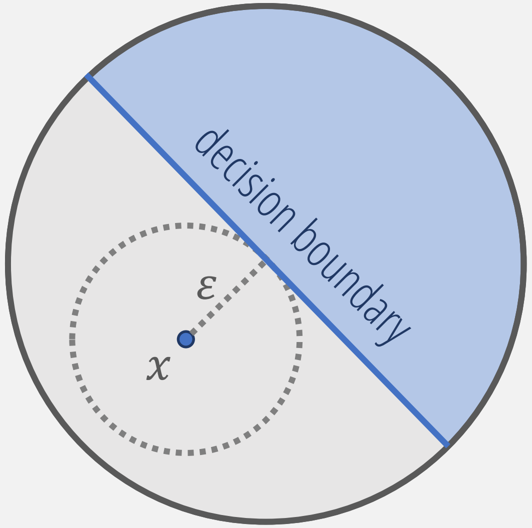
Fast Geometric Projections for Local Robustness Certification

Aymeric Fromherz*, **Klas Leino***, Matt Fredrikson, Bryan Parno, Corina Păsăreanu

Goal: Local Robustness

- A model $\boldsymbol{F}$ satisfies ***local robustness*** with robustness radius $\boldsymbol{\varepsilon}$ on a point $\boldsymbol{x}$ if

$$\forall x'. \|x - x'\|_p \leq \varepsilon \implies F(x) = F(x')$$

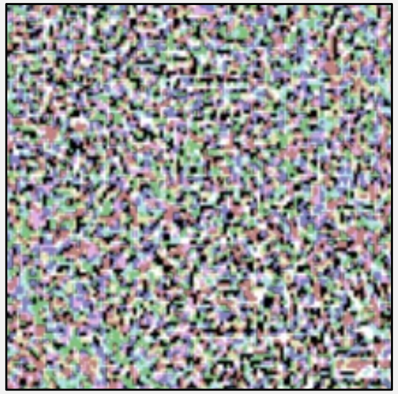


- Valid for any norm, but we focus on the ℓ_2 norm, which is less well-studied

Adversarial Examples

"panda"

+ $0.007\times$



adversarial perturbation

=

"gibbon"

Goodfellow et al. 2015

Defenses

Heuristic

- Adversarial training
- TRADES

Madry et al. 2018
Zhang et al. 2019

Certification

training procedure

- Kolter-Wong
- MMR

model-agnostic verification

- GeoCert
- MIP
- …

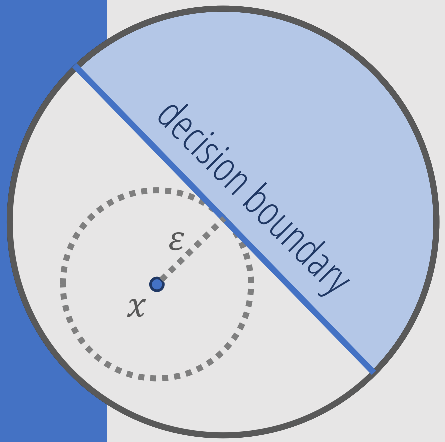
Wong & Kolter, 2018
Croce et al. 2019
Jordan et al. 2019
Tjeng & Tedrake, 2017

Probabilistic

- Randomized Smoothing

Cohen et al. 2019

How can We Certify Local Robustness?



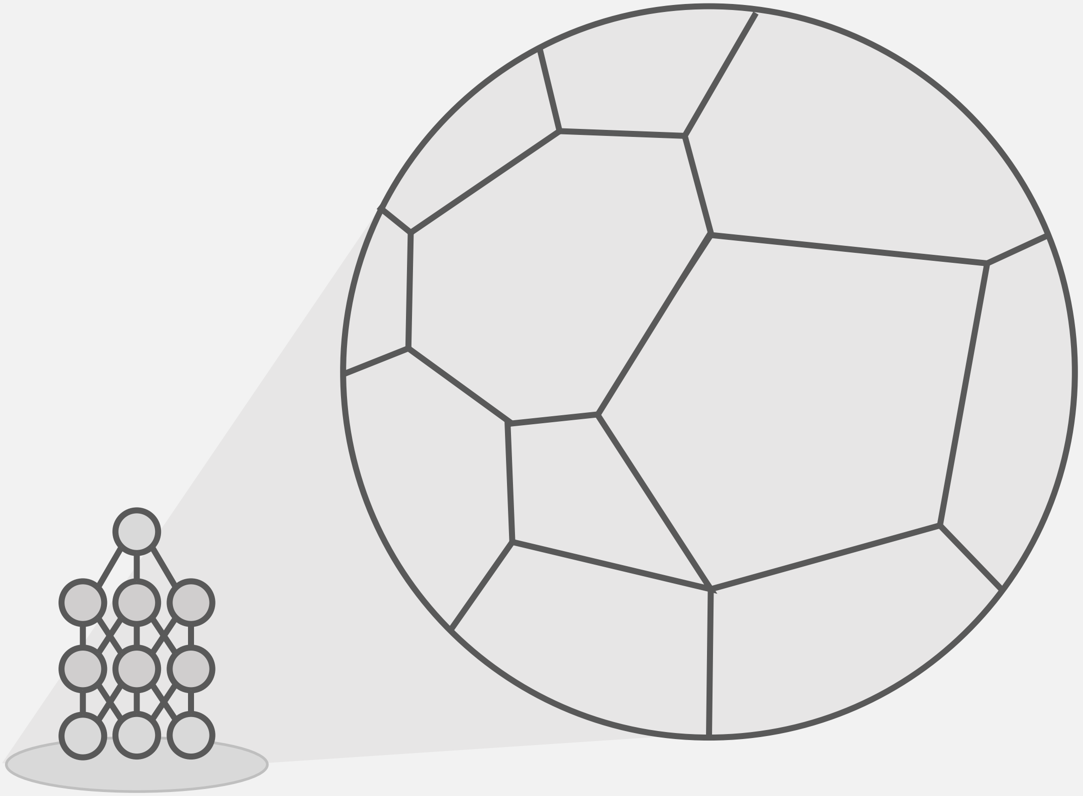
$$\forall x'.\ \|x - x'\|_p \le \varepsilon \implies F(x) = F(x')$$

Treating a NN as general function is too abstract

Idea: use a more refined understanding of the *geometry* of a class of networks

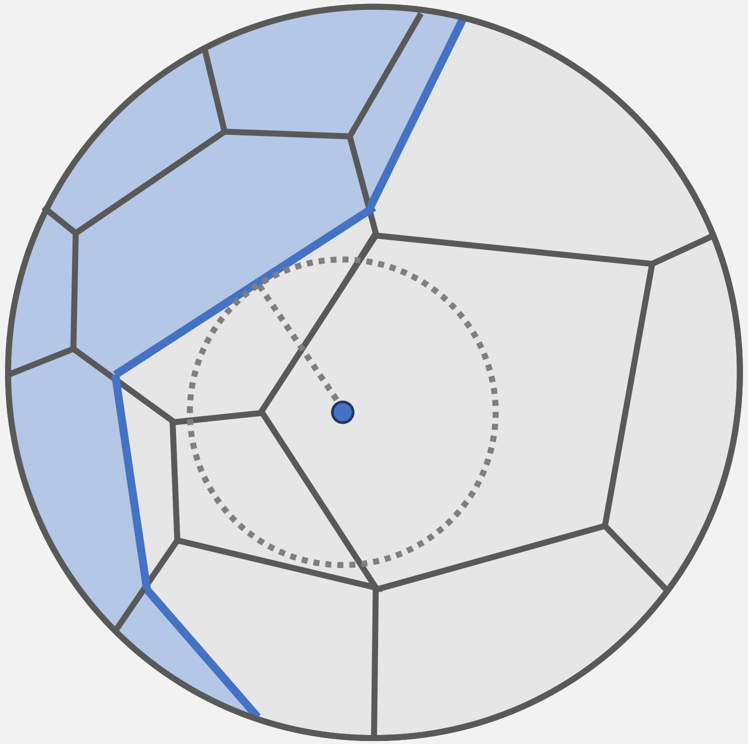
ReLU Networks as a Polyhedral Complex

- ReLU networks are ***piecewise-linear***
- Piecewise components partition input into a ***polyhedral complex***
- Regions correspond to ***activation patterns***
- Boundaries to regions can be computed using gradients



Constraint-Solving for Local Robustness Certification

- Each region may contain a decision boundary
- Given a point, can use constraint-solving to find distance to nearest boundary (e.g., GeoCert, MIP)
- This is expensive and doesn't scale

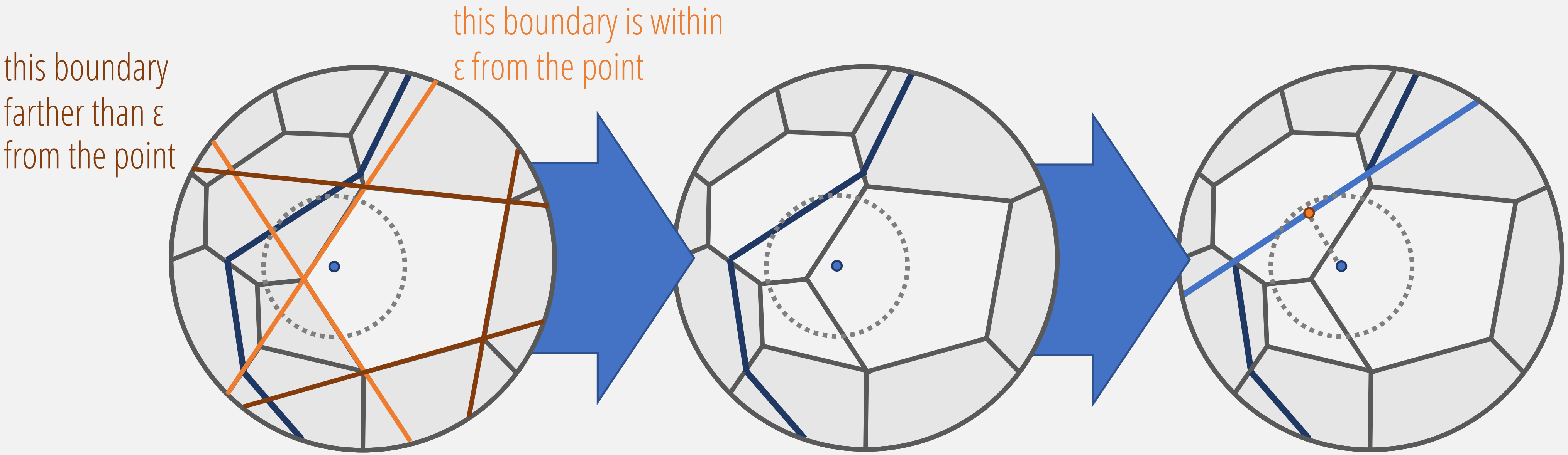


E.g., on a dense network with 120 neurons, the median certification time of these methods is **over one minute per instance**

Our contribution: algorithm that restricts analysis to **only fast primitives** that can be accelerated on GPUs

Fast Geometric Projections (FGP) Algorithm

Projections offer a fast, sound way to see which boundaries are within our ε-radius



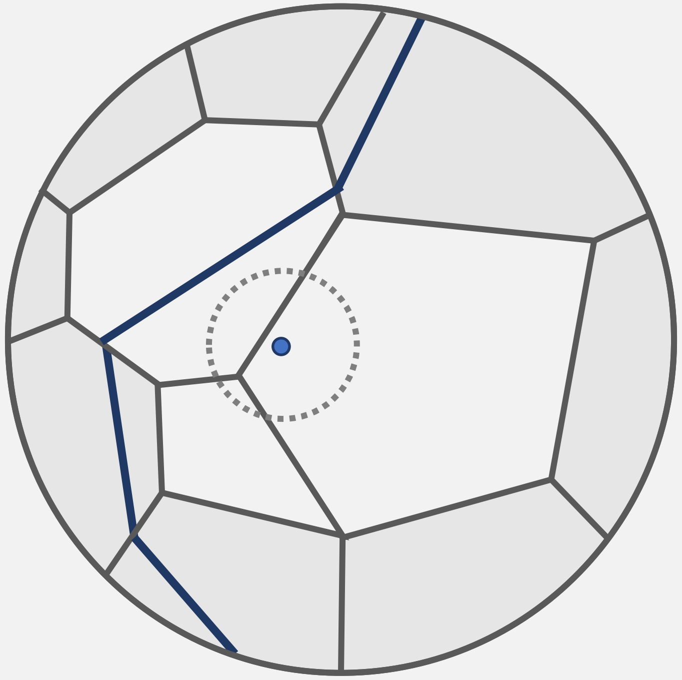
begin by *exploring* the starting region: for each boundary of starting region, check if the boundary is in the ε-ball

explore each of the neighboring regions whose boundaries were in the ε-ball

if a decision boundary is found, project onto it to verify an adversarial example was found

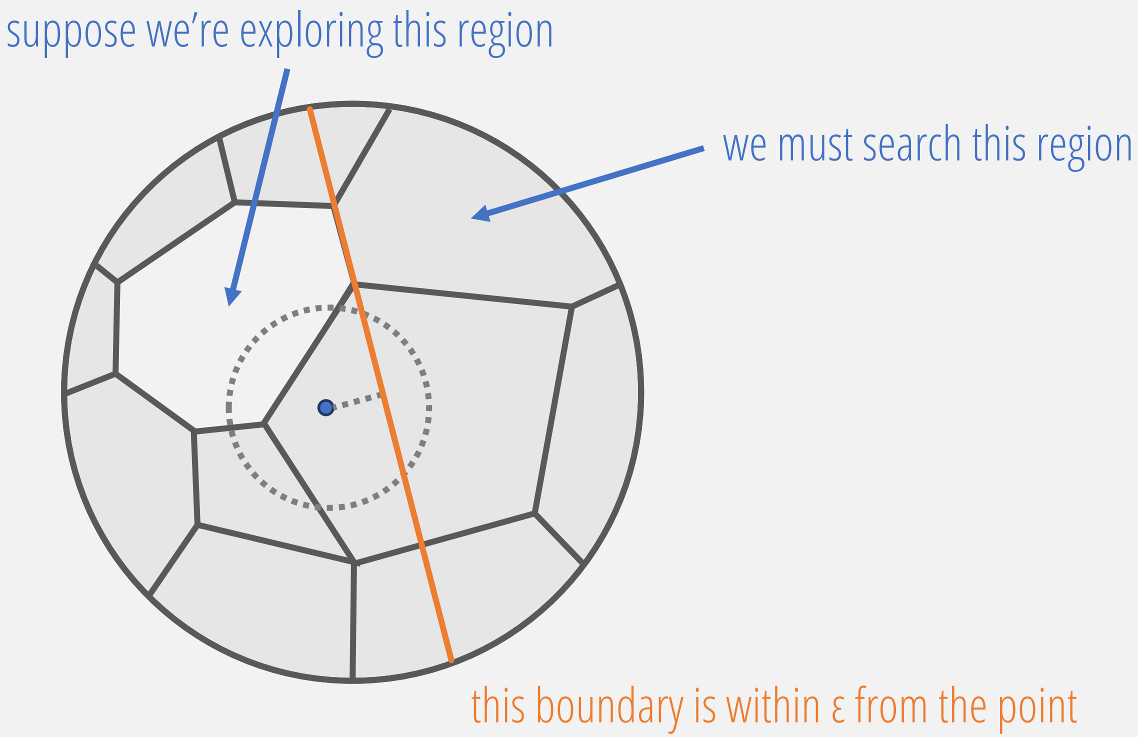
Fast Geometric Projections (FGP) Algorithm

If we run out of regions to explore and haven't encountered a decision boundary, we certify the point as ε-robust



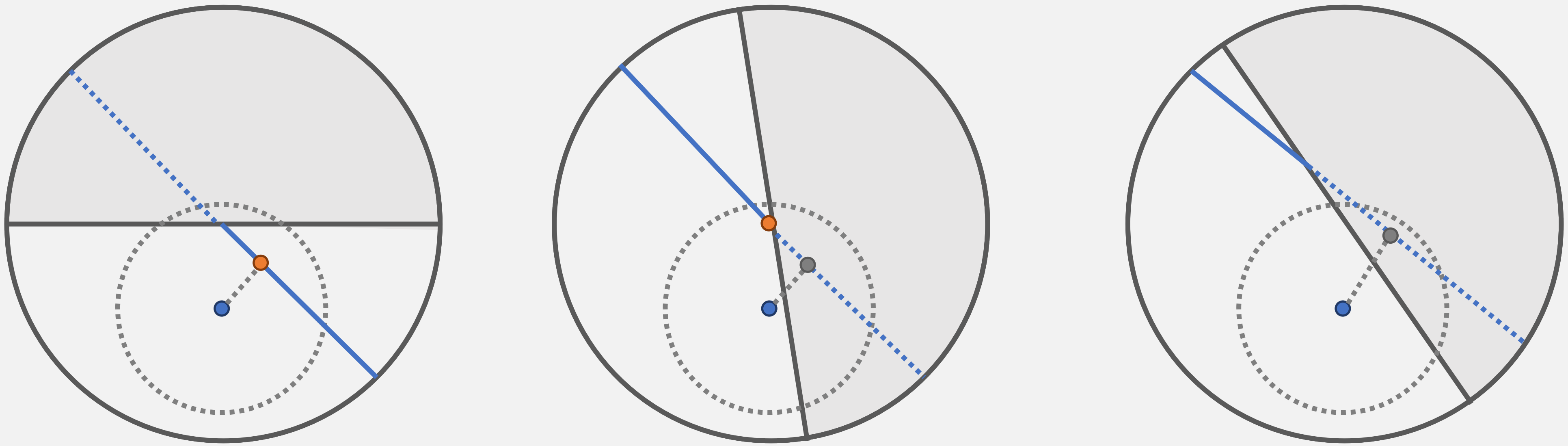
Region Exploration is an Overapproximation

- We compute a ***lower bound*** on distance from point to boundary (since we ignore that constraints are only valid on finite intervals)
- Thus we explore all regions that ***might*** be in the epsilon ball



Certification Edge Cases

FGP is ***sound*** but not ***complete***



projection onto decision boundary is in region, adversarial example exists

projection onto decision boundary is not in region, but adversarial example exists

projection onto decision boundary is not in region, no adversarial example exists

return NOT_ROBUST

can't distinguish these two cases

return UNKNOWN

Verification Results

On adversarially-trained dense networks, FGP outperforms GeoCert by **3 orders of magnitude** and MIP by **4 orders of magnitude**

UNKNOWN results account for **only 3-5% of cases**, while GeoCert and MIP time out (after 120s) on 10-100% of cases

Scalability

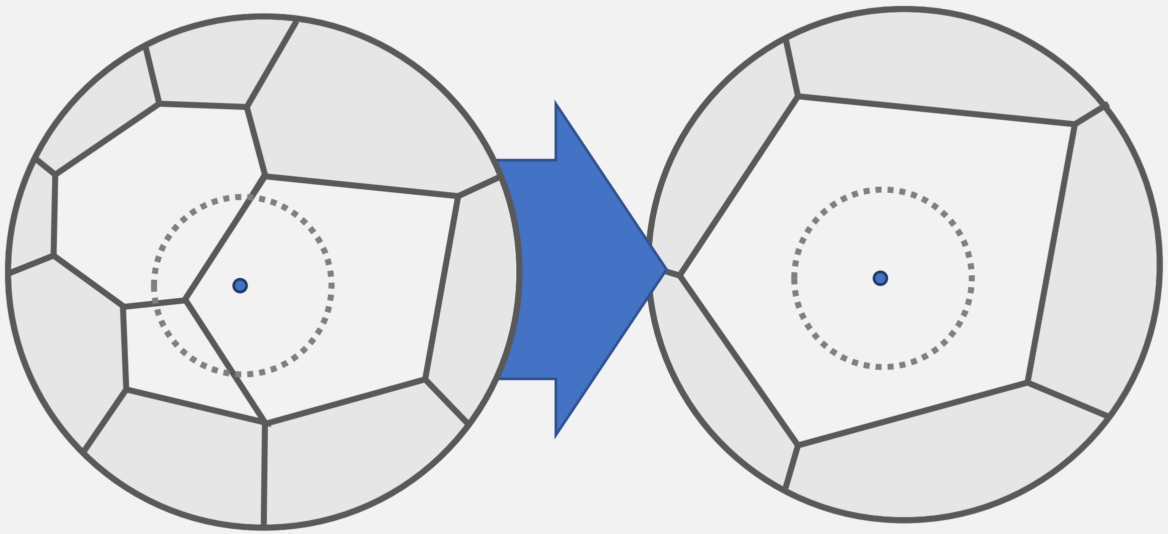
- Our time-per-region is about as small as it gets
  - Conservative search of regions outweighed by gain in speed compared to a more precise search
- Some networks will have too many regions to ever explore

A network with N neurons may have as many as 2^N possible regions. Exploring even a small fraction of the total regions would be impossible

Large networks will need to be regularized to have a smaller number of regions near points of interest

Training Networks for Verifiability

- Goal: Push region boundaries “further away” → fewer regions to explore
- We achieve better results when using regularization like Maximum Margin Regularization (MMR) and ReLU Stability (RS)



MMR: Croce et al. 2019
RS: Xiao et al. 2019

Verification on Larger Networks

	Time (s)	ROBUST	NOT ROBUST	UNKNOWN	TIMED OUT
MMR Dense Network 4 Layers	0.025	81%	14%	4%	1%
MMR Dense Network 20 Layers	0.057	86%	7%	7%	0%
ReLU Stability Convolutional Network 4 Layers	0.058	86%	14%	0%	0%

Conclusion

Looking Forward

Geometry provides a useful way of analyzing ReLU Networks

Focus on co-design between network training and verification for scaling certifiable robustness

Check Out Our Paper!

- Poster
- Paper on ArXiv
- Implementation on GitHub

full paper

https://tinyurl.com/fgp-iclr2021