

# Supplementary Material: Robustness Learning via Decision Tree Search Robust Optimisation

Yi-Ling Liu  
y.liu17@imperial.ac.uk

Imperial College London  
United Kingdom

Alessio Lomuscio  
a.lomuscio@imperial.ac.uk

In the supplementary material we provide:

- details about training network architectures,
- full implementation results with other attack approaches in terms of different metrics,
- training performance analysis with different algorithms,
- adversarial examples by DTS approach with various confidence values,
- Transferability analysis for FGSM, PGD and DTS.

## A Network Architecture

To investigate model capacity, we consider two training networks of simple and wide architectures for MNIST dataset [8], respectively. The simple network consists of two convolution layers of sizes 32 and 64 filters, and a fully connected layer of size 1024. The wide network consists of two convolution layers of sizes 64 and 128 filters, and also a fully connected layer of size 1024. Both networks are adversarially trained with FGSM, PGD and DTS methods. The test errors for these two network architectures are 2.3% and 1.8% respectively, and the detailed parameters for each layer are summarised in the first two rows of Table 1.

For the CIFAR10 dataset [9], we used the Resnet model [10] with different filters as the simple and wide networks. The network width is determined by factor  $k$  and the groups of convolutions are shown in brackets where  $N$  is the number of blocks in a group. For the simple network, the factor  $k$  is set as 1 while the number of blocks  $N$  is set as 5. For the wide network, a modified one is implemented via wider layers by a factor, resulting in a network of (16, 160, 320, 640) filters with 5 residual units in each convolution, where the factor  $k$  is set as 10. The resulting ResNet28-10 model [9] is adopted by an error rate of 4.17% and with 36.5M parameters at most. We also perform adversarial training with FGSM, PGD and DTS methods on these two architectures and investigated the resulting accuracies against white-box attack adversaries on different adversarial trained models. The details are enumerated in the last two rows of Table 1.

Dataset	Network Name	Network Architecture	Test Error
MNIST [8]	Simple Network (S. (MDL Training))	Conv1( [5, 5, 1, 32], 2×2) Conv2( [5, 5, 32, 64], 2×2) Full( [7 × 7 × 64, 1024]) Avg-pool( [-1, 7 × 7 × 64])	2.3
	Wide Network (W. (MDL Training))	Conv1( [5, 5, 1, 64], 2×2) Conv2( [5, 5, 64, 128], 2×2) Full( [7 × 7 × 128, 1024]) Avg-pool( [-1, 7 × 7 × 128])	1.8
CIFAR10 [9]	Simple Network (S. (MDL Training))	Conv1( [3 × 3, 16]) Conv2( $\begin{bmatrix} 3 \times 3 & 16 \times k \\ 3 \times 3 & 16 \times k \end{bmatrix} \times N, k:1 N:5$ ) Conv3( $\begin{bmatrix} 3 \times 3 & 32 \times k \\ 3 \times 3 & 32 \times k \end{bmatrix} \times N$ ) Conv4( $\begin{bmatrix} 3 \times 3 & 64 \times k \\ 3 \times 3 & 64 \times k \end{bmatrix} \times N$ ) Avg-pool( [8 × 8])	4.9
	Wide Network (W. (MDL Training))	Conv1( [3 × 3, 16]) Conv2( $\begin{bmatrix} 3 \times 3 & 16 \times k \\ 3 \times 3 & 16 \times k \end{bmatrix} \times N, k:10 N:5$ ) Conv3( $\begin{bmatrix} 3 \times 3 & 32 \times k \\ 3 \times 3 & 32 \times k \end{bmatrix} \times N$ ) Conv4( $\begin{bmatrix} 3 \times 3 & 64 \times k \\ 3 \times 3 & 64 \times k \end{bmatrix} \times N$ ) Avg-pool( [8 × 8])	4.2

Table 1: The full network architectures and parameters adopted for MNIST and CIFAR10. The MDL is represented as one of the FGSM, PGD and DTS. For MNIST dataset, the simple network consists of two convolution layers of sizes 32 and 64 filters and the wide network consists of two convolution layers of sizes 64 and 128 filters, following a fully connected layer of size 1024. For CIFAR10, the Resnet28-10 model [9] is implemented with an error rate of 4.17%. The network width is determined by factor k and the groups of convolutions are shown in brackets where N is a number of blocks in group [9].

In all our experiments we use Stochastic Gradient Descent (SGD) with momentum and cross-entropy loss. The initial learning rate is set to 0.1, weight decay to 0.0005, dampening to 0, momentum to 0.9 and minibatch size to 128. On CIFAR learning rate dropped by 0.1 at 10,000, 15,000 and 20,000 epochs and we train for total 30,000 epochs. We select the ResNet28-10 model [9] as it outperforms the original ResNet [10] by 0.92% (with the same minibatch size during training) on CIFAR-10.

## B Full Experiment Results

We report the detailed results obtained with the MNIST [8] and CIFAR10 [9] datasets. We evaluate the robustness obtained against different attack methods, namely FGSM, PGD, and

DTS. As mentioned previously, we first perform adversarial training with FGSM, PGD and DTS methods on two network architectures and investigate the resulting accuracies against white-box attack adversaries on different adversarial trained models. The experimental results are shown in Table 2 for MNIST and Table 3 for CIFAR10.

**MNIST.** We generated adversarial examples using the white-box attack method of FGSM, PGD and DTS with  $\epsilon_{\text{DIFF}} = 0.02$  and  $0.03$ , and then evaluated them on a target network, that was adversarially trained independently, with different methods. The target models consist of two different architectures, which are simple and wide networks. The first column (Nature) stands for the accuracy of each adversarially trained target network without attacks. The FGSM random attack (FGSM-R) was implemented according to [9], whereby small random perturbations are performed before applying FGSM. The PGD attack was evaluated using settings of 2 and 5 steps with step size 0.01 according to the constraints of  $\epsilon_{\text{DIFF}}$  to 0.02 and 0.03. The search time for DTS attack is constrained in 1 second and 2 seconds respectively; once an adversarial example is found, the process will be terminated as aforementioned.

From the results, the attack strength from strong to weak is DTS, PGD and then FGSM as the accuracies against DTS attack (DTS-X columns) are lower than PGD and FGSM columns. This means that DTSATTACK is a stronger attack, and thus the resulting accuracies among different trained networks are lower. In addition, the results show that DTS contributes to improved accuracies against different adversaries and maintains robustness even under the DTS attack itself. For example, the accuracy of S. (DTS training network) is 93.8, which is higher than other training networks 92.7 and 93.6 under the PGD attack (column PGD-2). Also, the accuracy of S. (DTS training network) under the DTS attack itself (column DTS-1) is 91.9, which is higher than other networks (e.g., S. (PGD training network) or S. (FGSM training network)). Moreover, changing the architecture from simple to wide networks also contributes to accuracies overall. For instance, the results of the fourth to sixth rows are higher than that of the first to third rows. Furthermore, the accuracies decrease only few percentages when increasing  $\epsilon_{\text{DIFF}}$  to 0.03 and the results from the seventh to last rows demonstrate this compared with the results in 0.02. In addition, S. (DTS training) and W. (DTS training) under  $\epsilon_{\text{DIFF}} = 0.03$  are more robust than S (PGD Training) and W (PGD Training) in general. This means that DTS remains robust while increasing the total perturbations. Also, the average distances in norm metrics of  $l_0$ ,  $l_1$  and  $l_2$  are reported in the last few columns of Table 2. The average distances required for DTS are smaller than the others as DTS searches mainly the most potential features. With these advantages, DTS does save training efforts during adversarial training.

**CIFAR10.** Table 3 summarises the resulting accuracies obtained on the CIFAR10 dataset. We generated adversarial examples using the same methods as previous with  $\epsilon_{\text{DIFF}} = 0.02$  and  $0.03$ , and then evaluated them on a target network, independently trained via the FGSM, PGD and DTS methods. From the results, the attack strength from strong to weak is DTS, PGD and then FGSM as aforementioned. Also, the results show that DTS contributes to improved accuracies against different adversaries and maintains robustness under the DTS attack itself. For example, the accuracy of W. (DTS training network) under the DTS attack itself (column DTS-2) is 91.5, which is higher than other networks with 91.0 or 89.5 (e.g., W. (PGD training network) or W. (FGSM training network)) with  $\epsilon_{\text{DIFF}} = 0.02$ . The results obtained also demonstrate that a strong adversary can help to improve model accuracies. Moreover, Comparing the results against different network architectures reveals that changing the architecture from simple to wide networks can contribute to accuracies generally. For instance, the results of the fourth to sixth rows are higher than that of the first to third rows. Furthermore, the accuracies decrease few percentages when increasing  $\epsilon_{\text{DIFF}}$  to 0.03, similar

Adversary Target Model	Nature	FGSM	FGSM-R	PGD-2	PGD-5	DTS-1	DTS-2	$l_0$	$l_1$	$l_2$	$\epsilon_{\text{DIFF}}$
S. (FGSM Training)	96.2	95.1	94.8	92.7	91.4	90.8	89.6	13.9	12.7	3.87	0.02
S. (PGD Training)	95.8	95.2	94.2	93.6	92.7	91.5	90.8	13.5	12.1	3.71	
S. (DTS Training)	<b>97.7</b>	<b>96.3</b>	<b>95.1</b>	<b>93.8</b>	<b>93.3</b>	<b>91.9</b>	<b>91.6</b>	11.8	11.4	3.27	
W. (FGSM Training)	97.4	96.7	96.4	94.6	94.1	92.8	92.3	13.7	13.4	3.95	
W. (PGD Training)	96.9	95.7	95.3	95.1	94.8	93.5	93.1	12.1	12.5	3.86	
W. (DTS Training)	<b>97.8</b>	<b>96.8</b>	<b>96.3</b>	<b>95.8</b>	<b>95.2</b>	<b>93.9</b>	<b>93.6</b>	12.4	12.1	3.79	
S. (PGD Training)	94.8	92.7	91.9	90.7	89.9	89.5	88.6	18.8	17.9	4.17	0.03
S. (DTS Training)	<b>96.5</b>	<b>94.1</b>	<b>93.2</b>	<b>92.7</b>	<b>92.3</b>	<b>90.5</b>	<b>90.1</b>	17.9	17.3	4.04	
W. (PGD Training)	96.1	92.8	92.1	91.3	90.3	90.1	89.3	19.2	19.3	4.28	
W. (DTS Training)	<b>97.3</b>	<b>94.3</b>	<b>93.7</b>	<b>93.1</b>	<b>92.5</b>	<b>91.8</b>	<b>91.2</b>	18.9	17.6	4.21	

Table 2: The resulting accuracy of nature training, FGSM, PGD and DTS methods against white-box adversarial attacks with  $\epsilon_{\text{DIFF}} = 0.02$  and  $0.03$  on MNIST dataset.

Adversary Target Model	Nature	FGSM	FGSM-R	PGD-2	PGD-5	DTS-1	DTS-2	$l_0$	$l_1$	$l_2$	$\epsilon_{\text{DIFF}}$
S. (FGSM Training)	93.8	93.0	92.5	89.5	89.1	87.9	87.0	14.5	13.6	3.93	0.02
S. (PGD Training)	93.3	92.4	92.0	91.6	91.2	90.1	88.9	13.8	12.9	3.82	
S. (DTS Training)	<b>94.8</b>	<b>93.2</b>	<b>92.8</b>	<b>91.9</b>	<b>90.7</b>	<b>90.5</b>	<b>89.8</b>	12.3	11.3	3.74	
W. (FGSM Training)	94.2	93.9	93.3	91.4	90.8	90.3	89.5	14.2	13.9	3.91	
W. (PGD Training)	94.0	93.7	93.4	92.9	92.3	91.6	91.0	13.9	13.5	3.84	
W. (DTS Training)	<b>94.9</b>	<b>94.1</b>	<b>93.7</b>	<b>93.2</b>	<b>92.5</b>	<b>91.7</b>	<b>91.5</b>	12.4	11.9	3.79	
S. (PGD Training)	92.4	89.8	88.4	87.6	86.9	86.4	85.9	20.8	18.9	4.22	0.03
S. (DTS Training)	<b>93.2</b>	<b>92.5</b>	<b>91.8</b>	<b>90.7</b>	<b>89.7</b>	<b>88.6</b>	<b>87.4</b>	19.2	18.3	4.04	
W. (PGD Training)	93.4	90.1	88.9	88.1	87.4	87.1	86.2	20.3	19.2	4.34	
W. (DTS Training)	<b>93.8</b>	<b>92.5</b>	<b>92.1</b>	<b>91.4</b>	<b>90.2</b>	<b>89.3</b>	<b>88.6</b>	19.5	18.6	4.21	

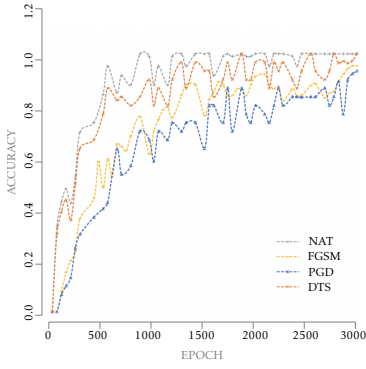
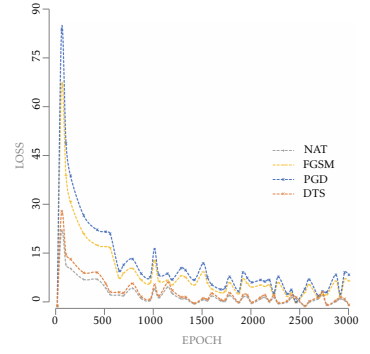
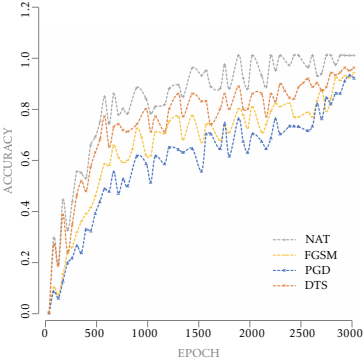
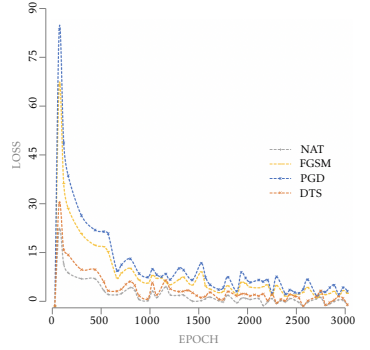
Table 3: The resulting accuracy of nature training, FGSM, PGD and DTS methods against white-box adversarial attacks with  $\epsilon_{\text{DIFF}} = 0.02$  and  $0.03$  on CIFAR10 dataset.

to the results of MNIST. In addition, S. (DTS training) and W. (DTS training) under  $\epsilon_{\text{DIFF}} = 0.03$  are more robust than S (PGD Training) and W (PGD Training) overall. This also shows that DTS remains robust while increasing the total perturbations. The average distances in norm metrics of  $l_0, l_1$  and  $l_2$  are also reported in the last few columns of Table 3. The average distances required for DTS are smaller than FGSM and PGD as DTS only searches for the most potential features. With these advantages, DTS also save training efforts during adversarial training in CIFAR10 dataset.

## C Performance Analysis

In this section, we compare the performance of adversarial training with different methods in Figure 1 and Figure 2. The figures show the accuracy and loss trends of nature, FGSM, PGD and DTS under the constraints of  $\epsilon_{\text{DIFF}} = 0.02$  and  $0.03$ .

**MNIST.** In Figure 1, we report the accuracy and loss trends for different adversarial training methods over the first 3,000 epochs with  $\epsilon_{\text{DIFF}} = 0.02$  in Figure 1(a) & (b) and  $0.03$  in Figure 1(c) & (d). In the constraints of  $0.02$ , the accuracies reach  $0.6$  before 500 epochs and DTS reach  $0.6$  around 300 epochs. The losses of different methods downgrade under  $30$

(a) Accuracy ( $\epsilon_{0.02}$ ).(b) Loss ( $\epsilon_{0.02}$ ).(c) Accuracy ( $\epsilon_{0.03}$ ).(d) Loss ( $\epsilon_{0.03}$ ).Figure 1: MNIST accuracy and loss with  $\epsilon_{\text{DIFF}} = 0.02$  ((a) & (b)) and  $0.03$  ((c) & (d)).

around 500 epochs and DTS converges faster than the others. In the constraints of  $0.03$ , the accuracy and loss of DTS perform with similar trends as  $0.02$  and the overall accuracies reach to  $0.6$  around 700 epochs. In addition, we find that robust optimisation with DTS converges faster than the other two SoA methods with  $14.2\%$  on average when compared with PGD method. The reason for this gain of DTS is that we only search important features as opposed to adding perturbations with randomly distributed methods. With these advantages, DTS can save training efforts and contribute to robustness during adversarial training.

**CIFAR10.** As before, we report the accuracy and loss trends for different adversarial training methods over the first 30,000 epochs with  $\epsilon_{\text{DIFF}} = 0.02$  in Figure 2(a) & (b) and  $0.03$  in Figure 2(c) & (d). In the constraints of  $0.02$ , the accuracies reach  $0.6$  before 13,000 epochs and DTS reach  $0.6$  around 10,000 epochs. The losses of different methods downgrade under 20 around 10,000 epochs and DTS converges faster than the others. In the constraints of  $0.03$ , the accuracy and loss of DTS perform with similar trends as  $0.02$  and the overall accuracies reach to  $0.6$  around 15,000 epochs. In addition, we find that robust optimisation with DTS converges faster than the other two SoA methods with  $10.3\%$  on average when compared with PGD method. From these results, DTS can save training efforts and contribute to robustness during adversarial training for both MNIST and CIFAR10 datasets.

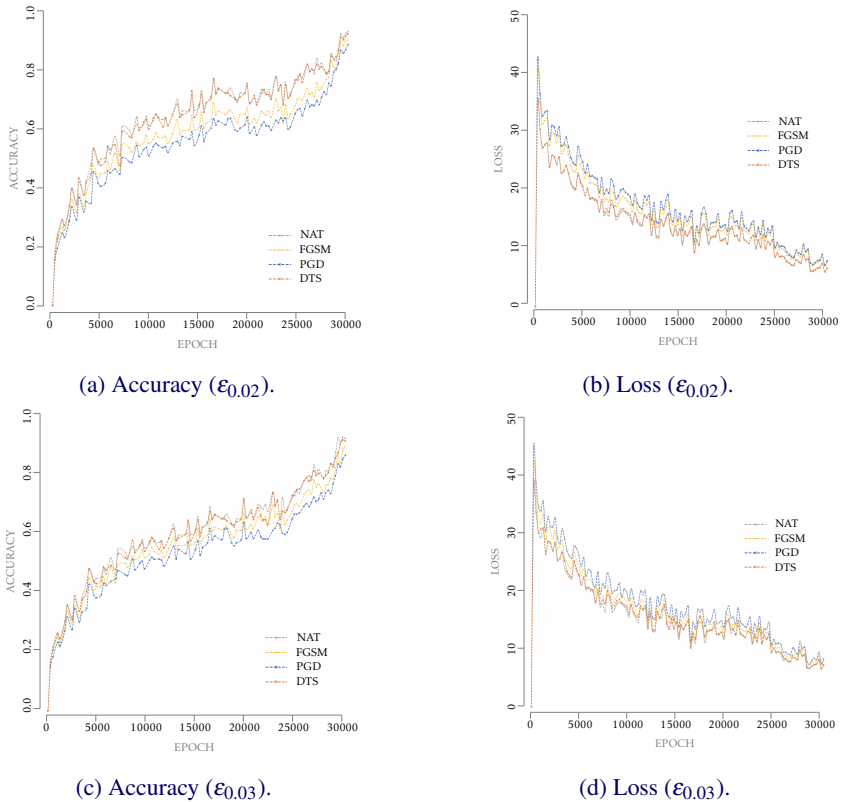


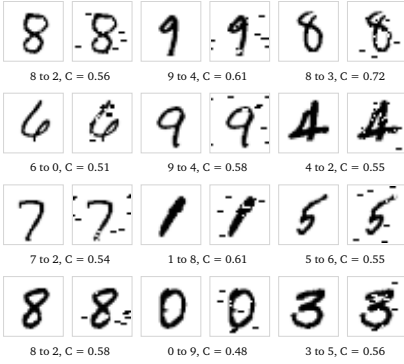
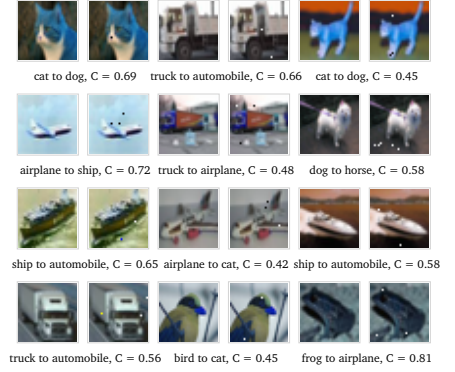
Figure 2: CIFAR10 accuracy and loss with  $\epsilon_{\text{DIFF}} = 0.02$  ((a) & (b)) and 0.03 ((c) & (d)).

## D Some Adversarial Examples with DTS

In this section we present some adversarial examples obtained with DTSATTACK for MNIST and CIFAR10 in Figure 3 and Figure 4. In Figure 3, we show adversarial examples with different confidence values under  $\epsilon_{\text{DIFF}} = 0.02$ ; in Figure 4, more adversarial examples for MNIST and CIFAR10 are presented under the same constraints.

**MNIST.** Some adversarial examples found using DTSATTACK with  $\epsilon_{\text{DIFF}} = 0.02$  are reported in Fig. 3a. Through DTSATTACK, only minor perturbations, where  $\epsilon_{\text{DIFF}} = 0.02$ , are required to generate attacks. The odd columns are the original images with correct classes and the even columns are their corresponding adversarial examples. In the figure, most confidence values are over 0.5 while generating adversarial examples. From the results, the adversarial ones are still clear to distinguish from the correct classes. Some of these attacks are not easy to detect by humans; see for example the one reported for class 4. In addition, these attacks are effective when used for adversarial training. As we show in Section C, the resulting training process becomes efficient. We report more adversarial examples for MNIST under the same constraints in Figure 4a.

**CIFAR10.** As MNIST, some adversarial examples obtained via DTSATTACK with  $\epsilon_{\text{DIFF}} = 0.02$  are presented in Fig. 3b. With DTSATTACK method, only minor perturbations are required to generate effective attacks. The odd columns are the original images with orig-

(a) MNIST adversarial images ( $\epsilon_{0.02}$ ).(b) CIFAR10 adversarial images ( $\epsilon_{0.02}$ ).Figure 3: Some adversarial examples for MNIST and CIFAR10 under  $\epsilon_{\text{DIFF}} = 0.02$ .

inal classes and the even columns are their corresponding adversarial examples. In these examples, most confidence values to acquire adversarial images are above 0.5. From the results, the adversarial ones are barely distinguishable from the correct classes. Some of these attacks are hard to detect by humans; see for example the ones reported for class cat to class dog with confidence values of 0.69 and 0.45. In addition, these attacks are effective when used for adversarial training. As we show in Section C, the resulting training process converges faster than the other SoA methods. We also report more adversarial examples for CIFAR10 under the same constraints in Figure 4b.

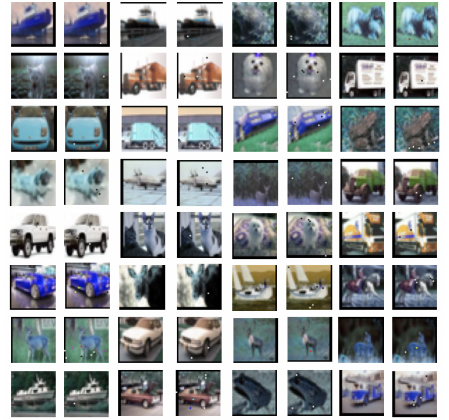
(a) Adversarial examples on MNIST ( $\epsilon_{0.02}$ ).(b) Adversarial examples on CIFAR10 ( $\epsilon_{0.02}$ ).

Figure 4: Adversarial examples on MNIST and CIFAR10 with  $\epsilon_{\text{DIFF}} = 0.02$  in the  $l_\infty$  norm. The odd columns are the clean images and the even columns are their corresponding adversarial examples.



Source \ Target	Nature Training	FGSM Training	PGD Training	DTS Training
Nature Training	98.3	55.6	45.8	36.2
FGSM Training	90.3	84.3	83.4	81.7
PGD Training	91.2	84.4	85.9	82.1
DTS Training	92.8	89.2	88.7	89.9

Table 4: The transferability comparison of nature training, FGSM, PGD and DTS methods with  $\epsilon_{\text{DIFF}} = 0.02$  from the source networks on MNIST.

Source \ Target	Nature Training	FGSM Training	PGD Training	DTS Training
Nature Training	95.2	53.4	40.4	35.3
FGSM Training	88.3	82.4	81.2	80.6
PGD Training	89.6	82.8	83.7	84.4
DTS Training	91.3	86.1	86.7	88.5

Table 5: The transferability comparison between nature training, FGSM, PGD and DTS methods with  $\epsilon_{\text{DIFF}} = 0.02$  from the source networks on CIFAR10.

## E Transferability

In this section, we evaluate the transferability between different adversarial training models using FGSM, PGD, and DTS methods. The evaluation basis for transferability is classified as cross-technique transferability which concerns misclassifications (caused by a set of attacks) on models trained on different learning methods. The results are summarised in Table 4 for MNIST and Table 5 for CIFAR10.

**MNIST.** Table 4 summarises the resulting accuracies obtained on the MNIST dataset. We generated adversarial examples based on one trained model of FGSM, PGD or DTS with  $\epsilon_{\text{DIFF}} = 0.02$ , and then evaluated them on a target network, to check if the target model can resist these adversarial examples. The network architecture is the same as the simple network previously and each model is trained with 3,000 epochs. The accuracies of the first row degrade a lot while encountering different adversarially trained attacks. In addition, the results obtained show that a strong adversary generally reduces transferability and increases robustness. For example, the result with source trained model (PGD Training) paired with target (DTS Training) is 88.7; this is higher than 85.9, which is the value from the same source but target (PGD Training).

**CIFAR10.** Table 5 summarises the resulting accuracies obtained on the CIFAR10 dataset. We also generated adversarial examples using the same methods as previous with  $\epsilon_{\text{DIFF}} = 0.02$ , and then evaluated them on a target network, independently trained via the FGSM, PGD and DTS methods. The network architecture is the same as the simple network used for CIFAR10 and we train each model with 30,000 epochs. The results obtained also demonstrate that a strong adversary can help to improve model accuracies and reduce transferability. In addition, DTS contributes to improving the accuracies against different adversaries while retaining robustness against the DTS attack itself.



## References

- [1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.
- [2] A. Krizhevsky and G. Hinton. Convolutional deep belief networks on cifar-10 <https://www.cs.toronto.edu/~kriz/cifar.html>, 2010.
- [3] Y. LeCun and C. Cortes. Mnist handwritten digit database <http://yann.lecun.com/exdb/mnist/>, 1998.
- [4] F. Tramer, A. Kurakin, N. Papernot, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. In *arXiv:1705.07204*, 2017.
- [5] S. Zagoruyko and N. Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12, 2016.