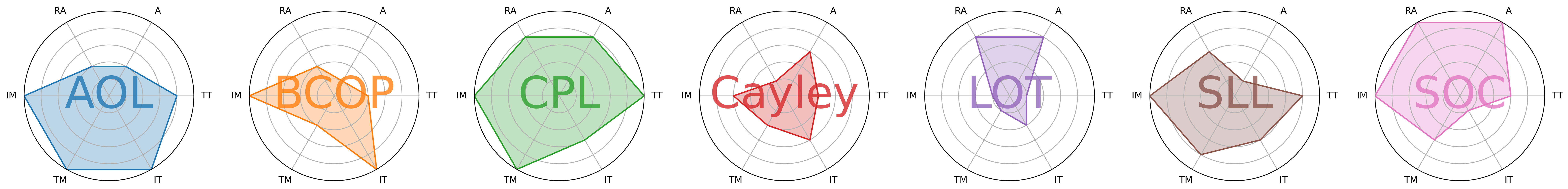
1-Lipschitz Layers Compared: Memory, Speed and Certifiable Robustness

Bernd Prach*, Fabio Brau*, Giorgio Buttazzo, and Christoph H. Lampert

Institute of Science and Technology Austria

Sant'Anna School of Advanced Studies – Pisa

Overview



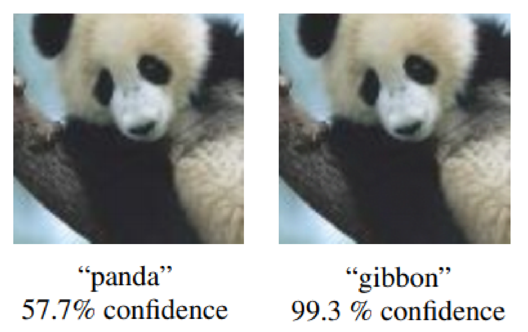
Rating robust accuracy (RA), accuracy (A), training time (TT), inference time (IT), training memory (TM) and inference memory (IM).

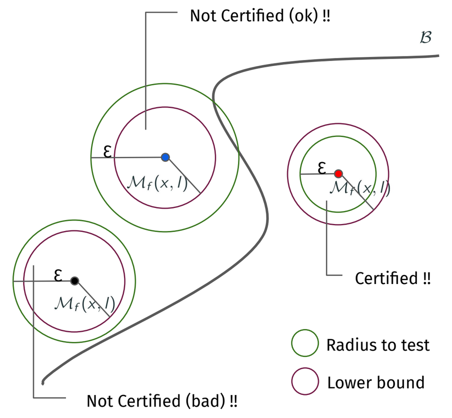
Summary

Comparison of 7 methods for creating 1-Lipschitz convolutions from the literature using 6 metrics on 4 datasets with 4 model sizes and 4 training time budgets.

Introduction

Adversarial Examples: [1]





Definition 1. *We call f 1-Lipschitz, if for all x and y*

$$\|f(x) - f(y)\|_2 \leq \|x - y\|_2.$$

(difference of outputs) *(difference of inputs)*

Lemma 1. *An input x is classified robustly by a classifier f for perturbations of size up to $\mathcal{M}_f(x)$, where*

$$\mathcal{M}_f(x) = \frac{1}{\sqrt{2}} \left[f_l(x) - \max_{i \neq l} f_i(x) \right]_+$$

[1] C. Szegedy, 2014, Intriguing properties of neural networks

Theoretical Analysis

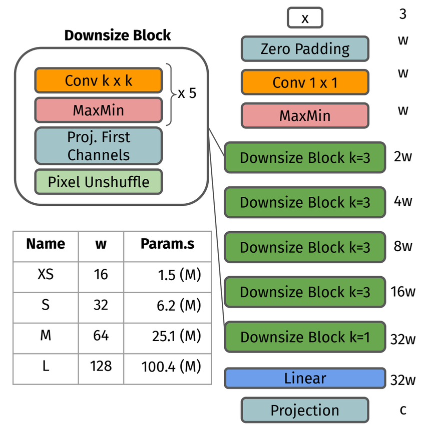
For batch size b and input size $s \times s \times c$:

Method	FLOPs $\mathcal{O}(\cdot)$	Memory $\mathcal{O}(\cdot)$
Standard	bs^2c^2	$bs^2c + c^2$
AOL[1]	$bs^2c^2 + c^3$	$bs^2c + c^2$
BCOP[2]	$bs^2c^2 + c^3$	$bs^2c + c^2$
Cayley[3]	$bs^2c^2 + s^2c^3$	$bs^2c + s^2c^2$
CPL[4]	bs^2c^2	$bs^2c + c^2$
LOT[5]	$bs^2c^2 + s^2c^3$	$bs^2c + s^2c^2$
SLL[6]	$bs^2c^2 + c^3$	$bs^2c + c^2$
SOC[7]	bs^2c^2	$bs^2c + c^2$

[1] B.Prach, 2022, Almost-orthogonal layers for efficient general-purpose Lipschitz networks
[2] Q. Li, 2019, Preventing gradient attenuation in Lipschitz constrained convolutional networks
[3] A.Trockman, 2021, Orthogonalizing convolutional layers with the Cayley transform
[4] L. Meunier, 2022, A dynamical system perspective for Lipschitz neural networks
[5] X. Xu, 2022, Lot: Layer-wise orthogonal training on improving L2 certified robustness
[6] A. Araujo, 2023, A unified algebraic perspective on Lipschitz neural networks
[7] S. Singla, 2021, Skew orthogonal convolutions

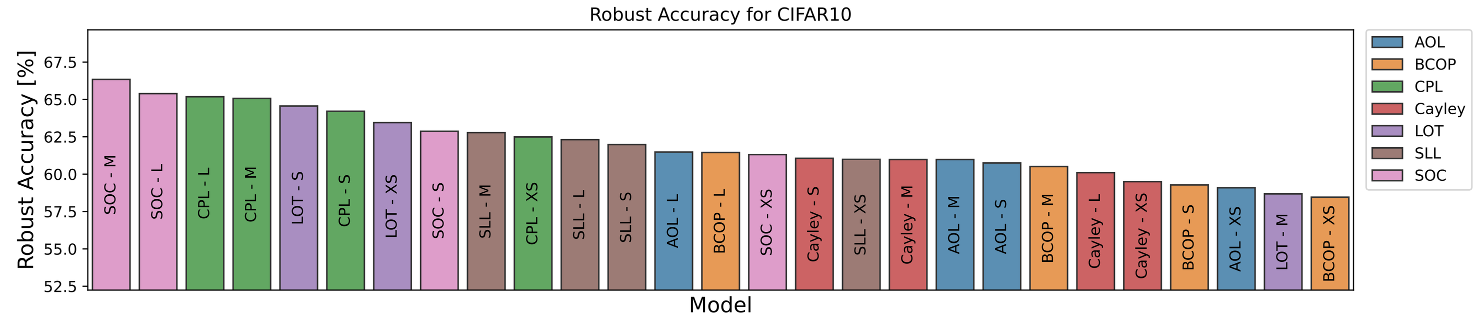
Architecture

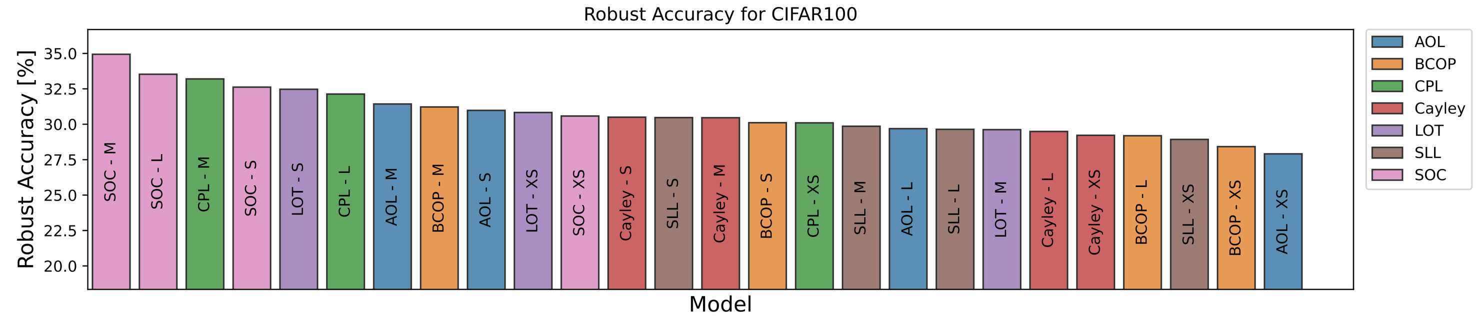
1. MaxMin activation instead of ReLU
2. Squared Convolutions ($c_{in} = c_{out}$)
3. Pixel Unshuffle reduces spatial dimension increasing channels ($\times 4$)
4. Proj. First reduces channels ($\div 2$)
5. Final Projection on first c-channels

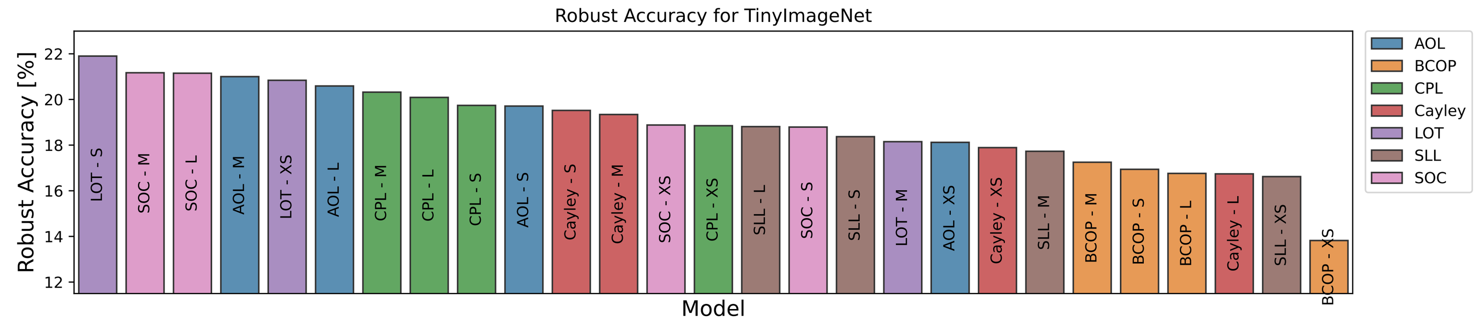


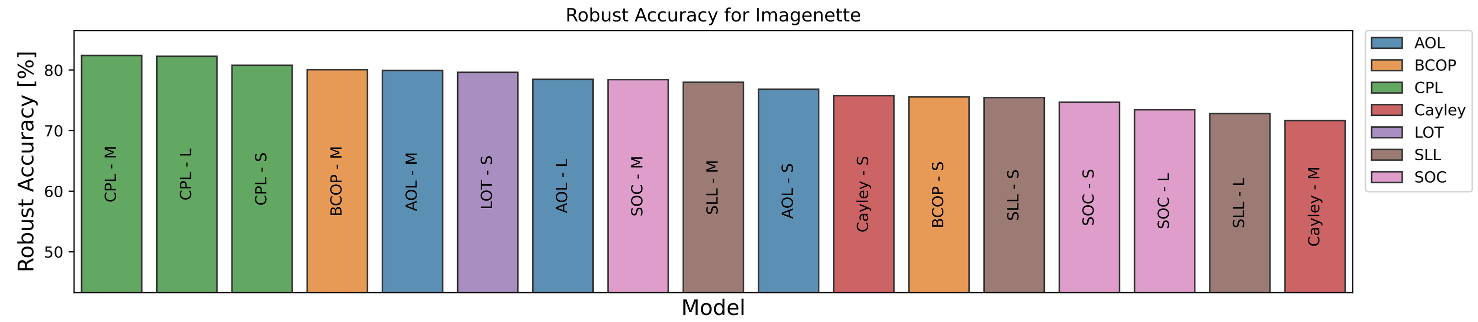
Certifiable Robust Accuracy

Certified robust accuracy for perturbations of size up to $\epsilon = 36/255$, for a training time of 24h:

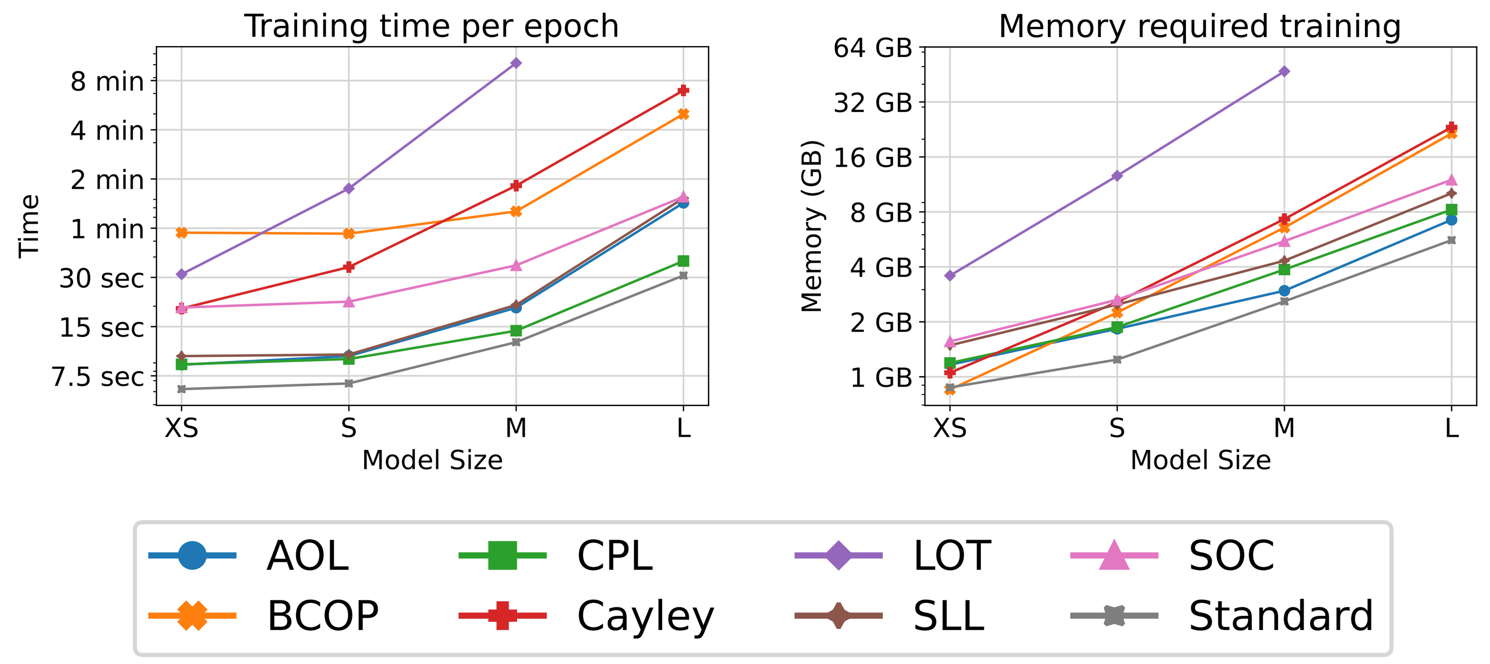








Memory and Speed



Conclusion & Interpretation

→ CPL seems most promising, followed by SOC.
→ Skip connections or identity initialization seem useful.
→ Computation on kernels helps with larger input resolution.

Contact Information

Web: https://berndprach.github.io/
https://fabiobrau.github.io/
Email: bprach@ist.ac.at
f.brau@santannapisa.it

Paper: →