs-berzins/relu_edge_subdivision

steps 1-4 are linear
step 5 is log-linear due to sorting

# Implementation

PyTorch

github.com/arturs-berzins/relu_edge_subdivision

steps 1-4 are linear

step 5 is log-linear due to sorting

can be improved to linear with hash-tables

# Implementation

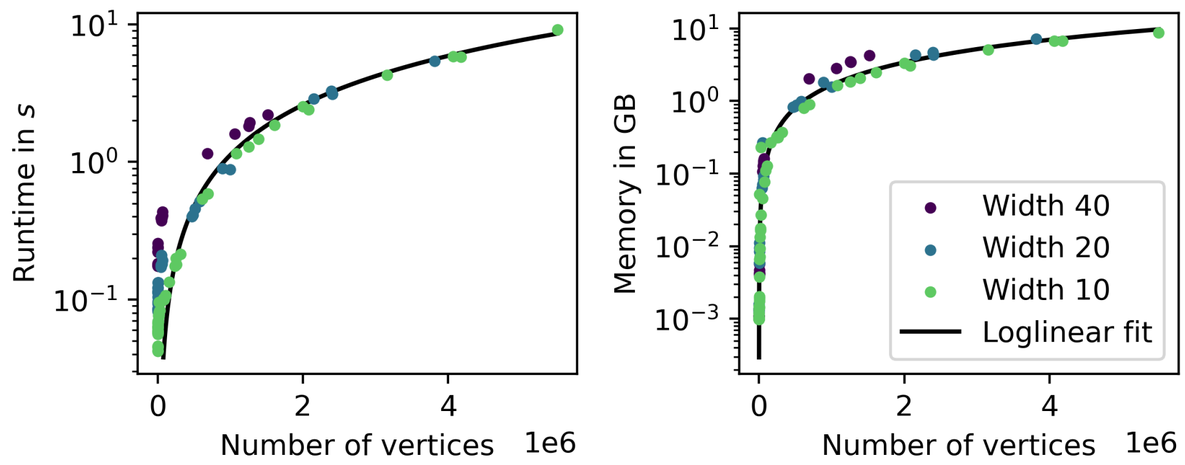
PyTorch

github.com/arturs-berzins/relu_edge_subdivision

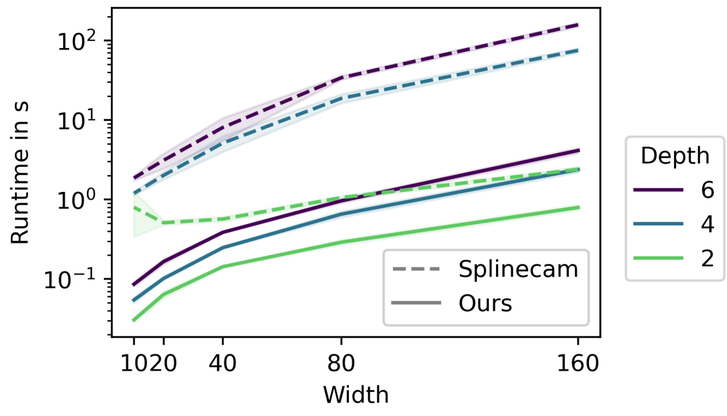
steps 1-4 are linear

step 5 is log-linear due to sorting

can be improved to linear with hash-tables

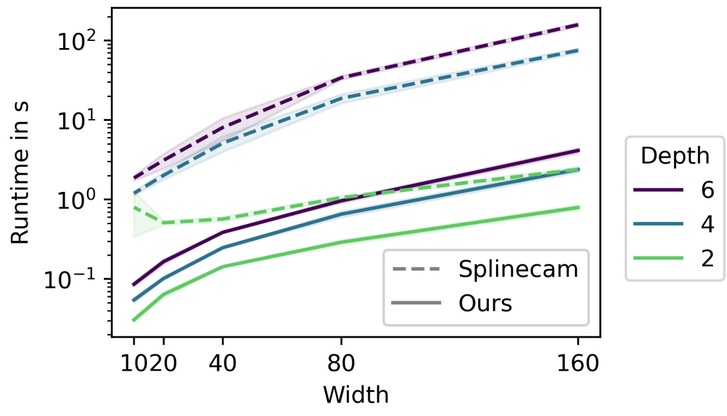


# Benchmark

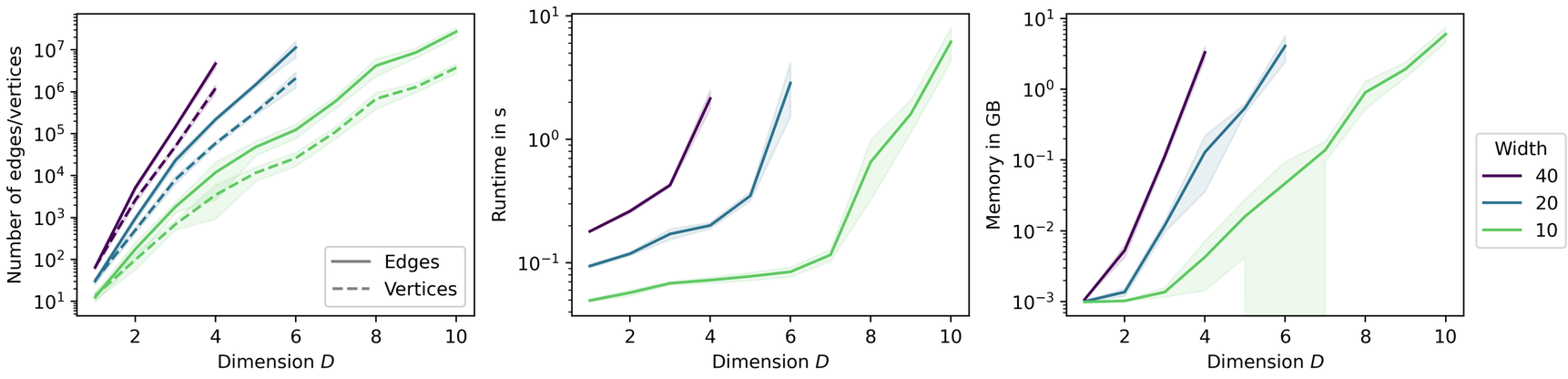



Humayun, A. I., Balestriero, R., Balakrishnan, G., and Baraniuk, R.
Splinecam: Exact visualization and characterization of deep network geometry and decision boundaries.
CVPR2023.

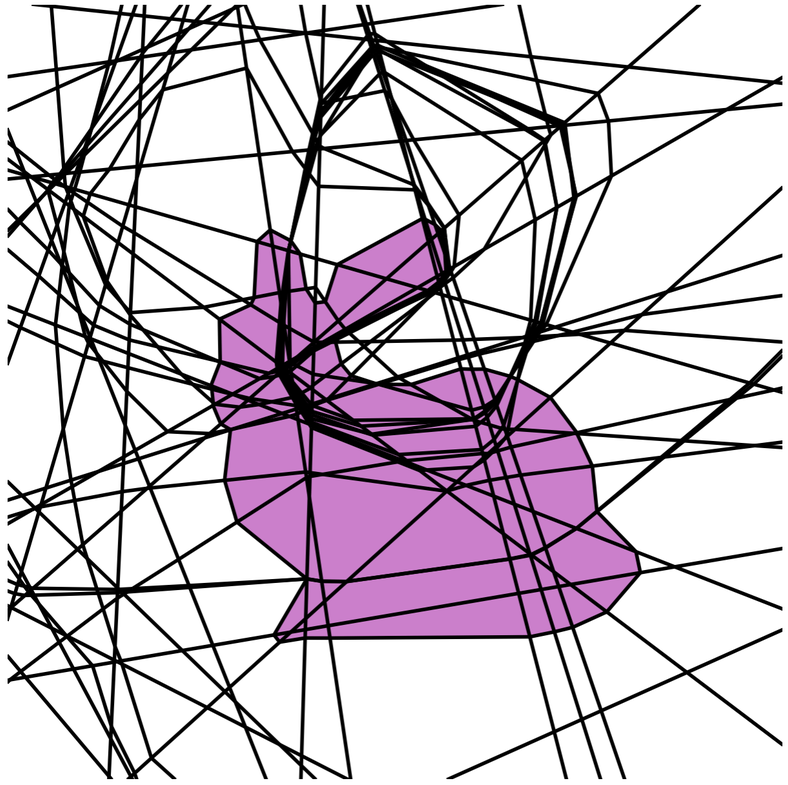
# Benchmark




Humayun, A. I., Balestriero, R., Balakrishnan, G., and Baraniuk, R.
Splinecam: Exact visualization and characterization of deep network geometry and decision boundaries.
CVPR2023.



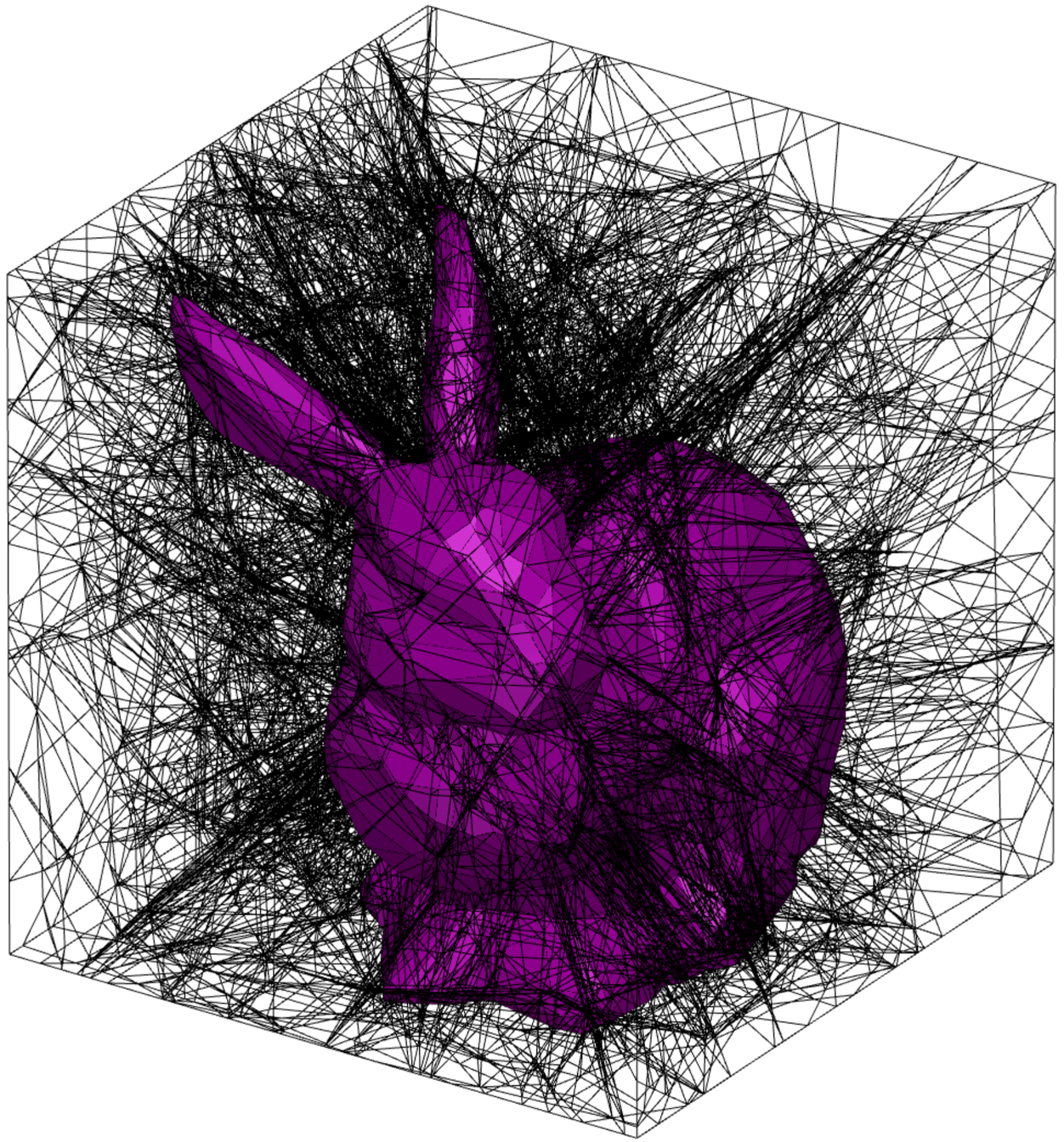
# Optimizing the complex



shape compactness

in general, any vertex position based objective

(edge lengths, angles, areas, volumes, curvatures)



github.com/arturs-berzins/relu_edge_subdivision