Distribution Models for Falsification and Verification of DNNs

APPENDIX

In this section we describe the DNN properties used in our study.

A. FashionMNIST

The properties for Fashion MNIST consists in comparing different pieces of clothes in a way that the difference between clothes with similar shapes are smaller that others with different shapes. E.g. the difference between a t-shirt/top and a shirt should be smaller than the difference between a t-shirt/top and a sneaker. There are two types of properties:

(A) Specify that the output class must be one of the classes being compared.

a) Property $\phi_{A,0}$.:

$$\forall x.((x \in [0,1]^n) \wedge (argmax(\mathcal{N}(x)) = 7)) \rightarrow (|\mathcal{N}(x)_7 - \mathcal{N}(x)_6| > |\mathcal{N}(x)_7 - \mathcal{N}(x)_5|)$$

b) Property $\phi_{A,1}$.:

$$\forall x.((x \in [0,1]^n) \wedge (argmax(\mathcal{N}(x)) = 6)) \rightarrow (|\mathcal{N}(x)_6 - \mathcal{N}(x)_9| > |\mathcal{N}(x)_6 - \mathcal{N}(x)_2|)$$

c) Property $\phi_{A,2}$.:

$$\forall x.((x \in [0,1]^n) \wedge (argmax(\mathcal{N}(x)) = 5)) \rightarrow (|\mathcal{N}(x)_5 - \mathcal{N}(x)_8| > |\mathcal{N}(x)_5 - \mathcal{N}(x)_7|)$$

d) Property $\phi_{A,3}$.:

$$\forall x.((x \in [0,1]^n) \wedge (argmax(\mathcal{N}(x)) = 4)) \rightarrow (|\mathcal{N}(x)_4 - \mathcal{N}(x)_1| > |\mathcal{N}(x)_4 - \mathcal{N}(x)_6|)$$

e) Property $\phi_{A,4}$.:

$$\forall x.((x \in [0,1]^n) \wedge (argmax(\mathcal{N}(x)) = 3)) \rightarrow (|\mathcal{N}(x)_3 - \mathcal{N}(x)_7| > |\mathcal{N}(x)_3 - \mathcal{N}(x)_0|)$$

f) Property $\phi_{A,5}$.:

$$\forall x.((x \in [0,1]^n) \wedge (argmax(\mathcal{N}(x)) = 9)) \rightarrow (|\mathcal{N}(x)_9 - \mathcal{N}(x)_0| > |\mathcal{N}(x)_9 - \mathcal{N}(x)_7|)$$

g) Property $\phi_{A,6}$.:

$$\forall x.((x \in [0,1]^n) \wedge (argmax(\mathcal{N}(x)) = 2)) \rightarrow (|\mathcal{N}(x)_2 - \mathcal{N}(x)_1| > |\mathcal{N}(x)_2 - \mathcal{N}(x)_4|)$$

h) Property $\phi_{A,7}$.:

$$\forall x.((x \in [0,1]^n) \wedge (argmax(\mathcal{N}(x)) = 5)) \rightarrow (|\mathcal{N}(x)_5 - \mathcal{N}(x)_2| > |\mathcal{N}(x)_5 - \mathcal{N}(x)_9|)$$

i) Property $\phi_{A,8}$.:

$$\forall x.((x \in [0,1]^n) \wedge (argmax(\mathcal{N}(x)) = 0)) \rightarrow (|\mathcal{N}(x)_0 - \mathcal{N}(x)_8| > |\mathcal{N}(x)_0 - \mathcal{N}(x)_6|)$$

j) Property $\phi_{A,9}$.:

$$\forall x.((x \in [0,1]^n) \wedge (argmax(\mathcal{N}(x)) = 1)) \rightarrow (|\mathcal{N}(x)_1 - \mathcal{N}(x)_7| > |\mathcal{N}(x)_1 - \mathcal{N}(x)_3|)$$

(B) Do not specify any output class.

k) Property $\phi_{B,0}$.:

$$\forall x.(x \in [0,1]^n) \rightarrow (|\mathcal{N}(x)_7 - \mathcal{N}(x)_6| > |\mathcal{N}(x)_7 - \mathcal{N}(x)_5|)$$

l) Property $\phi_{B,1}$.:

$$\forall x.(x \in [0,1]^n) \rightarrow (|\mathcal{N}(x)_6 - \mathcal{N}(x)_9| > |\mathcal{N}(x)_6 - \mathcal{N}(x)_2|)$$

m) Property $\phi_{B,2}$.:

$$\forall x.(x \in [0,1]^n) \rightarrow (|\mathcal{N}(x)_5 - \mathcal{N}(x)_8| > |\mathcal{N}(x)_5 - \mathcal{N}(x)_7|)$$

n) Property $\phi_{B,3}$.:

$$\forall x.(x \in [0,1]^n) \rightarrow (|\mathcal{N}(x)_4 - \mathcal{N}(x)_1| > |\mathcal{N}(x)_4 - \mathcal{N}(x)_6|)$$

o) Property $\phi_{B,4}$.:

$$\forall x.(x \in [0,1]^n) \rightarrow (|\mathcal{N}(x)_3 - \mathcal{N}(x)_7| > |\mathcal{N}(x)_3 - \mathcal{N}(x)_0|)$$

p) Property $\phi_{B,5}$.:

$$\forall x.(x \in [0,1]^n) \rightarrow (|\mathcal{N}(x)_7 - \mathcal{N}(x)_2| > |\mathcal{N}(x)_7 - \mathcal{N}(x)_9|)$$

q) Property $\phi_{B,6}$.:

$$\forall x.(x \in [0,1]^n) \rightarrow (|\mathcal{N}(x)_6 - \mathcal{N}(x)_5| > |\mathcal{N}(x)_6 - \mathcal{N}(x)_4|)$$

r) Property $\phi_{B,7}$.:

$$\forall x.(x \in [0,1]^n) \rightarrow (|\mathcal{N}(x)_5 - \mathcal{N}(x)_1| > |\mathcal{N}(x)_5 - \mathcal{N}(x)_7|)$$

s) Property $\phi_{B,8}$.:

$$\forall x.(x \in [0,1]^n) \rightarrow (|\mathcal{N}(x)_4 - \mathcal{N}(x)_8| > |\mathcal{N}(x)_4 - \mathcal{N}(x)_2|)$$

t) Property $\phi_{B,9}$.:

$$\forall x.(x \in [0,1]^n) \rightarrow (|\mathcal{N}(x)_3 - \mathcal{N}(x)_9| > |\mathcal{N}(x)_3 - \mathcal{N}(x)_0|)$$

B. DroNet

The network used for the GHPR-DroNet benchmark is the DroNet network[1] [1] for autonomous quadrotor control. This network is based on a ResNet type architecture, with 3 residual blocks. It is comprised of 475131 neurons and 320226 parameters.

The properties for DroNet codify the desired behavior that, if the probability for collision is low, the system should not make sharp turns. The DroNet properties are of the form: for all inputs, if the probability of collision is between p_{min} and p_{max}, then the steering angle is within d degrees of 0.

a) Property ϕ_0.:

$$\forall x.((x \in [0,1]^n) \wedge (0 < \mathcal{N}(x)_P \leq 0.1)) \rightarrow (-5^\circ \leq \mathcal{N}(x)_S \leq 5^\circ)$$

b) Property ϕ_1.:

$$\forall x.((x \in [0,1]^n) \wedge (0.1 < \mathcal{N}(x)_P \leq 0.2)) \rightarrow (-10^\circ \leq \mathcal{N}(x)_S \leq 10^\circ)$$

c) Property ϕ_2.:

$$\forall x.((x \in [0,1]^n) \wedge (0.2 < \mathcal{N}(x)_P \leq 0.3)) \rightarrow (-20^\circ \leq \mathcal{N}(x)_S \leq 20^\circ)$$

d) Property ϕ_3.:

$$\forall x.((x \in [0,1]^n) \wedge (0.3 < \mathcal{N}(x)_P \leq 0.4)) \rightarrow (-30^\circ \leq \mathcal{N}(x)_S \leq 30^\circ)$$

e) Property ϕ_4.:

$$\forall x.((x \in [0,1]^n) \wedge (0.4 < \mathcal{N}(x)_P \leq 0.5)) \rightarrow (-40^\circ \leq \mathcal{N}(x)_S \leq 40^\circ)$$

f) Property ϕ_5.:

$$\forall x.((x \in [0,1]^n) \wedge (0.5 < \mathcal{N}(x)_P \leq 0.6)) \rightarrow (-50^\circ \leq \mathcal{N}(x)_S \leq 50^\circ)$$

g) Property ϕ_6.:

$$\forall x.((x \in [0,1]^n) \wedge (0.6 < \mathcal{N}(x)_P \leq 0.7)) \rightarrow (-60^\circ \leq \mathcal{N}(x)_S \leq 60^\circ)$$

h) Property ϕ_7.:

$$\forall x.((x \in [0,1]^n) \wedge (0.7 < \mathcal{N}(x)_P \leq 0.8)) \rightarrow (-70^\circ \leq \mathcal{N}(x)_S \leq 70^\circ)$$

i) Property ϕ_8.:

$$\forall x.((x \in [0,1]^n) \wedge (0.8 < \mathcal{N}(x)_P \leq 0.9)) \rightarrow (-80^\circ \leq \mathcal{N}(x)_S \leq 80^\circ)$$

[1]https://github.com/uzh-rpg/rpg_public_dronet

TABLE III: A count of the results produced by each tool when running on properties without DFV.

	Result				
Tool	sat	unsat	unknown	timeout	error
DeepFool	74	0	26	0	0
BIM	73	0	27	0	0
FGSM	71	0	29	0	0
PGD	85	0	0	15	0
Neurify	59	0	0	40	1
nnenum	61	0	0	0	39
VeriNet	49	0	0	51	0

TABLE IV: A count of the results produced by each tool when running on properties with DFV.

	Result				
Tool	sat	unsat	unknown	timeout	error
DeepFool	56	0	44	0	0
BIM	53	0	47	0	0
FGSM	48	0	52	0	0
PGD	71	0	0	29	0
Neurify	7	0	0	93	0
nnenum	64	0	25	0	11
VeriNet	2	0	0	98	0

j) Property ϕ_9.:

$$\forall x.((x \in [0,1]^n) \wedge (0.9 < \mathcal{N}(x)_P \leq 1.0)) \rightarrow (-90^\circ \leq \mathcal{N}(x)_S \leq 90^\circ)$$

In this section we present additional results and data from the experiments for our first research question. Table III shows the number of results of each type produced by each tool on the FashionMNIST model alone, without using DFV. Similarly, Table IV shows the number of results of each type produced by each tool on the FashionMNIST when DFV is used with a simple VAE as the environment model. As expected, DFV reduces the number of `sat` results as it restricts tools to report counter-examples within the distribution.

In this section we report additional plots and data from the experiments executed to address our second research question.

Fig. 13 shows the mean reconstruction similarity of each counter-example found by PGD across all of the latent space sizes, number of layers, and number of neurons per layer explored. Each latent space size is shown in a different plot, with a latent space of dimension 1 in the top plot and dimension 32 in the bottom plot.

We also show the same plots but using the encoder stochastic reconstruction error (ESRE) in Fig. 14. This value is computed as the mean of the mean squared error of 100 reconstructions of each counter-example using VAE_{MRS}.

In addition to the quality measures for each counter-example, we present the times to find each counter-example across the 90 VAE configurations explored in RQ2 in Fig. 15.

Finally, Figure 16 presents the times to find each counter-example across the 16 different radii explored in the second part of RQ2.

In this section we report addition plots and data from the experiments run to address our third research question.

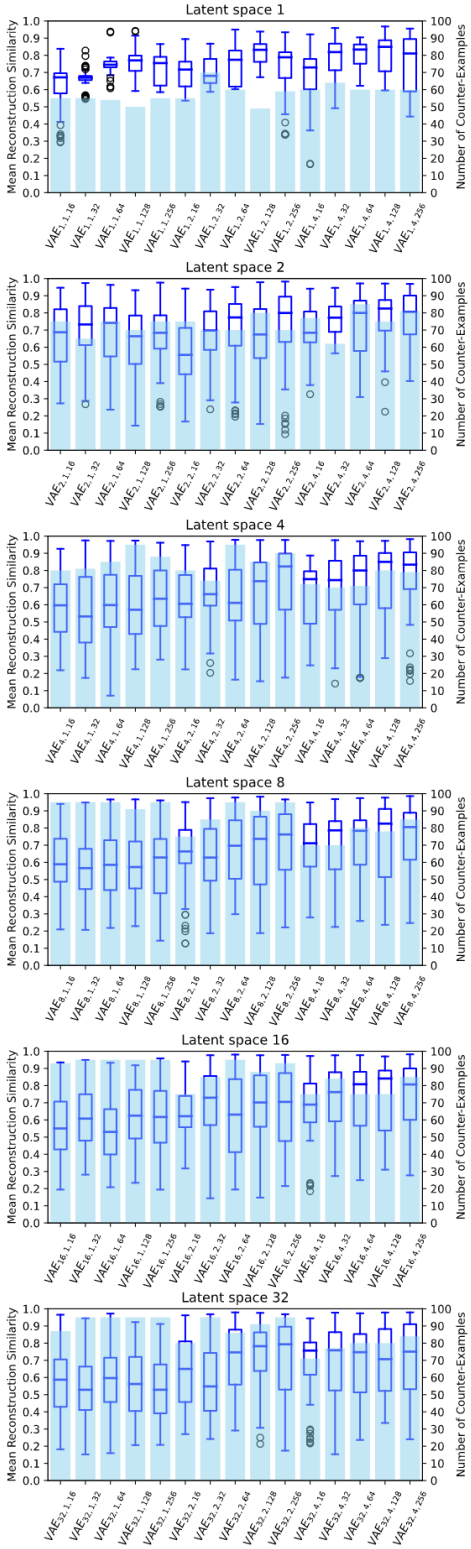

Fig. 13: MRS of counter-examples found using PGD across all latent space sizes, number of layers, and number of neurons per layer. The MRS was computed with the VAE_{MRS} model using SSIM similarity.

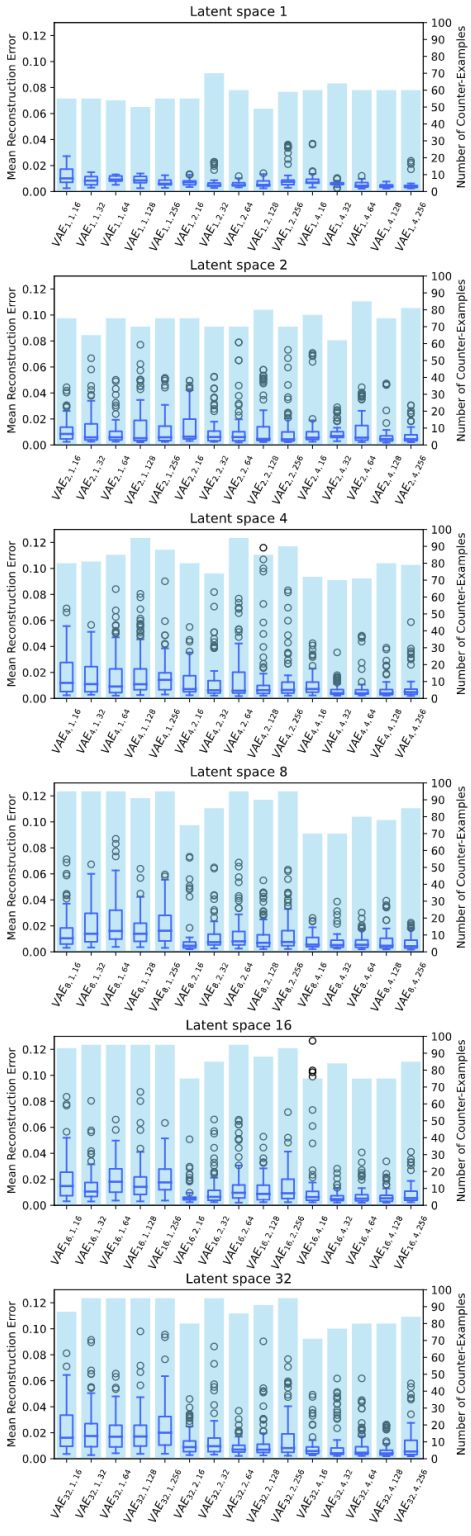

Fig. 14: MRS of counter-examples found using PGD across all latent space sizes, number of layers, and number of neurons per layer. The error was computed with the VAE_{MRS} model using the Mean Squared Error (MSE).

Figure 17 presents the encoder stochastic reconstruction error (ESRE) for each counter-example found. This value is computed as the mean of the mean squared error of 100 reconstructions of each counter-example using Conv-VAE_{DroNet}.

We also present all of the counter-examples found for the DroNet properties, both with (Figures 19 and 20) and without (Fig. 18) DFV.

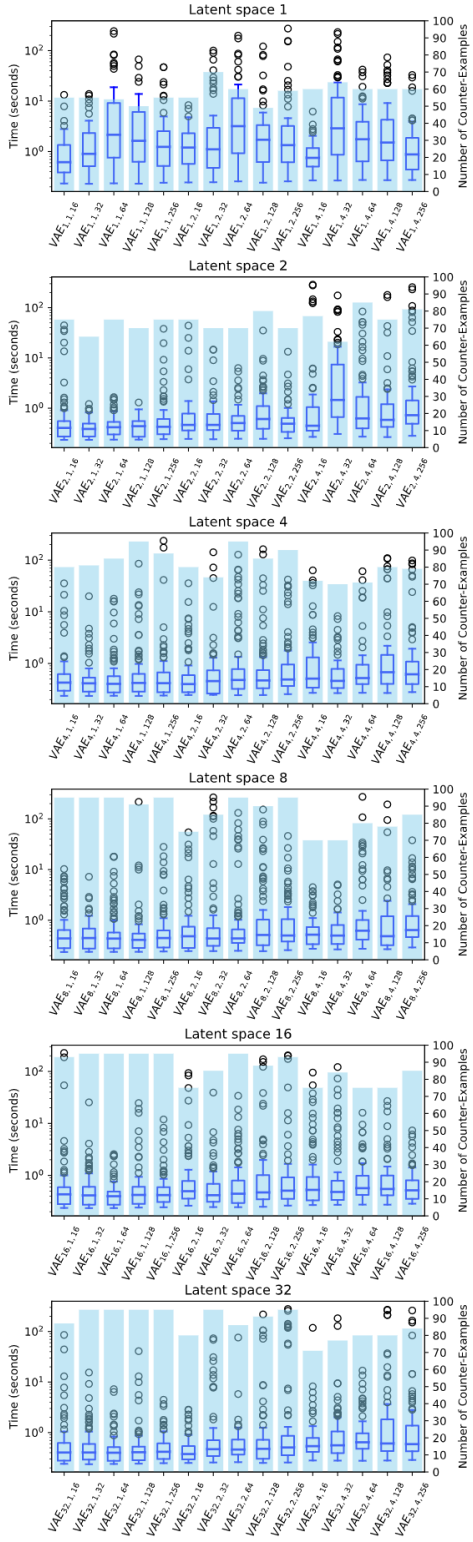

Fig. 15: Time spent by PGD to find counter-examples for each model explored in RQ2.

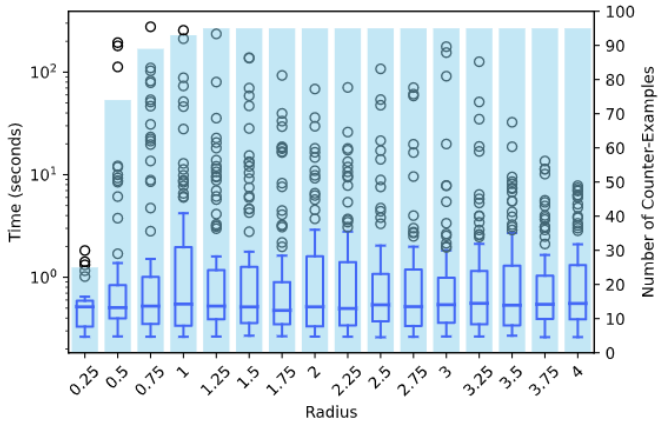

Fig. 16: Time spent by PGD to find counter-examples using different radii.

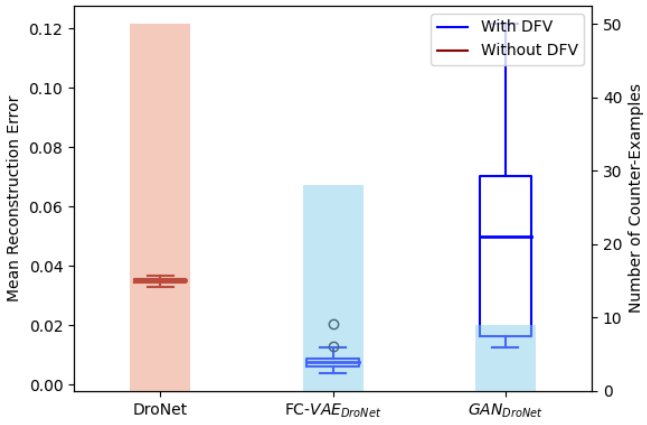

Fig. 17: A plot of the reconstruction error for each counter-example found. The Mean Squared Error (MSE) is used to measure reconstruction error, and we take the mean of 100 reconstructions using Conv-VAE_{DroNet}.

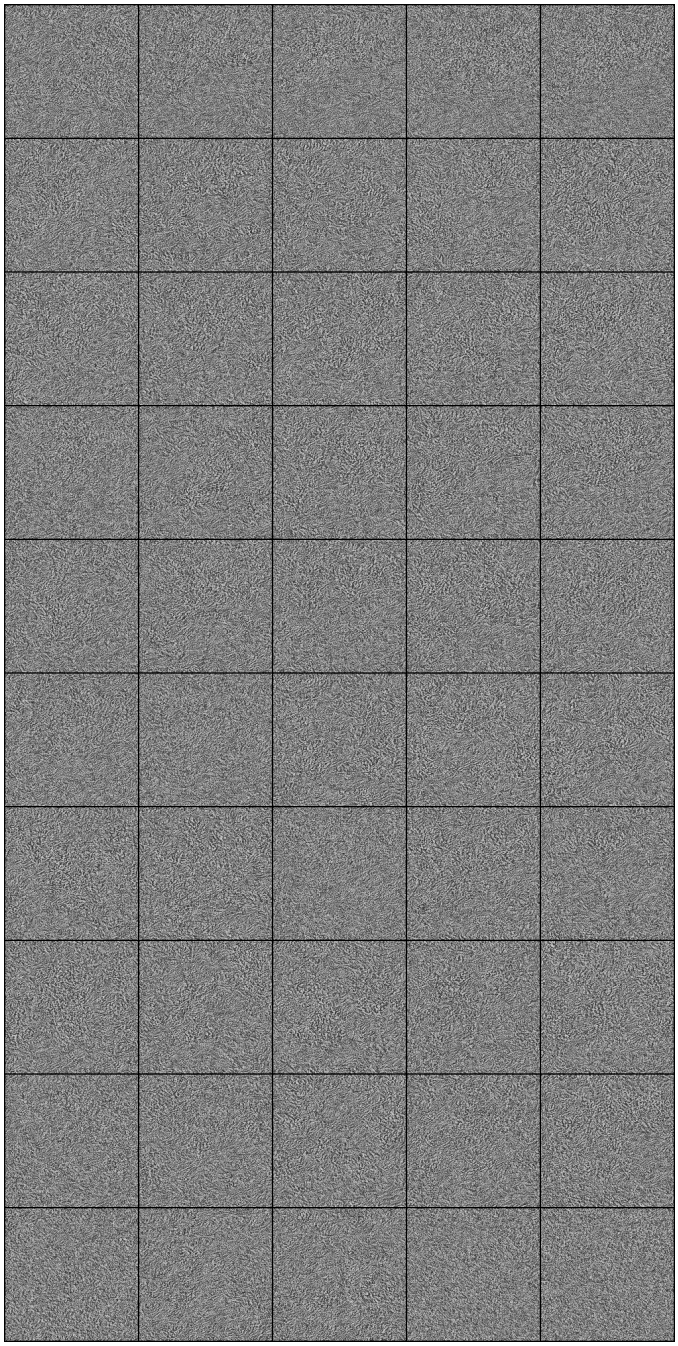

Fig. 18: The counter-examples found by PGD for each of the 10 properties of the DroNet DNN without using DFV. Each row corresponds to one property and each column is a separate run of PGD on the property and DroNet network.

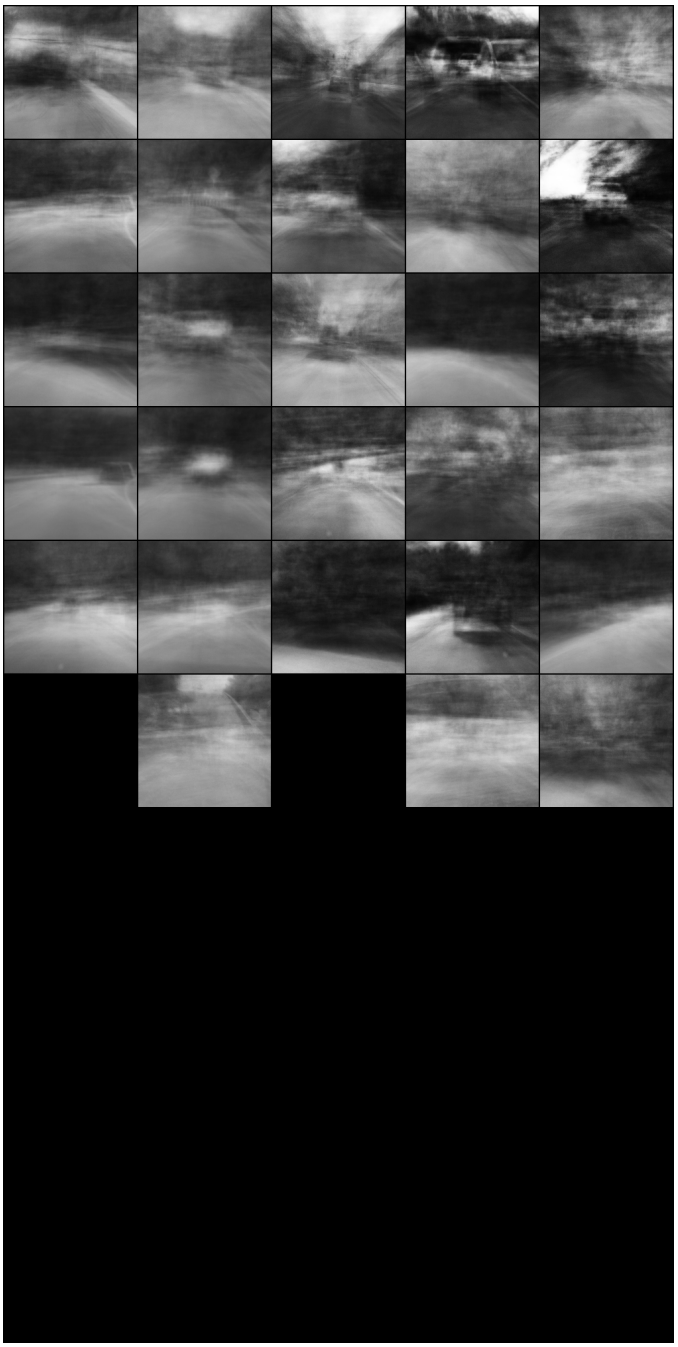

Fig. 19: The counter-examples found by PGD for each of the 10 properties of the DroNet DNN using DFV with FC-VAE_{Dronet}. Each row corresponds to one property and each column is a separate run of PGD on the property and DroNet network.

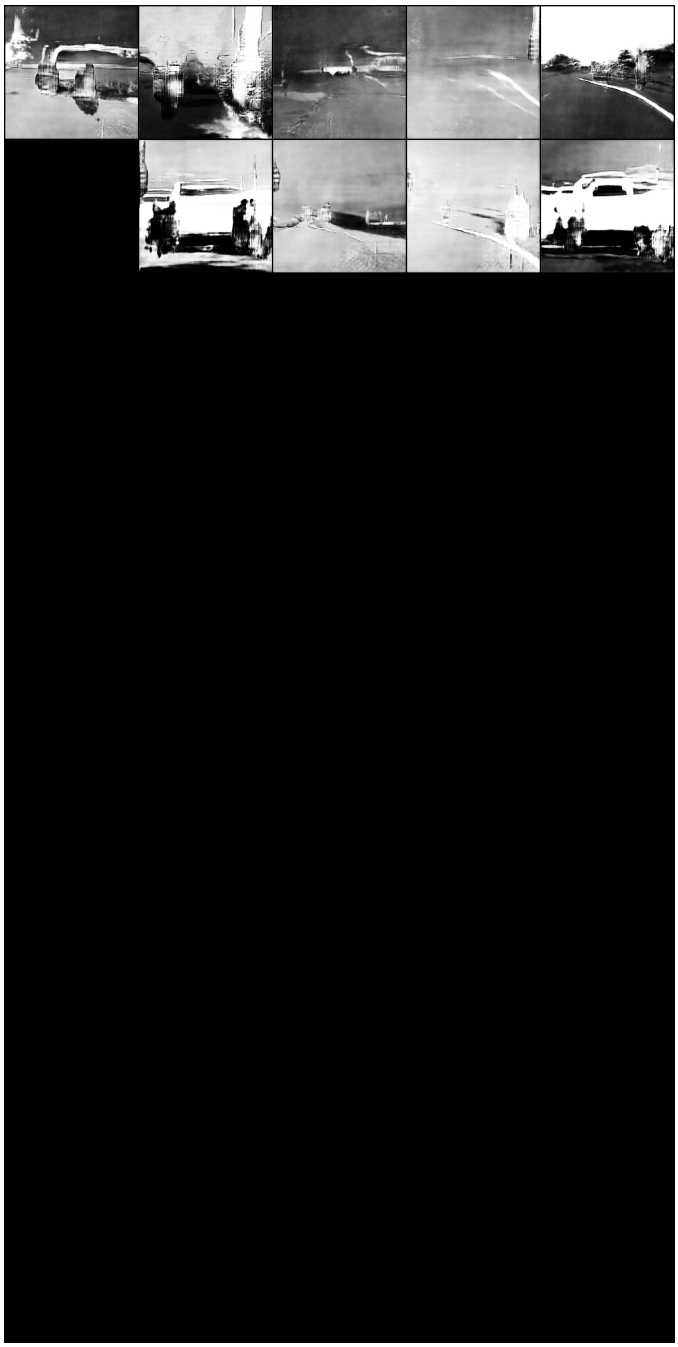

Fig. 20: The counter-examples found by PGD for each of the 10 properties of the DroNet DNN using DFV with GAN_{DroNet}. Each row corresponds to one property and each column is a separate run of PGD on the property and DroNet network.

REFERENCES

[1] A. Loquercio, A. I. Maqueda, C. R. D. Blanco, and D. Scaramuzza, "Dronet: Learning to fly by driving," *IEEE Robotics and Automation Letters*, 2018.