Understanding Robust Overfitting of Adversarial Training and Beyond

Chaojian Yu[1], Bo Han[2], Li Shen[3], Jun Yu[4], Chen Gong[5], Mingming Gong[6], Tongliang Liu[1]

[1]TML Lab, Sydney AI Centre, The University of Sydney

[2]Department of Computer Science, Hong Kong Baptist University

[3]JD Explore Academy

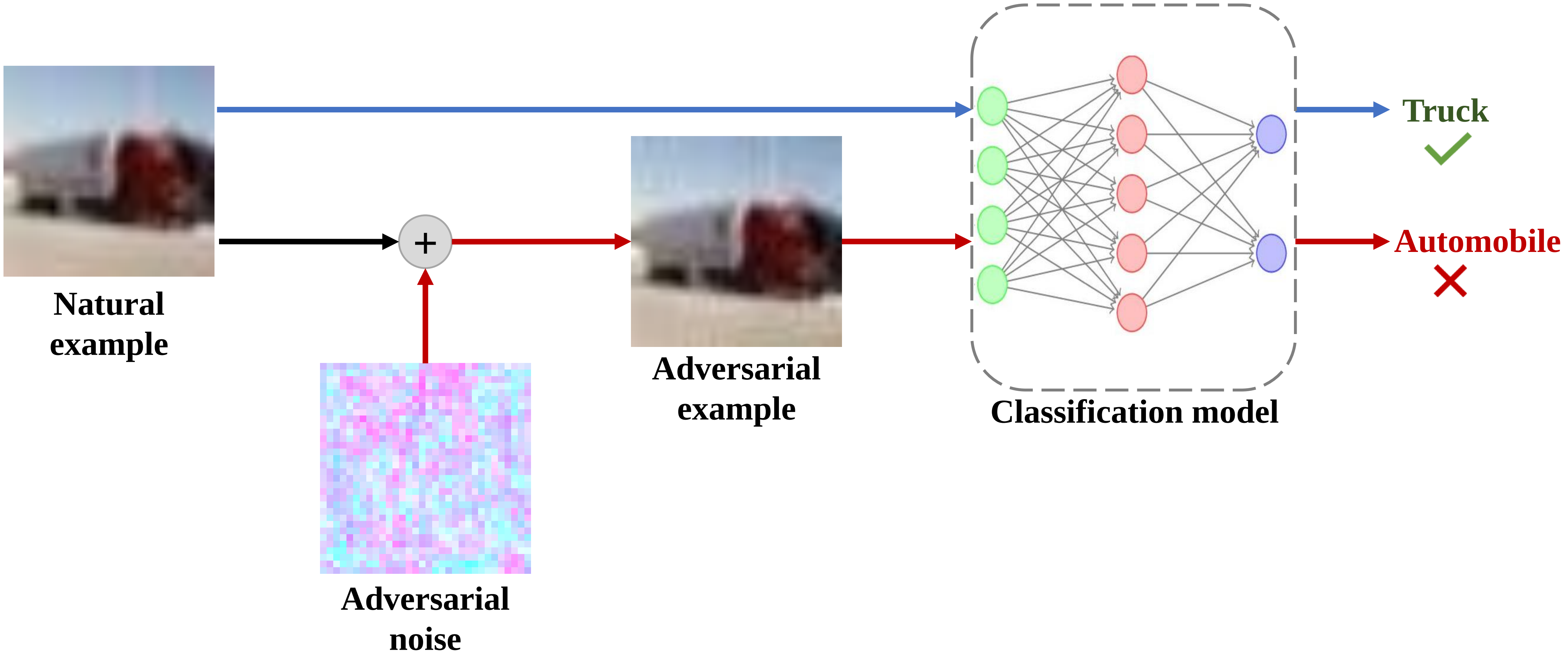
[4]Department of Automation, University of Science and Technology of China

[5]School of Computer Science and Engineering, Nanjing University of Science and Technology

[6]School of Mathematics and Statistics, The University of Melbourne


Machine Learning And AI Is Everywhere.

Deep Neural Networks are vulnerable to adversarial examples.

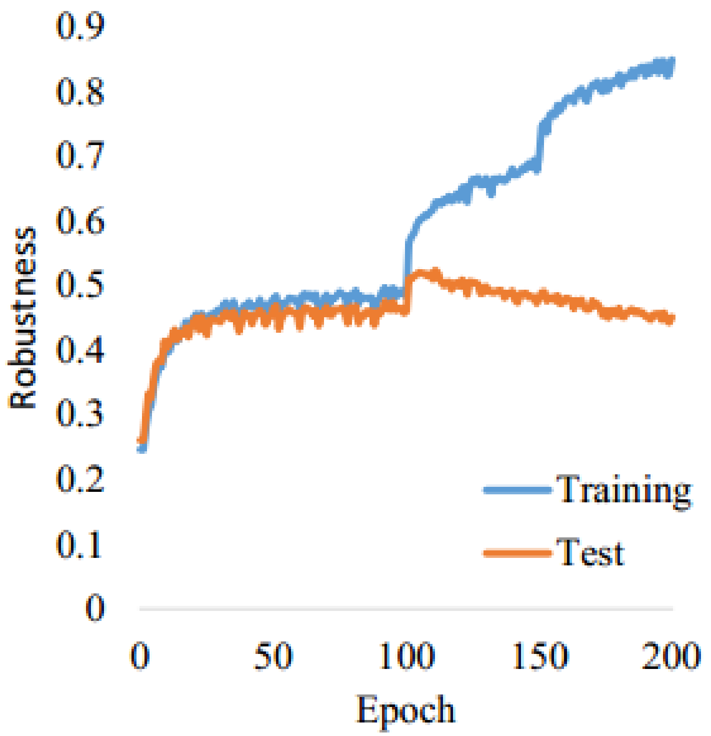


Introduction

- Adversarial training (AT), one of the most effective defenses, can be formulated as a min-max optimization problem:

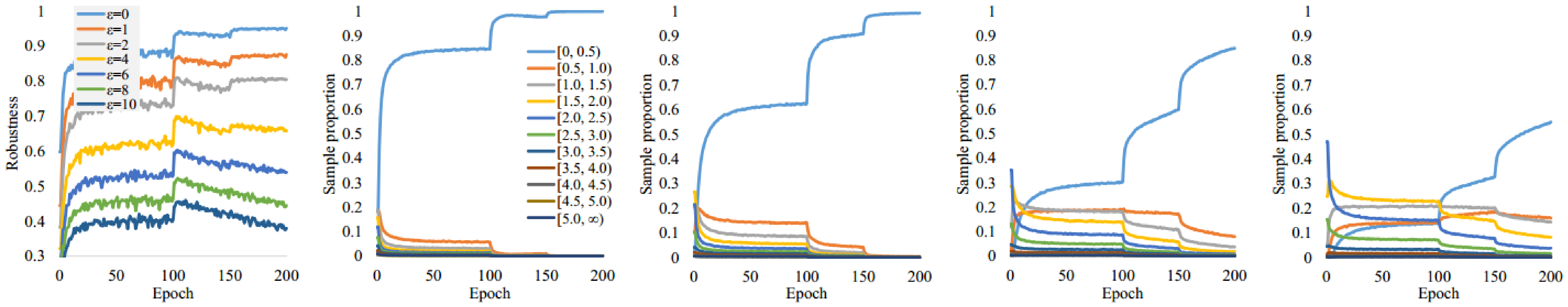
$$\min_{w} \frac{1}{n} \sum_{i=1}^{n} \max_{||x_i' - x_i||_p \leq \epsilon} \ell(f_w(x_i'), y_i)$$

- Robust overfitting: the robust accuracy on test data will continue to degrade with further training. *The underlying reasons for this are still not completely understood*.



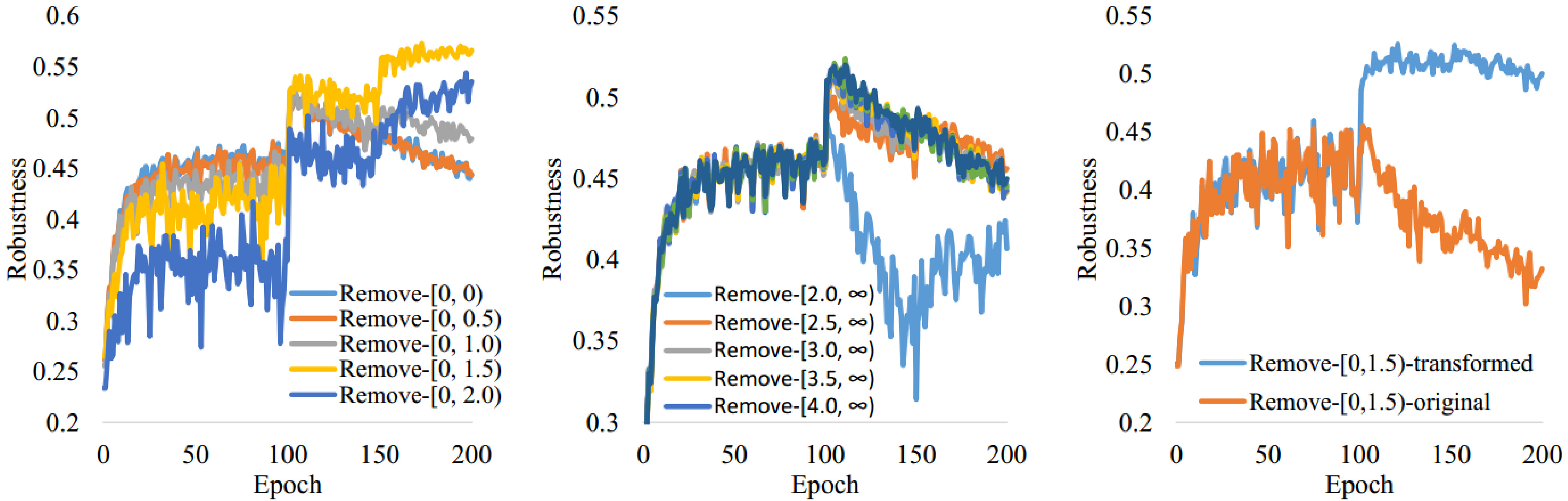
[1] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In ICLR, 2018.
[2] Rice, L., Wong, E., and Kolter, J. Overfitting in adversarially robust deep learning. In ICML, 2020.

Data Distribution Perspective



- The data distribution of overfitted AT is mismatched with that of non-overfit AT.
- Q1: if we suppress the large-loss data in overfitted AT to align the data distribution of non-overfit AT, will it eliminate robust overfitting?
- Q2: if we suppress the small-loss data in overfitted AT that does not match the strength of adversary, will it eliminate robust overfitting?

Causes of Robust Overfitting



- Removing large-loss data: aligning to the data distribution of non-overfit AT is invalid.
- Removing small-loss data: identifying that some small-loss data cause robust overfitting.
- Explanation: network becomes more robust as the adversarial training progresses, making some generated adversarial data relatively less aggressive, and when their loss drops to a certain level, these adversarial data eventually lead to robust overfitting.

MLCAT Prototype

- Learn large-loss data as usual.
- Adopt additional measure to increase the loss of the small-loss data.
- Versatile: loss adjustment strategy S and minimum loss condition ℓ_{min} can be flexibly implemented depend on base AT algorithm.

Turning waste into treasure

Algorithm 1 MLCAT-prototype (in a mini-batch).

Require: base adversarial training algorithm $\mathcal{A}$, optimizer $\mathfrak{O}$, network f_w, training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, mini-batch $\mathcal{B}$, batch size m, minimum loss conditions ℓ_{min} for $\mathcal{A}$, loss adjustment strategy $\mathcal{S}$

Sample a mini-batch $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{m}$ from $\mathcal{D}$
$\mathcal{B}' = \mathcal{A}$.inner_maximization$(f_w, \mathcal{B})$
$\{\ell_i\}_{i=1}^{m} \leftarrow \ell(f_w, \mathcal{B}')$
$\ell_{\mathcal{B}'} \leftarrow 0$ # initialize loss accumulator
for $i = 1, \ldots, m$ **do**
  if $\ell_i \geq \ell_{min}$ **then**
    $\ell_{\mathcal{B}'} = \ell_{\mathcal{B}'} + \ell_i$
  else
    $\ell_i^{\mathcal{S}} \leftarrow \mathcal{S}(f_w, x_i', \ell_{min})$ # adjust loss
    $\ell_{\mathcal{B}'} = \ell_{\mathcal{B}'} + \ell_i^{\mathcal{S}}$ # accumulate adjusted loss
  end if
end for
$\ell_{\mathcal{B}'} \leftarrow \ell_{\mathcal{B}'} / m$ # average accumulated loss
$\nabla_w \leftarrow \mathcal{A}$.outer_minimization$(f_w, \ell_{\mathcal{B}'})$
$\mathfrak{O}$.step(∇_w)

Two Realizations of MLCAT

- Loss Scaling (MLCAT_{LS}): create a corrected loss from original loss and then trains the network based on the corrected loss.

$$\ell_i^S = \frac{\ell_{min}}{\ell_i} \cdot \ell_i = \ell_{min}$$

- Weight Perturbation (MLCAT_{WP}): generate perturbation to the model weights, and trains the network on the perturbative parameters.

$$v = \nabla_w \sum_i \mathbb{1}(\ell_i \leq \ell_{min})\, \ell_i$$

$$\ell_i^S = \ell(f_{w+v}(x_i'), y_i)$$

[1] Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassifed examples. In ICLR, 2020.
[2] Wu, D., Xia, S., and Wang, Y. Adversarial Weight Perturbation Helps Robust Generalization. In NeurIPS, 2020.

Experiments

Table 1. Test robustness (%) on CIFAR10. We omit the standard deviations of 5 runs as they are very small ($< 0.6\%$).

Network	Threat Model	Method	PGD-20 Best	PGD-20 Last	PGD-20 Diff	AA Best	AA Last	AA Diff
PreAct ResNet-18	L_∞	AT	52.29	44.43	-7.86	47.99	42.08	-5.91
		MLCAT_{LS}	56.90	56.87	**-0.03**	28.12	26.93	-1.19
		MLCAT_{WP}	**58.48**	**57.65**	-0.83	**50.70**	**50.32**	**-0.38**
	L_2	AT	69.27	65.86	-3.41	67.70	64.64	-3.06
		MLCAT_{LS}	73.16	72.48	-0.68	49.7	48.94	-0.76
		MLCAT_{WP}	**74.38**	**73.86**	**-0.52**	**70.46**	**70.15**	**-0.31**
Wide ResNet-34-10	L_∞	AT	55.57	47.37	-8.20	52.13	46.09	-6.04
		MLCAT_{LS}	**64.73**	**63.94**	-0.79	35.00	34.51	-0.49
		MLCAT_{WP}	62.50	61.91	**-0.59**	**54.65**	**54.56**	**-0.09**
	L_2	AT	71.57	69.99	-1.58	70.44	68.92	-1.52
		MLCAT_{LS}	75.05	74.97	**-0.08**	55.31	55.11	**-0.20**
		MLCAT_{WP}	**76.92**	**76.55**	-0.37	**74.35**	**73.97**	-0.38

Experiments

Table 3. Test robustness (%) on CIFAR100. We omit the standard deviations of 5 runs as they are very small (< 0.6%).

Network	Threat Model	Method	PGD-20 Best	PGD-20 Last	PGD-20 Diff	AA Best	AA Last	AA Diff
PreAct ResNet-18	L_∞	AT	28.01	20.39	-7.62	23.61	18.41	-5.20
		$\mathrm{MLCAT}_{\mathrm{LS}}$	20.09	18.14	-1.95	13.41	11.35	-2.06
		$\mathrm{MLCAT}_{\mathrm{WP}}$	**31.27**	**30.57**	**-0.70**	**25.66**	**25.28**	**-0.38**
	L_2	AT	41.38	35.34	-6.04	37.94	33.58	-4.36
		$\mathrm{MLCAT}_{\mathrm{LS}}$	31.23	30.80	**-0.43**	22.06	21.72	-0.34
		$\mathrm{MLCAT}_{\mathrm{WP}}$	**45.49**	**44.84**	-0.65	**41.22**	**41.15**	**-0.07**
Wide ResNet-34-10	L_∞	AT	30.74	24.89	-5.85	26.98	23.07	-3.91
		$\mathrm{MLCAT}_{\mathrm{LS}}$	22.86	22.18	-0.68	14.61	14.05	-0.56
		$\mathrm{MLCAT}_{\mathrm{WP}}$	**34.97**	**34.64**	**-0.33**	**29.49**	**29.25**	**-0.24**
	L_2	AT	44.12	41.29	-2.83	41.39	39.34	-2.05
		$\mathrm{MLCAT}_{\mathrm{LS}}$	34.09	33.66	**-0.43**	25.06	24.31	-0.75
		$\mathrm{MLCAT}_{\mathrm{WP}}$	**50.17**	**49.51**	-0.66	**46.05**	**45.77**	**-0.28**