

The many faces of reliability of visual perception for autonomous driving

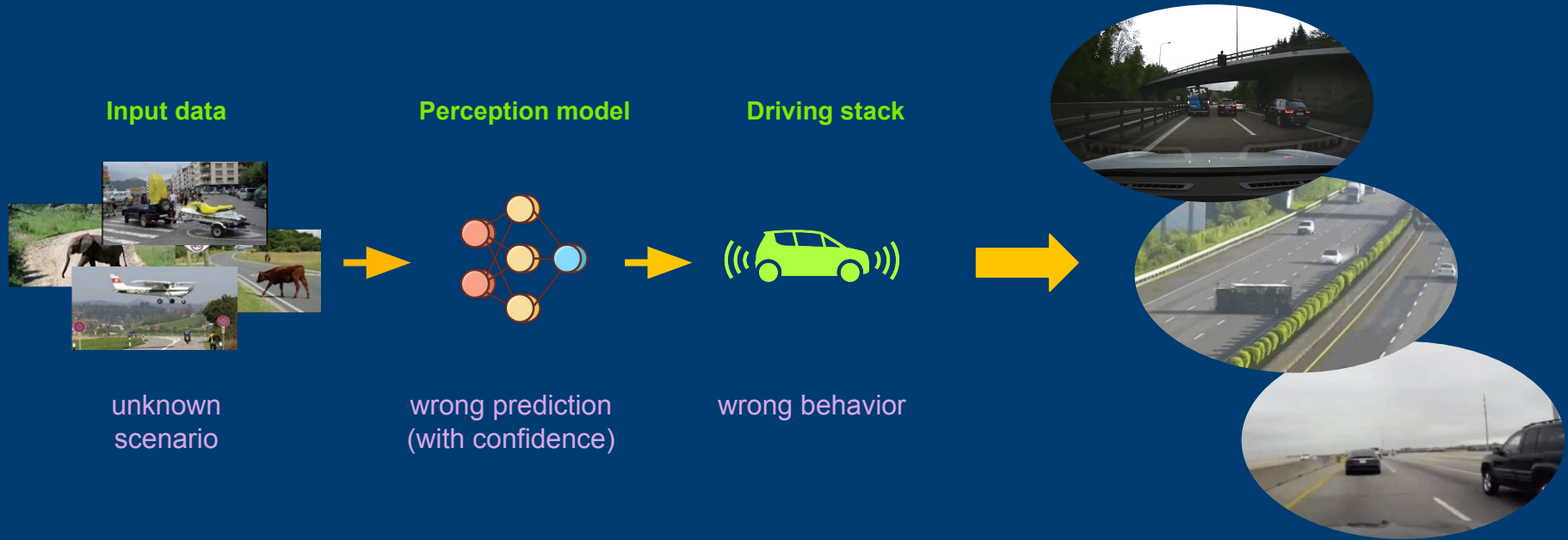
Performance monitoring

Andrei Bursuc

valeo.ai

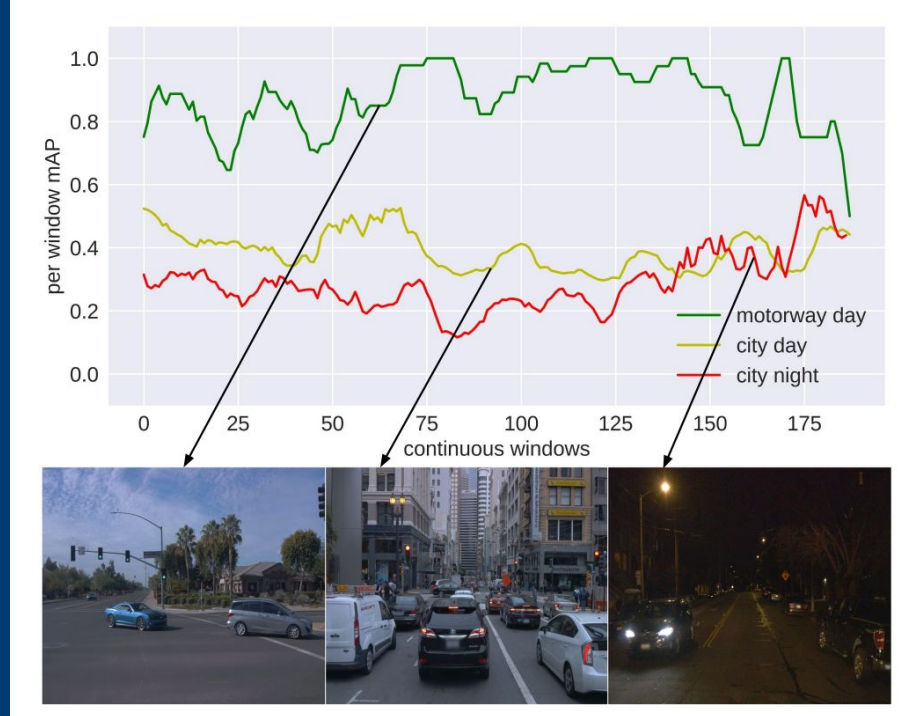
Learning to identify complex situations

Challenges of driving automation



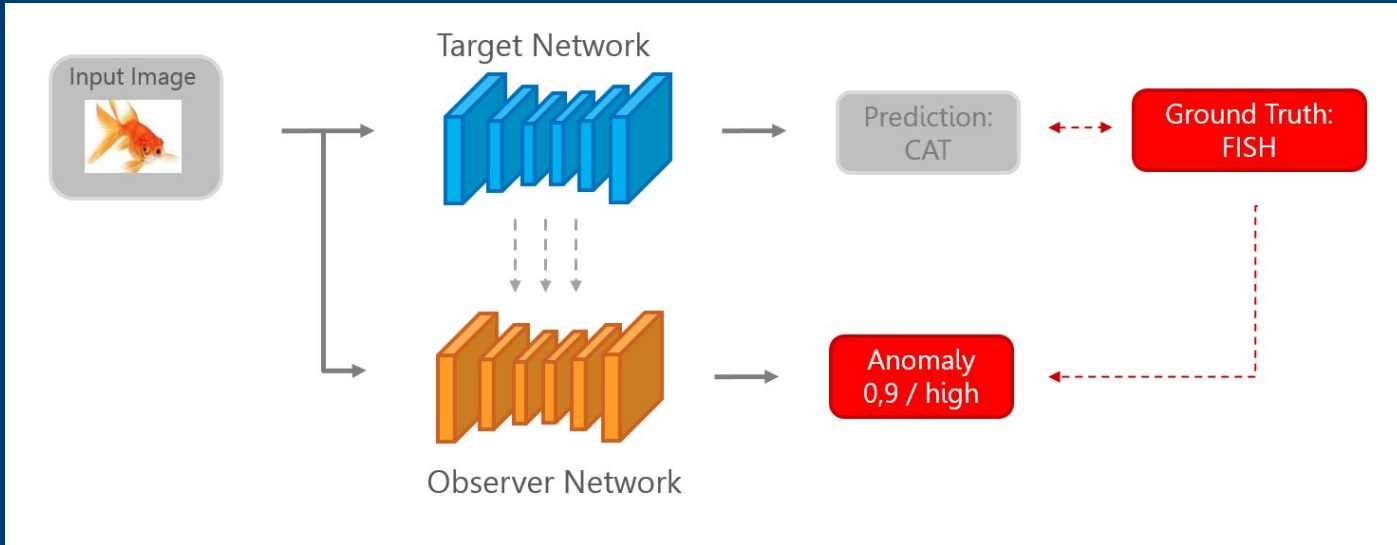
How to identify/prevent incorrect predictions that can cause system failures?

Challenges of driving automation



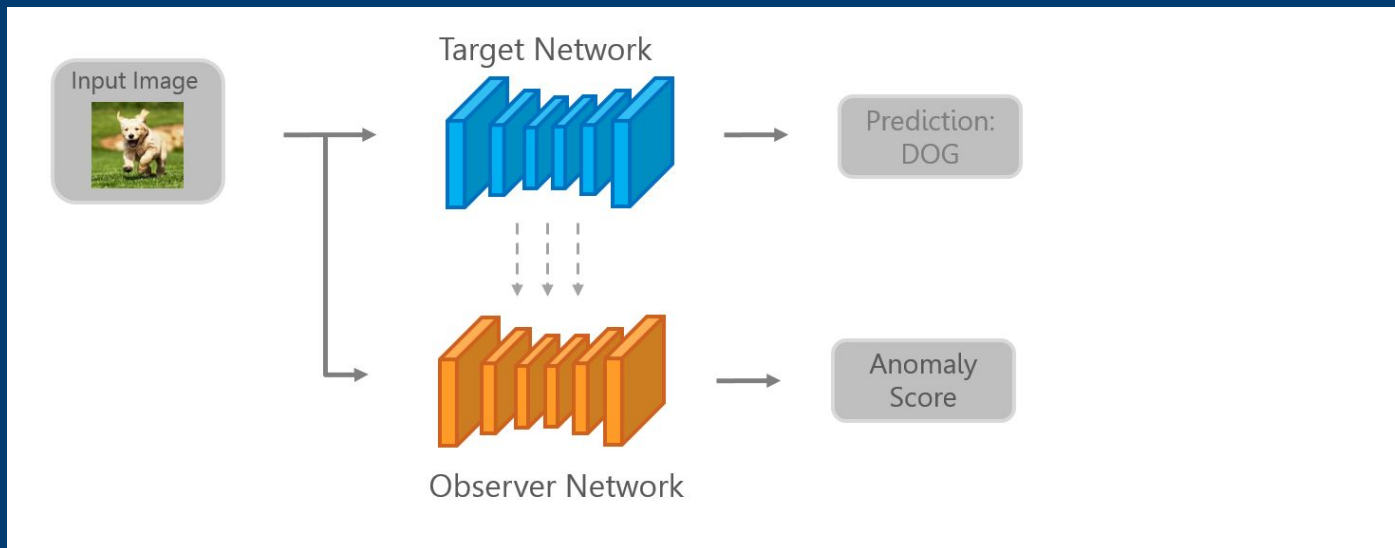
Performance can fluctuate depending on conditions and traditional engineered monitoring solutions cannot deal alone with the complexity of the world.

Neural monitoring - Observer Networks



- **Target Network:** (pre-trained) neural network for a task of interest
- **Observer Network (ObsNet):** auxiliary network connected to Target Network
 - Can have access to **internal activations and predictions** of Target
 - Trained to **predict failures** of Target Network
 - Produces confidence/failure/anomaly score

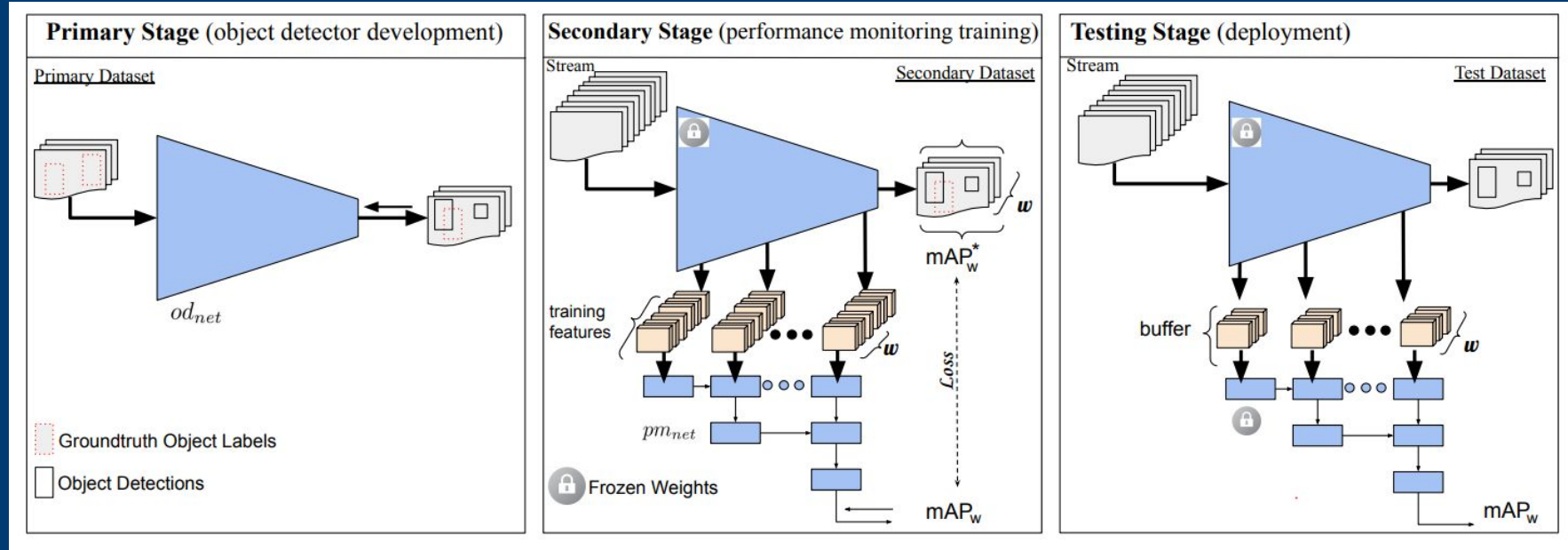
Neural monitoring - Observer Networks



- **Benefits:**
 - Generic, flexible, fast, memory-efficient
- **Drawbacks:**
 - Needs a dedicated train set (Target Network makes few errors)
 - May not generalize to OOD data, not available at train time

A major challenge with Observer Networks is related to the availability of hard training data.

Monitoring object detection performance

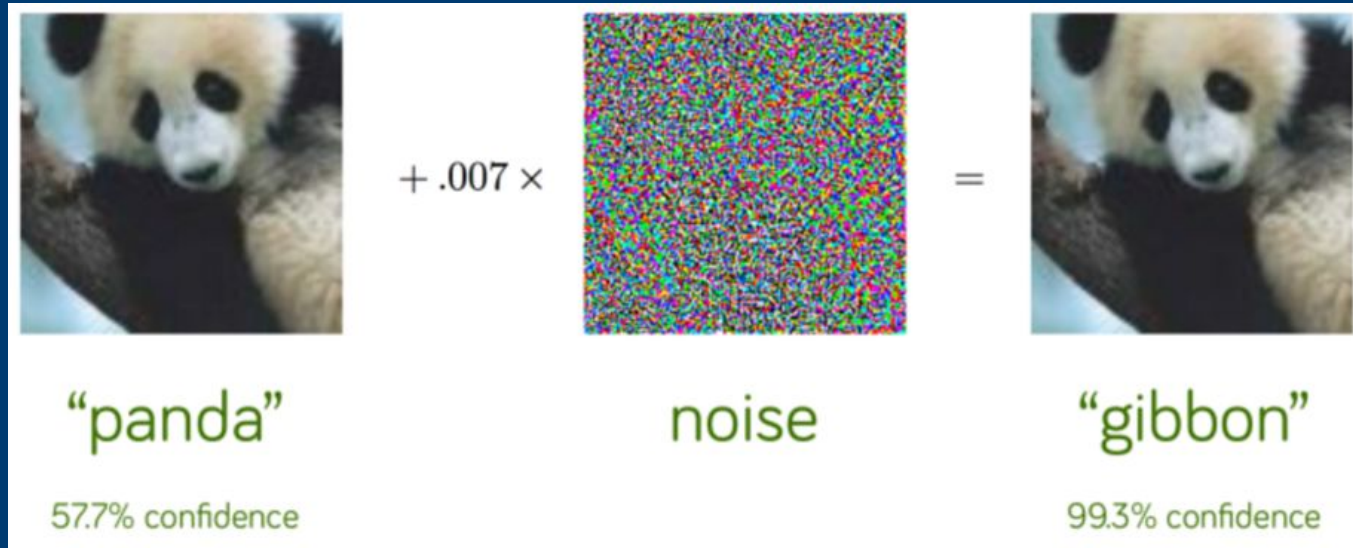


Earlier approaches leveraged temporal information to compile per sequence statistics and predict mAP



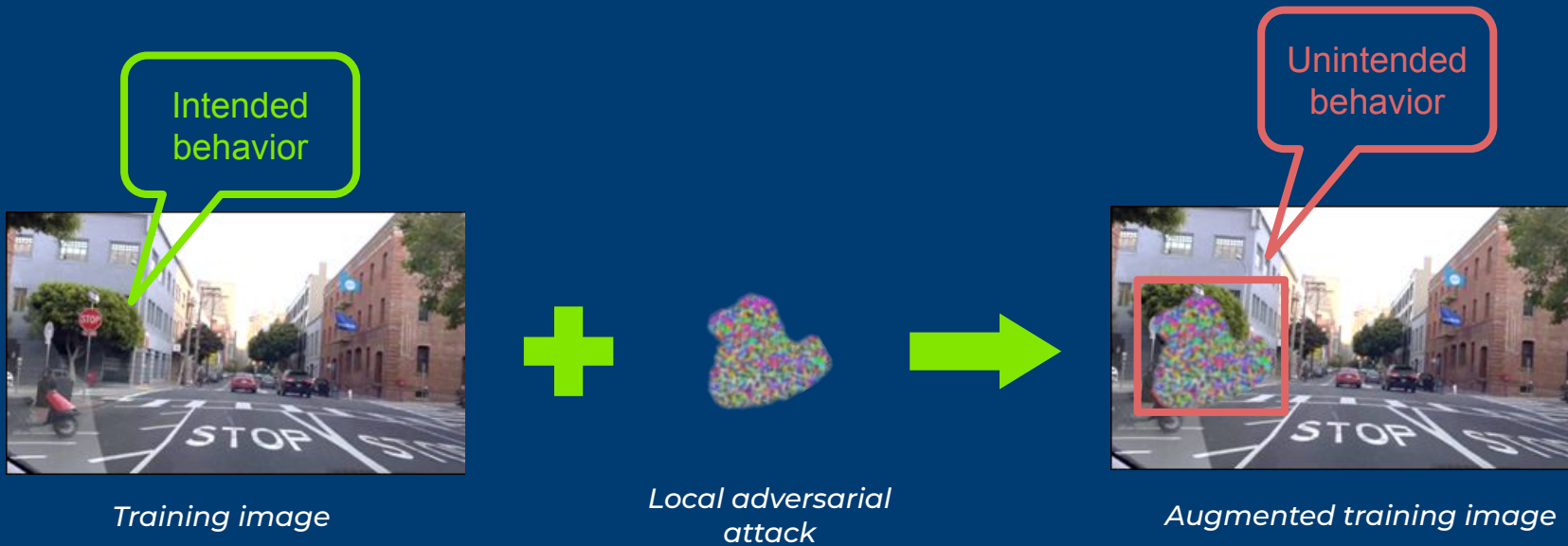
What if we make the Target fail and learn from that?

Adversarial Attacks



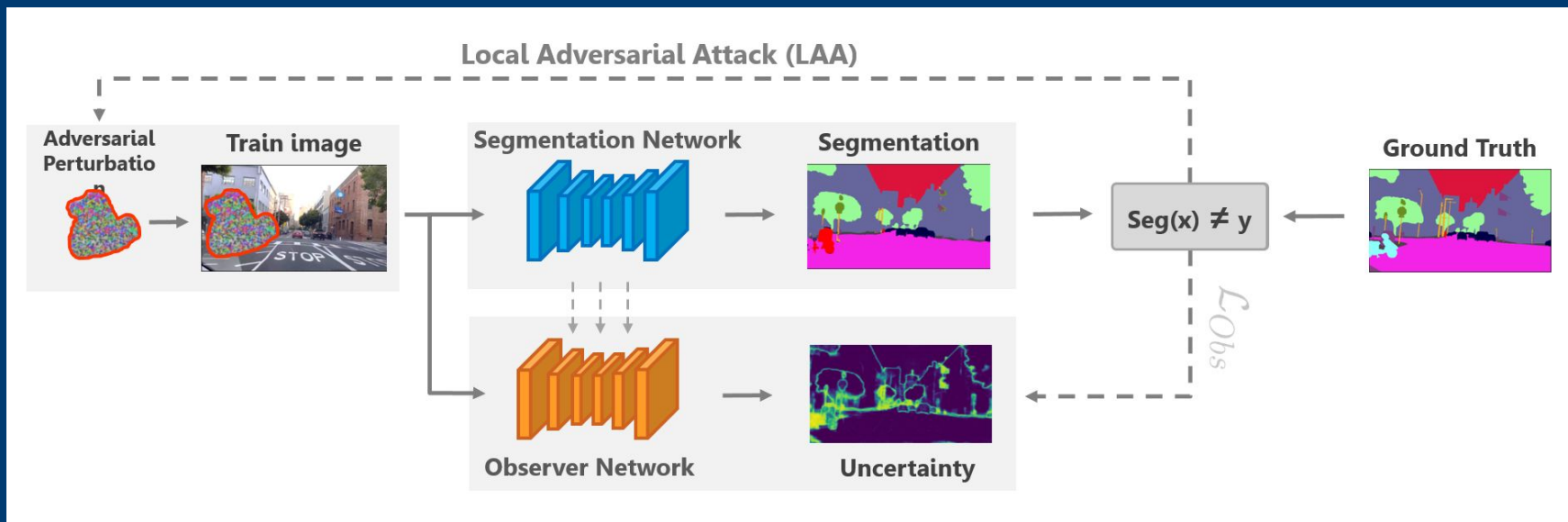
- Neural Networks can be fooled by perturbing the input image with constructed noise
- We use Adversarial Attacks in order to **trigger failures** of the target network

Local Adversarial Attacks



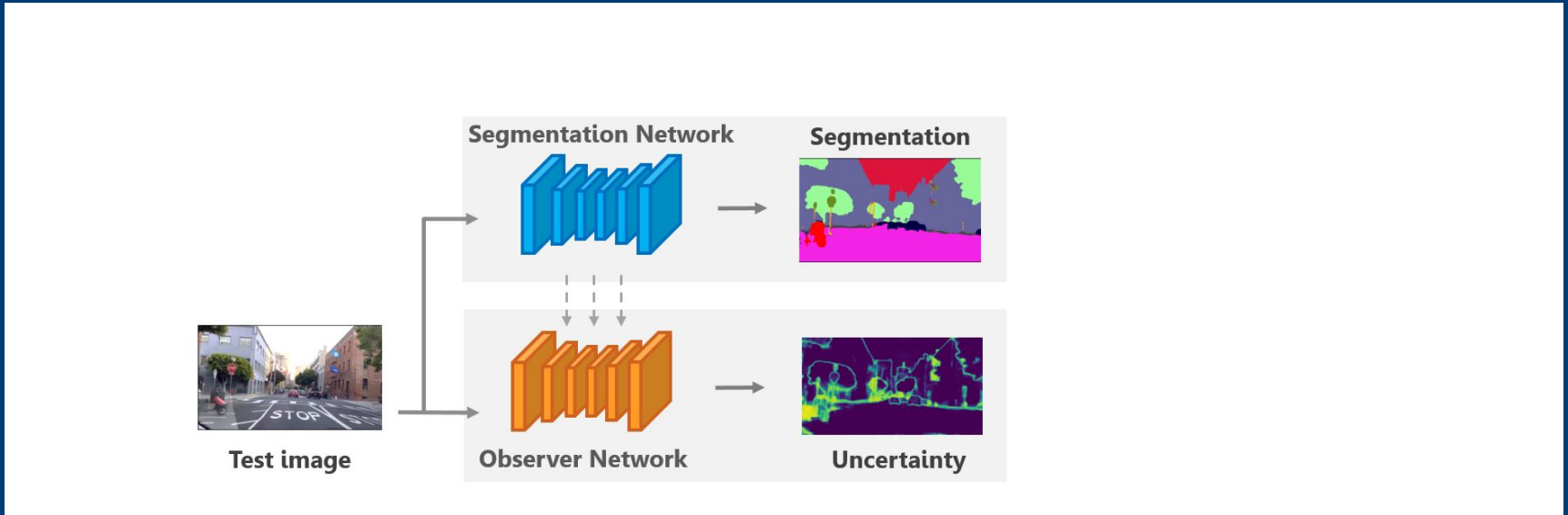
- Use Local Adversarial Attacks (LAA) to **“hallucinate”** new class
- Edit a part of the image to decrease the target prediction in this location
- Encapsulate attack in random shape as **proxy for unknown objects**

ObsNet - training setup



- The Observer learns failure behavior patterns of Target under attacks

ObsNet - at runtime



- Generate classification predictions from Target and uncertainty from Observer

ObsNet Results

Method	Fpr95Tpr ↓	AuPR ↑	AuRoc ↑	ACE ↓
Softmax [HG17]	63.5	95.4	80.1	0.633
Void [BSN+19]	68.1	92.4	75.3	0.499
AE [HG17]	92.1	88.0	53.1	0.832
MCDA [AB18]	61.9	95.8	82.0	0.411
Temp. Scale [GPSW17]	61.8	95.8	81.9	0.287
ODIN [LSL18]	<u>60.6</u>	95.7	81.7	0.353
ConfidNet [CTBH+19]	61.6	95.9	81.9	0.367
Gauss P [MAG+20]	61.3	96.0	82.5	0.384
Deep Ensemble [LPB17]	60.3	<u>96.1</u>	82.3	0.375
MCDropout [GG16]	61.1	96.0	82.6	0.394
ObsNet + LAA	60.3	96.2	82.8	<u>0.345</u>

Method	Fpr95Tpr ↓	AuPR ↑	AuRoc ↑	ACE ↓
Softmax [HG17]	65.5	94.7	80.8	0.463
Void [BSN+19]	69.3	93.6	73.5	0.492
AE [HG17]	84.6	92.7	67.3	0.712
MCDA [AB18]	69.9	97.1	82.7	0.409
Temp. Scale [GPSW17]	65.3	94.9	81.6	0.323
ODIN [LSL18]	61.3	95.0	82.3	0.414
ConfidNet [CTBH+19]	60.1	98.1	90.3	0.399
Gauss P [MAG+20]	48.7	98.5	90.7	0.449
Deep Ensemble [LPB17]	51.7	98.3	88.9	0.437
MCDropout [GG16]	45.7	98.8	92.2	0.429
ObsNet + LAA	44.7	98.9	92.7	<u>0.383</u>

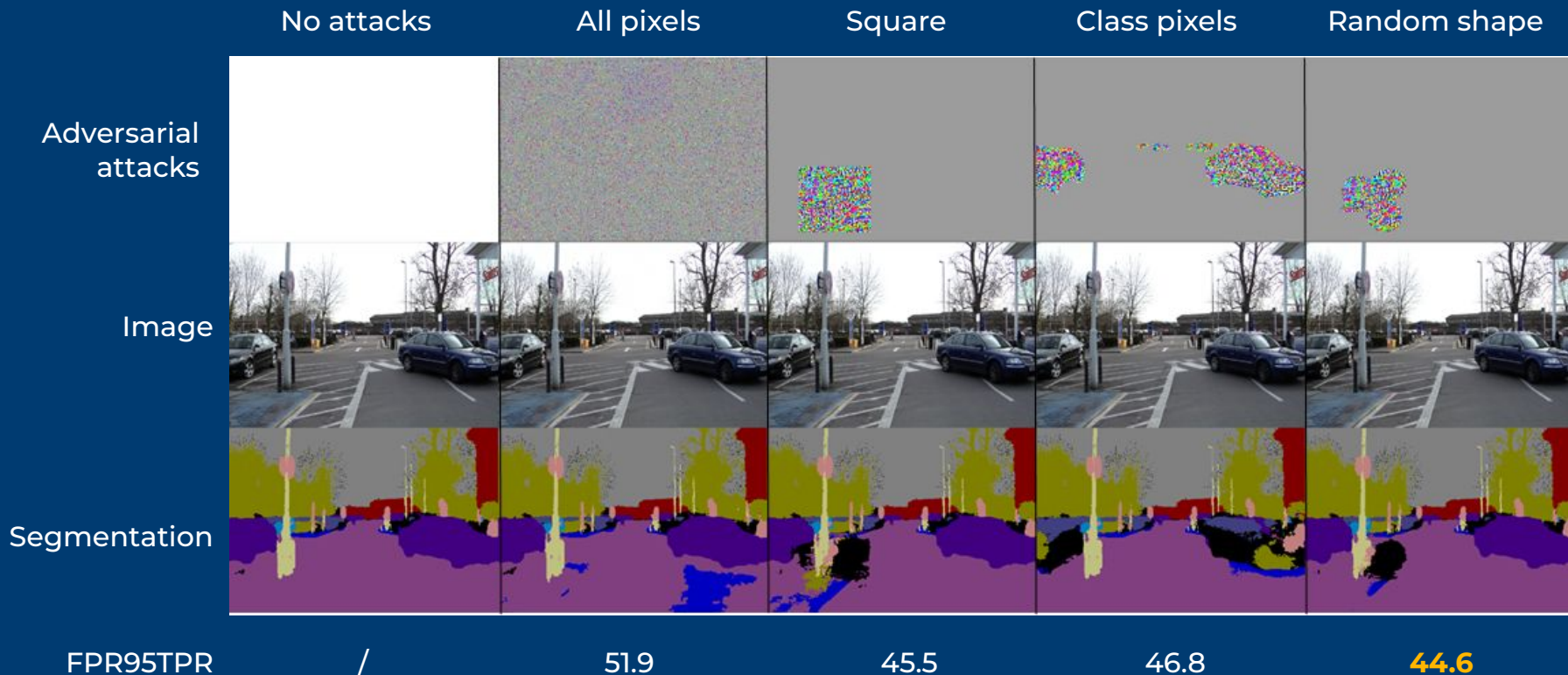
BDD Anomaly (OOD: train, motorcycle)



StreetHazards



Impact of attack shape



ObsNet Quantitative Results

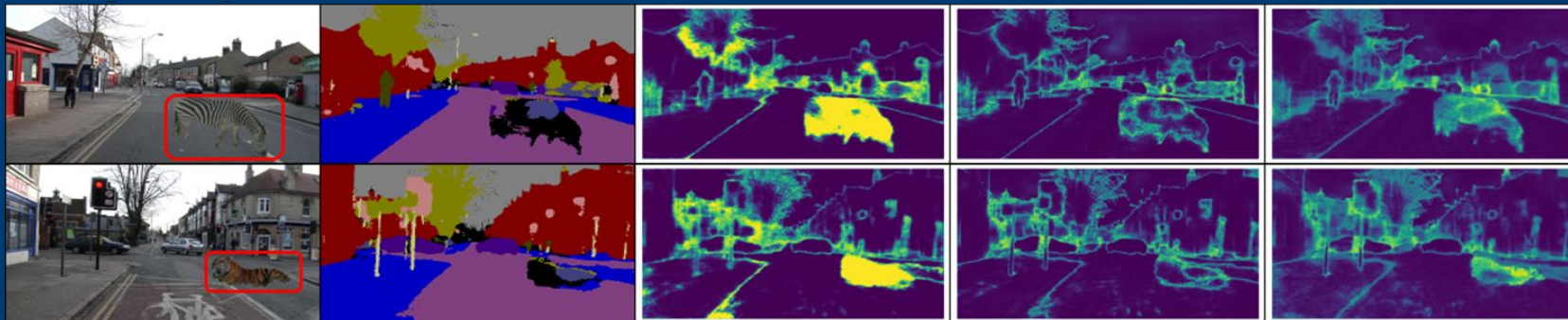
Input image

Segmentation

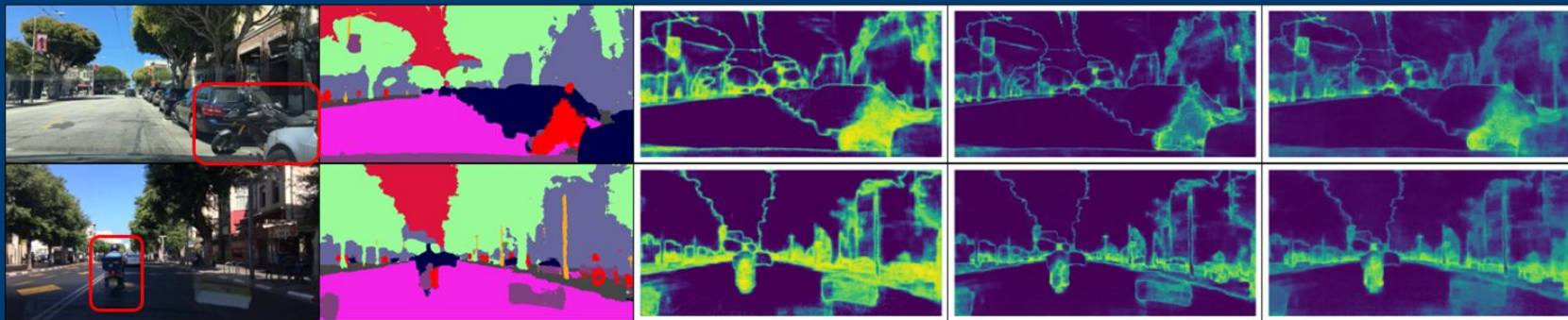
ObsNet

Softmax

MCDropout



CamVid OOD

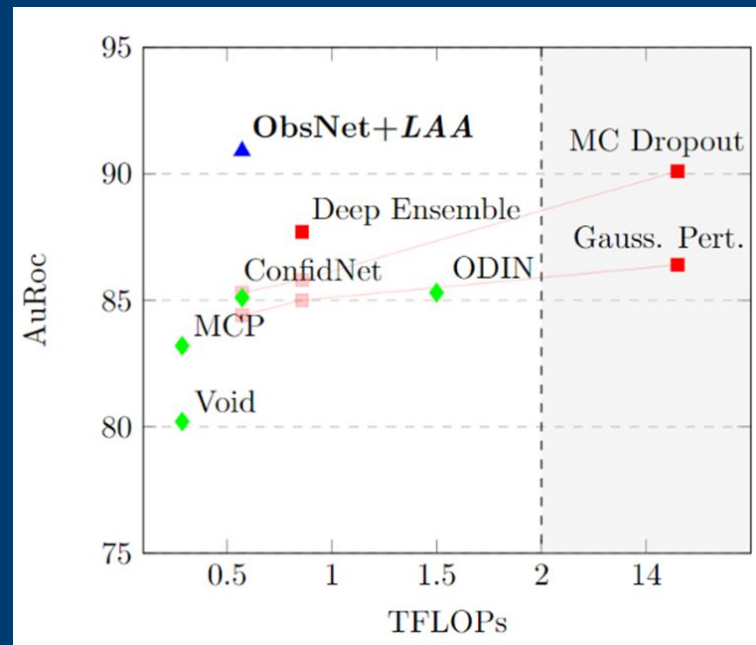


BDD Anomaly

ObsNet

Takeaways

- Leverage adversarial attacks to **find blind spots** in the Target Network
- Focus on localized regions to mimic unknown objects
- Can generate **infinite negative** examples
- Idea can be applied for **regression**, e.g., SLURP
- **Cannot localize precisely** the anomalous object
- The predicted error is generic, **not easy to match a specific type of uncertainty**



Precision vs test-time computational cost

The end.