Efficient Loss Function by Minimizing the Detrimental Effect of Floating-point Errors on Gradient-based Attacks

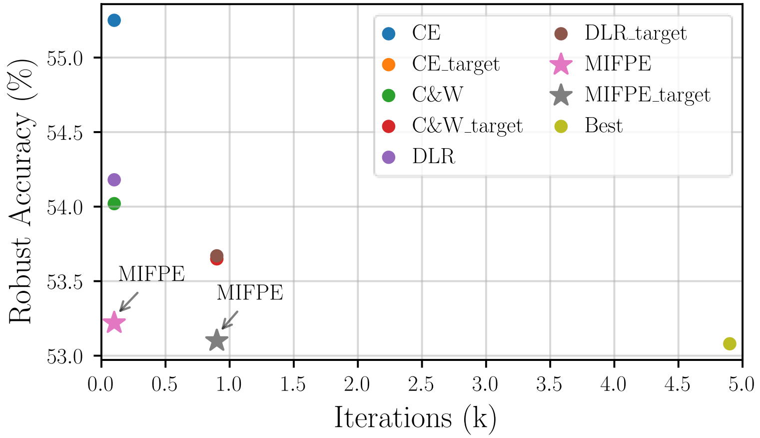
Yunrui Yu

Chengzhong Xu

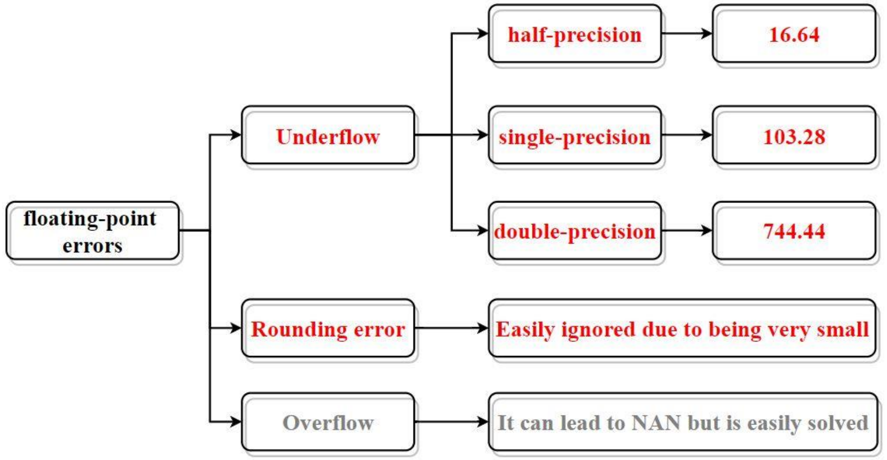
State key lab of IoTSC University of Macau

Paper Tag: **TUE-AM-387**

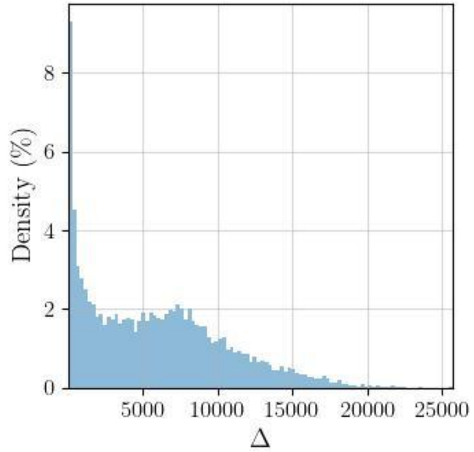
Quick Review

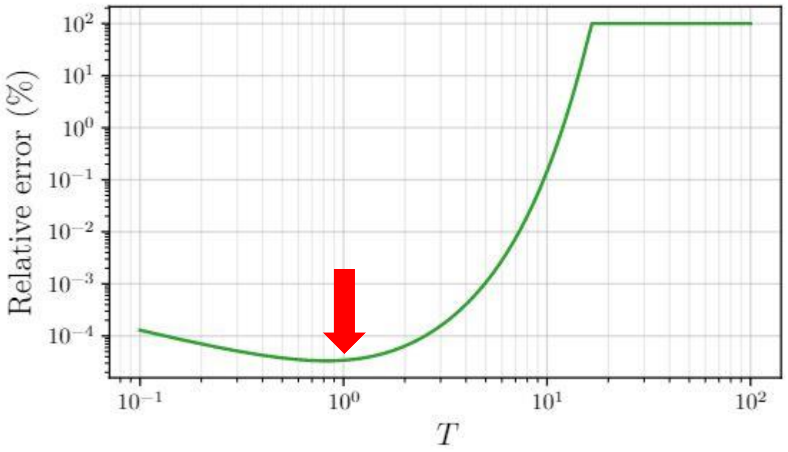


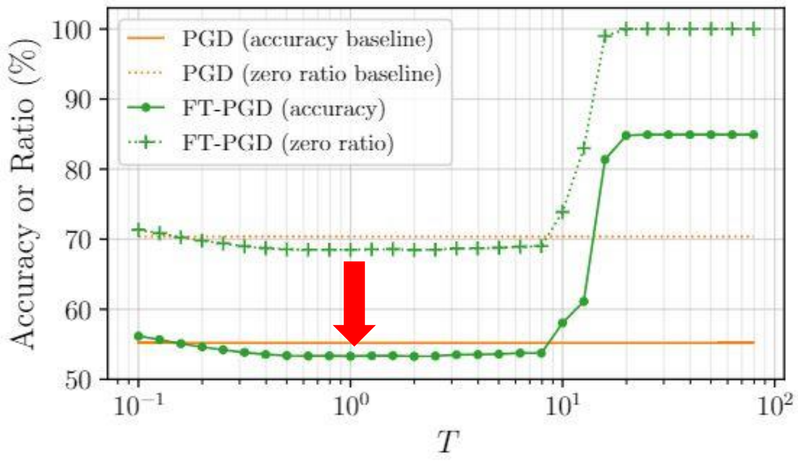
$e^{-\lambda}$



$\Delta = z_{\pi 1} - z_{\pi 2} \geq 0$







$$\mathcal{L}^{\text{MIFPE}}(\mathbf{z}, y) \triangleq \mathcal{L}^{\text{ce}}(T\mathbf{z}/\Delta_{\text{detach}}, y),$$

$$\mathcal{L}^{\text{MIFPE}}_{\text{target}}(\mathbf{z}, y_t) = -\mathcal{L}^{\text{ce}}(T\mathbf{z}/\Delta_{\text{detach}}, y_t),$$

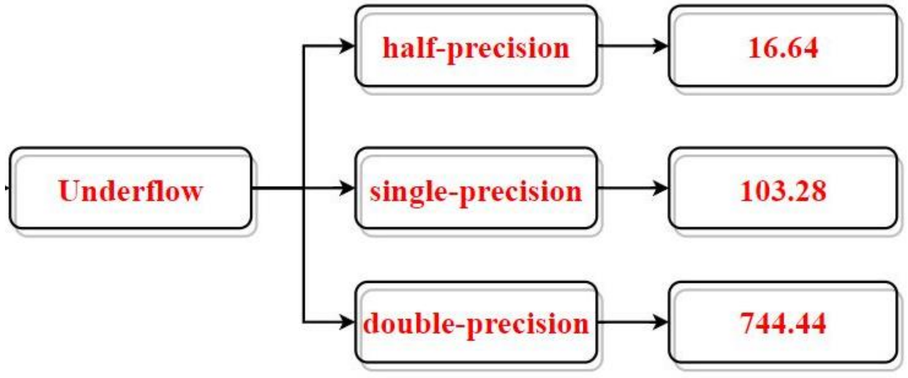
Motivation - Floating-point errors

$$CE(\boldsymbol{z}, y) = -log p_y = -log \frac{e^{\boldsymbol{z}_y - \boldsymbol{z}_{\pi 1}}}{\sum_{i=1}^{K} e^{\boldsymbol{z}_i - \boldsymbol{z}_{\pi 1}}}$$

where $p_i = \frac{e^{\boldsymbol{z}_i - \boldsymbol{z}_{\pi 1}}}{\sum_{j=1}^{K} e^{\boldsymbol{z}_j - \boldsymbol{z}_{\pi 1}}}, i \in \{1,2, \dots, K\}.$

$$\nabla_{\widehat{\mathbf{X}}} \mathrm{CE}(\boldsymbol{z}, y) = (-1 + p_y) \nabla_{\widehat{\mathbf{X}}} (\boldsymbol{z}_y - \boldsymbol{z}_{\pi 1}) + \sum_{i \neq y} p_i \nabla_{\widehat{\mathbf{X}}} (\boldsymbol{z}_i - \boldsymbol{z}_{\pi 1})$$

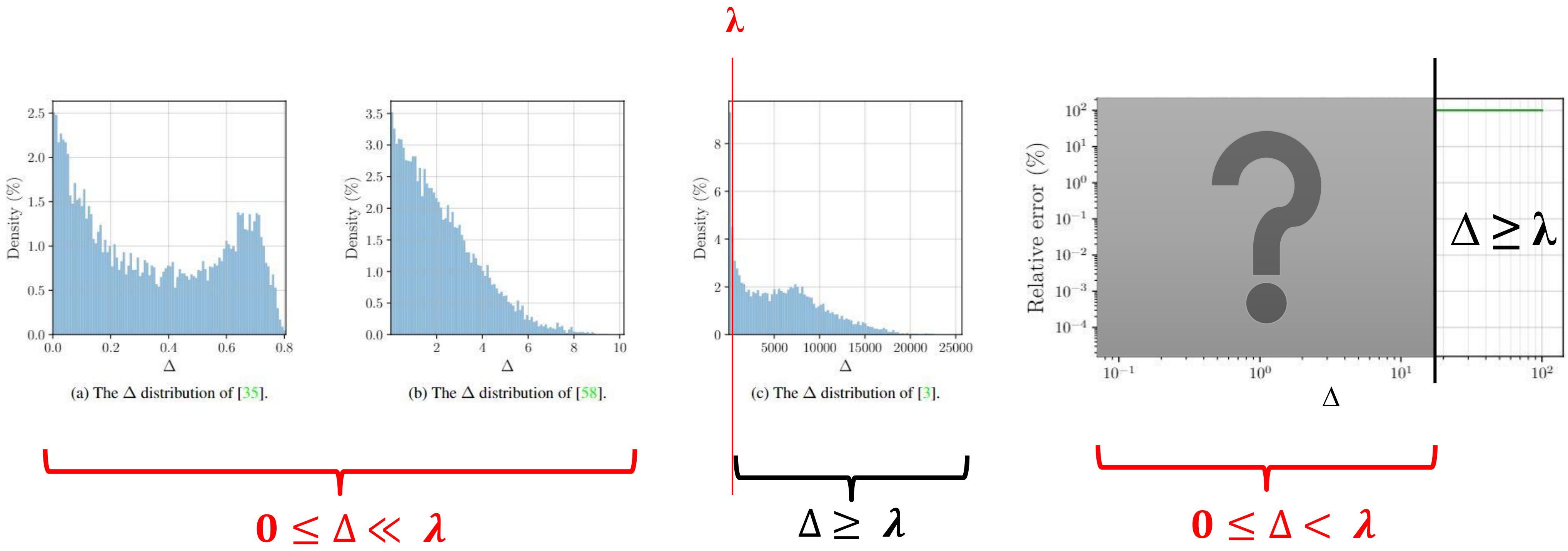
$\boldsymbol{e^{-\lambda}}$



when $\Delta = \boldsymbol{z}_{\pi 1} - \boldsymbol{z}_{\pi 2} \geq \boldsymbol{\lambda}$ and $\boldsymbol{z}_y = \boldsymbol{z}_{\pi 1}$ ➡ $p_y = 1, p_{i \neq y} = 0$ ➡ $\nabla_{\hat{x}} CE\ (\boldsymbol{z}, y) = 0$

The relative error of the calculated gradient $\delta_{CE} = \delta(\nabla_{\hat{x}} CE\ (\boldsymbol{z}, y)) = 100\%.$

Underflow can not explain all cases



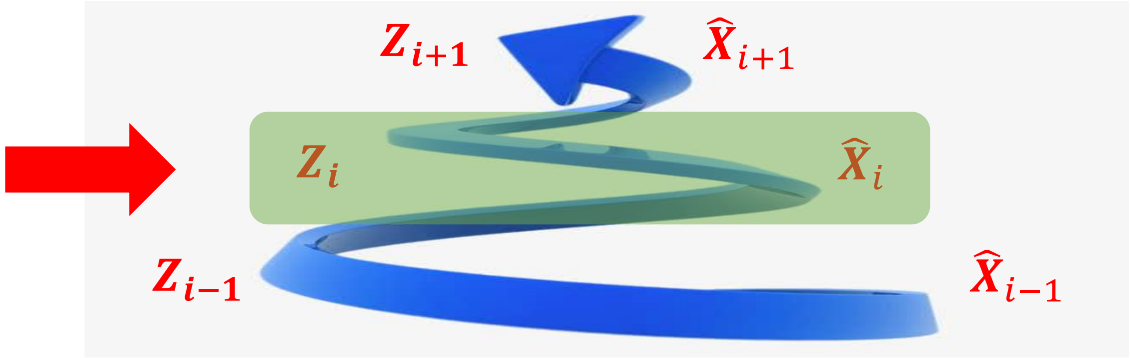
(a) The Δ distribution of [35]. (b) The Δ distribution of [58]. (c) The Δ distribution of [3].

But not all values of Δ in the model are greater than λ , **so what does the relative error δ_{CE} look like when $0 \leq \Delta < \lambda$?**

Rounding error

when $\boldsymbol{z}_y = \boldsymbol{z}_{\pi 1}$

$\nabla_{\widehat{X}}\mathrm{CE}(\boldsymbol{z}, y) = \sum_{i \neq y} p_i \nabla_{\widehat{X}}(\boldsymbol{z}_i - \boldsymbol{z}_{\pi 1})$



After we introduce a scaling factor c

where $c = T/\Delta_{detach}$

$$\nabla_{\widehat{X}}\mathrm{CE}(c\boldsymbol{z}, y) = c \sum_{i \neq y} p_i^c \nabla_{\widehat{X}}(\boldsymbol{z}_i - \boldsymbol{z}_{\pi 1})$$

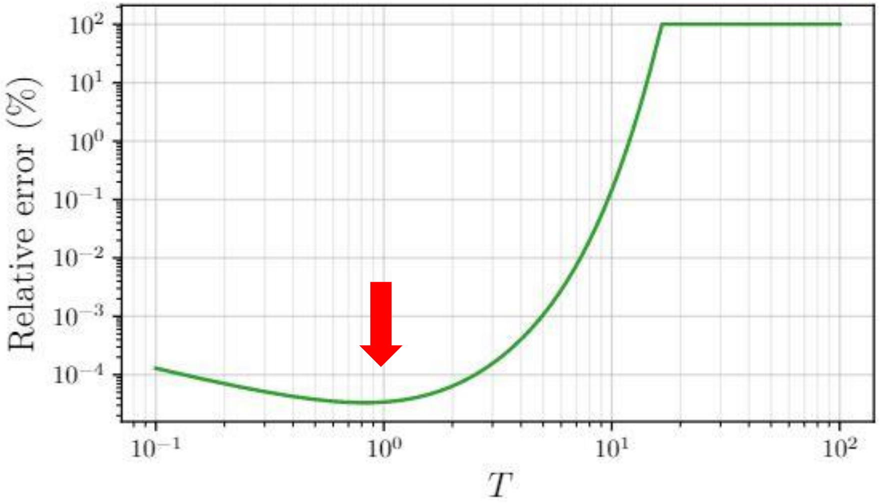
where $p_i^c = \frac{e^{c(\boldsymbol{z}_i - \boldsymbol{z}_{\pi 1})}}{\sum_{j=1}^{K} e^{c(\boldsymbol{z}_j - \boldsymbol{z}_{\pi 1})}}, i \in \{1,2,\ldots,K\}$ and $i \neq y$

Strong correlation

- when $K = 2$

$$\nabla_{\hat{X}}\mathrm{CE}(c\mathbf{z}, y) = cp_2^c\, \nabla_{\hat{X}}(\mathbf{z}_{\pi 2} - \mathbf{z}_{\pi 1}) \propto cp_2^c$$

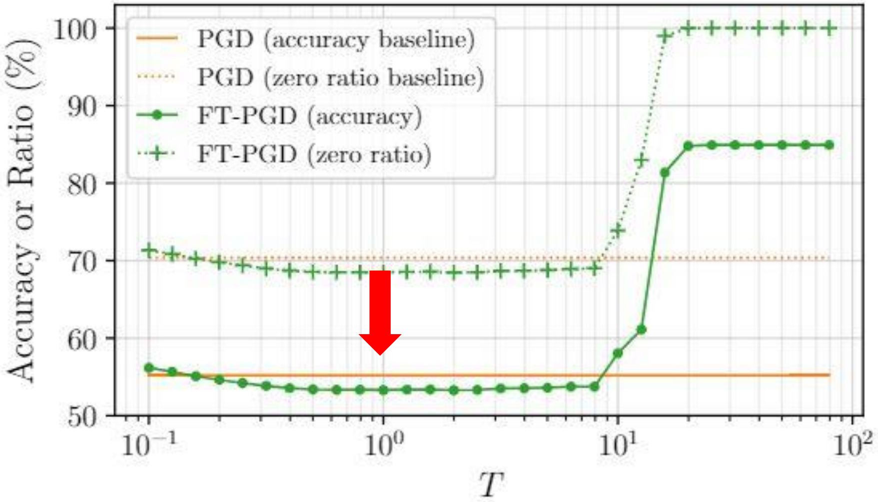
$$\delta_{CE} \propto \delta(cp_2^c)$$



- Following the same operation, we add a scale factor c to Δ and hold $\mathrm{T} = c\Delta$ constant during each iteration of the multi-iteration attack

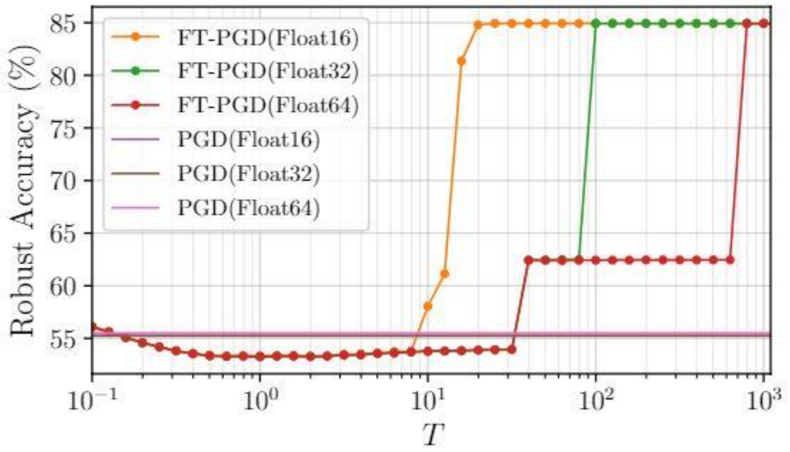
$$\mathcal{L}^{\mathrm{MIFPE}}(\mathbf{z}, y) \triangleq \mathcal{L}^{\mathrm{ce}}\left(T\mathbf{z}/\Delta_{\mathrm{detach}}, y\right),$$

$$\mathcal{L}^{\mathrm{MIFPE}}_{\mathrm{target}}(\mathbf{z}, y_t) = -\mathcal{L}^{\mathrm{ce}}\left(T\mathbf{z}/\Delta_{\mathrm{detach}}, y_t\right),$$



Other solutions

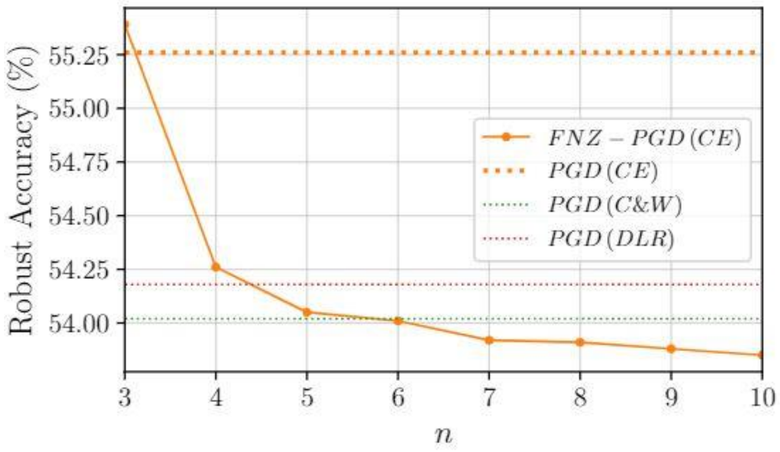
- Increasing the floating-point precision



- Surrogate loss functions

$$\mathcal{L}^{\text{cw}}(\mathbf{z},\mathrm{y}) = -\mathbf{z}_y + \max_{i \neq y} \mathbf{z}_i,$$

$$\mathcal{L}^{\text{dlr}}(\mathbf{z},\mathrm{y}) = \frac{-\mathbf{z}_y + \max_{i \neq y} \mathbf{z}_i}{\mathbf{z}_{\pi 1} - \mathbf{z}_{\pi 3}},$$

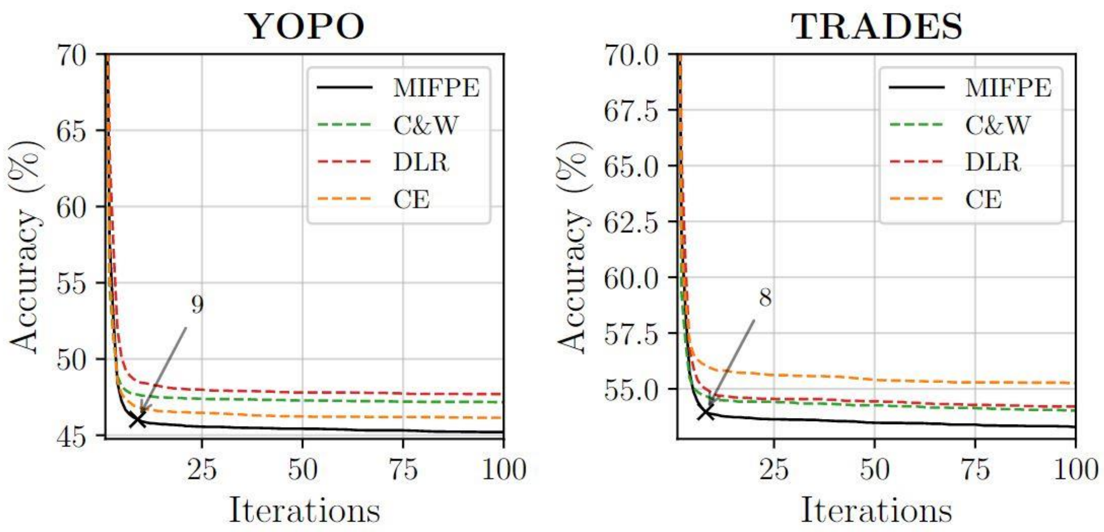


Experiment

Defense method	Architecture	Clean	CE ($\mathcal{L}^{\text{sce}}$) 100	C&W ($\mathcal{L}^{\text{cw}}$) 100		DLR ($\mathcal{L}^{\text{dlr}}$) 100		GAMA_PGD 100		MIFPE ($\mathcal{L}^{\text{LNSCE}}$) 100		Best 4900
MNIST, ℓ_∞, $\varepsilon = 0.3$												
Uncovering limits [123]	WRN-28-10	99.26	96.55	96.64	(+0.09)	96.71	(+0.16)	96.69	(+0.14)	**96.53**	(−0.02)	96.31
MMA training [35] †	LeNet5Madry	98.98	95.66	95.60	(-0.06)	95.56	(-0.10)	95.96	(+0.13)	**95.50**	(−0.16)	93.51
MMA training [35]	LeNet5Madry	98.95	95.09	95.33	(+0.24)	95.59	(+0.50)	95.74	(+0.19)	**94.88**	(−0.21)	91.40
Neural level sets [15]	SmallCNN	99.35	99.28	94.68	(-4.60)	95.09	(-4.19)	99.29	(+0.01)	**94.67**	(−4.61)	90.85
TRADES [14]	SmallCNN	99.48	93.69	93.88	(+0.19)	94.49	(+0.80)	93.82	(+0.13)	**93.67**	(−0.02)	92.71
Robust optimization [1]	SmallCNN	99.35	93.06	93.19	(+0.13)	93.63	(+0.57)	93.39	(+0.33)	**92.88**	(−0.18)	90.85
Fast adversarial training [26]	SmallCNN	98.50	86.82	86.96	(+0.14)	87.42	(+0.60)	87.62	(+0.80)	**86.57**	(−0.25)	82.93
CIFAR-10, ℓ_∞, $\varepsilon = 8/255$												
Uncovering limits [123]†	WRN-70-16	91.10	67.96	66.70	(-1.26)	66.78	(-1.18)	66.08	(-1.88)	**65.96**	(−2.00)	65.87
Fixing data augmentation [131]	WRN-106-16	88.50	67.57	65.55	(-2.02)	65.61	(-1.96)	64.94	(-2.63)	**64.75**	(−2.82)	64.58
Fixing data augmentation [131]	WRN-70-16	88.54	67.27	65.23	(-2.04)	65.32	(-1.95)	64.57	(-2.70)	**64.46**	(−2.81)	64.20
Proper definition [133]	WRN-70-16	89.01	66.66	63.94	(-2.72)	64.01	(-2.65)	63.65	(-3.01)	**63.49**	(−3.17)	63.35
Uncovering limits [123] †	WRN-28-10	89.48	65.59	63.62	(-1.97)	63.82	(-1.77)	63.05	(-2.90)	**62.96**	(−2.63)	62.76
Proper definition [133]	WRN-28-10	88.61	64.66	61.55	(-3.11)	61.62	(-3.04)	61.19	(-3.47)	**61.12**	(−3.54)	61.04
Adversarial weight perturbation [32]†	WRN-28-10	88.25	63.18	60.51	(-2.67)	60.60	(-2.58)	60.18	(-3.00)	**60.09**	(−3.09)	60.04
Unlabeled data [22]†	WRN-28-10	89.69	61.60	60.47	(-1.13)	60.67	(-0.93)	59.82	(-1.78)	**59.72**	(−1.88)	59.53
HYDRA [7]†	WRN-28-10	88.98	59.53	58.21	(-1.32)	58.30	(-1.23)	57.52	(-2.01)	**57.38**	(−2.15)	57.14
Misclassification-aware [25]	WRN-28-10	87.50	61.60	58.03	(-3.57)	58.73	(-2.87)	57.20	(-4.40)	**56.88**	(−4.72)	56.29
Pre-training [24]†	WRN-28-10	87.11	57.07	56.27	(-0.80)	57.07	(0.00)	55.22	(-1.85)	**55.10**	(−1.97)	54.92
Hypersphere embedding [31]	WRN-34-20	85.14	61.43	55.35	(-6.08)	56.21	(-5.22)	54.37	(-7.06)	**53.85**	(−7.58)	53.74
Overfitting [33]	WRN-34-20	85.34	56.85	55.22	(-1.63)	55.97	(-0.88)	53.87	(-2.98)	**53.62**	(−3.23)	53.42
Self-adaptive training [104]‡	WRN-34-10	83.48	56.12	54.30	(-1.82)	54.73	(-1.39)	53.64	(-2.48)	**53.48**	(−2.64)	53.34
TRADES [14]‡	WRN-34-10	84.92	55.21	53.94	(-1.27)	54.11	(-1.10)	53.38	(-1.83)	**53.22**	(−1.99)	53.08
Robustness library [34]	RN-50	87.03	51.56	52.07	(+0.51)	52.81	(+1.25)	50.04	(-1.52)	**49.84**	(−1.72)	49.25
Neural level sets [15]‡	RN-18	81.30	79.12	40.07	(-39.05)	45.10	(-34.02)	79.69	(+0.57)	**40.06**	(−39.06)	39.77
YOPO [30]	WRN-34-10	87.20	46.05	47.02	(+0.97)	47.55	(+1.50)	45.30	(-0.75)	**45.19**	(−0.86)	44.83
Fast adversarial training [26]	RN-18	83.34	45.75	45.81	(+0.06)	46.89	(+1.14)	43.71	(-2.04)	**43.57**	(−2.18)	43.21

Ablation study

- Convergence speed

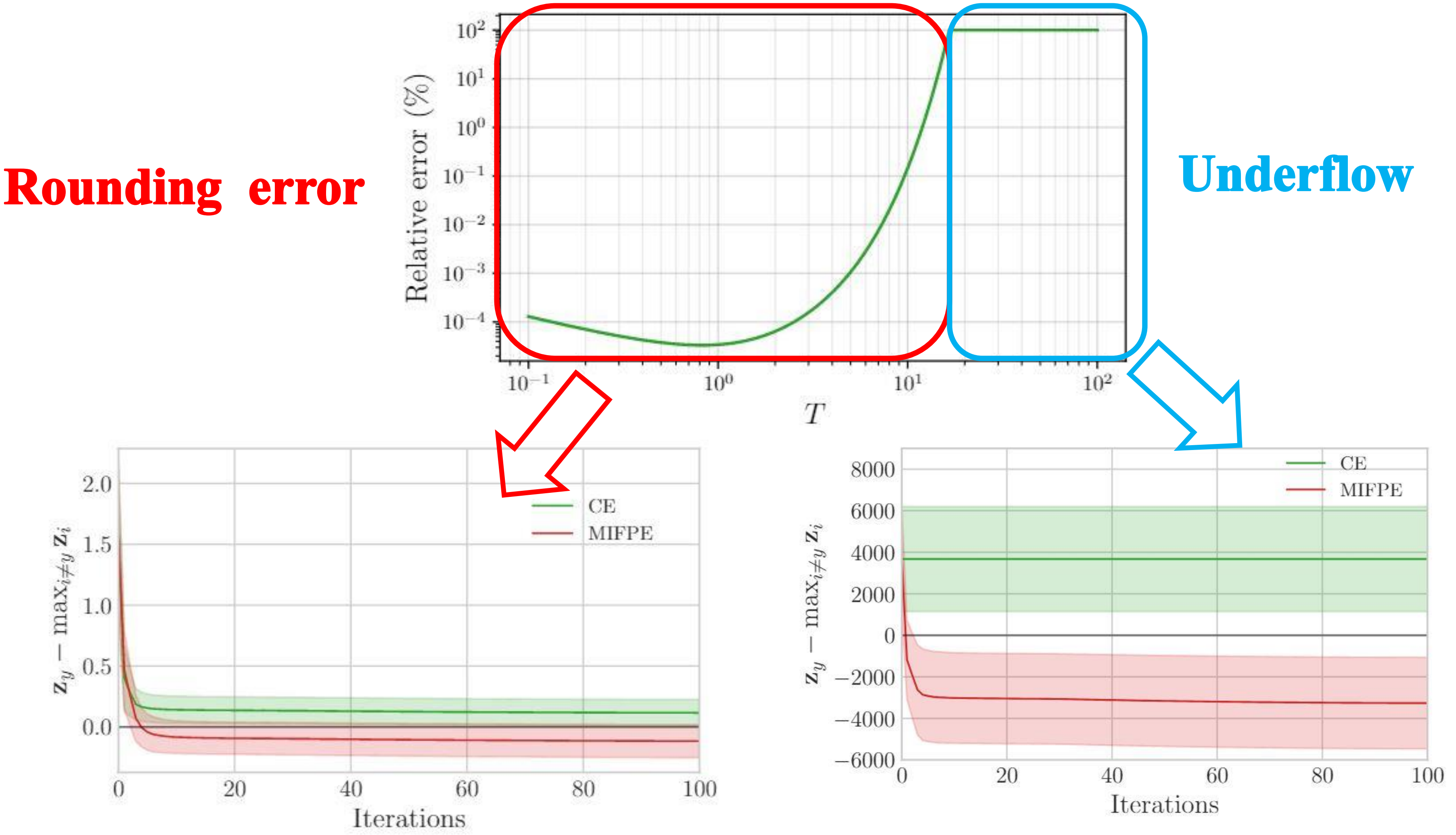


- Boost the capability of existing attack strategies

Attack iteration	FGSM 1	PGD 100	APGD_DLR 100	AA 4900
Original	79.83	79.79	45.90	40.22
MIFPE	**49.76**	**40.06**	**40.49**	**39.89**
∇	30.07	39.73	5.41	0.33

Ablation study

$\mathbb{Z} = z_y - \max_{i \neq y} z_i$ Its sign indicates if the attack is successful or not.



Thanks for listening!