Provable adversarial ℓ_2 robustness by propagating ellipsoids

Ezra Winston with Eric Wong

Summary

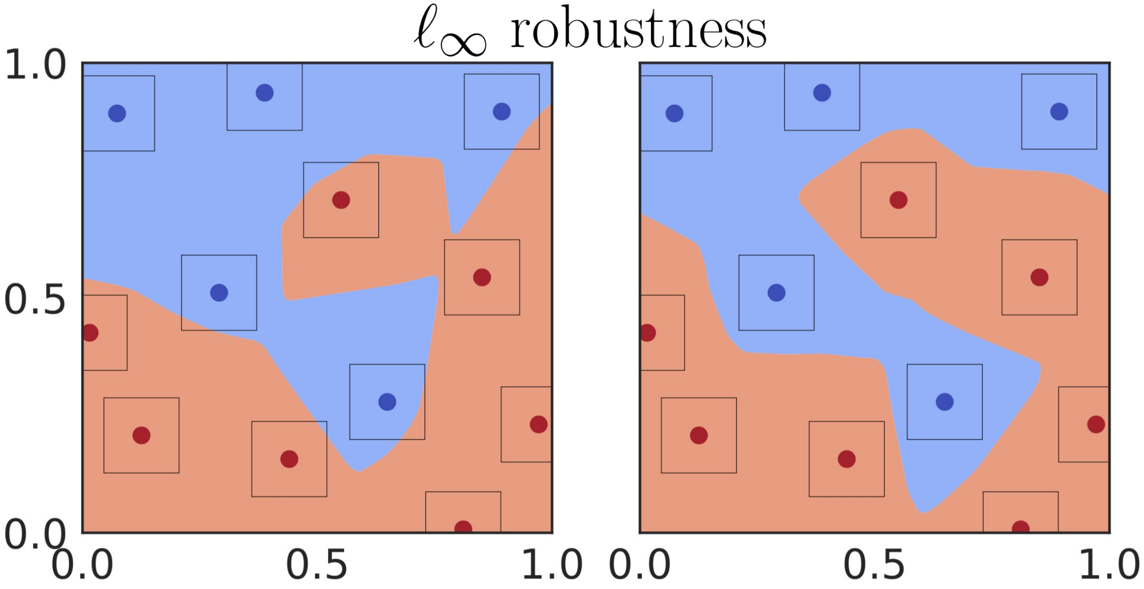
- Background: adversarial ℓ_2 robustness
- Ellipsoid Propagation
- Current Results
- Next Step: Efficient Computation?

Adversarial Robustness

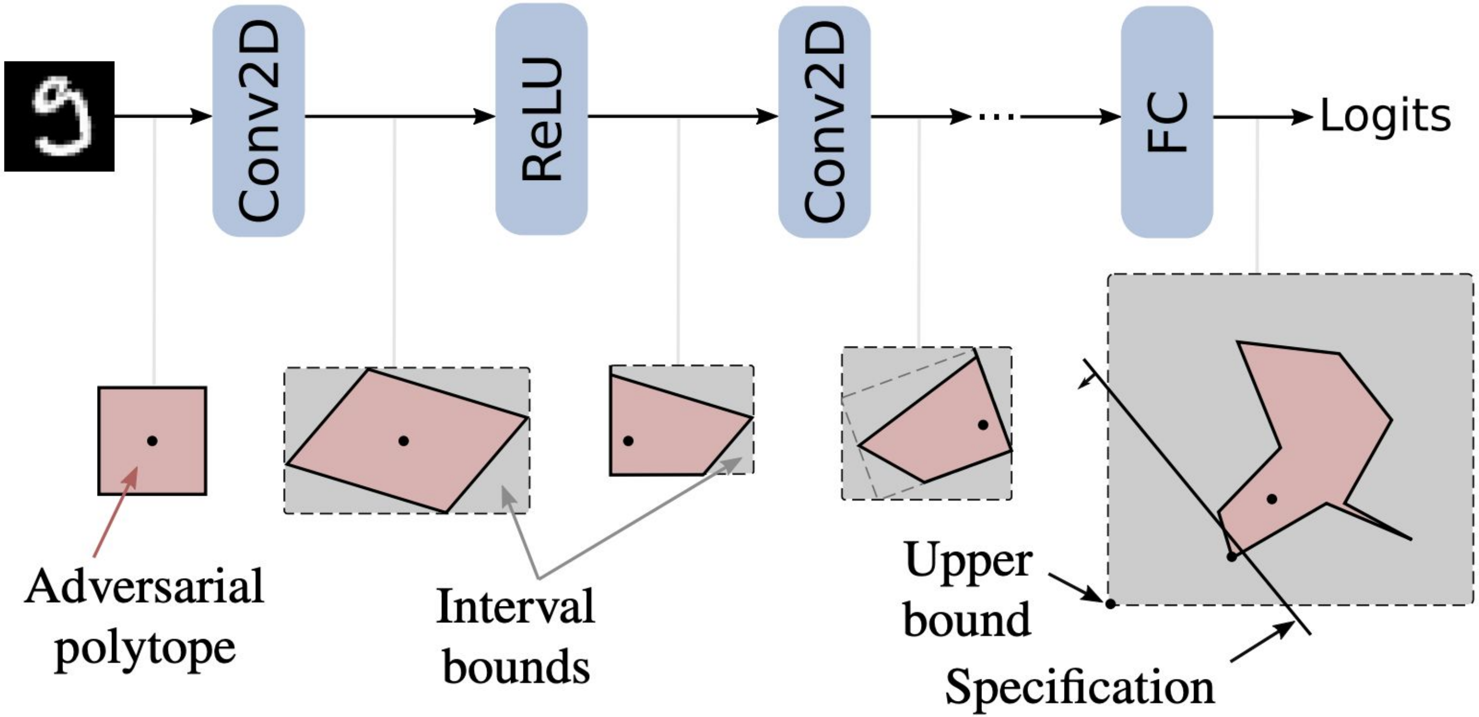
$$\min_{\theta} \sum_{i} \max_{\delta \in \Delta} \ell(f(x_i + \delta; \theta), y_i)$$

ℓ_p robustness:

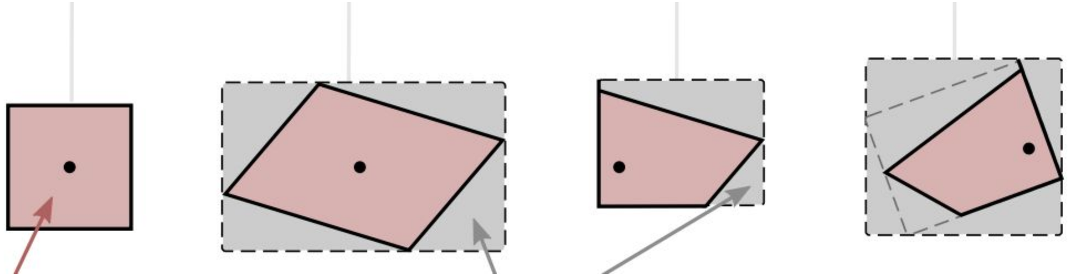
$$\Delta = \{\delta : \|\delta\|_p \leq \epsilon\}$$



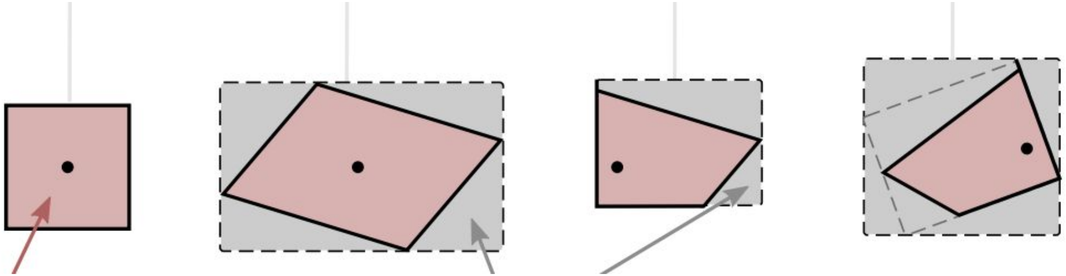
Interval bound propagation



Interval bound propagation

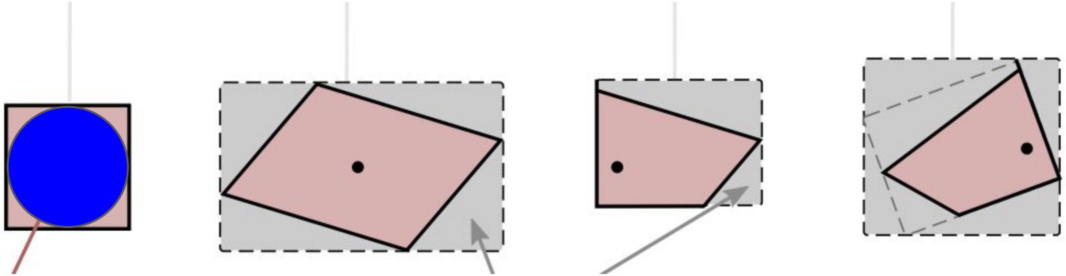


Interval bound propagation



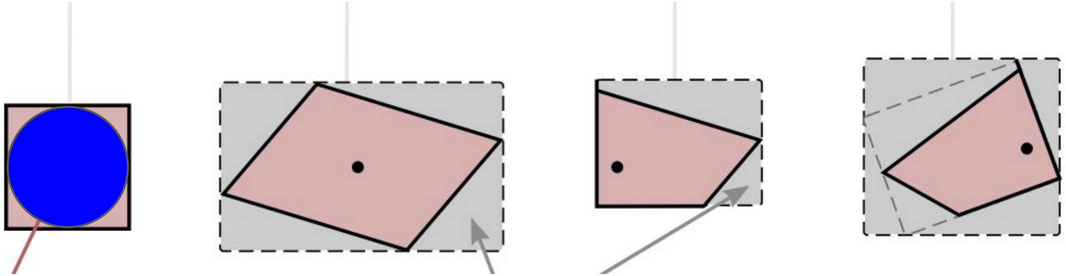
- Doesn’t maintain correlations

Interval bound propagation



- Doesn’t maintain correlations
- Obviously doesn’t work for ℓ_2 balls -- converts to ℓ_∞

Interval bound propagation



- Doesn't maintain correlations
- Obviously doesn't work for ℓ_2 balls -- converts to ℓ_∞
- Dual LP method of Wong et al. also relies on interval bounds on ReLU activations

ℓ_2 seems harder

ℓ_∞

Dataset	Epsilon	Robust Error	Standard Error
MNIST	0.1	3.67%	1.08%
CIFAR10	2/255	46.11%	31.28%

ℓ_2

Dataset	Epsilon	Robust Error	Standard Error
MNIST	1.58	55.47%	11.86%
CIFAR10	36/255	48.04%	38.80%

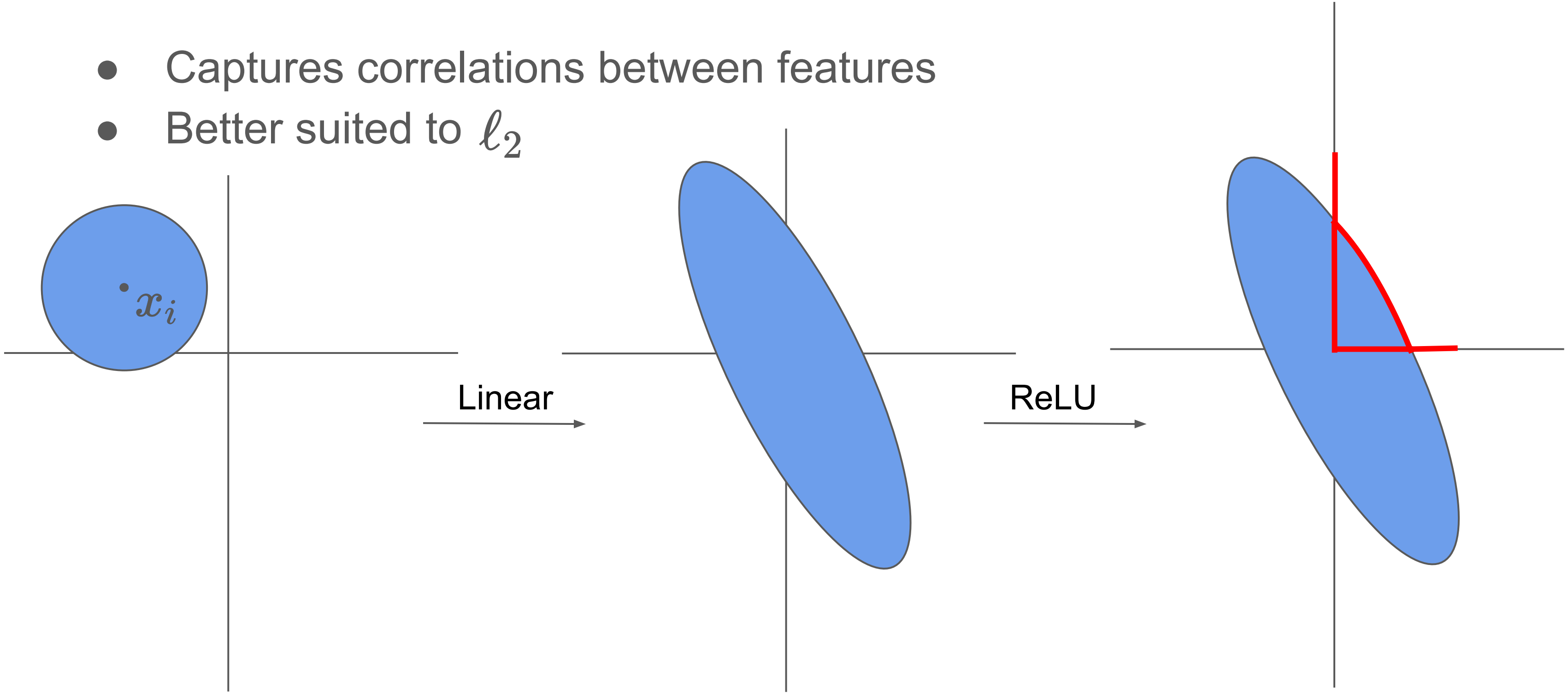
Summary

- Background: adversarial ℓ_2 robustness
- Ellipsoid Propagation
- Current Results
- Next Step: Efficient Computation?

Ellipsoid Propagation

Replace interval bounds with ellipsoids

- Captures correlations between features
- Better suited to ℓ_2

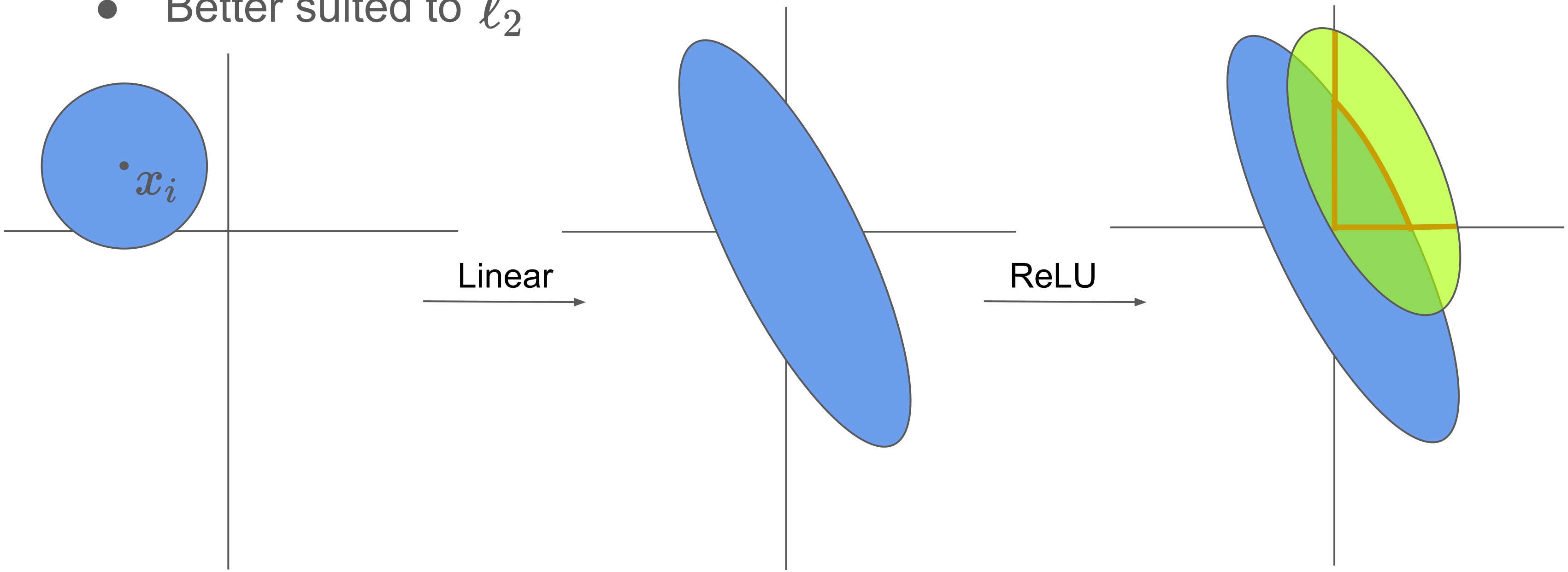


Ellipsoid Propagation

Replace interval bounds with ellipsoids

- Captures correlations between features
- Better suited to ℓ_2

Find new Minimum Volume Ellipsoid (MVE) containing output of ReLU



Halfspace projection

Problem: How to find Minimum Volume Ellipsoid (MVE) containing projection of ellipsoid after ReLU?

Instead: Use different nonlinearity consisting of projection onto a single halfspace.

- ReLU is projection onto each axis-aligned halfspace

$$\max(0, x) = \underset{z:z^T e_m \geq 0}{\mathrm{proj}} \left(\dots \underset{z:z^T e_1 \geq 0}{\mathrm{proj}} (x) \right)$$

- Reasonable performance on MNIST with much fewer than m projections

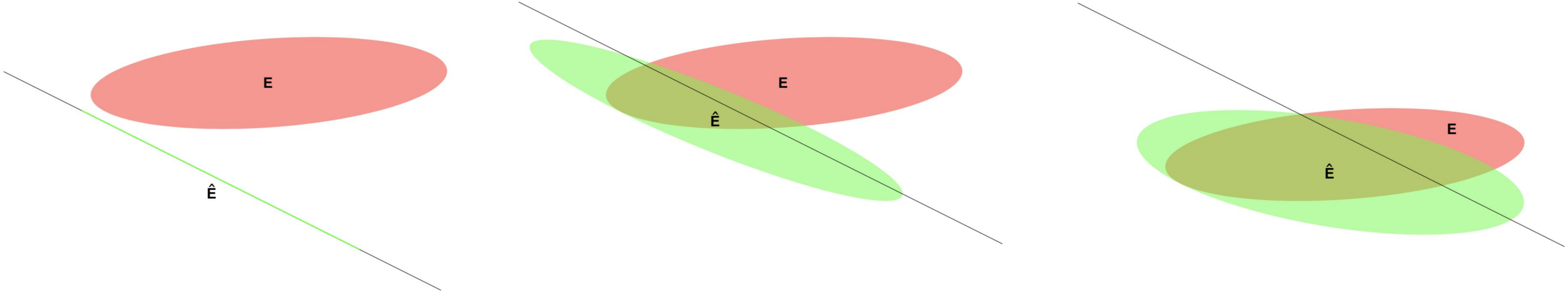
n	MNIST error
1	5.1%
5	2.5%
10	2.0%
50	1.4%

m: hidden dim = 6256
n: num halfspaces

Ellipsoid Propagation

Ellipsoid with center q and PSD matrix Q: $\mathcal{E}(q, Q) = \{x : (x-q)^\top Q^{-1}(x-q) \leq 1\}$

- Start with ℓ_2 ball which is $\mathcal{E}(x_i, I)$
- Linear layer propagation $A\mathcal{E}(q, Q) + b = \mathcal{E}(Aq + b, AQA^\top)$
- Projection onto halfspace H $MVE(Proj_H(\mathcal{E}))$



Computing $MVE(Proj_H(\mathcal{E}))$

Ellipsoid Method: Classic method for solving system of linear inequalities using a sequence of ellipsoids

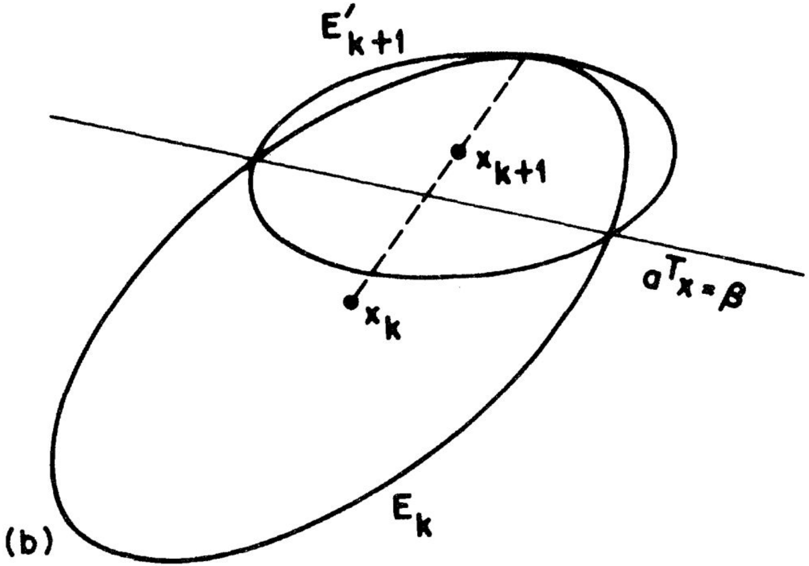
- We just use the fact that it gives an efficient way to compute

$$MVE(\mathcal{E} \cap H)$$

as a rank-one update $Q \leftarrow Q - aa^\top$

- This will help us compute

$$MVE(Proj_H(\mathcal{E}))$$



Computing $MVE(Proj_H(\mathcal{E}))$

Want $$\begin{aligned}\mathcal{E}^* &= MVE(Proj_H(\mathcal{E})) \\ &= MVE(Proj_{HP}(\mathcal{E} \cap H^c) \cup (\mathcal{E} \cap H))\end{aligned}$$

Computing $MVE(Proj_H(\mathcal{E}))$

Part of ellipsoid already in H

Want
$$
\begin{aligned}
\mathcal{E}^* &= MVE(Proj_H(\mathcal{E})) \\
&= MVE(Proj_{HP}(\mathcal{E} \cap H^c) \cup (\mathcal{E} \cap H))
\end{aligned}
$$

Projection of part of ellipsoid not in H
onto the dividing hyperplane HP

Computing $MVE(Proj_H(\mathcal{E}))$

Part of ellipsoid already in H

Want $$\mathcal{E}^* = MVE(Proj_H(\mathcal{E}))$$
$$= MVE(\underline{Proj_{HP}(\mathcal{E} \cap H^c)} \cup \overline{(\mathcal{E} \cap H)})$$

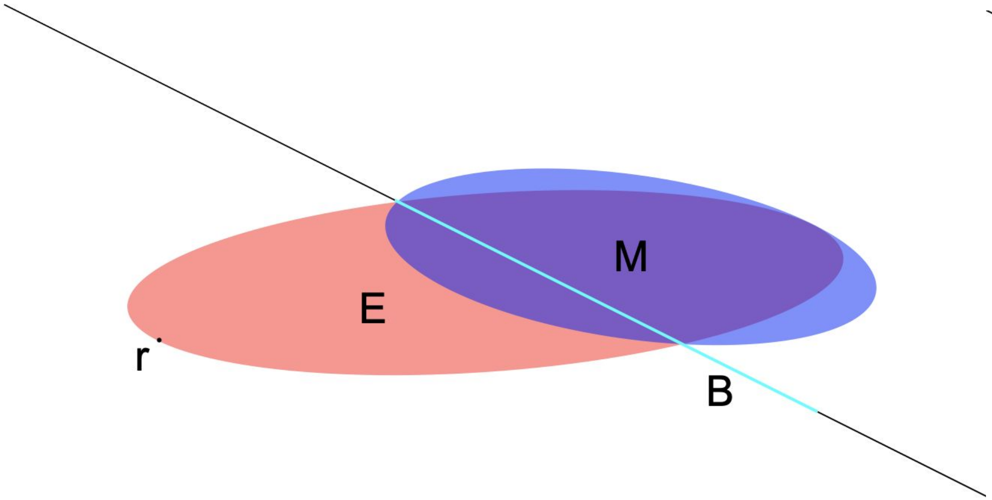
Projection of part of ellipsoid not in H
onto the dividing hyperplane HP

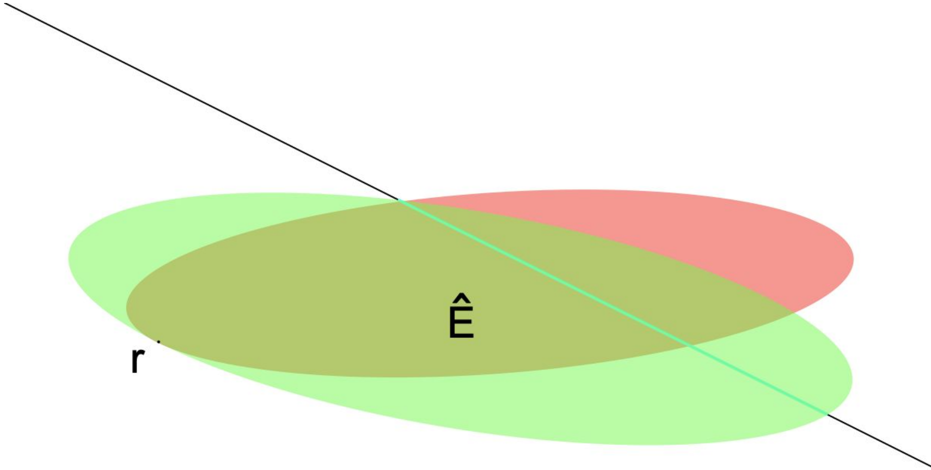
Approximate $\mathcal{E} \cap H^c$ by ellipsoid $MVE(\mathcal{E} \cap H^c)$, which is easy to project onto HP and easy to compute with Ellipsoid Method

$$\mathcal{E}^* \subseteq \hat{\mathcal{E}} := MVE(Proj_{HP}(\underline{MVE(\mathcal{E} \cap H^c)}) \cup (\mathcal{E} \cap H))$$

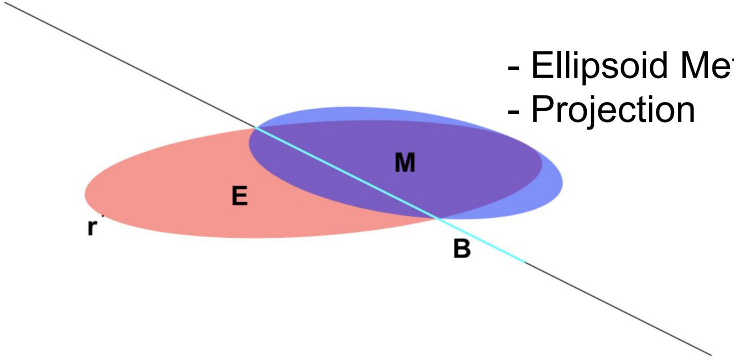
Computing $MVE(Proj_H(\mathcal{E}))$

$$\mathcal{E}^* \subseteq \hat{\mathcal{E}} := MVE(\overbrace{Proj_{HP}(\underbrace{MVE(\mathcal{E} \cap H^c)}_{M})}^{B} \cup (\mathcal{E} \cap H))$$

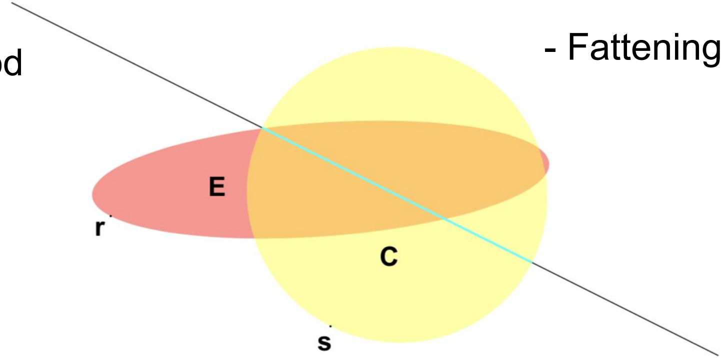




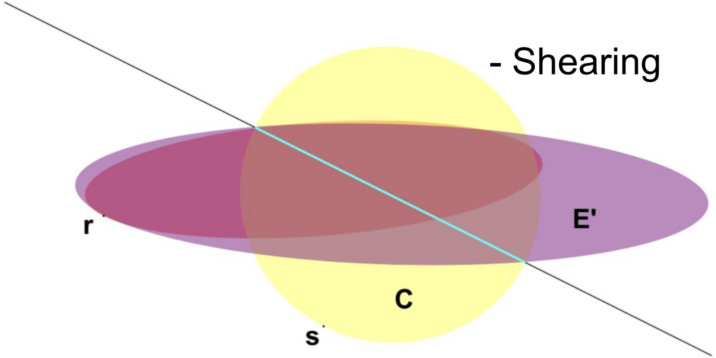
Closed-form computation



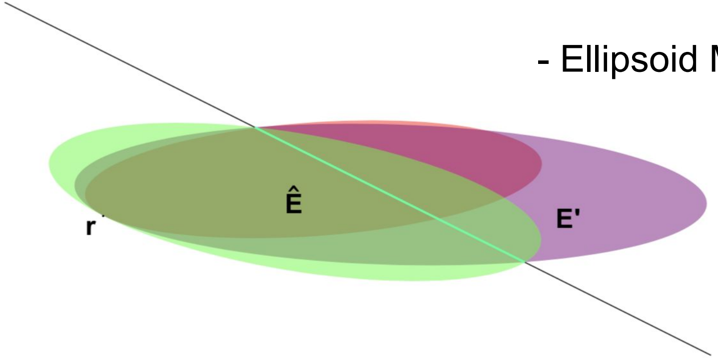
- Ellipsoid Method
- Projection



- Fattening



- Shearing



- Ellipsoid Method

Updates have simple form

The updates all have simple forms. New Q is

$$\rho v_1 v_1^\top + \phi(I - v_2 v_2^\top)(\psi v_3 v_3^\top + \omega Q)(I - v_2 v_2^\top)$$

Case depends on value of Qc, where c is the halfspace normal vector. Qc then needs to be computed at each halfspace

Case	v_1	v_2	v_3
1	0	0	0
2	0	c	0
3	$(c^\top q - s - \gamma)c - HQc/s$	c	0
4	$q - Qc/s - H(q + \tau(-\alpha)Qc/s) - \gamma c$	c	Qc
5	$q - Qc/s - H(q + \tau(-\alpha)Qc/s) - \gamma c$	c	Qc

Summary

- Background: adversarial ℓ_2 robustness
- Ellipsoid Propagation
- Current Results
- Next Step: Efficient Computation?

Current results

- We can train an MNIST network using dual LP method, then verify with ellipsoids *and obtain same certified accuracy* as dual verification.
- Like dual method, ellipsoids unable to verify PGD-trained network since bound becomes too loose with many halfspace crossings.
- Using ellipsoid propagation during training performs better than dual network on MNIST.

Method		**Robust Error**	**Standard Error**
Dual LP		55.47%	11.86%
Ellipsoids		42.50%	4.36%

Summary

- Background: adversarial ℓ_2 robustness
- Ellipsoid Propagation
- Current Results
- Next Step: Efficient Computation?

Next step: Efficient Computation?

$$\rho v_1 v_1^\top + \phi(I - v_2 v_2^\top)(\psi v_3 v_3^\top + \omega Q)(I - v_2 v_2^\top)$$

For hidden dim = n, number of halfspaces = k

- Naive implementation: O(k n^2)
  - For each halfspace update Q using matrix-vector multiplication
- Current implementation: O(k^2 n)
  - Keep Q as chain of operations, only collapses when computing Qc at each halfspace
  - Only vector-vector multiplications
  - At halfspace j, collapse chain of length j-1
- Scaling to CIFAR10
  - Approx. updates done over whole ReLU in parallel?   Matrix sketching?   Dual SDP?