Measuring and Reducing Joint Vulnerability of Models to Adversarial Examples

Mainuddin Ahmad Jonas and David Evans

University of Virginia

Motivation

Let f_1 and f_2 trained to perform the same task. For any given input x, we call f_1 and f_2 jointly vulnerable, if an adversarial example x' exists around x, such that $f_1(x') = f_2(x')$ and $f_1(x) \neq f_1(x')$. In Figure 1, we visualize the adversarial vulnerability of single models and joint models around a given input point x.

In this project, our aim is to:

1. Propose a good metric of joint robustness of two models. Our initial experiments with a model diversity-based metric shows promising result.
2. Propose a way to train multiple models to achieve joint robustness.
3. Certify the joint robustness of models against adversarial examples.

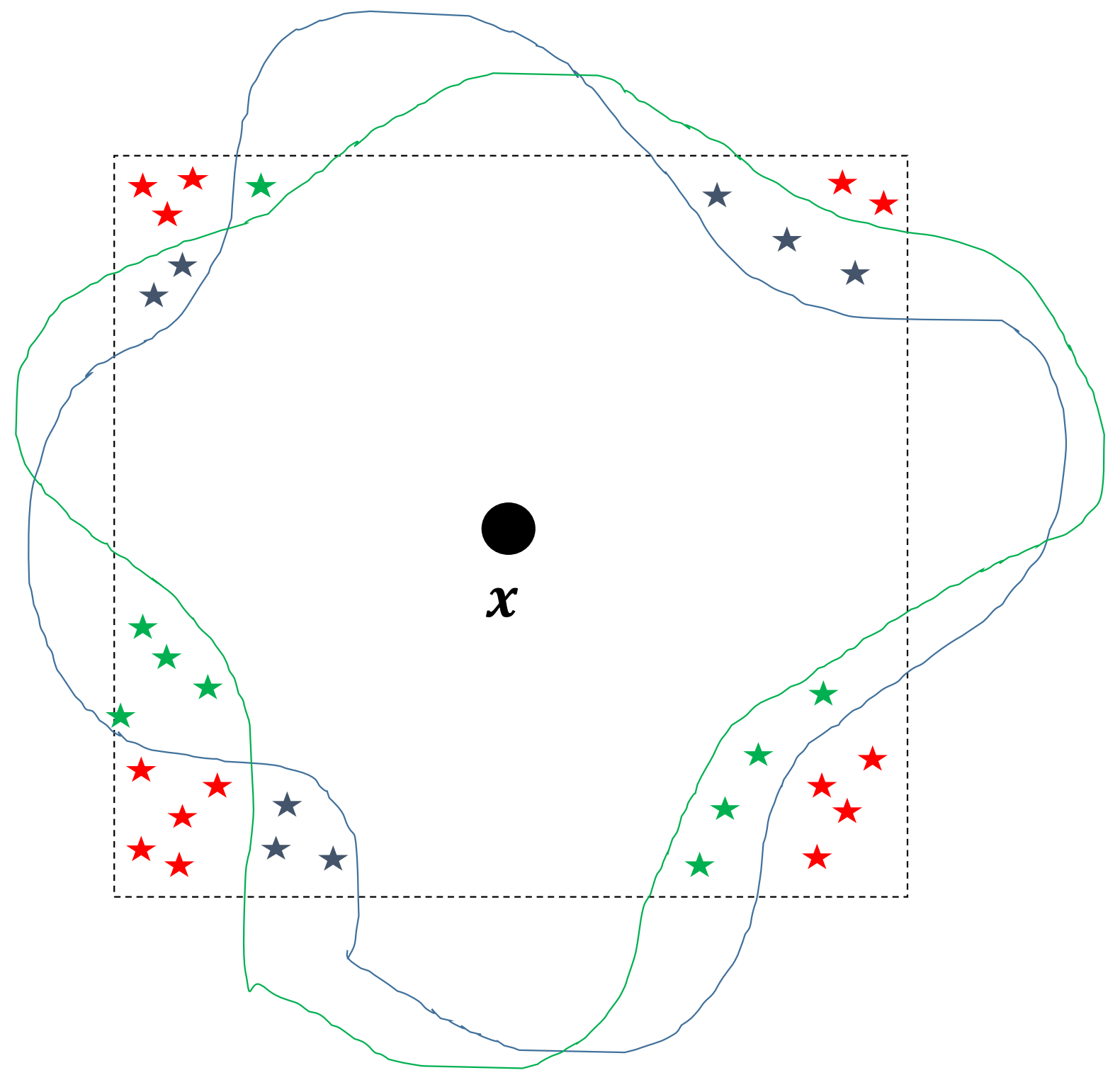


Figure 1. The central black dot represents the input x, and the dashed rectangle around it represents the allowable adversarial perturbation region around it. The decision boundary of model f_1 around x is depicted by the blue line, and the decision boundary of model f_2 is depicted by the green line. The blue stars are some adversarial examples against model f_1, the green stars are some adversarial examples against model f_2, and the red stars are some joint adversarial examples against both models.

Measuring Diversity of Models in an Ensemble

Given two models f_1 and f_2 trained to perform the same task, we aim to define a metric $d(f_1, f_2)$ that represents the adversarial diversity between the models. This metric should have the property that the higher the diversity, the lower the joint vulnerability of the models.

Metric for measuring diversity:
In our experiments, we first focused on the sign of the gradients of the output loss with respect to the input features. We found that both the signs and magnitudes are useful for the metric. Thus, we used the following metric:

$$d(f_1, f_2, X) = -\sum_{x \in X} \sum_{i \in D} \frac{\delta J(f_1, x)}{\delta x} \frac{\delta J(f_2, x)}{\delta x}$$

We show the results of our experiment to the right. It suggests that our metric is capturing joint robustness.

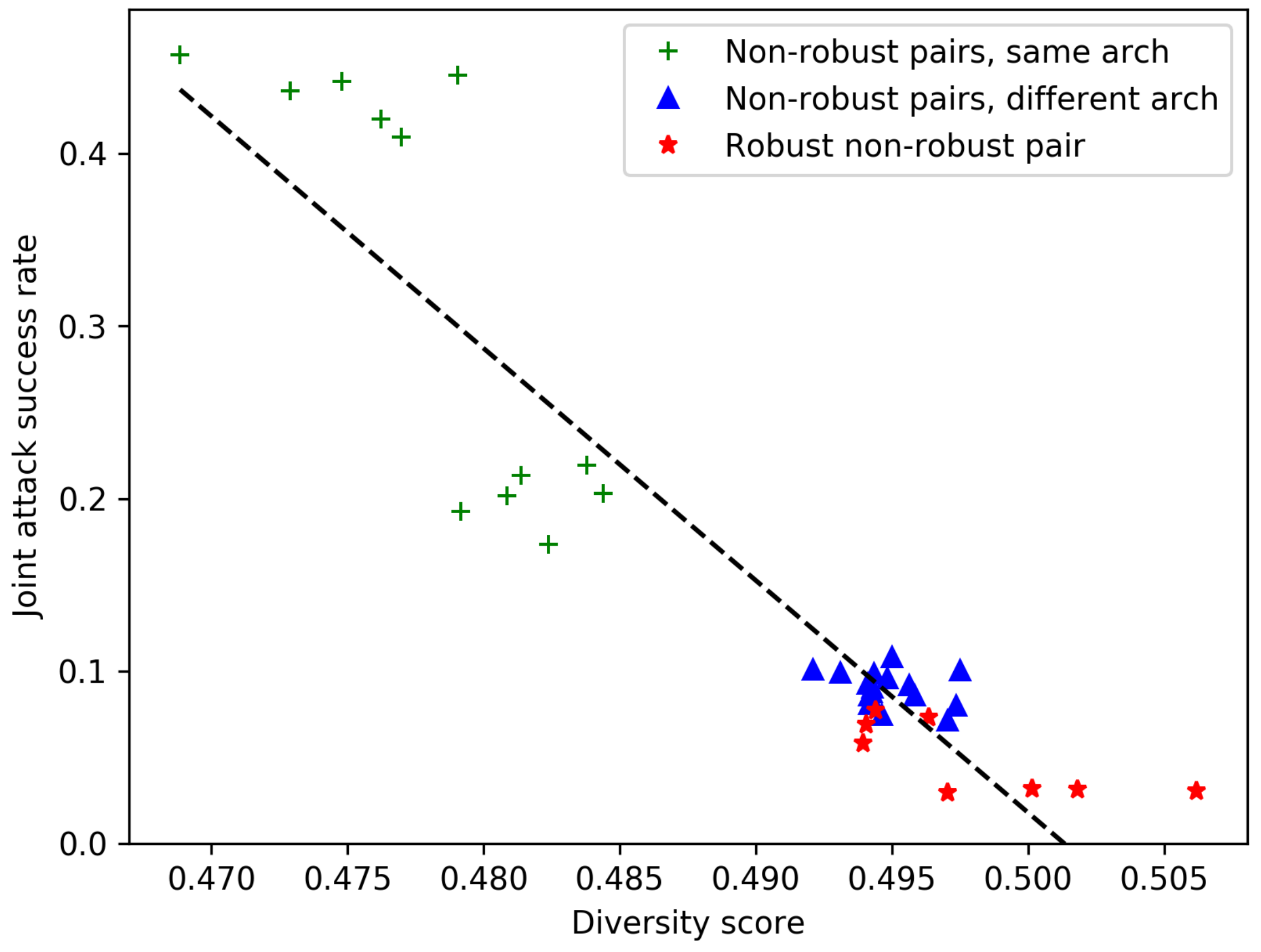


Figure 2: Success rate of PGD transfer attacks against pairs of models vs the diversity score of the model pair. It can be seen that the higher the diversity score, the lower the joint vulnerability as measured by transfer attack success.

Training for Diversity

We trained for diversity using two approaches: first through maximizing the diversity score of the two models, and second through cost-sensitive robust models.

Maximizing diversity score:

We took a normally trained MNIST model f_1, and trained another model f_2 to maximize the diversity score according to the following metric:

$$d(f_1, f_2, X) = \sum_{x \in X} \cos\left(\frac{\delta J(f_1, x)}{\delta x}, \frac{\delta J(f_2, x)}{\delta x}\right)$$

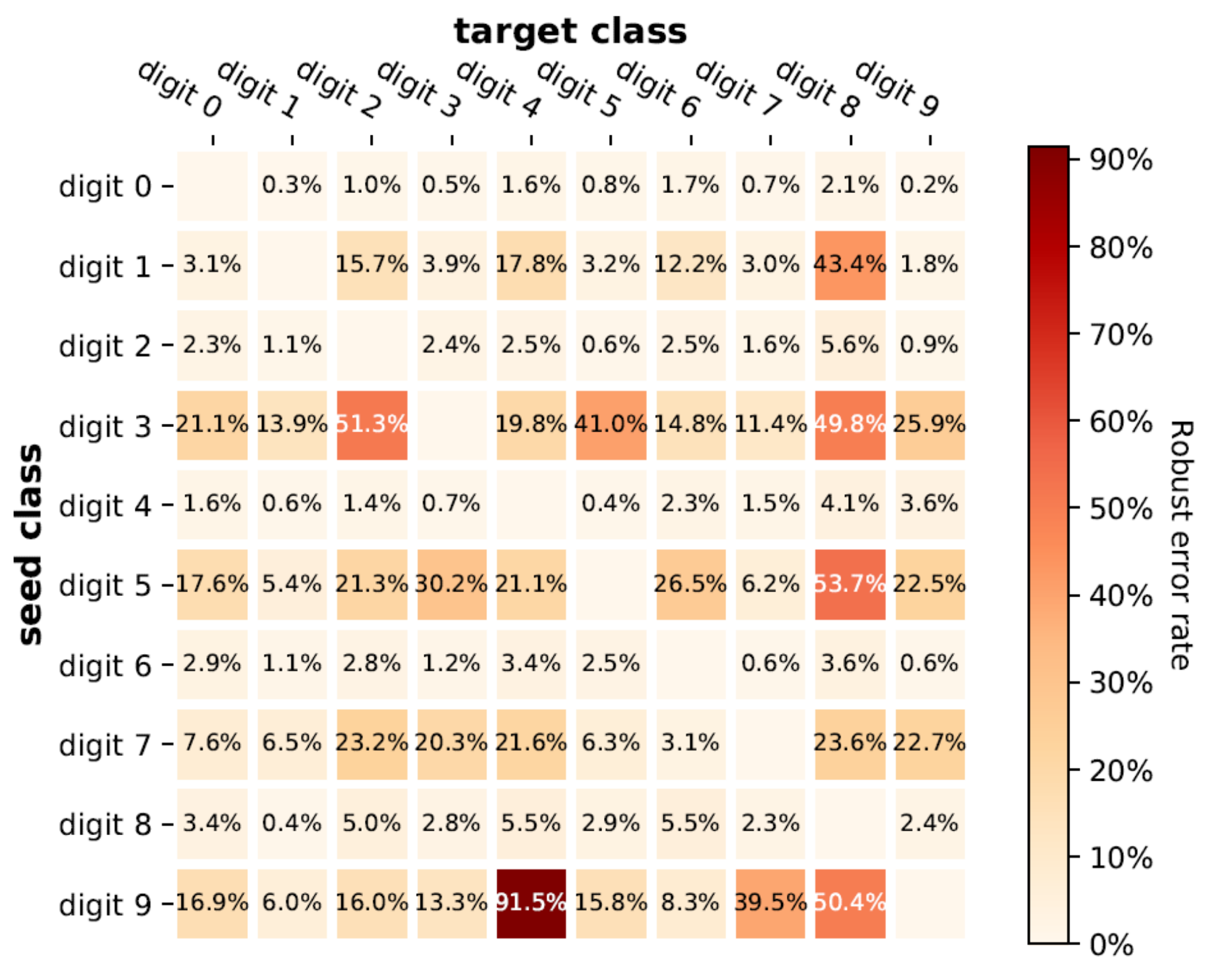
Then we compared the transfer attack success rate of the diversely trained model with the original transfer rates. We show the results of this in Table 1.

Table 1. PGD transfer attack success rate and diversity scores for normally trained and diversely trained models

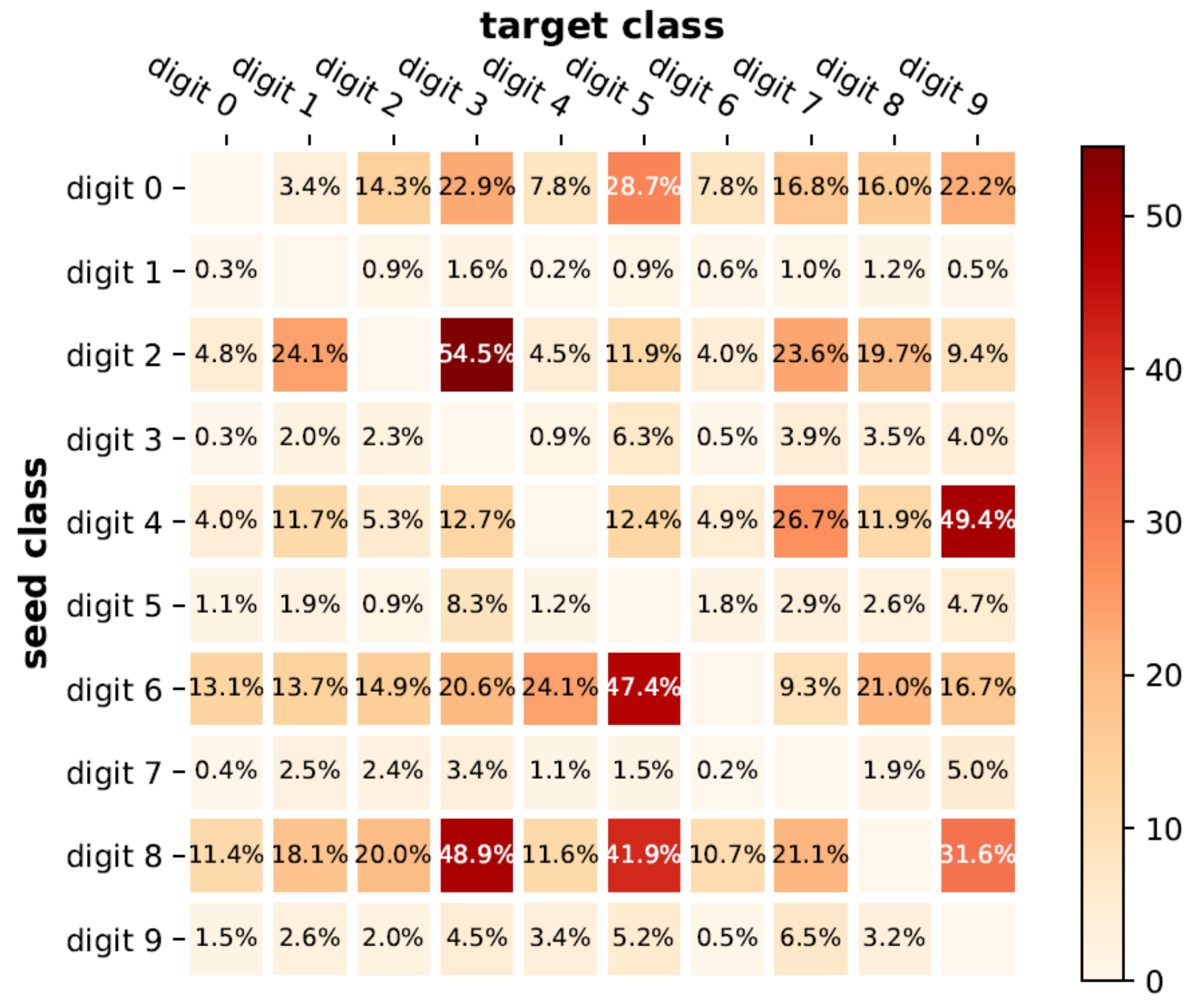
	Normal training	Diverse training
Transfer rate	63%	3%
Diversity score	0.02	0.14

Certifying Joint Robustness through Cost Sensitive Training

We took two models from the Cost-Sensitive Robustness work by Zhang et al. (ICLR 2019) – one trained to be robust on even digits of the MNIST dataset and the other on odd digits, and computed the robustness of the joint models. The two models are visualized below.

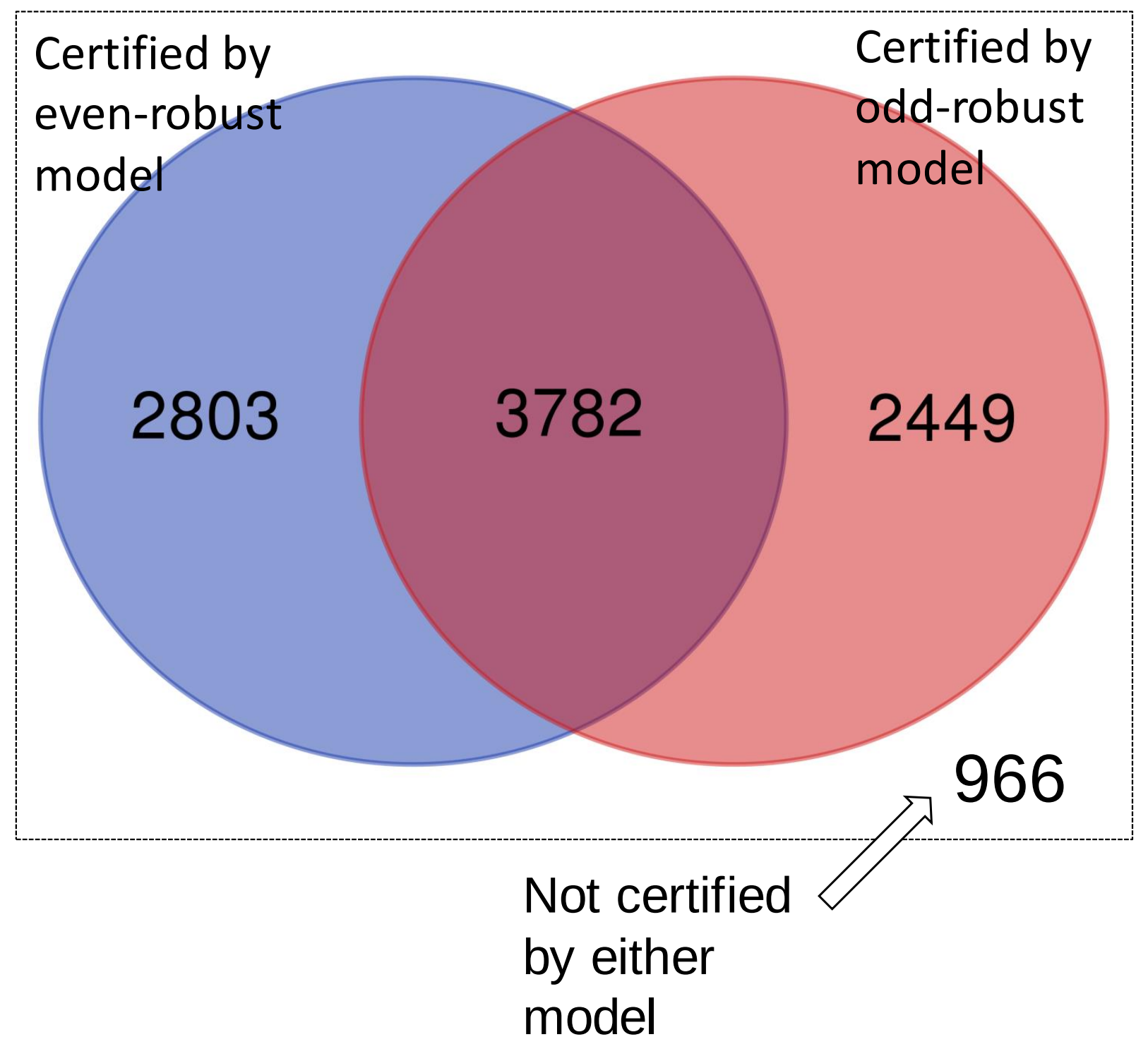


	digit 0	digit 1	digit 2	digit 3	digit 4	digit 5	digit 6	digit 7	digit 8	digit 9
digit 0		0.3%	1.0%	0.5%	1.6%	0.8%	1.7%	0.7%	2.1%	0.2%
digit 1	3.1%		15.7%	3.9%	17.8%	3.2%	12.2%	3.0%	43.4%	1.8%
digit 2	2.3%	1.1%		2.4%	2.5%	0.6%	2.5%	1.6%	5.6%	0.9%
digit 3	21.1%	13.9%	51.3%		19.8%	41.0%	14.8%	11.4%	49.8%	25.9%
digit 4	1.6%	0.6%	1.4%	0.7%		0.4%	2.3%	1.5%	4.1%	3.6%
digit 5	17.6%	5.4%	21.3%	30.2%	21.1%		26.5%	6.2%	53.7%	22.5%
digit 6	2.9%	1.1%	2.8%	1.2%	3.4%	2.5%		0.6%	3.6%	0.6%
digit 7	7.6%	6.5%	23.2%	20.3%	21.6%	6.3%	3.1%		23.6%	22.7%
digit 8	3.4%	0.4%	5.0%	2.8%	5.5%	2.9%	5.5%	2.3%		2.4%
digit 9	16.9%	6.0%	16.0%	13.3%	91.5%	15.8%	8.3%	39.5%	50.4%	



	digit 0	digit 1	digit 2	digit 3	digit 4	digit 5	digit 6	digit 7	digit 8	digit 9
digit 0		3.4%	14.3%	22.9%	7.8%	28.7%	7.8%	16.8%	16.0%	22.2%
digit 1	0.3%		0.9%	1.6%	0.2%	0.9%	0.6%	1.0%	1.2%	0.5%
digit 2	4.8%	24.1%		54.5%	4.5%	11.9%	4.0%	23.6%	19.7%	9.4%
digit 3	0.3%	2.0%	2.3%		0.9%	6.3%	0.5%	3.9%	3.5%	4.0%
digit 4	4.0%	11.7%	5.3%	12.7%		12.4%	4.9%	26.7%	11.9%	49.4%
digit 5	1.1%	1.9%	0.9%	8.3%	1.2%		1.8%	2.9%	2.6%	4.7%
digit 6	13.1%	13.7%	14.9%	20.6%	24.1%	47.4%		9.3%	21.0%	16.7%
digit 7	0.4%	2.5%	2.4%	3.4%	1.1%	1.5%	0.2%		1.9%	5.0%
digit 8	11.4%	18.1%	20.0%	48.9%	11.6%	41.9%	10.7%	21.1%		31.6%
digit 9	1.5%	2.6%	2.0%	4.5%	3.4%	5.2%	0.5%	6.5%	3.2%	

On the 10,000 test examples of the MNIST dataset, we used the two cost-sensitive models to certify robustness of each example. The results are visualized in the Venn diagram below. Out of 10,000 seeds, only 966 were not possible to certify. If a single robust model is trained, 1380 are not possible to certify. This suggests using two diverse models could be useful for certification.

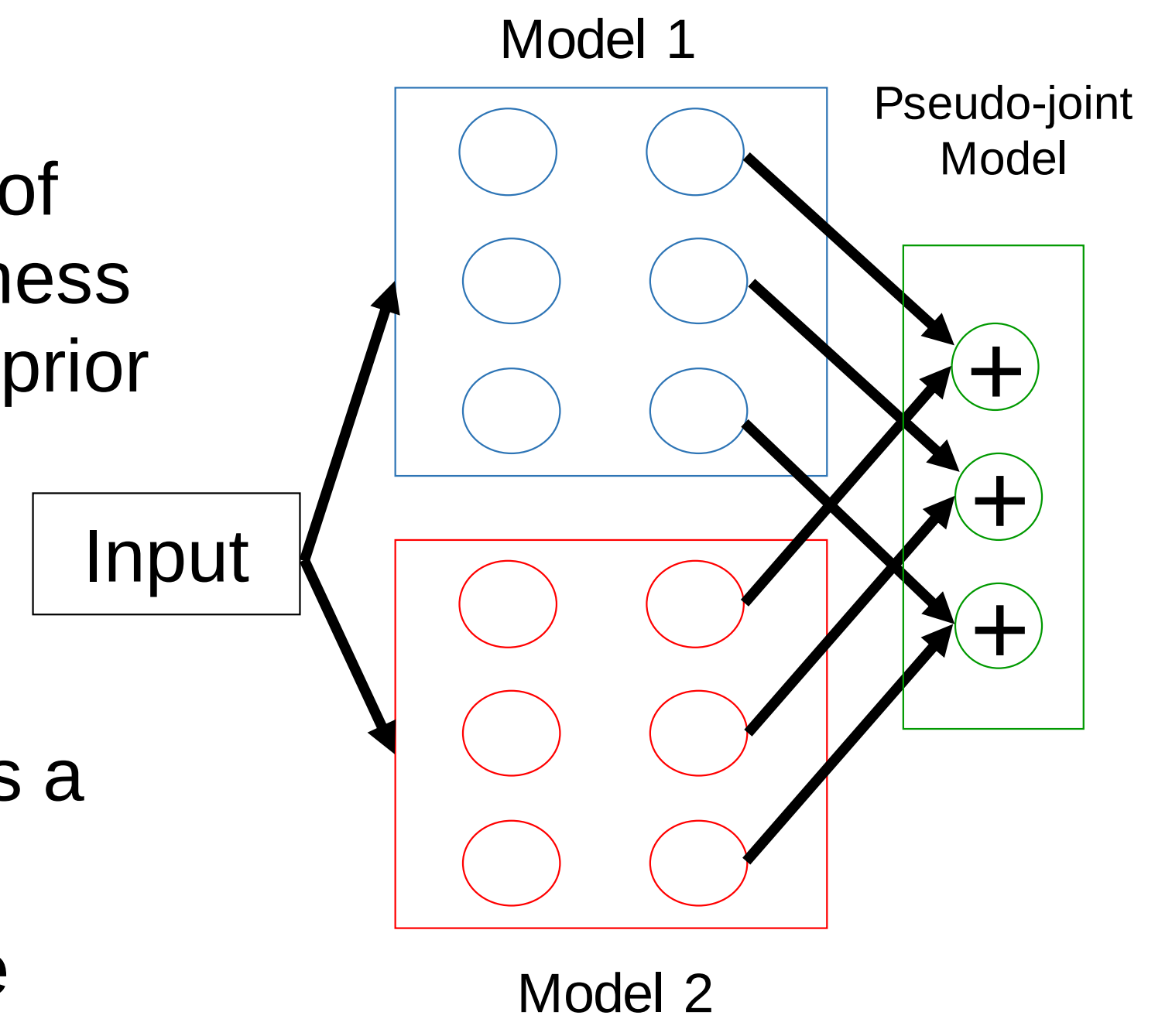

Certified by even-robust model
Certified by odd-robust model
2803
3782
2449
966
Not certified by either model

Improving Bound of Robust Certification

Let f_1 and f_2 be the two models we're trying to certify joint robustness for, and x be the input. Then we can define a pseudo-joint model *f:*

$$f(x) = f_1(x) + f_2(x)$$

It is trivial to prove that certifying the robustness of .f implies the joint robustness of f_1 and f_2. We can use prior work by Tjeng et al. of mixed integer linear programming to certify f. This approach will give us a tighter bound of joint robustness compared the previous approach.


Model 1
Pseudo-joint Model
Input
+
+
+
Model 2