Expressive Losses for Verified Robustness via Convex Combinations

Alessandro De Palma — Inria, ENS | PSL

Rudy Bunel, Krishnamurthy Dvijotham, M. Pawan Kumar, Robert Stanforth — Google DeepMind

Alessio Lomuscio — Imperial College London

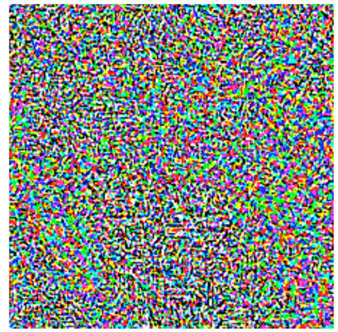
(Verified) Adversarial Robustness

“panda”
57.7% confidence

(Verified) Adversarial Robustness

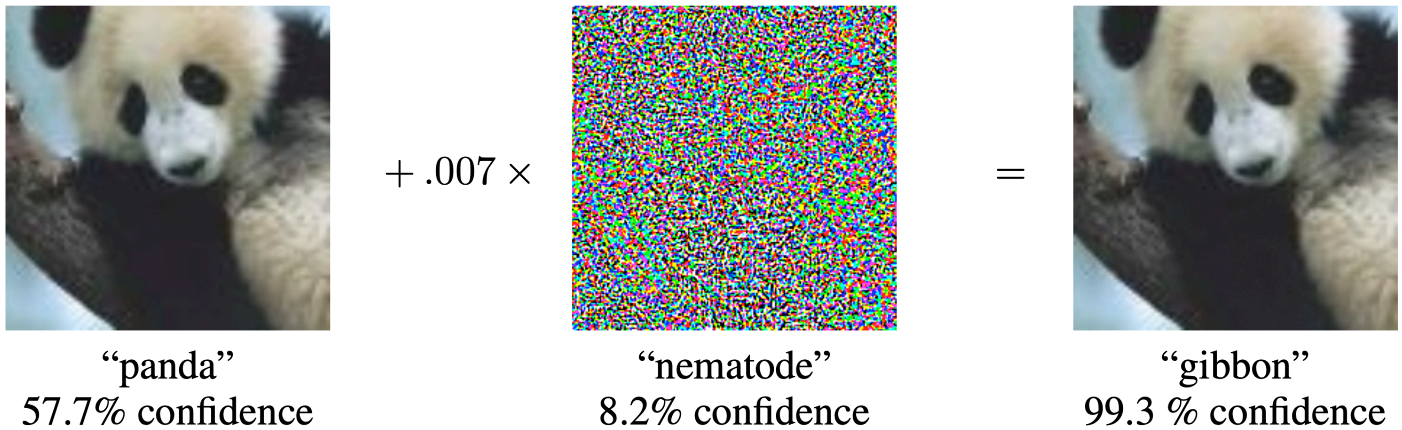
"panda"
57.7% confidence

$+ .007 \times$



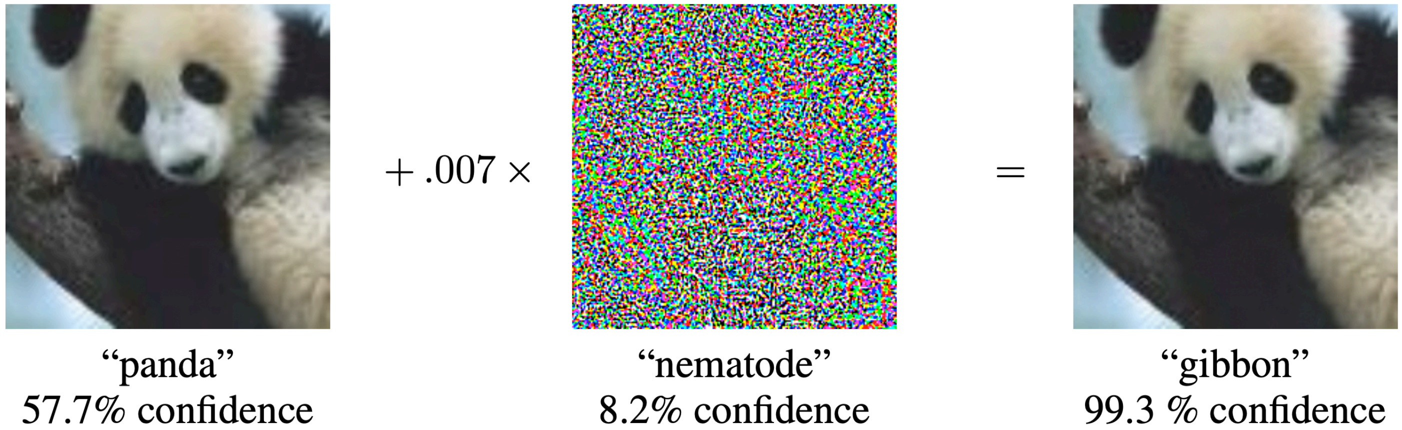
"nematode"
8.2% confidence

(Verified) Adversarial Robustness

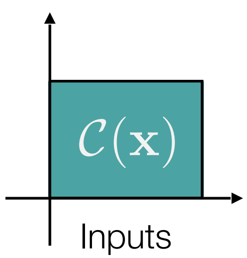


[Goodfellow et al., 2015]

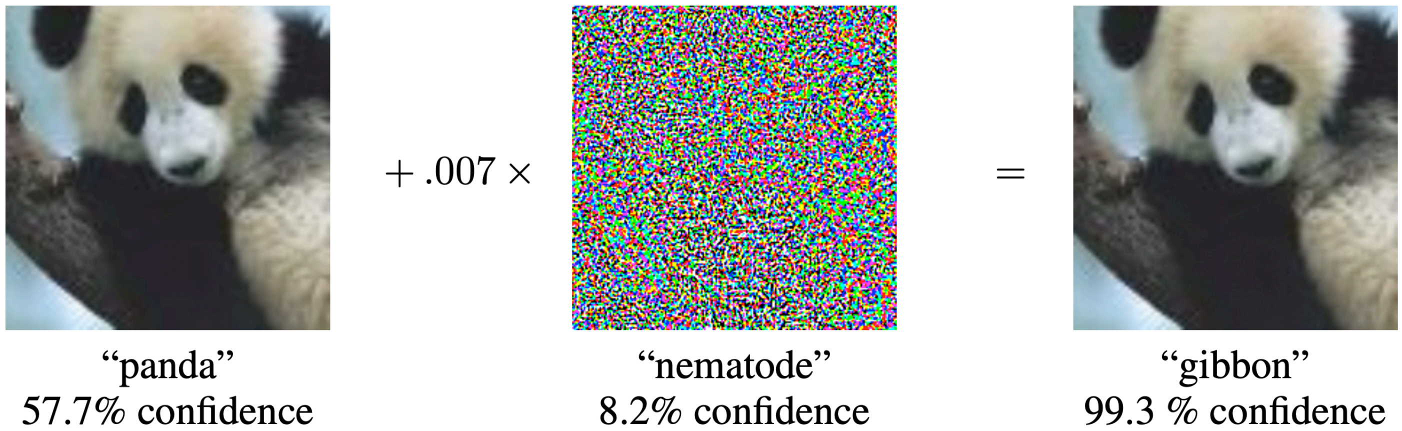
(Verified) Adversarial Robustness



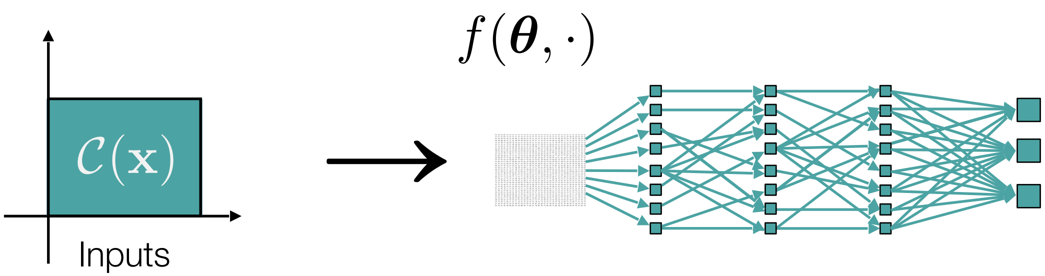
[Goodfellow et al., 2015]



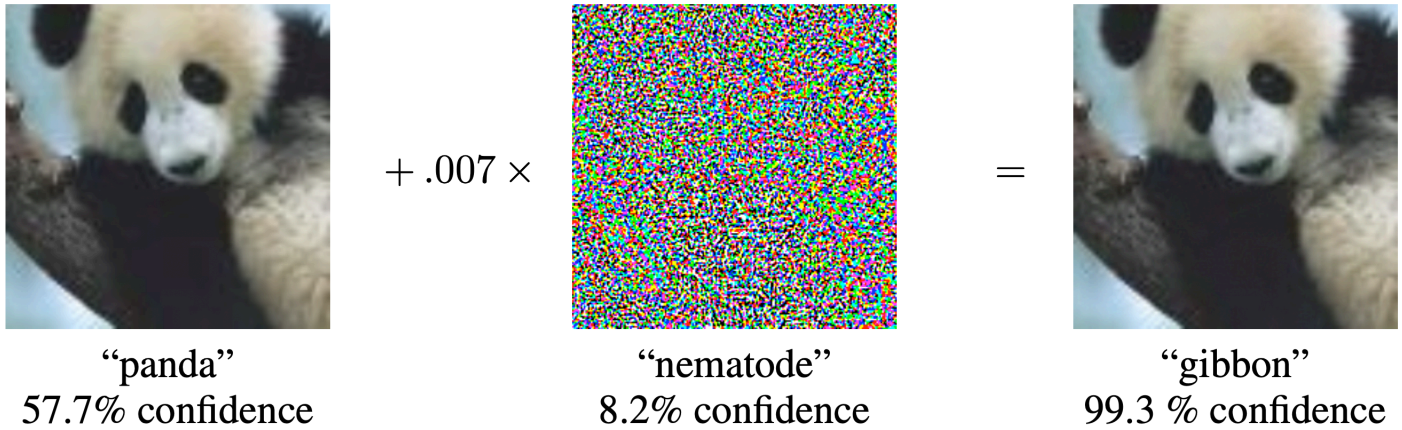
(Verified) Adversarial Robustness



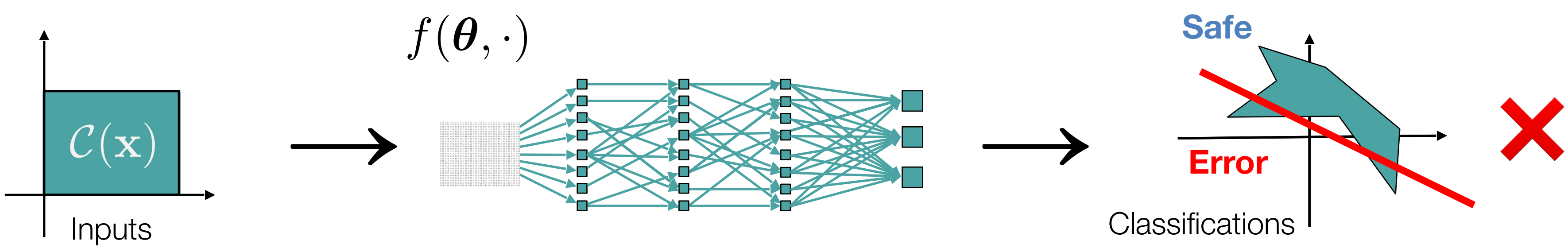
[Goodfellow et al., 2015]



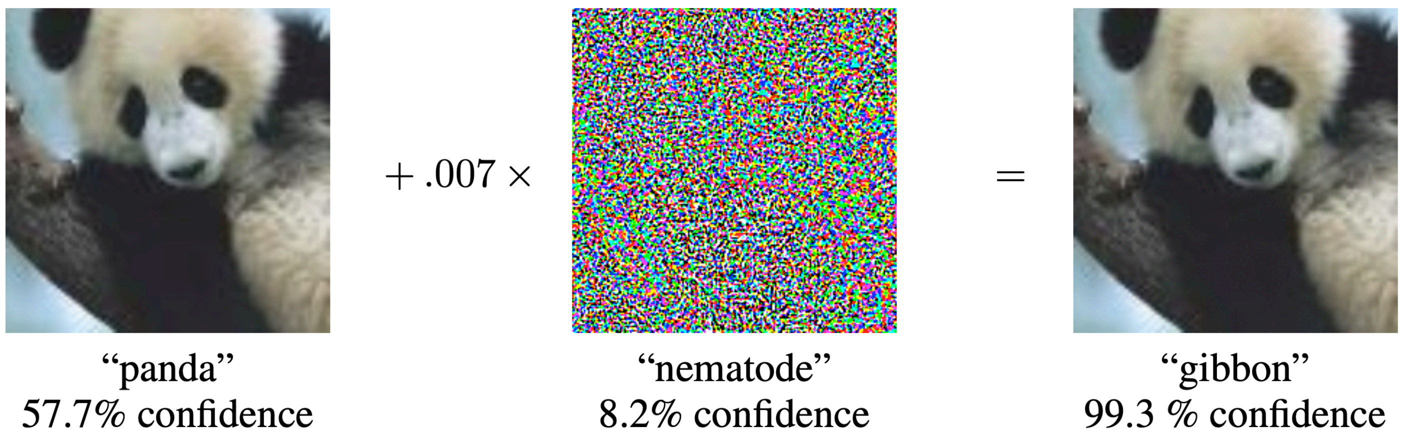
(Verified) Adversarial Robustness



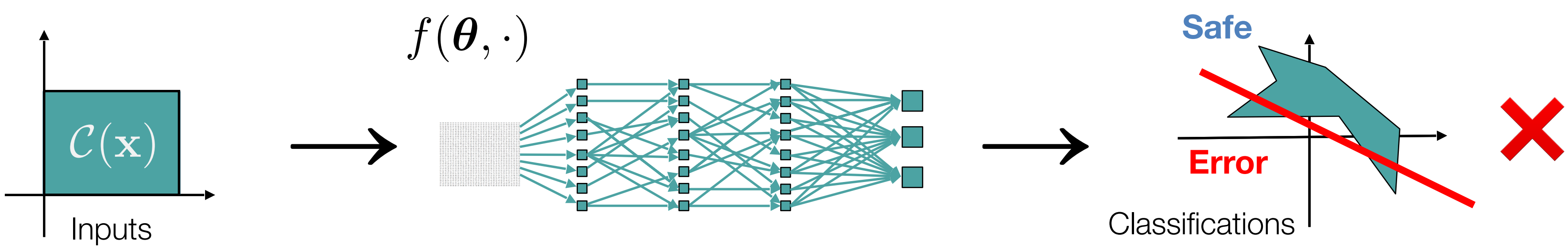
[Goodfellow et al., 2015]



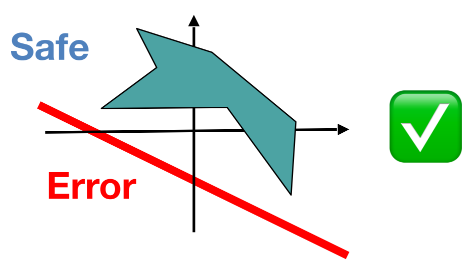
(Verified) Adversarial Robustness



[Goodfellow et al., 2015]



Verified training:



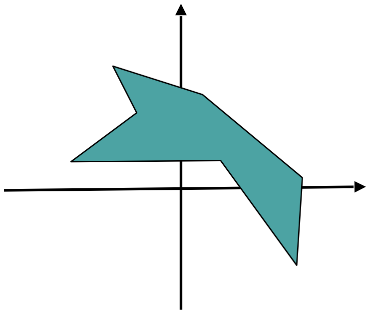
Robust Training

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \left[\max_{\mathbf{x}'\in\mathcal{C}(\mathbf{x})} \mathcal{L}(f(\boldsymbol{\theta},\mathbf{x}'),\mathbf{y}) \right]$$

Robust Training

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \left[\max_{\mathbf{x}'\in\mathcal{C}(\mathbf{x})} \mathcal{L}(f(\boldsymbol{\theta},\mathbf{x}'),\mathbf{y}) \right]$$

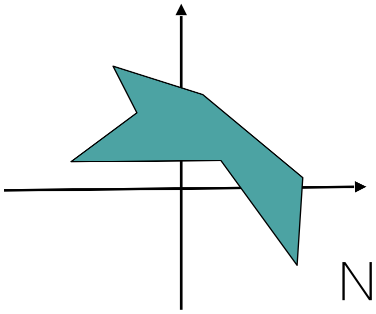
$$\mathcal{L}^*(f(\boldsymbol{\theta},\mathbf{x}),y)$$



Robust Training

$$\min_{\boldsymbol{\theta}} \underset{(\mathbf{x},\mathbf{y})\in\mathcal{D}}{\mathbb{E}}\left[\max_{\mathbf{x}'\in\mathcal{C}(\mathbf{x})} \mathcal{L}(f(\boldsymbol{\theta},\mathbf{x}'),\mathbf{y})\right]$$

$$\mathcal{L}^*(f(\boldsymbol{\theta},\mathbf{x}),y)$$



NP-Complete

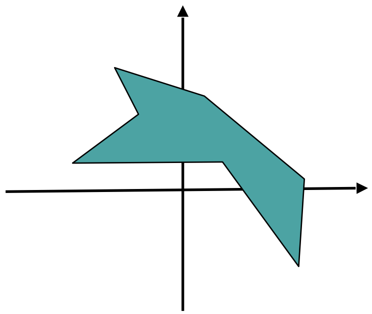
[Katz et al., 2017]

Adversarial Training

$$\text{Lower bound} \rightarrow \text{adversarial training}$$

[Madry et al., 2018, Wong et al., 2020]

$$\mathcal{L}^*(f(\boldsymbol{\theta}, \mathbf{x}), y)$$

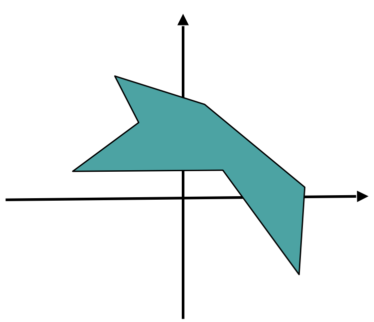


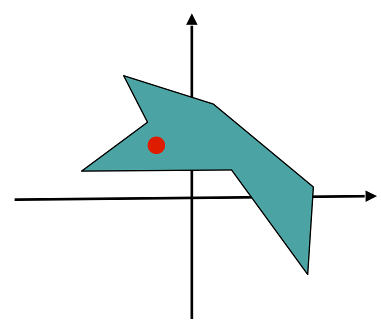
Adversarial Training

Lower bound $\rightarrow$ adversarial training

[Madry et al., 2018, Wong et al., 2020]

$$\mathcal{L}^*(f(\boldsymbol{\theta}, \mathbf{x}), y) \quad \geq \quad \mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}_{\mathrm{adv}}), y)$$



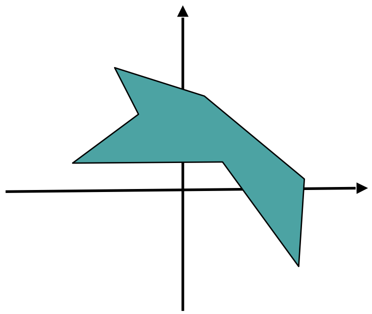


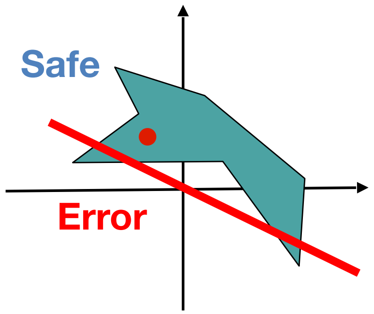
Adversarial Training

Lower bound $\rightarrow$ adversarial training

[Madry et al., 2018, Wong et al., 2020]

$$\mathcal{L}^*(f(\boldsymbol{\theta}, \mathbf{x}), y) \geq \mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}_{\mathrm{adv}}), y)$$



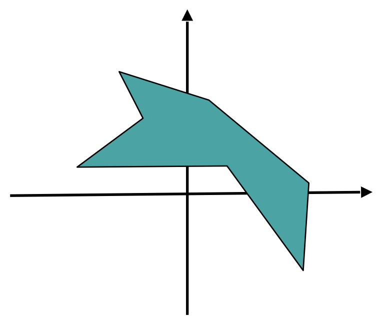


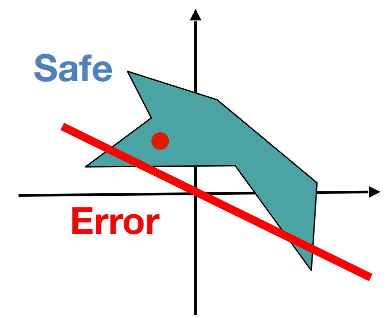
Adversarial Training

Lower bound $\rightarrow$ adversarial training

[Madry et al., 2018, Wong et al., 2020]

$$\mathcal{L}^*(f(\boldsymbol{\theta}, \mathbf{x}), y) \quad \geq \quad \mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}_{\text{adv}}), y)$$





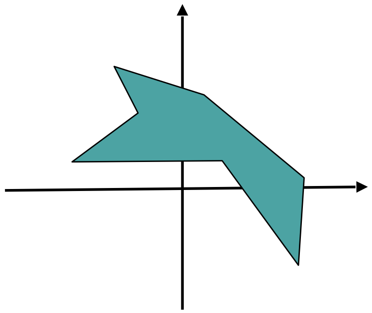
formal guarantees?

Verified Training

Upper bound $\rightarrow$ certified training

[Gowal et al., 2018, Zhang et al., 2020, Shi et al., 2021]

$$\mathcal{L}^*(f(\boldsymbol{\theta}, \mathbf{x}), y)$$

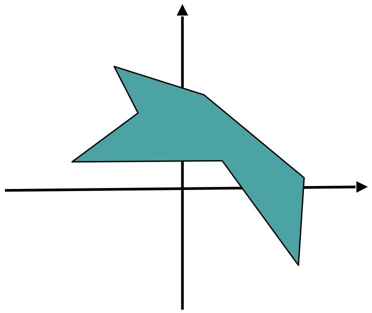


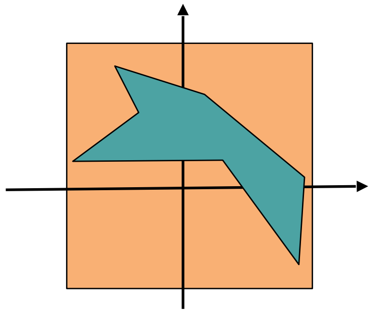
Verified Training

Upper bound → certified training

[Gowal et al., 2018, Zhang et al., 2020, Shi et al., 2021]

$$\mathcal{L}^*(f(\boldsymbol{\theta}, \mathbf{x}), y) \leq \mathcal{L}_{\text{ver}}(f(\boldsymbol{\theta}, \mathbf{x}), y)$$



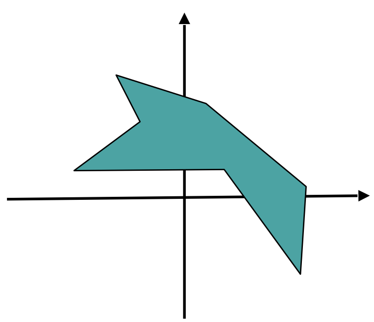


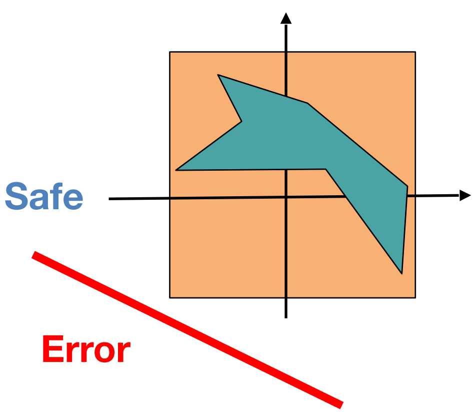
Verified Training

Upper bound $\rightarrow$ certified training

[Gowal et al., 2018, Zhang et al., 2020, Shi et al., 2021]

$$\mathcal{L}^*(f(\boldsymbol{\theta}, \mathbf{x}), y) \quad \leq \quad \mathcal{L}_{\text{ver}}(f(\boldsymbol{\theta}, \mathbf{x}), y)$$



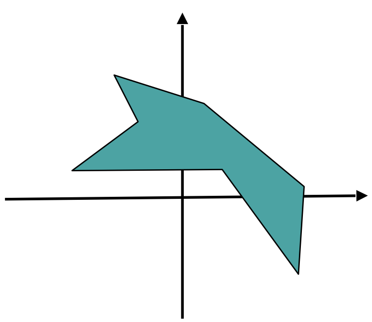


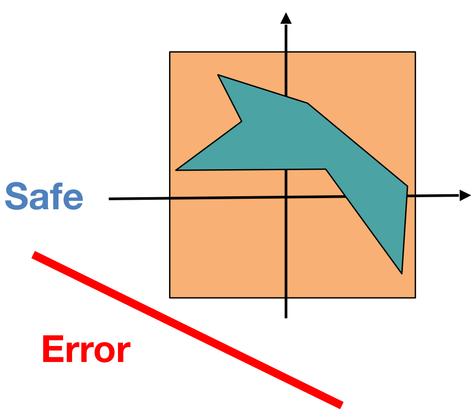
Verified Training

Upper bound $\rightarrow$ certified training

[Gowal et al., 2018, Zhang et al., 2020, Shi et al., 2021]

$$\mathcal{L}^*(f(\boldsymbol{\theta}, \mathbf{x}), y) \quad \leq \quad \mathcal{L}_{\text{ver}}(f(\boldsymbol{\theta}, \mathbf{x}), y)$$



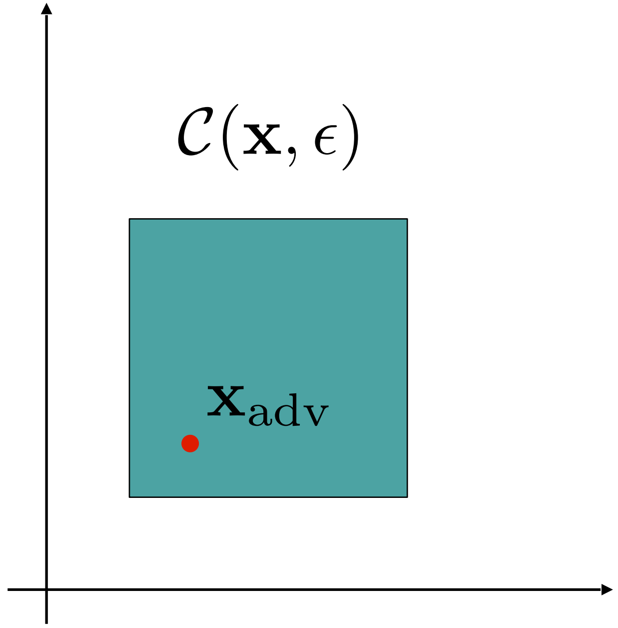


large cost in standard performance

Hybrid Training: SABR

[Müller et al., 2023]

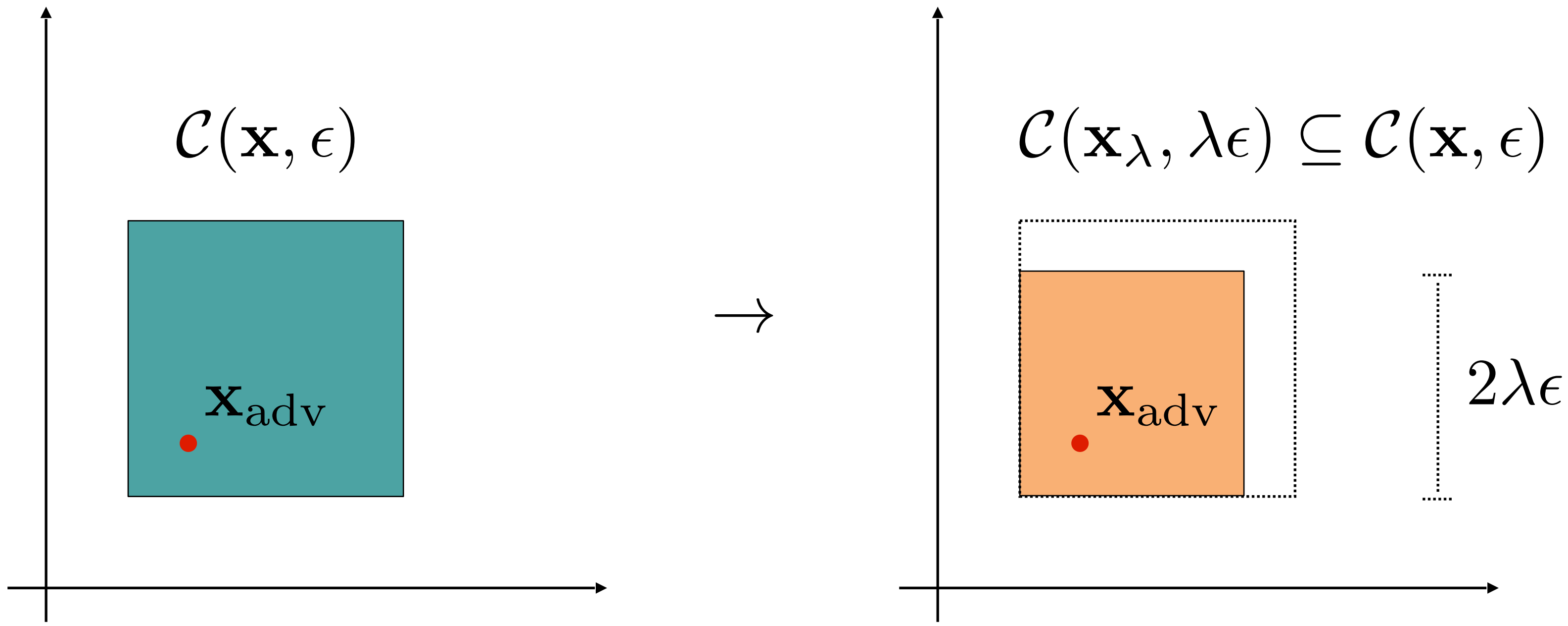
Compute over-approximation over a parametrized subset of the input domain that includes an adversarial attack.



Hybrid Training: SABR

[Müller et al., 2023]

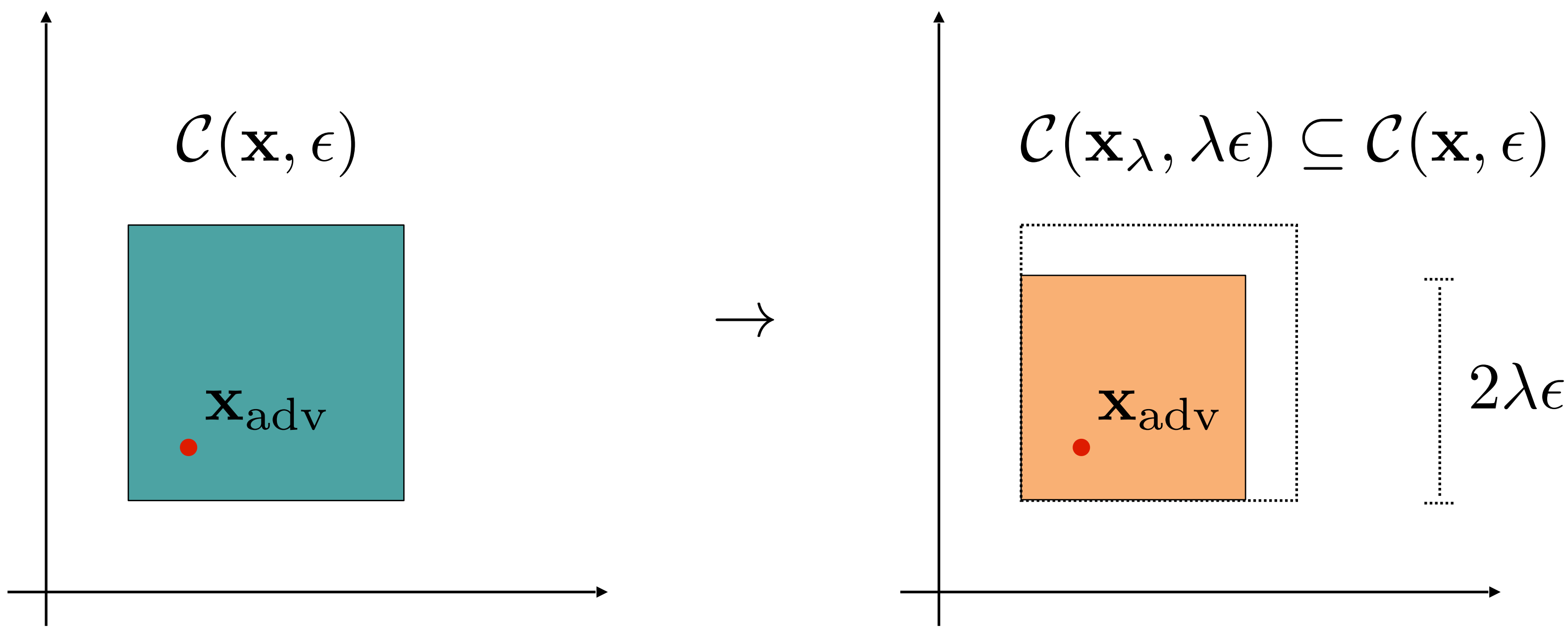
Compute over-approximation over a parametrized subset of the input domain that includes an adversarial attack.



Hybrid Training: SABR

[Müller et al., 2023]

Compute over-approximation over a parametrized subset of the input domain that includes an adversarial attack.



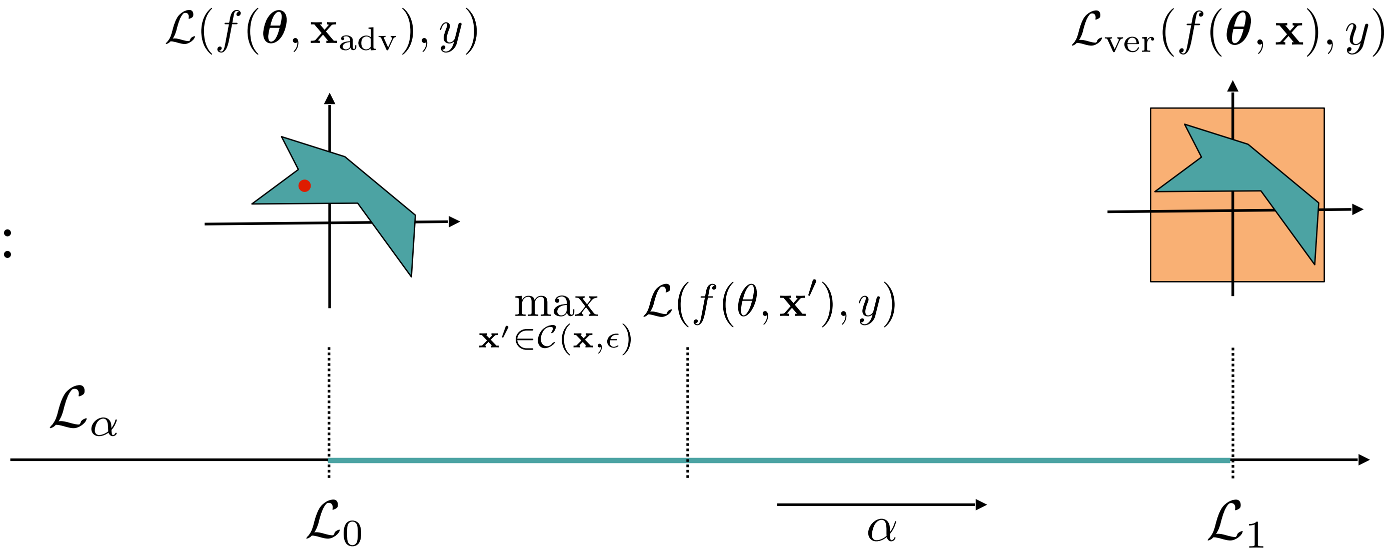
state-of-the-art results

💡 Loss Expressivity

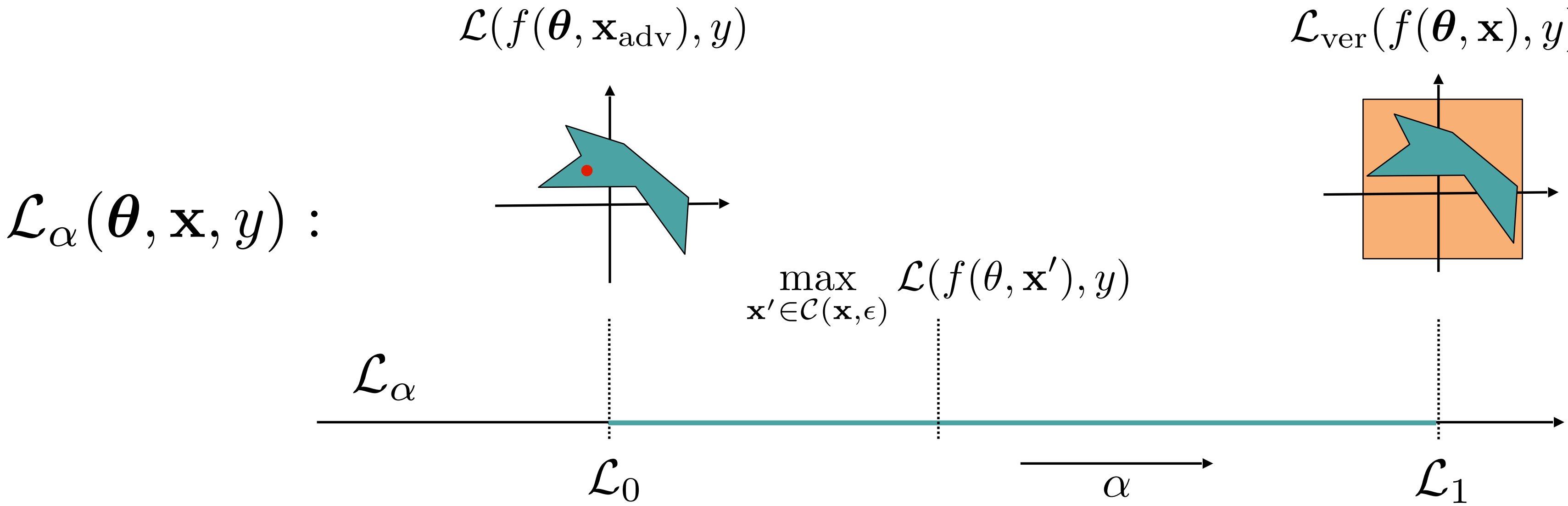
$$\mathcal{L}_\alpha(\boldsymbol{\theta}, \mathbf{x}, y) :$$

Loss Expressivity

$$\mathcal{L}_\alpha(\boldsymbol{\theta}, \mathbf{x}, y) :$$



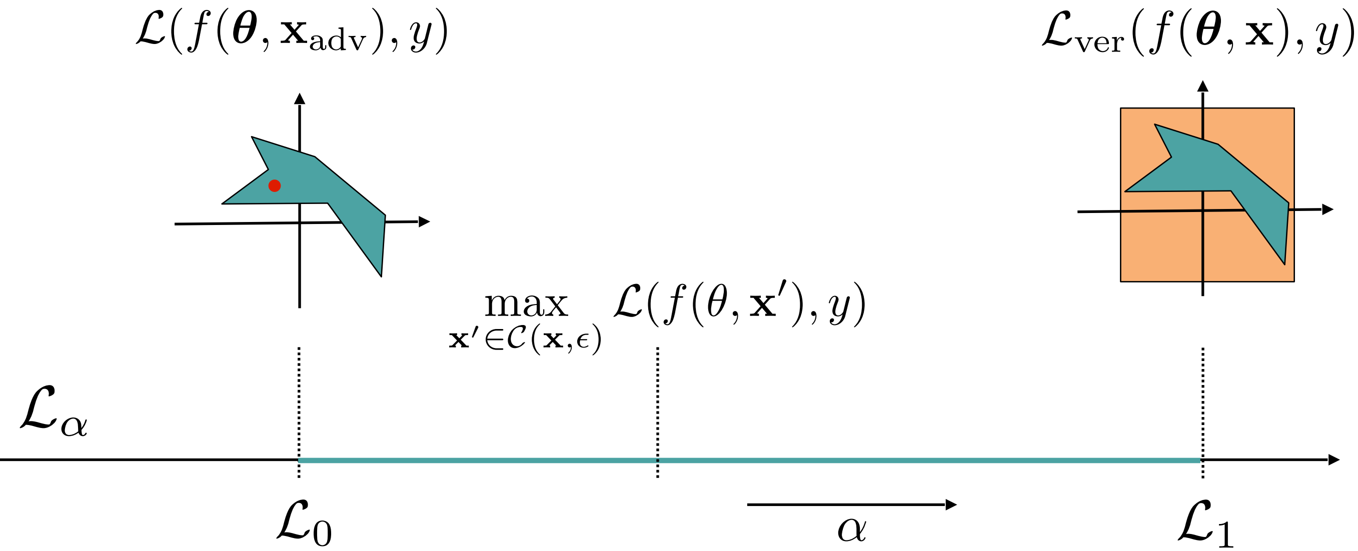
Loss Expressivity



- $\mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}_{\text{adv}}), y) \leq \mathcal{L}_\alpha(\boldsymbol{\theta}, \mathbf{x}, y) \leq \mathcal{L}_{\text{ver}}(f(\boldsymbol{\theta}, \mathbf{x}), y) \; \forall \; \alpha \in [0, 1]$;

💡 Loss Expressivity

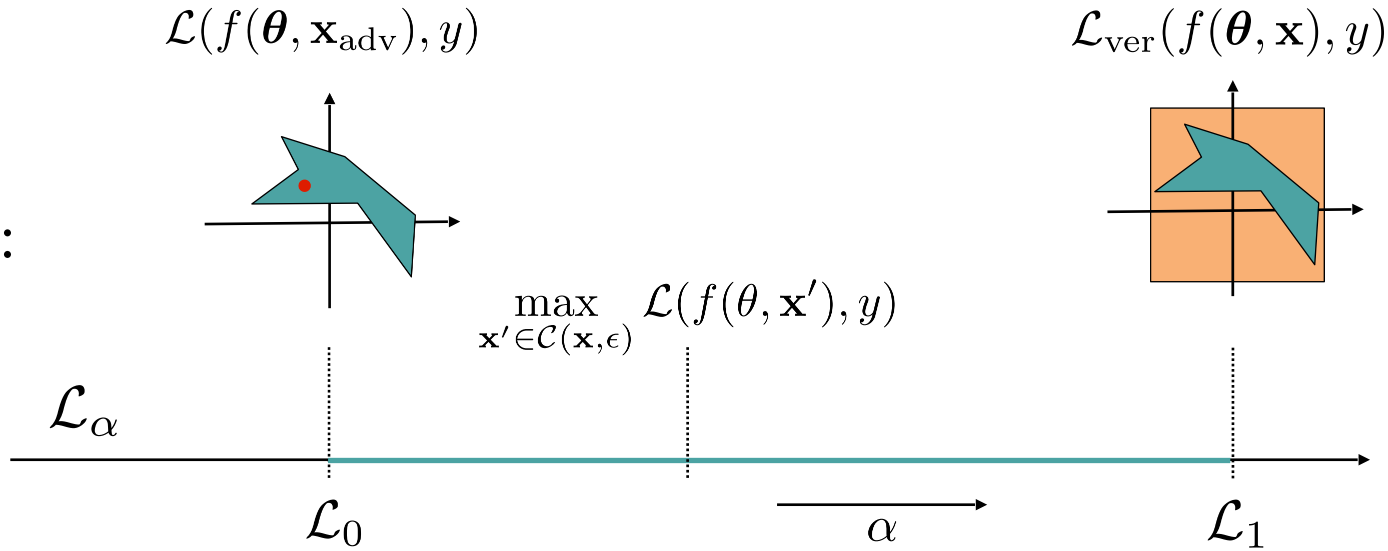
$\mathcal{L}_\alpha(\boldsymbol{\theta}, \mathbf{x}, y)$:



- $\mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}_{\text{adv}}), y) \leq \mathcal{L}_\alpha(\boldsymbol{\theta}, \mathbf{x}, y) \leq \mathcal{L}_{\text{ver}}(f(\boldsymbol{\theta}, \mathbf{x}), y) \; \forall \; \alpha \in [0, 1]$;
- $\mathcal{L}_\alpha(\boldsymbol{\theta}, \mathbf{x}, y)$ continuous and monotonically increasing for $\alpha \in [0, 1]$;

Loss Expressivity

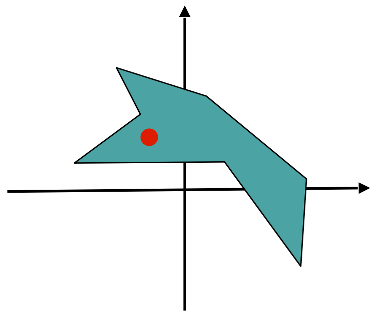
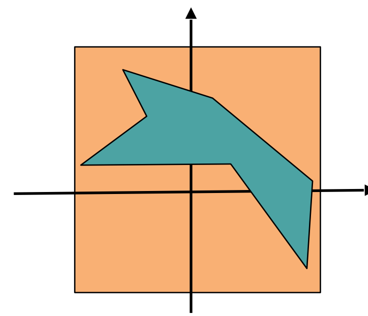
$\mathcal{L}_\alpha(\boldsymbol{\theta}, \mathbf{x}, y)$:



- $\mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}_{\text{adv}}), y) \leq \mathcal{L}_\alpha(\boldsymbol{\theta}, \mathbf{x}, y) \leq \mathcal{L}_{\text{ver}}(f(\boldsymbol{\theta}, \mathbf{x}), y) \ \forall \ \alpha \in [0, 1]$;
- $\mathcal{L}_\alpha(\boldsymbol{\theta}, \mathbf{x}, y)$ continuous and monotonically increasing for $\alpha \in [0, 1]$;
- $\mathcal{L}_0(\boldsymbol{\theta}, \mathbf{x}, y) = \mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}_{\text{adv}}), y)$;
- $\mathcal{L}_1(\boldsymbol{\theta}, \mathbf{x}, y) = \mathcal{L}_{\text{ver}}(f(\boldsymbol{\theta}, \mathbf{x}), y)$.

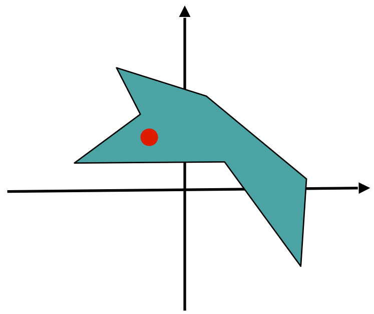
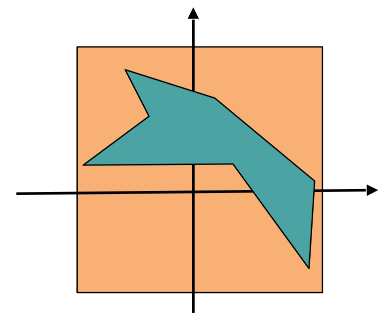
Expressivity via Convex Combinations

CC-IBP

$\mathcal{L}(-\,[(1-\alpha)$  $+\,\alpha$  $],y)$

Expressivity via Convex Combinations

CC-IBP

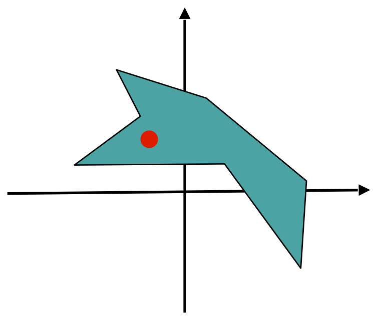
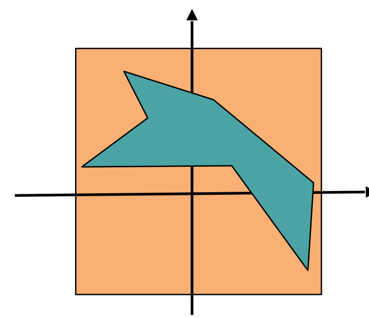
$\mathcal{L}(-[(1-\alpha)$  $+\,\alpha$  $], y)$

MTL-IBP

$$(1-\alpha)\mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}_{\text{adv}}), y) + \alpha\ \mathcal{L}_{\text{ver}}(f(\boldsymbol{\theta}, \mathbf{x}), y)$$

Expressivity via Convex Combinations

CC-IBP

$\mathcal{L}(-\,[(1-\alpha)$  $+\,\alpha$  $],\, y)$

MTL-IBP

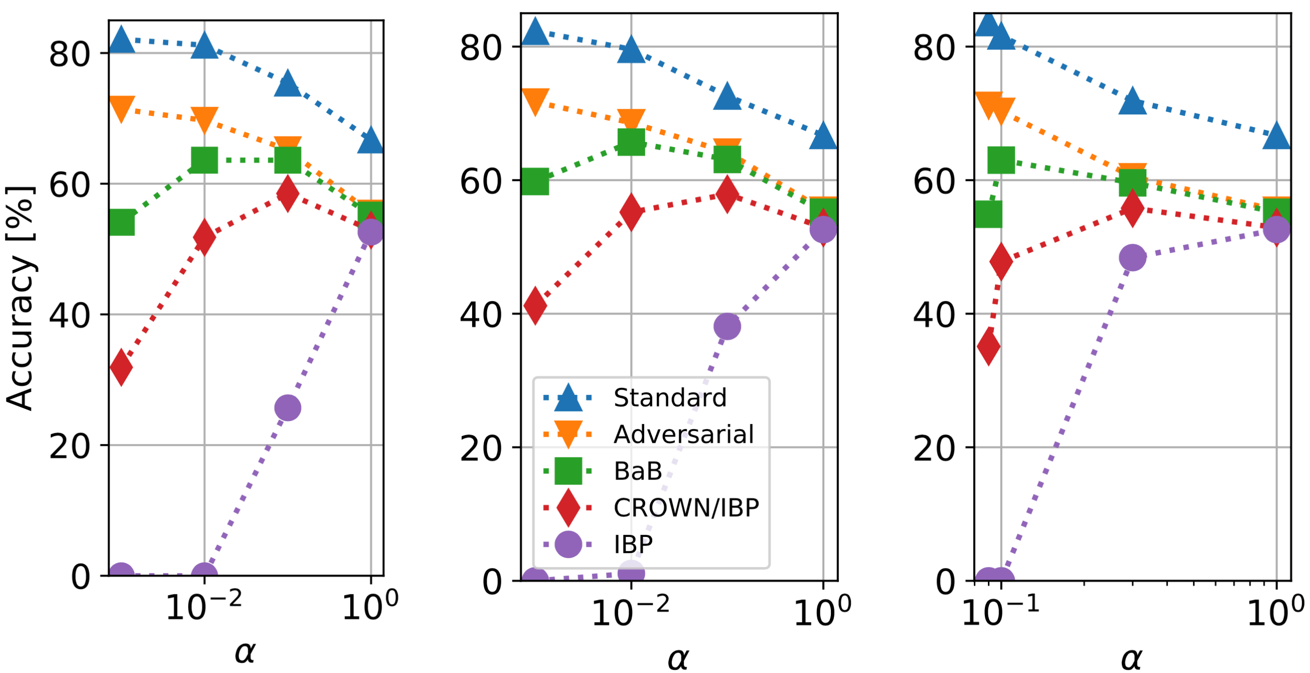
$$(1-\alpha)\mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}_{\text{adv}}), y) + \alpha\ \mathcal{L}_{\text{ver}}(f(\boldsymbol{\theta}, \mathbf{x}), y)$$

Exp-IBP

$$\mathcal{L}_{\alpha,\text{Exp}}(\boldsymbol{\theta}, \mathbf{x}, y) := \mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}_{\text{adv}}), y)^{(1-\alpha)}\ \mathcal{L}_{\text{ver}}(f(\boldsymbol{\theta}, \mathbf{x}), y)^{\alpha}$$

Loss Sensitivity

Sensitivity of CC-IBP, MTL-IBP and Exp-IBP to α for robustness to ℓ_∞ perturbations on CIFAR-10.



(a) CC-IBP, $\epsilon = {}^{2}/_{255}$.

(b) MTL-IBP, $\epsilon = {}^{2}/_{255}$.

(c) Exp-IBP, $\epsilon = {}^{2}/_{255}$.

Experimental Results

Performance of different verified training algorithms under ℓ_∞ norm perturbations on the TinyImageNet and downscaled (64×64) ImageNet datasets.

Dataset	ϵ	Method	Standard acc. [%]	Verified rob. acc. [%]
TinyImageNet	$\frac{1}{255}$	CC-IBP	38.61	26.39
		MTL-IBP	37.56	26.09
		Exp-IBP	38.71	26.18
		STAPS	28.98	22.16
		SABR	28.97	21.36
		SortNet	25.69	18.18
		IBP	25.40	19.92
		CROWN-IBP	25.62	17.93
ImageNet64	$\frac{1}{255}$	CC-IBP	19.62	11.87
		MTL-IBP	20.15	12.13
		Exp-IBP	22.73	13.30
		SortNet	14.79	9.54
		CROWN-IBP	16.23	8.73
		IBP	15.96	6.13

Conclusions

- Expressivity $\rightarrow$ state-of-the-art certified training;
- Expressivity easily obtained via convex combinations;
- Verified accuracy still comes at great cost in standard performance.

Code and models

https://github.com/alessandrodepalma/expressive-losses

alessandro.de-palma@inria.fr