S9932: LEARNING TO BOOST ROBUSTNESS FOR AUTONOMOUS DRIVING

Bernhard Firner, March 20, 2019

AUTONOMOUS DRIVING

Sounds simple

Actually pretty difficult

Can start with sub-domains to simplify

Robustness must be a goal from the start

“The first 90 percent of the code accounts for the first 90 percent of the development time. The remaining 10 percent of the code accounts for the other 90 percent of the development time.”

Tom Cargill, Bell Labs

NINETY-NINETY RULE

Without robustness we can be perpetually “almost there”

If testing is too hard it won’t happen until it is too late

There is always one more corner case left to take care of…

MANY CORNER CASES

NVIDIA NEW JERSEY GROUP

Located in historic Bell Labs building

Diverse roads nearby

Diverse weather in NJ

Early focus on **robustness** and **testing**

MAIN IDEAS IN THIS TALK

Train on lots of data and test often

Learn from human behavior with minimal hand-labeling

Diversity – Combines well with traditional systems to make them more robust

Scalability – Low effort to add more training data

Create new testing tools that are **fast**, **realistic**, and **reproducible**

PILOTNET

PILOTNET OVERVIEW

Use human driving to create training labels for neural network

Low labeling overhead means the approach can scale to required data diversity

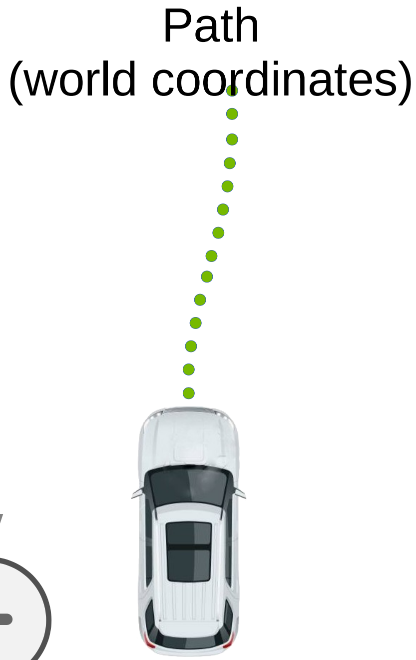
Ensemble with other approaches to get diversity and robustness

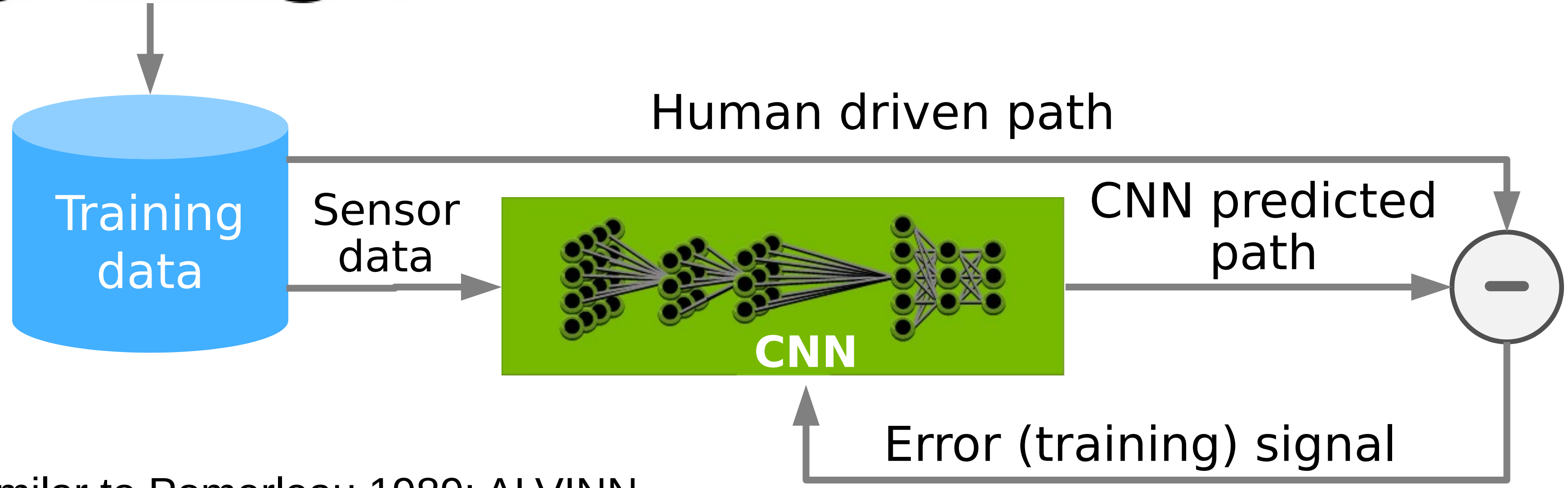
Just one piece of the autonomous system

THE PILOTNET APPROACH

Learning to predict a human path

Record data from lots of humans driving their cars:
- ➢ Sensor data
- ➢ Human driven path



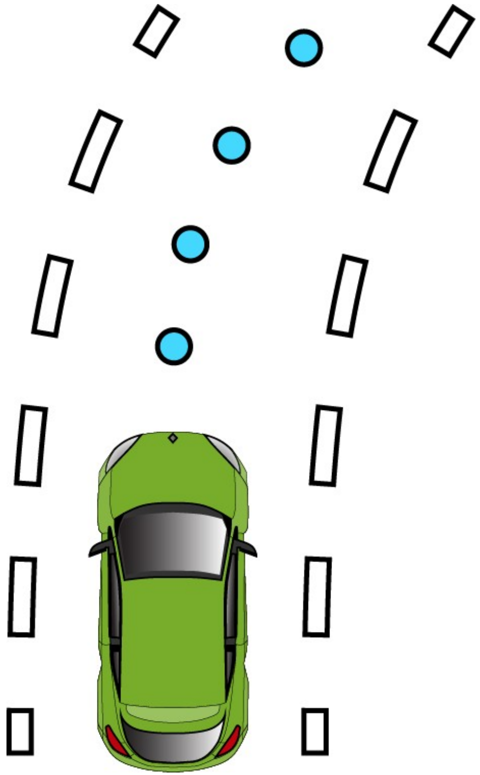


Similar to Pomerleau 1989: ALVINN
http://repository.cmu.edu/cgi/viewcontent.cgi?article=2874&context=compsci

TRAINING LABELS

World space coordinates

- Create a path (x,y,z) from egomotion using IMU and GPS
- Image input, world space predictions (e.g. meters)
- Predict where a human would drive the vehicle



TRAINING LABELS

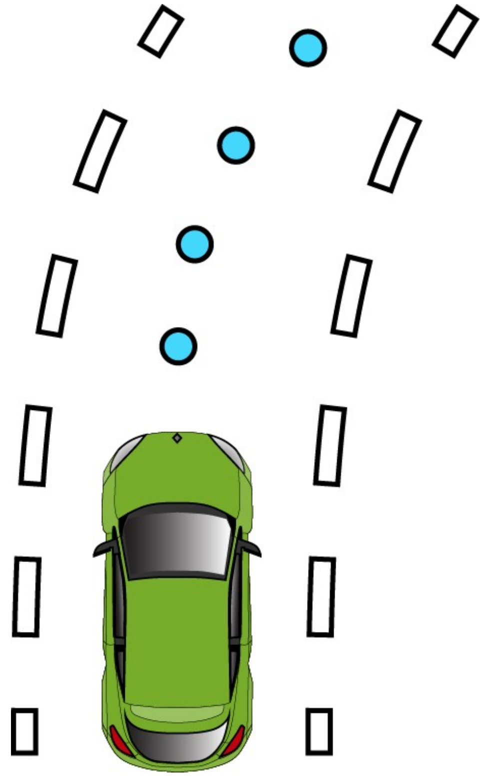
Dealing with different behaviors

Not all data is the same though!

Sequences are curated by driving quality and maneuver

For example, lane changes, splits, junk

The model can predict different paths for different objectives



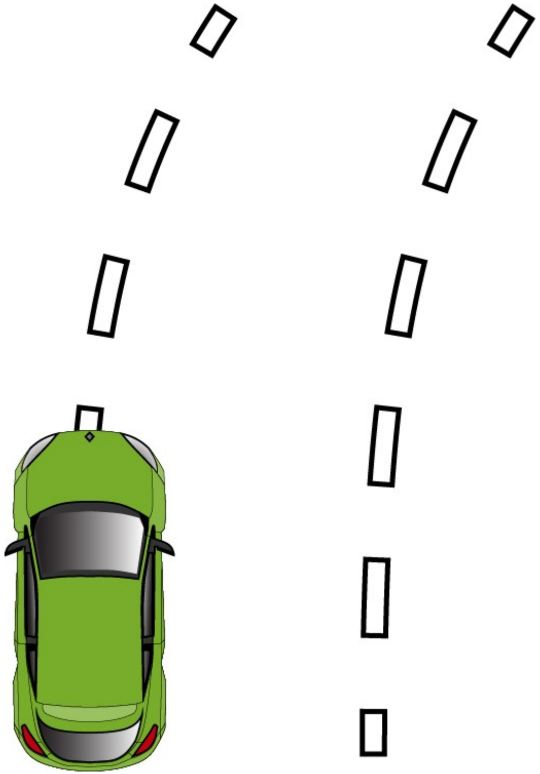
TRAINING LABELS

Dealing with the unexpected

Sometimes something bad happens

How can the neural network recover?

We can't collect training data with drivers drifting all over, that isn't safe.



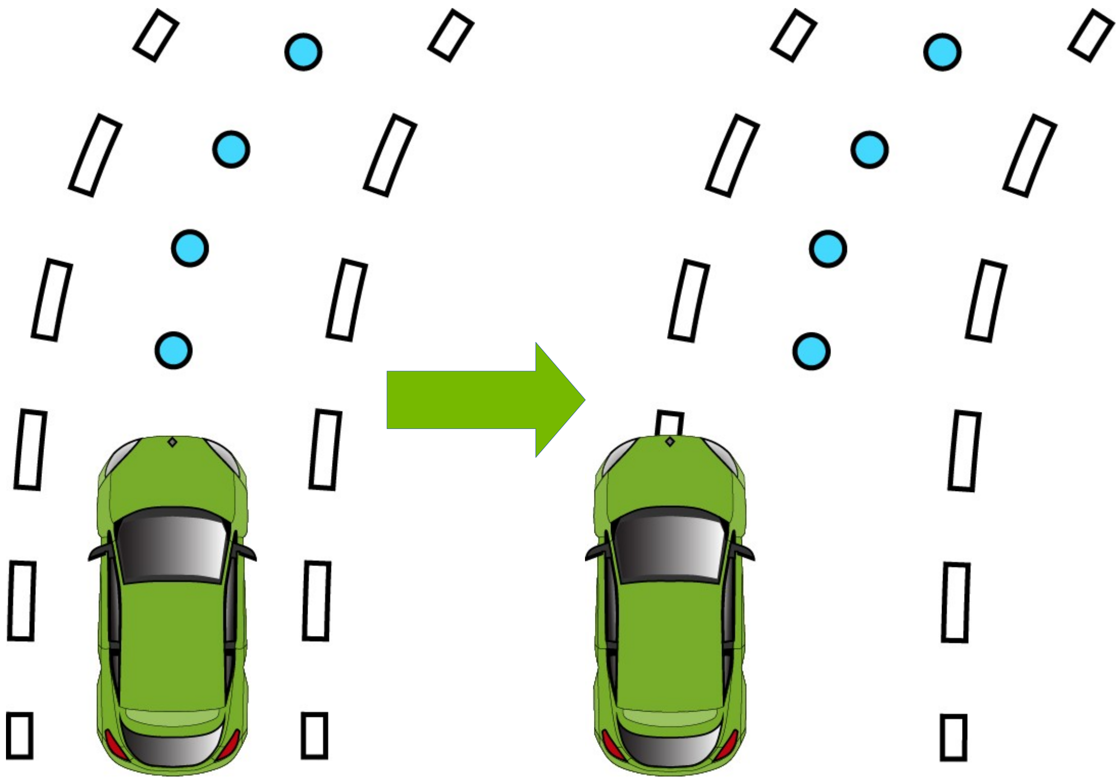
TRAINING LABELS

Augmenting the data

Start with good data

Perspective transform to a “bad” spot

Use the “good” coordinates as the label



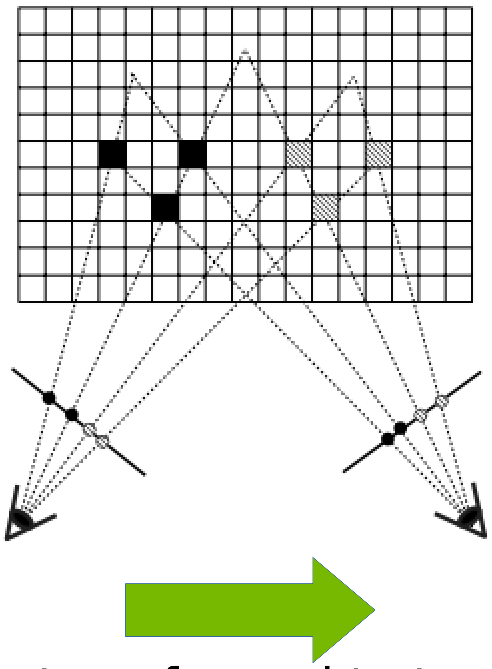
DATA AUGMENTATION

Under the hood

Image are transformed from the recorded views to a target camera using a viewport transform.

The image is rectified to a pinhole source, the perspective is transformed, then we re-warp to the target camera.

This assumes a flat world, so there is some distortion.



Source transformed to target

DATA AUGMENTATION

Under the hood

We collect with a camera that has a greater field of view than the driving camera

This allows us to simulate a field of view shifted to the side without running out of pixels

Also allows us to collect with a camera in one location and transform to a camera in another location for driving

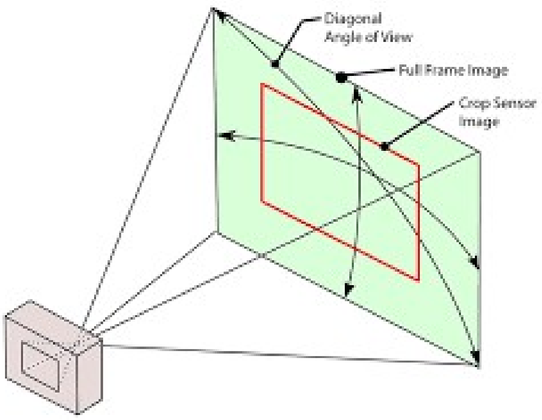


IMAGE AUGMENTATION EXAMPLE

Target camera

Source camera

IMAGE AUGMENTATION EXAMPLE

Target camera

Rectify

Warp

DRIVING WITH IMPLICIT RULES

Scale capabilities with data

WORLD SPACE PREDICTIONS

Address limitations of image-space predictions

2D Trajectory

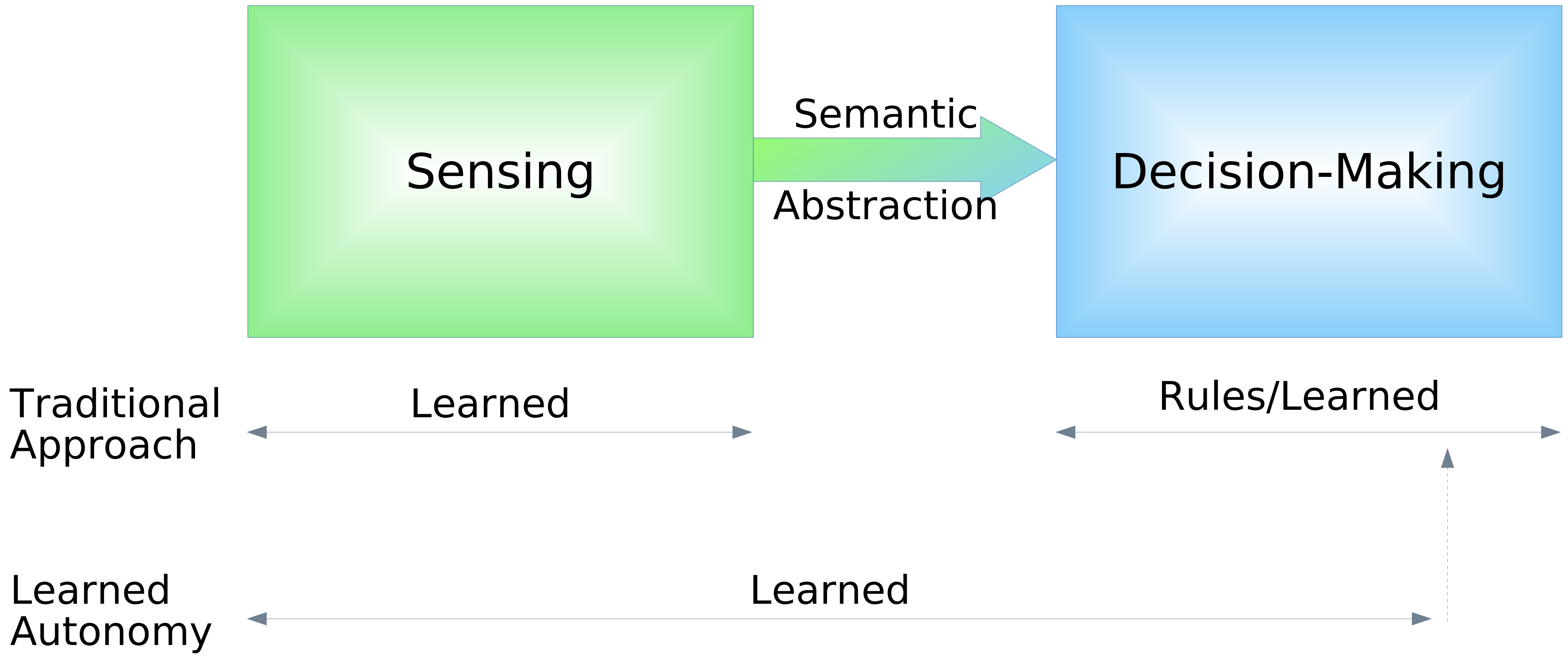
3D Trajectory

LEARNED TURNS

Learn different humans behaviors

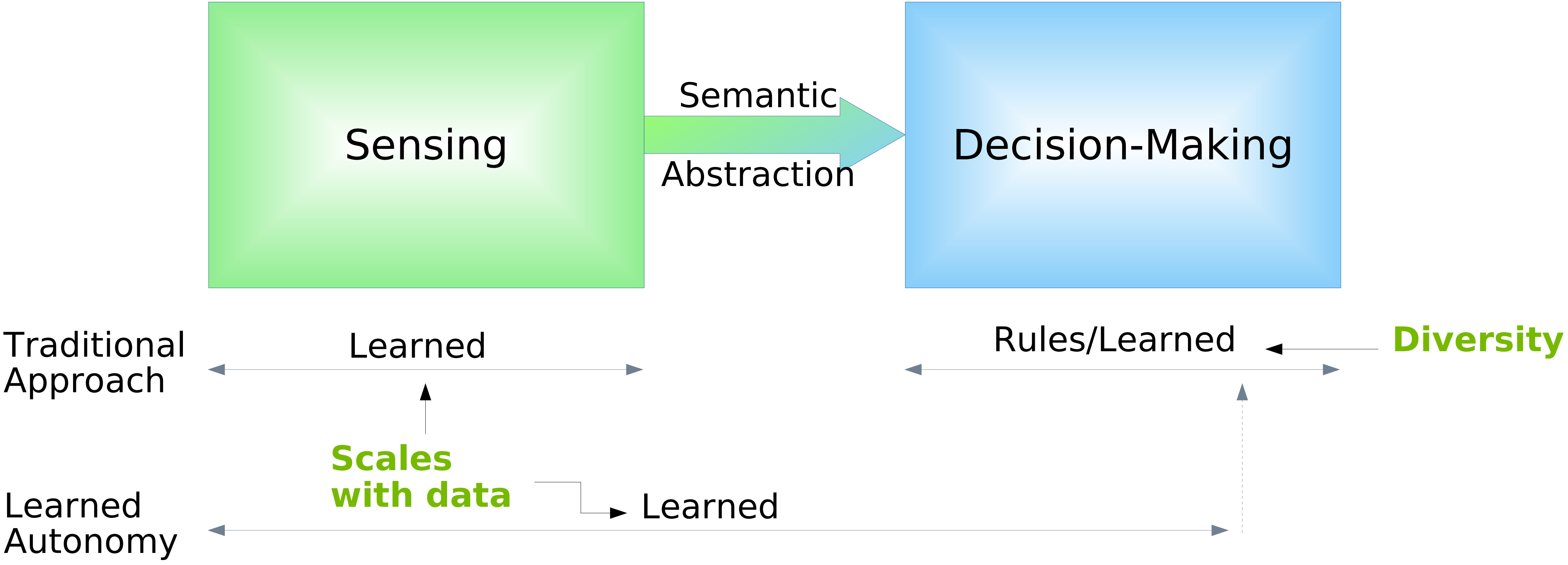
PILOTNET GOALS RECAP

Robustness through scalability and diversity



PILOTNET GOALS RECAP

Robustness through scalability and diversity



EVALUATION AND TESTING

TESTING

How do we know it’s working?

Real-world testing is the gold standard

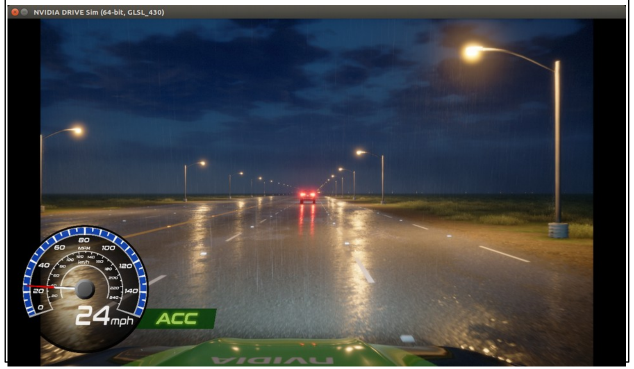
However, it can be **slow, dangerous,** and **expensive**

Results are also **subjective**

Simulation seems like a reasonable substitute

SIMULATION

- Addresses issues with real-world testing
- Test set can be created at will



SIMULATION

Let’s just create a photo-realistic world for testing

Safe, fast, and reproducible

SIMULATION DRAWBACKS

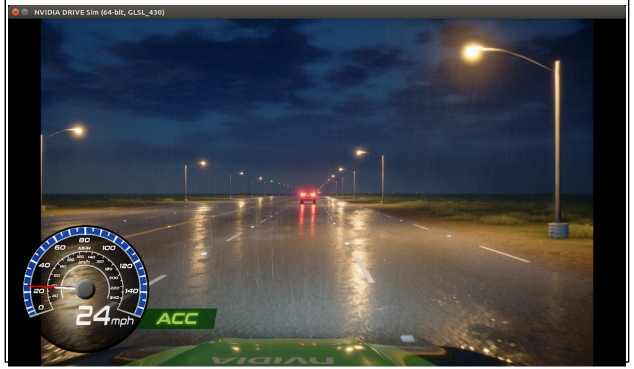
Will only test what we remember to simulate

We may remember one thing (rain) and forget another (snow melt)

SIMULATION

- Addresses issues with real-world testing
- Test set can be created at will

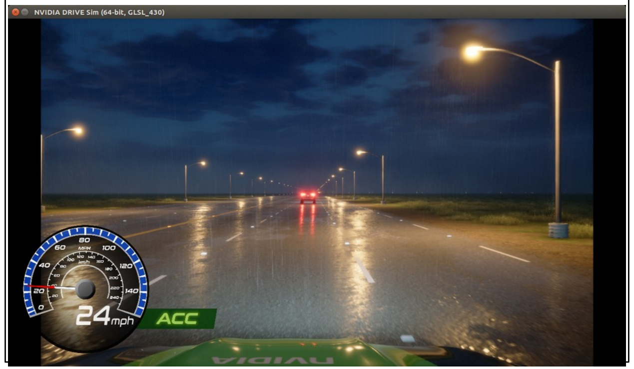
- Difficult to correctly model all behaviors and distributions



SIMULATION

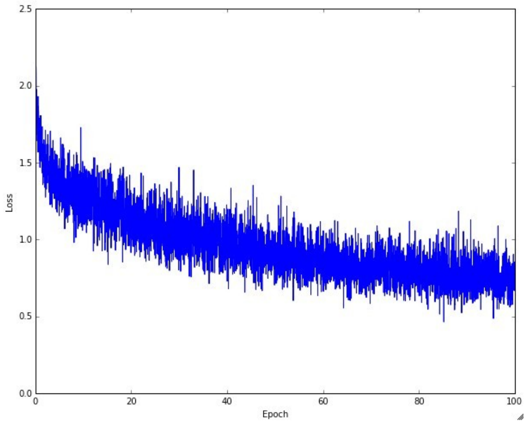
- Addresses issues with real-world testing
- Test set can be created at will

- Difficult to correctly model all behaviors and distributions

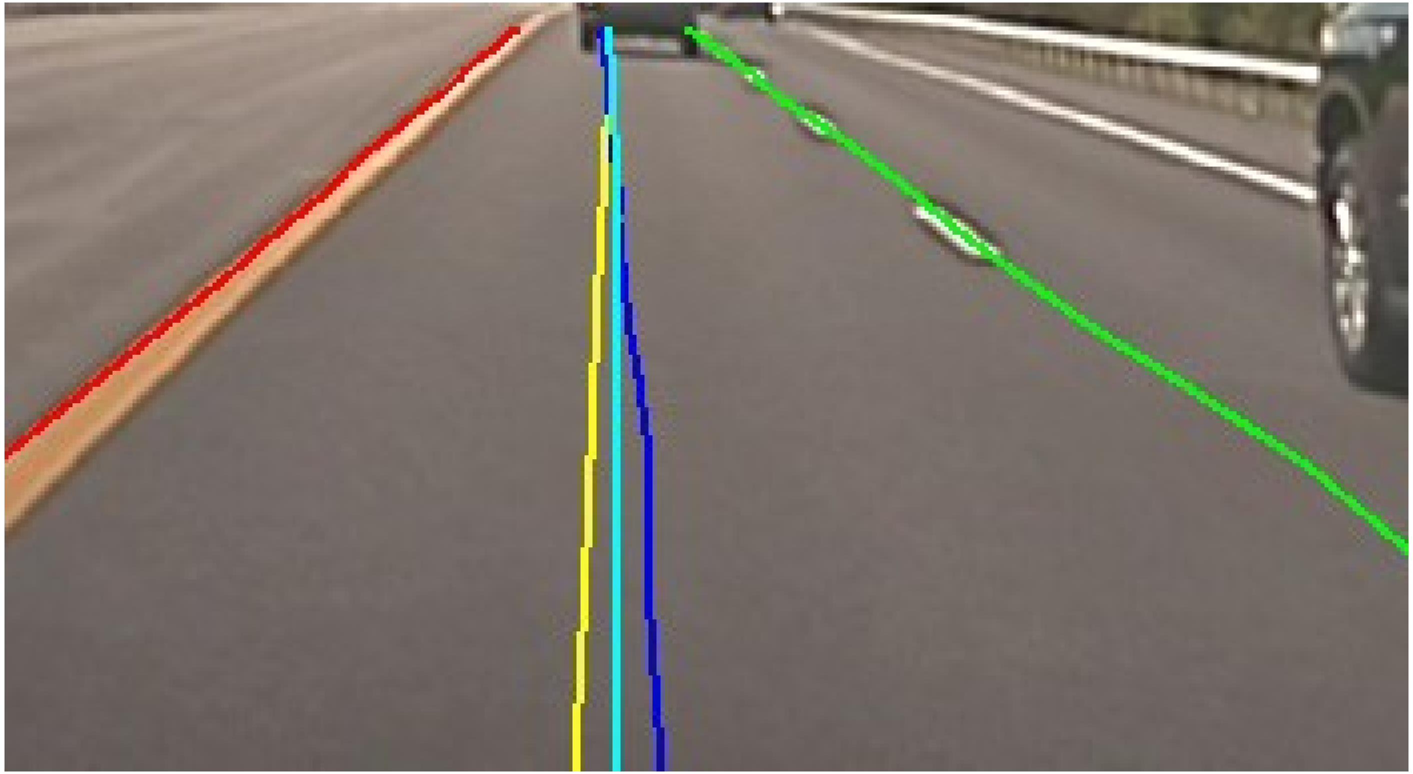


PREDICTION ERROR

- Uses real data!
- Simple! (Mean Squared Error, etc)

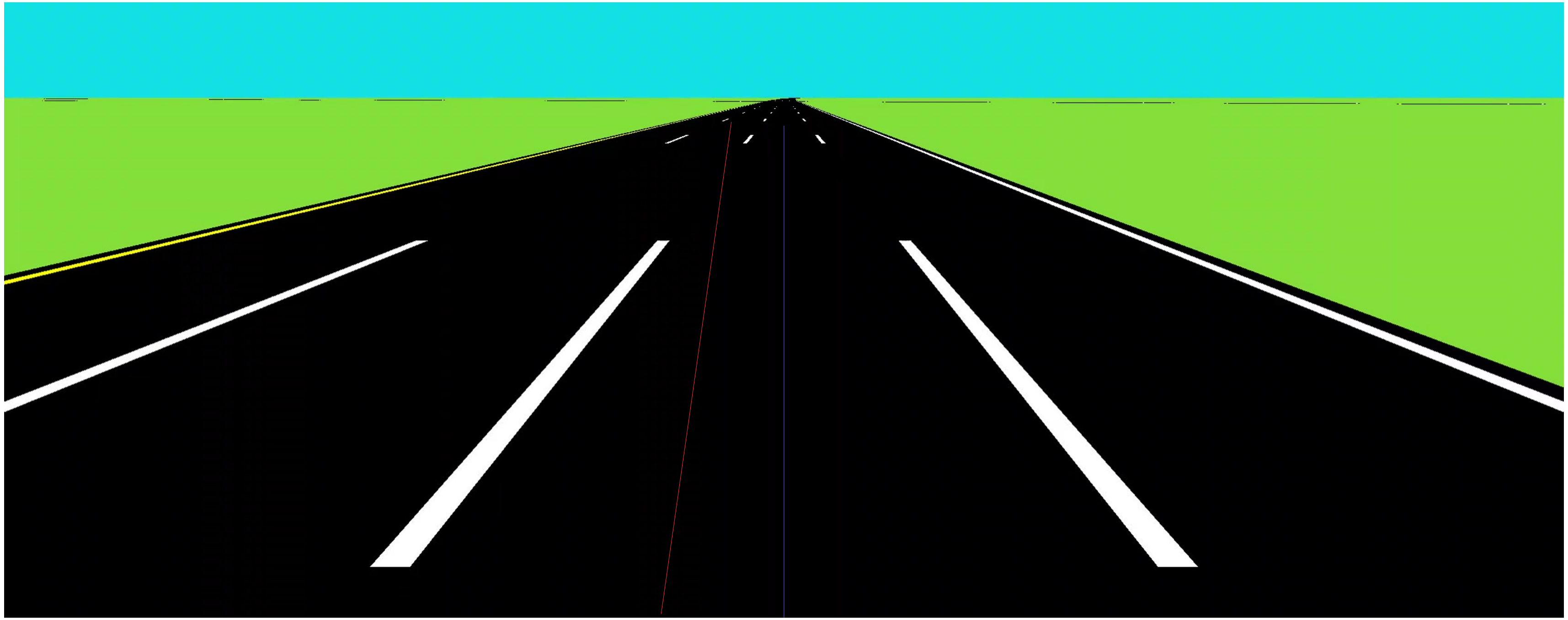


PREDICTION ERROR



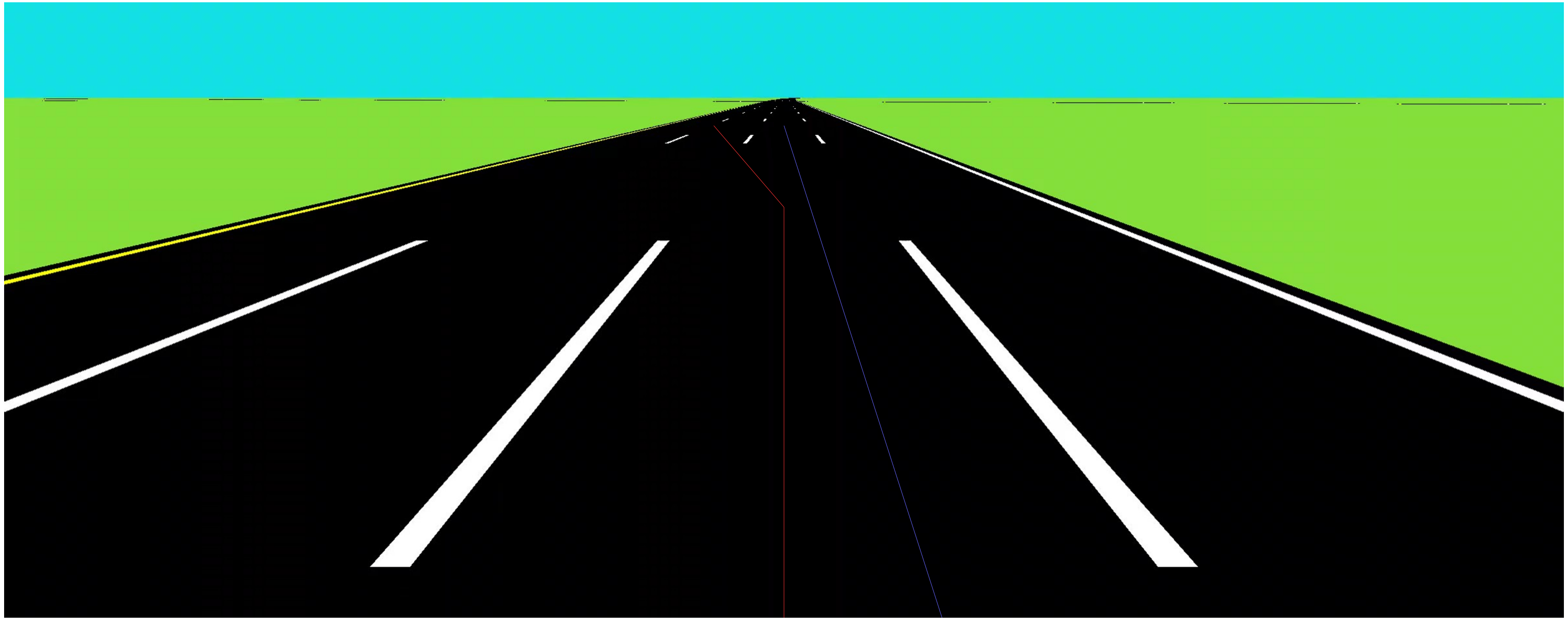
Simple and easy to understand (measure the distance between two lines)

PREDICTION ERROR DRAWBACKS



The blue line is obviously better than the red one

PREDICTION ERROR DRAWBACKS

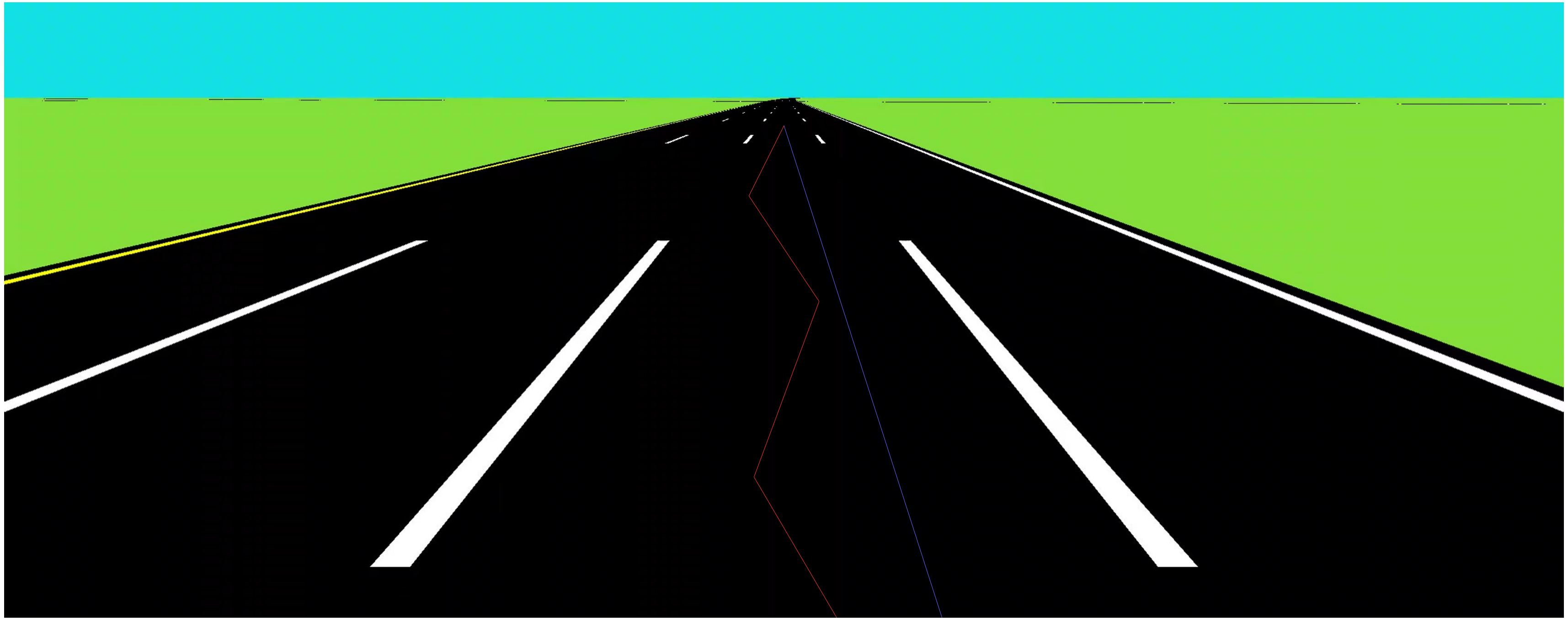


How about now?

The blue line is offset so the vehicle will be off-center.

The red line is closer to the center, but then leaves the lane.

PREDICTION ERROR DRAWBACKS



How about now?

The blue line is offset so the vehicle will be off-center.

The red line is closer to the center, but would be unpleasant.

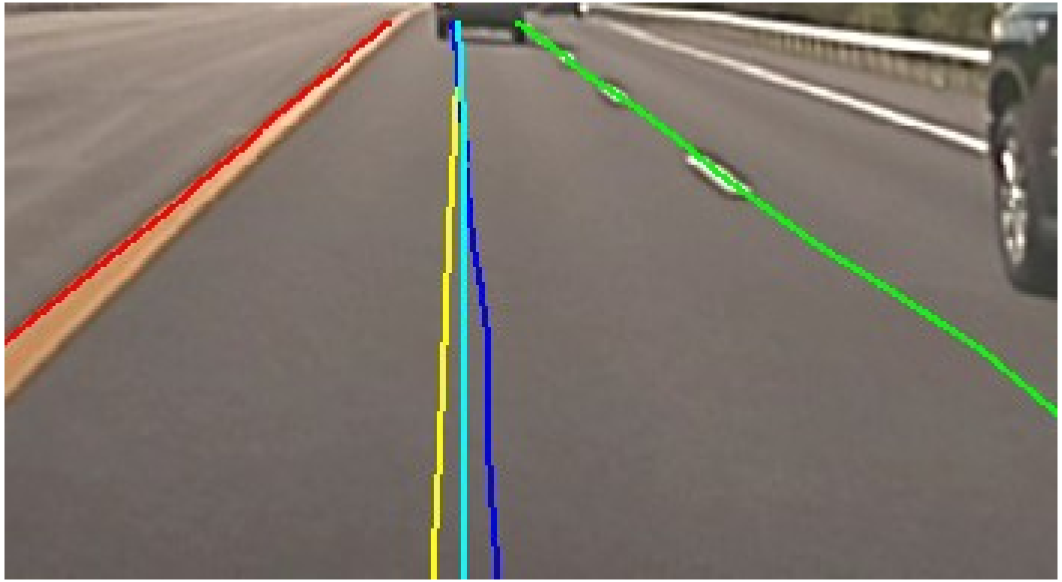
PREDICTION ERROR DRAWBACKS

Not a robust statistics

Looking at prediction error is very easy

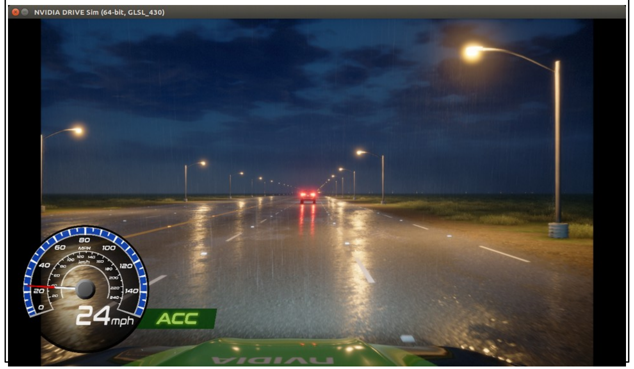
Not very robust

A prediction that is slightly off-center may be preferable to one that fails suddenly or is incredible uncomfortable



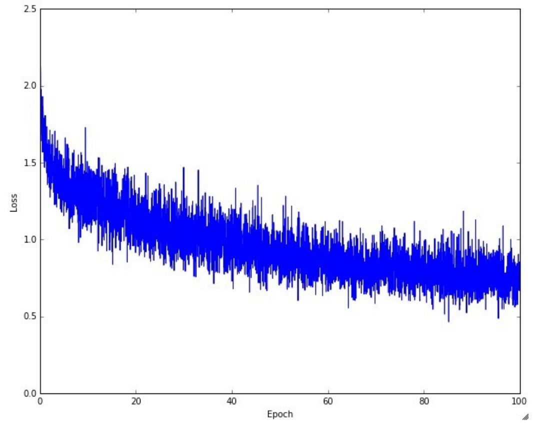
SIMULATION

- Addresses issues with real-world testing
- Test set can be created at will
- Difficult to correctly model all behaviors and distributions



PREDICTION ERROR

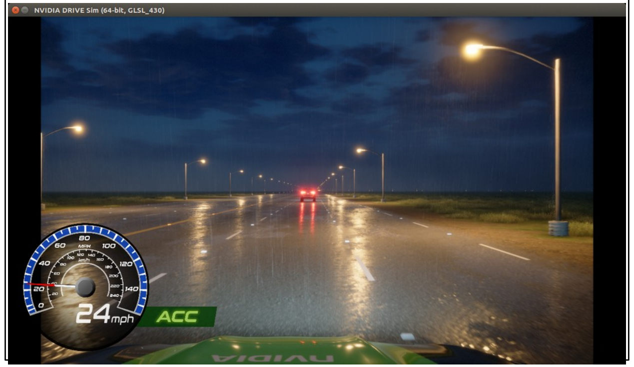
- Uses real data!
- Simple! (Mean Squared Error, etc)
- Good result doesn't mean good driving!



SIMULATION

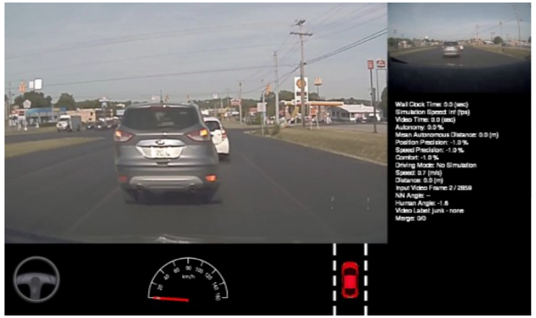
- Addresses issues with real-world testing
- Test set can be created at will

- Difficult to correctly model all behaviors and distributions



AUGMENTED RE-SIMULATION

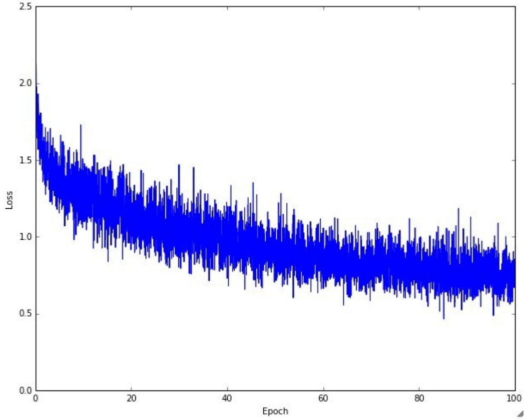
- Simulates using real data!
- Somewhat simple



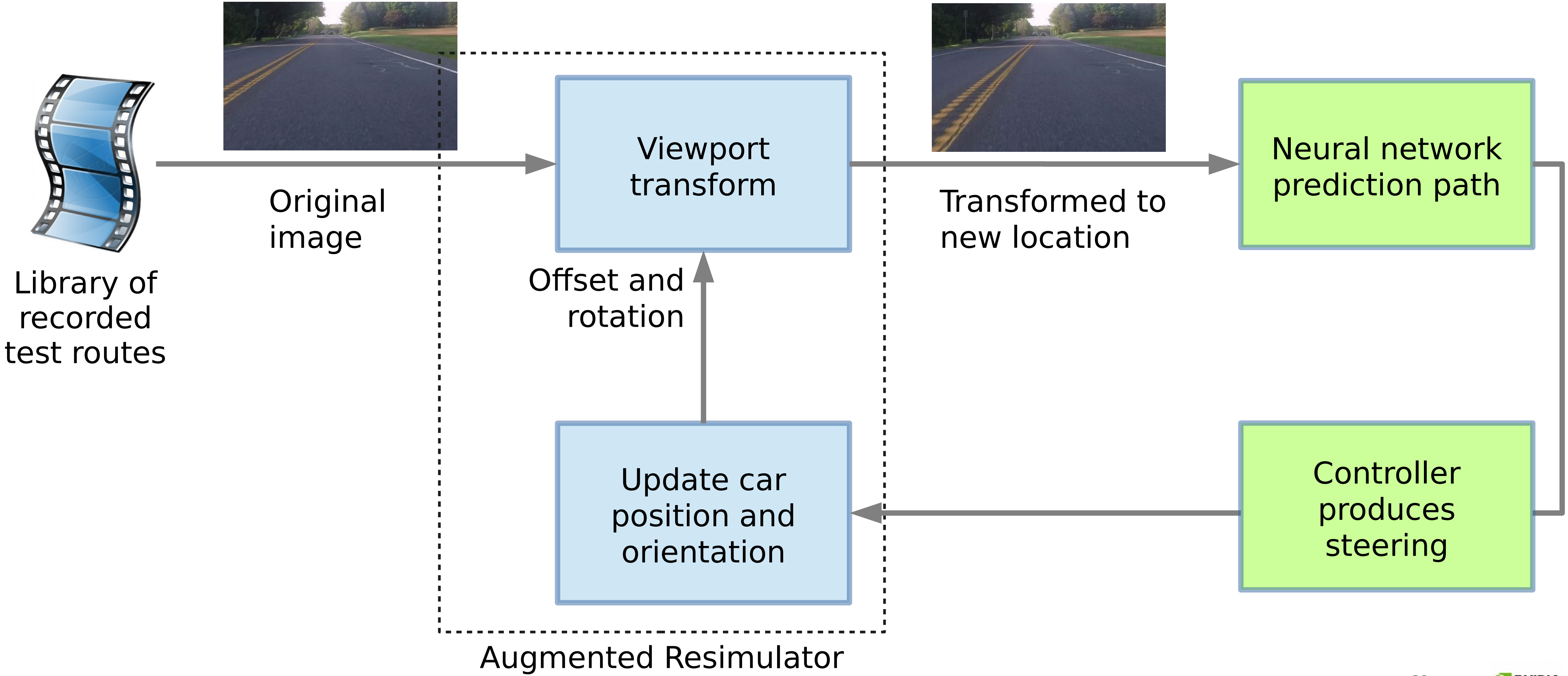
PREDICTION ERROR

- Uses real data!
- Simple! (Mean Squared Error, etc)

- Good result doesn’t mean good driving!



AUGMENTED RESIMULATION



AUGMENTED RESIM EXAMPLE

NVIDIA

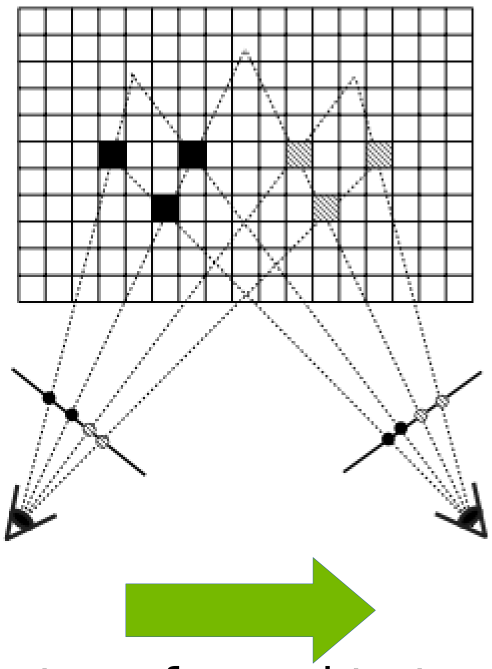
AUGMENTED RESIM ADVANTAGES

Safe, repeatable, and objective

Measure multiple statistics objectively:

MAD: Mean Autonomous Distance, or how far we can drive without failing
Comfort: How smooth is the ride?
Precision: Do we drive in the center of the road?

The three metrics are not necessarily correlated!



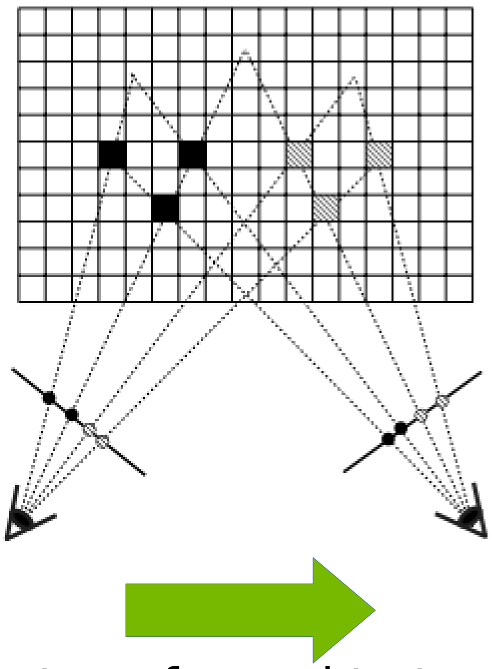
Source transformed to target

AUGMENTED RESIM DRAWBACKS

Not a perfect recreation

Image transformations introduce artifacts not seen in the real world

Data must be collected.

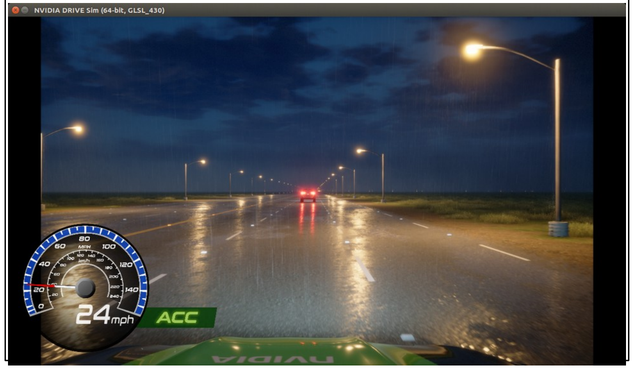


Source transformed to target

SIMULATION

- Addresses issues with real-world testing
- Test set can be created at will

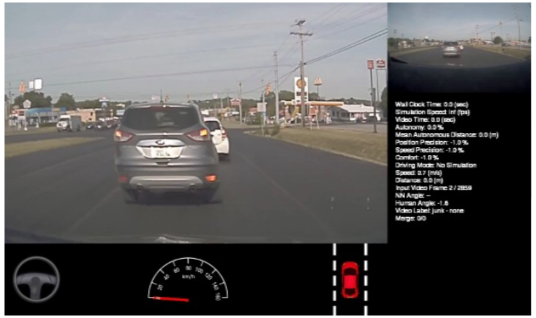
• Difficult to correctly model all behaviors and distributions



AUGMENTED RE-SIMULATION

- Simulates using real data!
- Somewhat simple

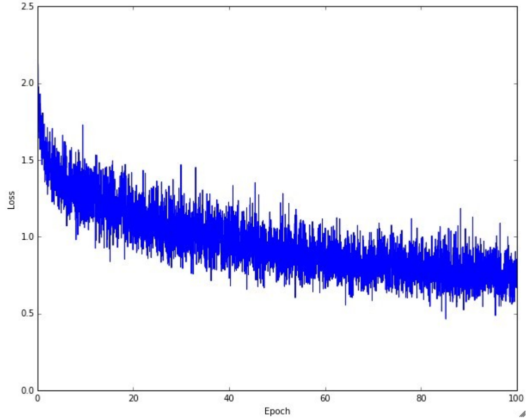
- Test set must be collected
- Artifacts



PREDICTION ERROR

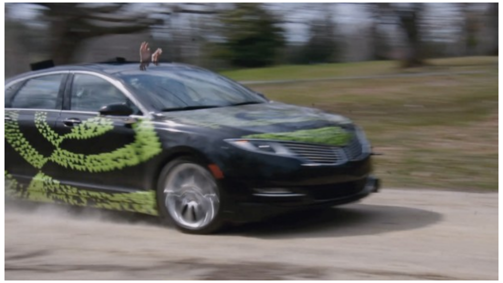
- Uses real data!
- Simple! (Mean Squared Error, etc)

- Good result doesn't mean good driving!



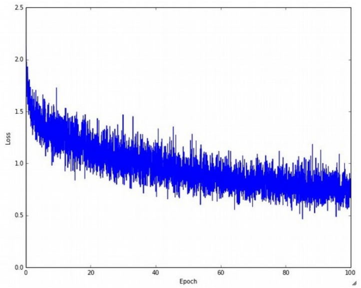
A COMBINATION OF TESTS IS BEST

REAL WORLD TESTS

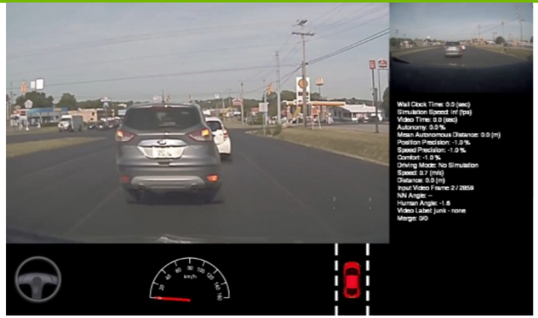


SIMULATION

PREDICTION ERROR



RE-SIMULATION



LESSONS LEARNED

TEST EARLY, TEST OFTEN

Lessons learned

Augmented resim and simulated data allow us to test early and often

It is important to catch a weakness in the current approach early for two reasons:

1. It may take a long time to address
2. It may require new kinds of sensor data

Frequent testing also gives a historical perspective about your rate of progress

REAL-WORLD TESTING IS AMBIGUOUS

We get into a lot of arguments

Real-world testing is biased by what is close to you. Someone in another location may have completely different results

People do not agree on how good or bad something feels or how two systems compare

It is very time-consuming to drive around searching for a failure

REPEATABILITY IS KEY

Stop arguing, start fixing

It is too hard to debug something if you can’t repeat it

This also allows you to develop metrics that capture the error

TAKEAWAYS

Applicable anywhere

Learning directly from human actions make labeling inexpensive

This allows us to **scale** as we collected more data

Since the labels are different than from a traditional approach we can combine them to increase **robustness**

Testing and evaluation should be done in multiple ways and as often as possible

Getting as close as possible to the real-world while still having repeatability is vital

NVIDIA®