# a Interpretability

In this section, we will evaluate the model's prediction to better understand it. We will find if there are any features in the patches that can confuse it and if the model's output is robust. First, we will introduce on technique used to highlight the region of the input image that contribute the most to the model predictions. Then, we will use it on the data from the *Quarry* test set to find out the patches were the model fails and analyze them.

Later, we will work with custom created patches with different features, walls, bumps, etc, to test the robustness of the model by comparing its predictions to the real data gathered from the simulator.

## a.1 Grad-CAM

Informally, Grad-CAM [**gradcam**] is a technique to produce "visual explanations" for convolutional neural networks. It uses the gradient flowing into the last convolution layer to highlight a regions of interested in the image that contribute the most to the final prediction.

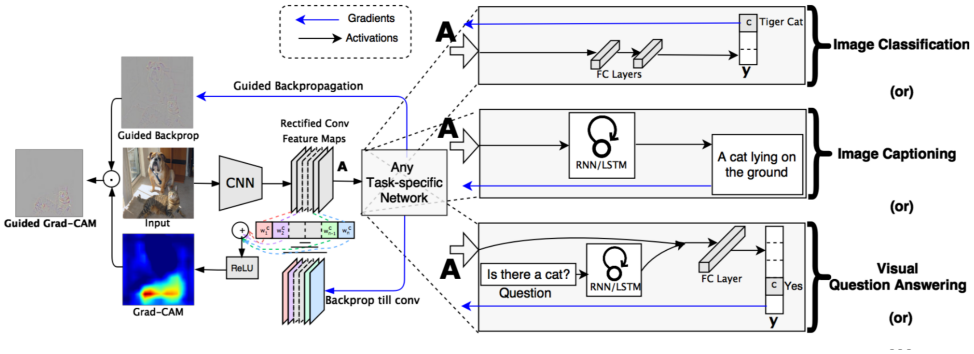



Figure 1. Grad-CAM procedure on an input image [**gradcam**].

In detail, the output with respect to a target class is backpropagate while storing the gradient and the output in the last convolution. Then global average is applyed the saved gradient keeping the channel dimension in order to get a 1-d tensor, this will represent the importance of each channel in the target convolutional layer. We each element of the convolutional layer outputs is multiplyed with the averaged gradients to create the grad cam. This whole procedure is fast and it is architecture independent.

## a.2 Robustness

To test the model's robustness we createed custom patches with different features, walls/bumps/ramps, and test the model prediction against the real robot advancement obtained from the simulator. According to the previous experiments, we used a threshold of 20cm and a time window of two seconds.

Wall in front of Krock

The most trivial test is to place a not traversable wall in front of *Krock* at an increasing distance from the its head. We will expect to reach a point where the model predict traversable even if the wall itself is too tall. Why? Because the robot will be able to travel more than the threshold before being stopped by the obstacle.

We created fifty-five different patches by first placing the wall exactly in front of the robot and then move it by 1cm at the time towards the end. It follows some of the input patches ordered by distance from the robot. We remind to the reader that Krock traverse the patch from left to right.

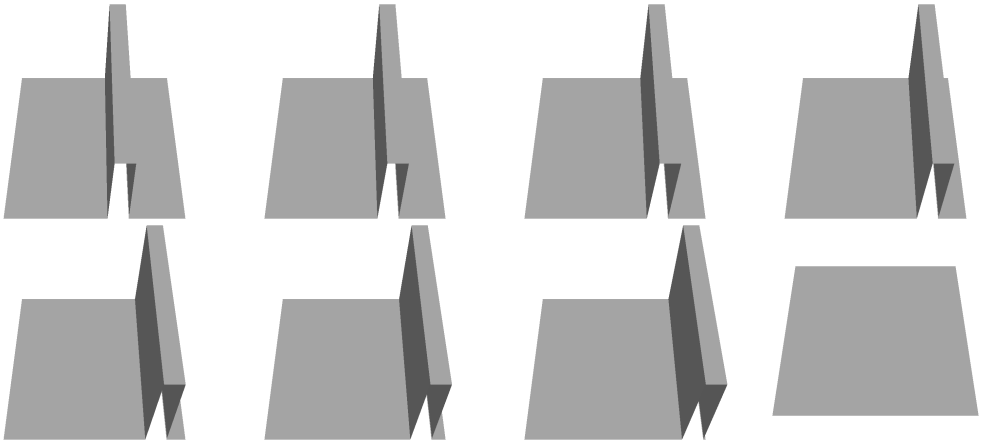


Figure 2. Some of the tested patches with a non traversable wall at increasing distance from Krock.

Given those inputs to the model, we get the following predictions.

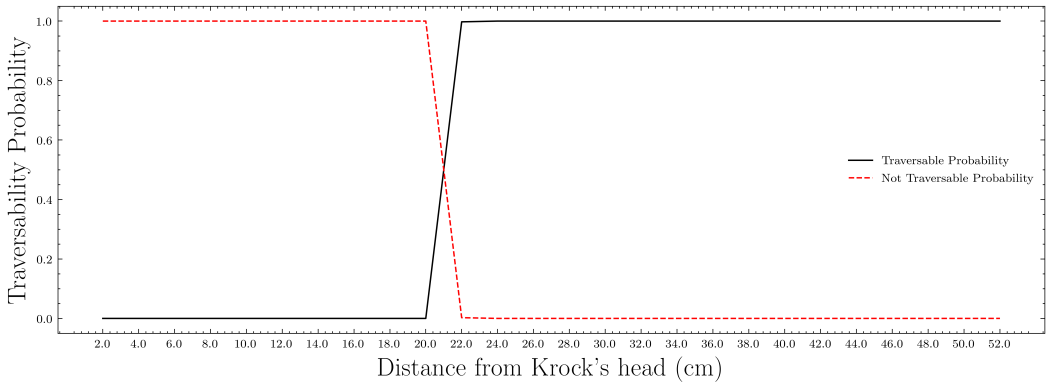



Figure 3. Traversability probabilities against wall distance from Krock's head.

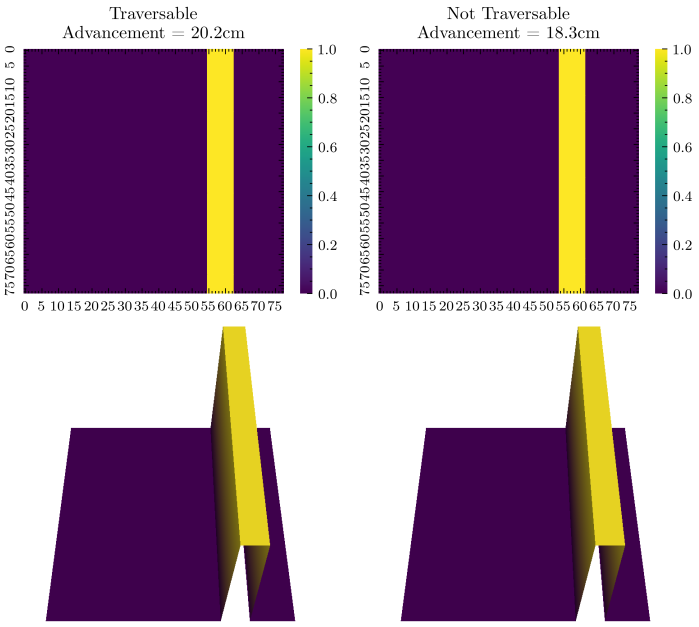
graph too tall, adjust the figure size

Summarized by the following table:

| Distance(cm) | Prediction |
|---|---|
| 0 - 20 | Not traversable |
| 20 - end | Traversable |

Table 1. Model prediction from the wall patches.

To be sure the results are correct, we run the last not traversable and the first traversable patch on the simulator to get real advancement. In the simulator, Krock advances 19.9cm on the not traversable patch ($a$) where the wall is at a distance of 19.6cm from the head. While, on the first traversable patch in which the wall is at a distance of 20.5cm, the robot was able to travel for 22.4cm. This shows the network ability to correct understanding that distance from the obstacle is more relevant than its height.

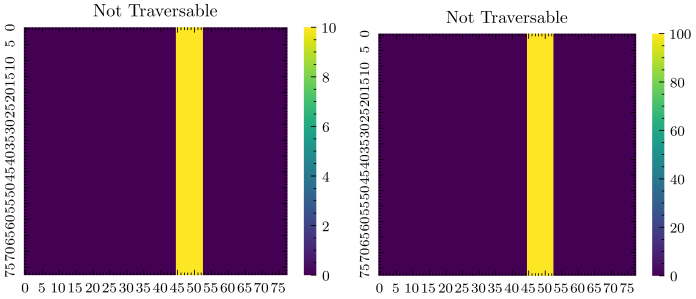



(a) Distance 20cm (b) Distance 22cm

Figure 4. The last non traversable and the first traversable patches with wall at a distance $< tr$ and $>$ tr.

Due to the patch resolution and minor changes in the spawing position, the robot may advance more than the distance between its head and the wall.

Furthermore, we can increase the wall size of the first traversable patch, ($b$), to 10 and to 100m to stress even more the ability of the model to look at the distance and not at the height.




(a) height = 10m (b) height = 100m

Figure 5. Two patches with a very tall wall at a distance $> tr$.

remove title, second colormap is wrong

Correctly, the model classifies the patches as traversable and was not confused by the enourmous height of the wall.

Increasing height walls

We can also evaluate the model's robustness by placing different wall of increasing heights in front of the robot to check whether the prediction matches the real data. We run forty patches in the simulator from a wall's height of 1cm to 20cm. The following figure shows some of the inputs.

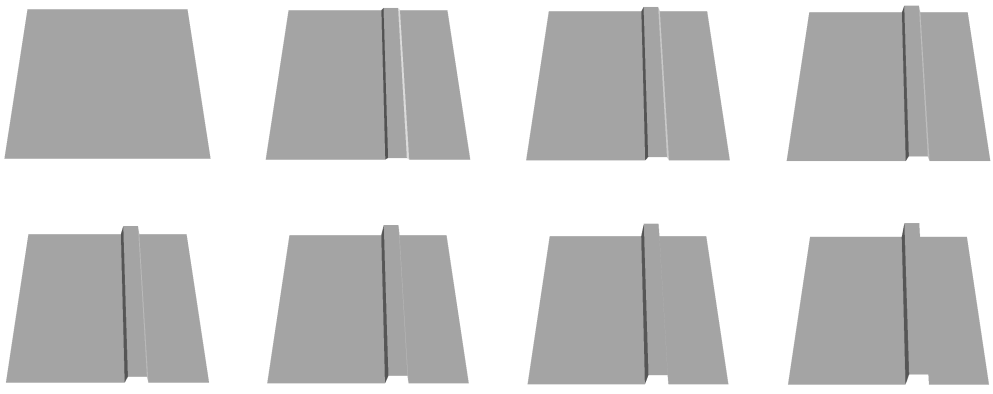


Figure 6. Some of the tested patches with a wall at increasing height ahead of Krock.

The models predicts that the walls under 10cm are traversable.

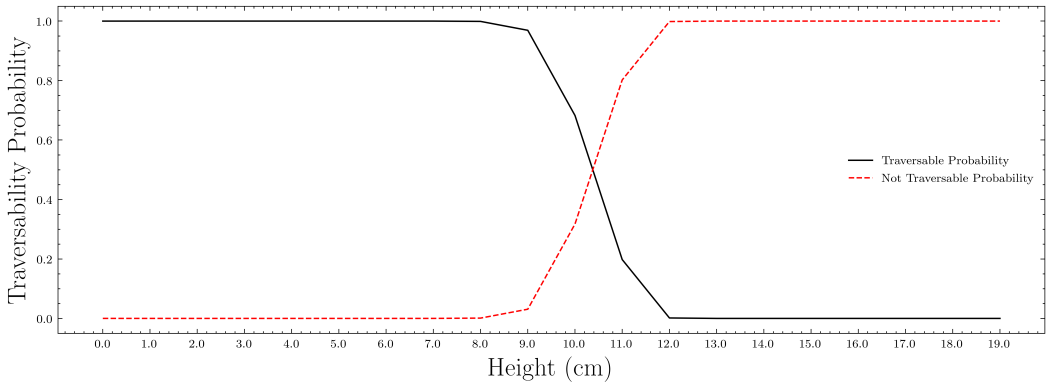
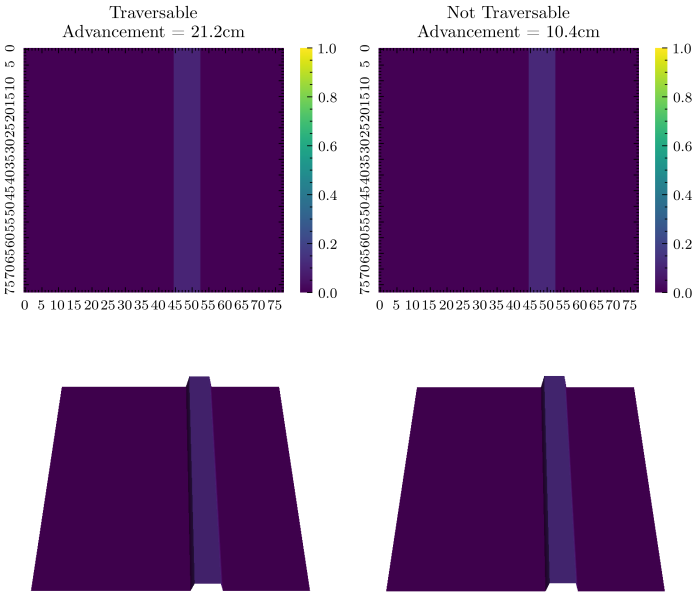



Figure 7. Traversability probabilities against walls height in front of Krock.

| Height(cm) | Prediction |
|---|---|
| 0 - 10 | Traversable |
| 10 - end | Not Traversable |

Table 2. Model prediction for the wall patches

We can compare the model's prediction with the advancement computed in the simulator using the same approach from the last section. The following figure shows the results from the last traversable patch and the first non traversable.




(a) height 9.7cm (b) Height 10cm

Figure 8. The last traversable and the first non traversable patches with a increasing height wall ahead of Krock.

In the first case, the simulator outputs and advancement of 39.5cm meaning that Krock was able to overcome the obstacle, while it fails in the second case. Correctly, the predictions match the real data.

Increasing height and distance walls

We can combine the previous experiments and test the model prediction against the ground truth for each height/distance combination. To reduce the number of samples and improve readability, we limit ourself to consider only patches with a wall tall between 5cm and 20cm, we know from previous sections patches with a value smaller and bigger obstacle are traversable and not traversable respectively. Similar, we set the wall's distance from Krock's head between 1cm to 30cm for the same reasons. The following image shows the traversability probability for each patch.

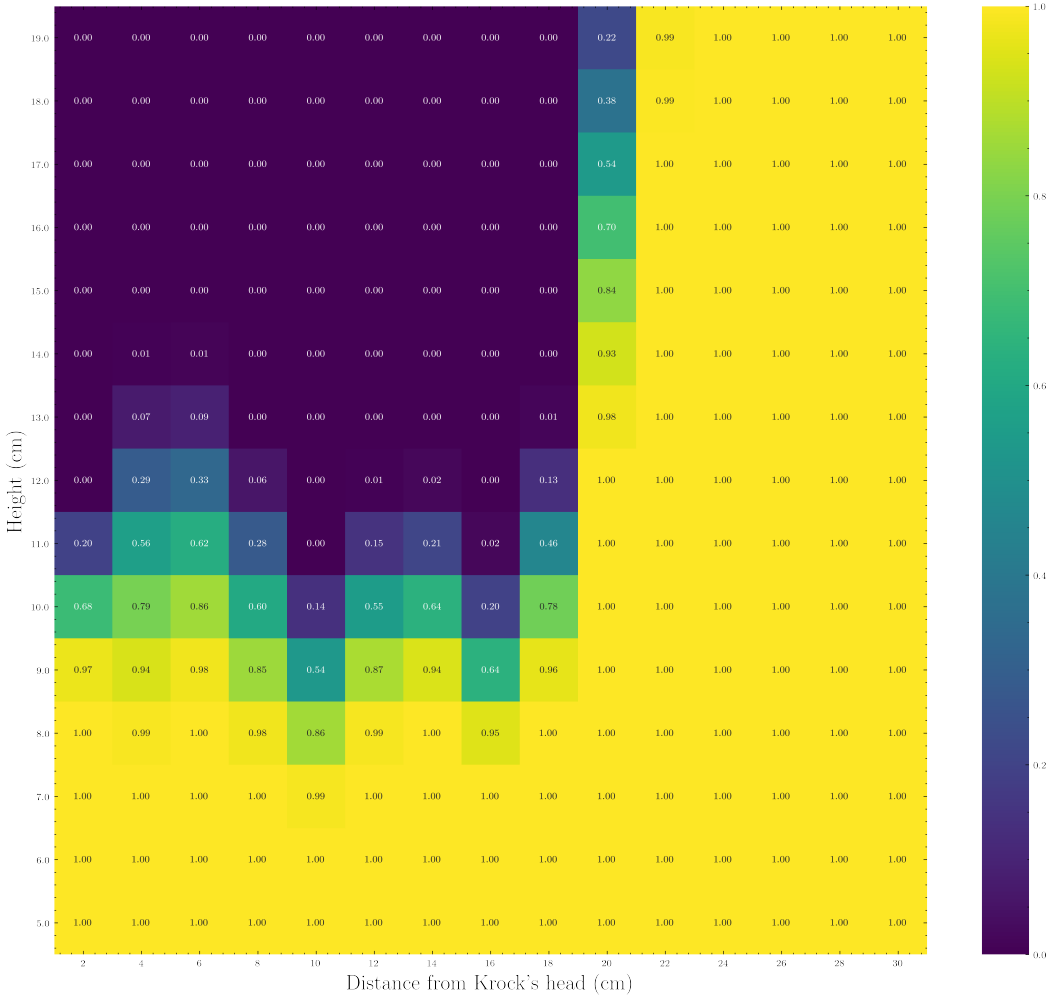



Figure 9. Traversability probabilities for patches with a wall of increasing height and distance form Krock's head.

missing ground truth heatmap

FINISH

Tunnel

do it

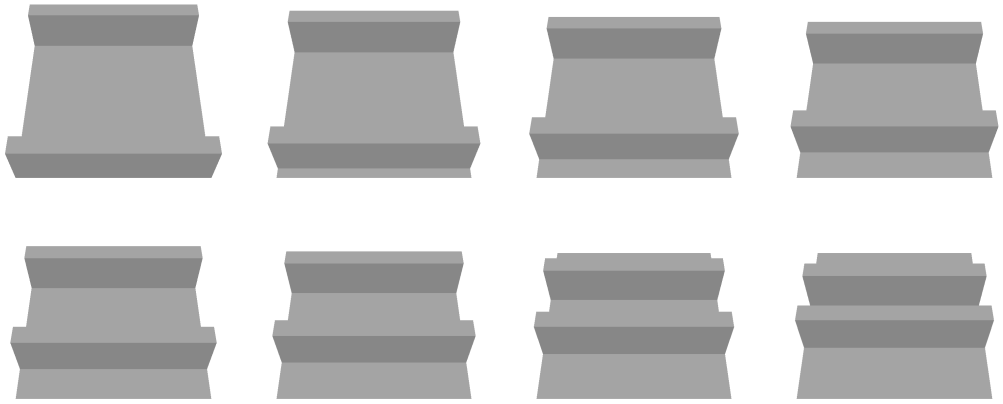


Figure 10. Some of the tested patches with tunnel at different distances.

Ramps

explain we had to square the linear ramps to create a small flat region

We generate twenty ramps with a maximum height from 0.25m to 4m. Below we plot the traversability probabilities against the maximum height of each ramp.

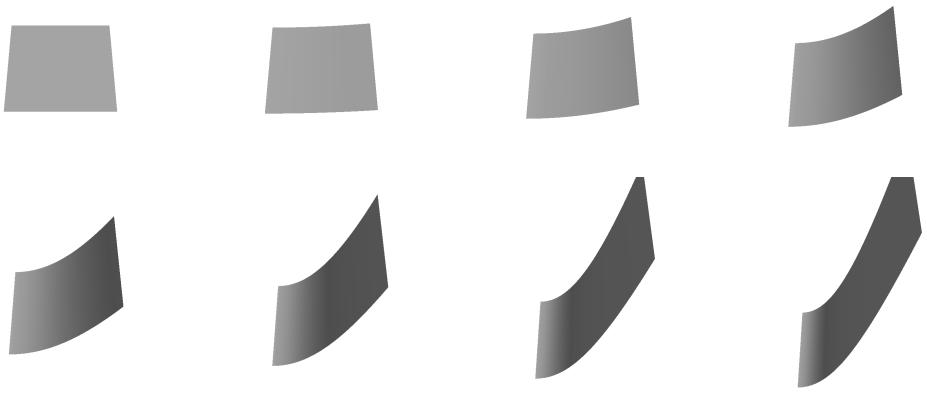


Figure 11. Some of the tested patches with steep ramps.

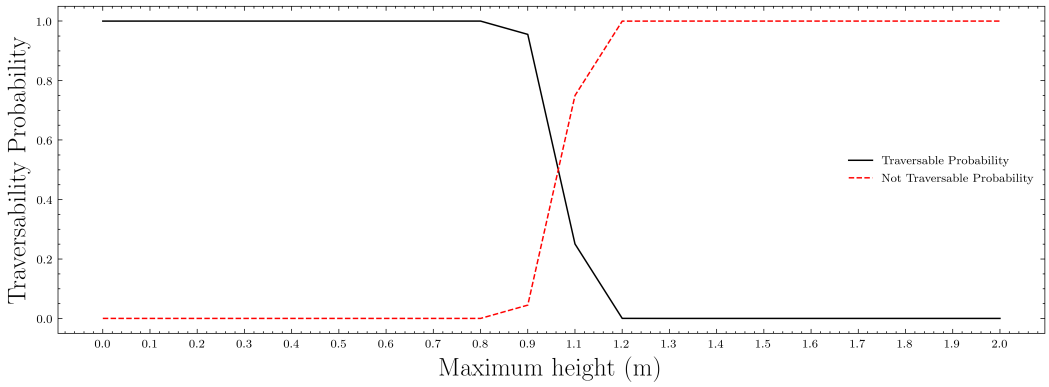



Figure 12. Traversability probabilities against maximum height of each ramp.

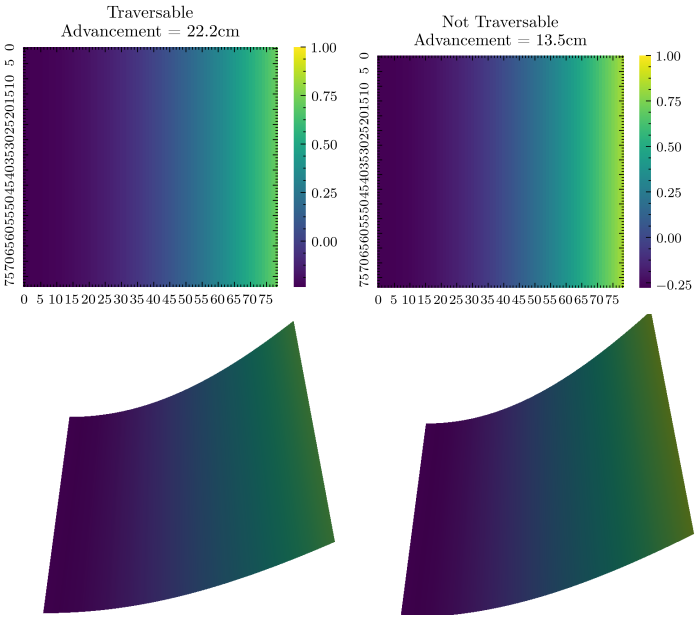
x labels are wrong, why?

The following table summarizes the results.

| Height(m) | Prediction |
|---|---|
| 0.5 - 1 | Traversable |
| 1 - end | Not traversable |

Table 3. Model prediction for the ramps patches

We test the last traversable patch and the first not traversable with the real advancement gather from the simulator.




(a) Height 0.94cm (b) height 1.1m

Figure 13. The last traversable and the first non traversable patches with a steep ramp ahead of Krock.

scale is wrong

Krock is able to traverse up to $1m$ height ramps, this is confirmed using the simulation.

We can add rocks to those patches to give Krock the ability to climb them better.

add rocks

Holes

do it

## a.3 Quarry Dataset

We start evaluating our model by using the *Quarry* map. We expect the model to correctly classify the patches with easy to see features such as big obstacles, steep ramps and holes. Unfortunately, the dataset is not trivial and most of the patches are challenging to classify even to human eye.

For instance, if look at patches, for some of them is not so easy to estimate the advancement by human eye. This is due to the specific robot locomotion that depends on the starting pose. Our goal in this section is to explain the model predictions on different inputs.

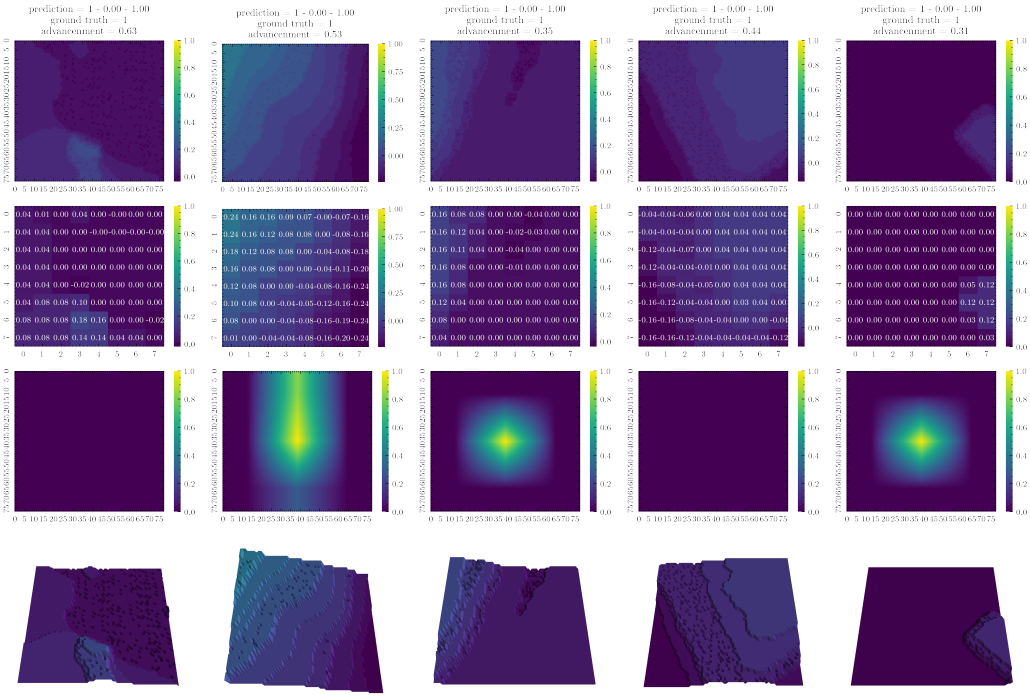



Figure 14. Best

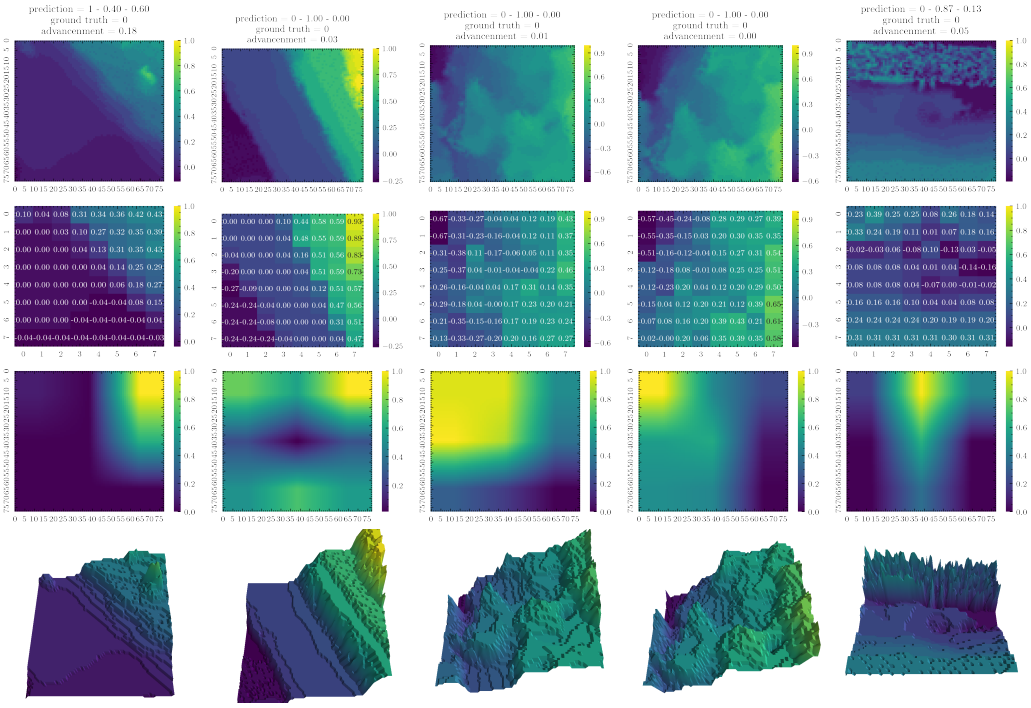



Figure 15. Worst

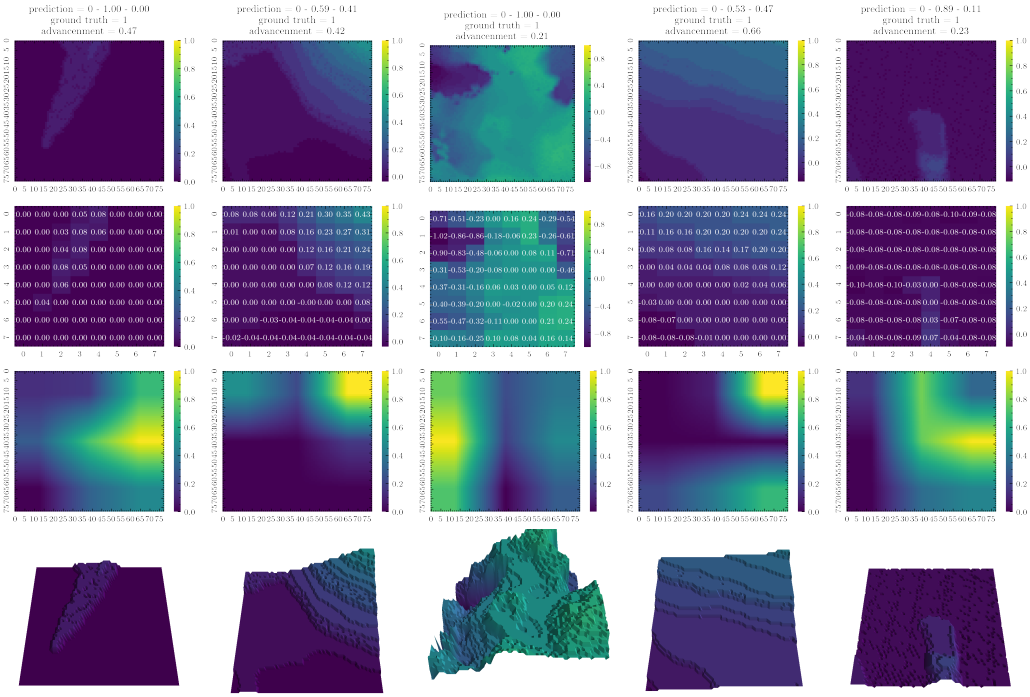



Figure 16. False Positive

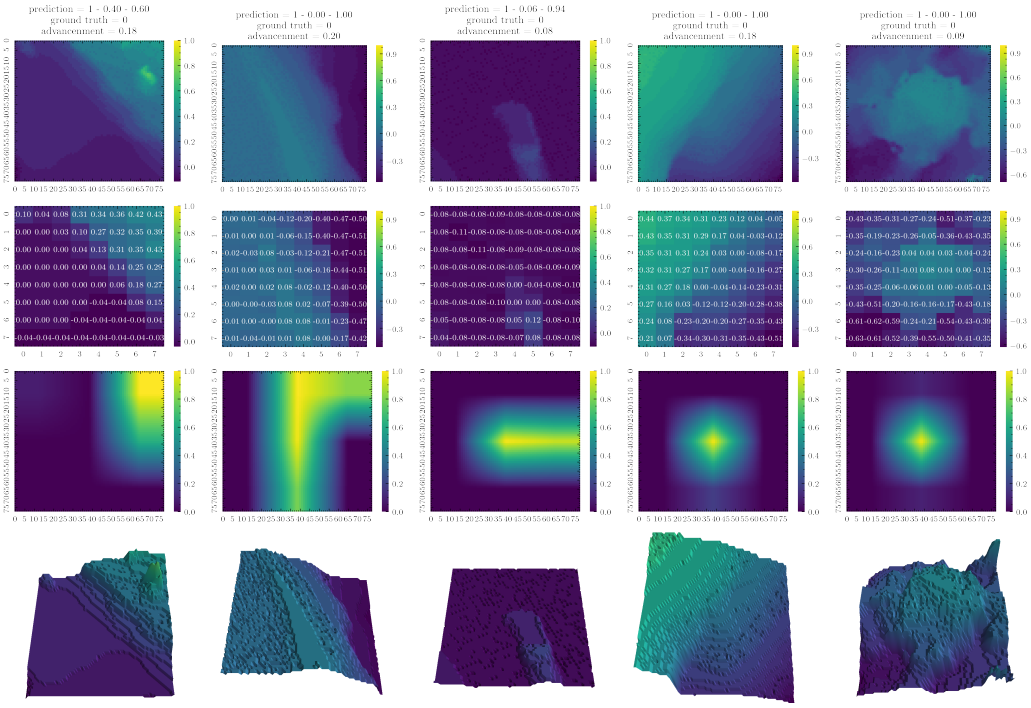



Figure 17. False Negative