

CS231A

Computer Vision: From 3D Reconstruction to Recognition



Optimal Estimation Cont'

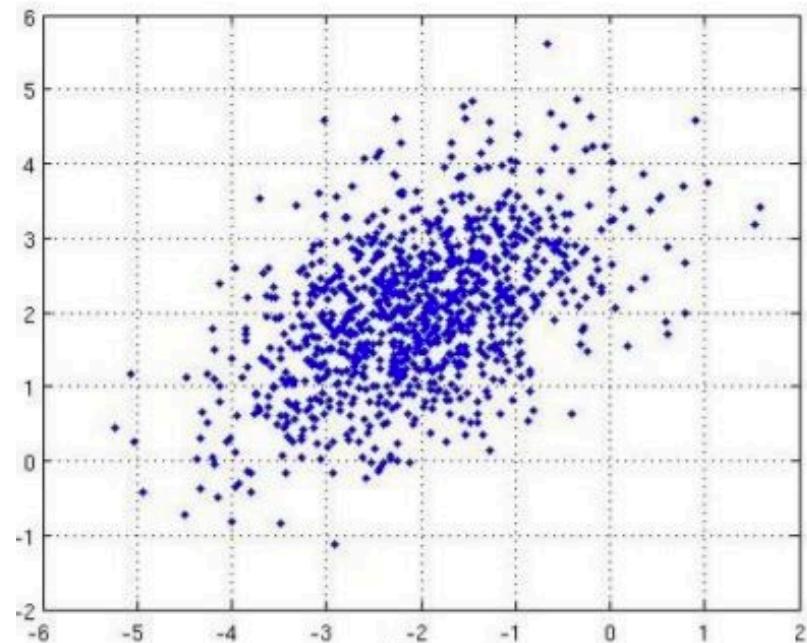
Recap

- Recursive Filter
- Kalman Filter
- Extended Kalman Filter

Nonparametric filters

- No fixed functional form of the posterior – can capture multimodality
- Instead: finite numbers of values
- Histogram filter: State = finitely many regions
- Particle filter: Distribution represented by samples

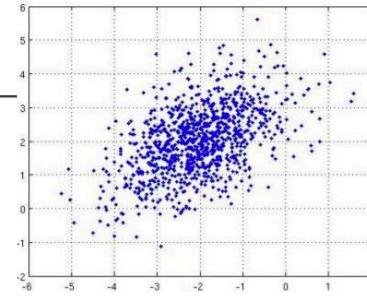
Particle Filter



$$p(x_t | z_{t:1}, u_{t:1}, x_0) \rightarrow X_t = \{x_t^0, \dots, x_t^N\}$$

The Particle filter algorithm

```
1: Algorithm Particle filter( $\mathcal{X}_{t-1}, u_t, z_t$ ):
2:    $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$ 
3:   for  $m = 1$  to  $M$  do
4:     sample  $x_t^{[m]} \sim p(x_t | u_t, x_{t-1}^{[m]})$ 
5:      $w_t^{[m]} = p(z_t | x_t^{[m]})$ 
6:      $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$ 
7:   endfor
8:   for  $m = 1$  to  $M$  do
9:     draw  $i$  with probability  $\propto w_t^{[i]}$ 
10:    add  $x_t^{[i]}$  to  $\mathcal{X}_t$ 
11:   endfor
12:   return  $\mathcal{X}_t$ 
```



Process Model

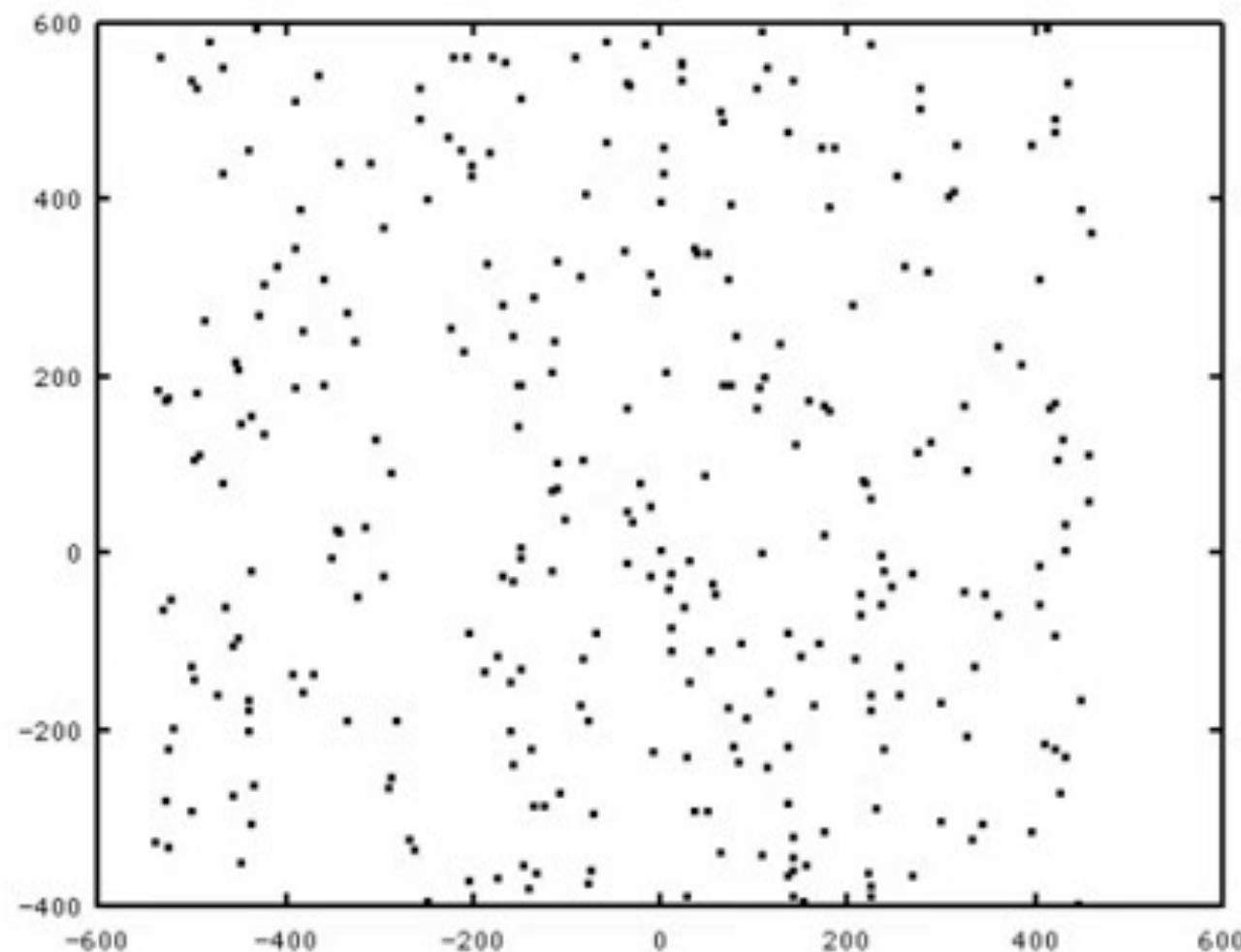
Measurement Model

Importance Sampling

$$\begin{aligned} \text{Before resampling} &\longrightarrow \bar{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx \\ \text{After resampling} &\longrightarrow bel(x_t) = \eta p(z_t | x_t) \bar{bel}(x_t) \end{aligned}$$



Figure 1



-360.227, -530.956

When to Use Each?

Bayes Filter

General Framework
No implementation!

Kalman Filter

Linear Models
Gaussian Distributions

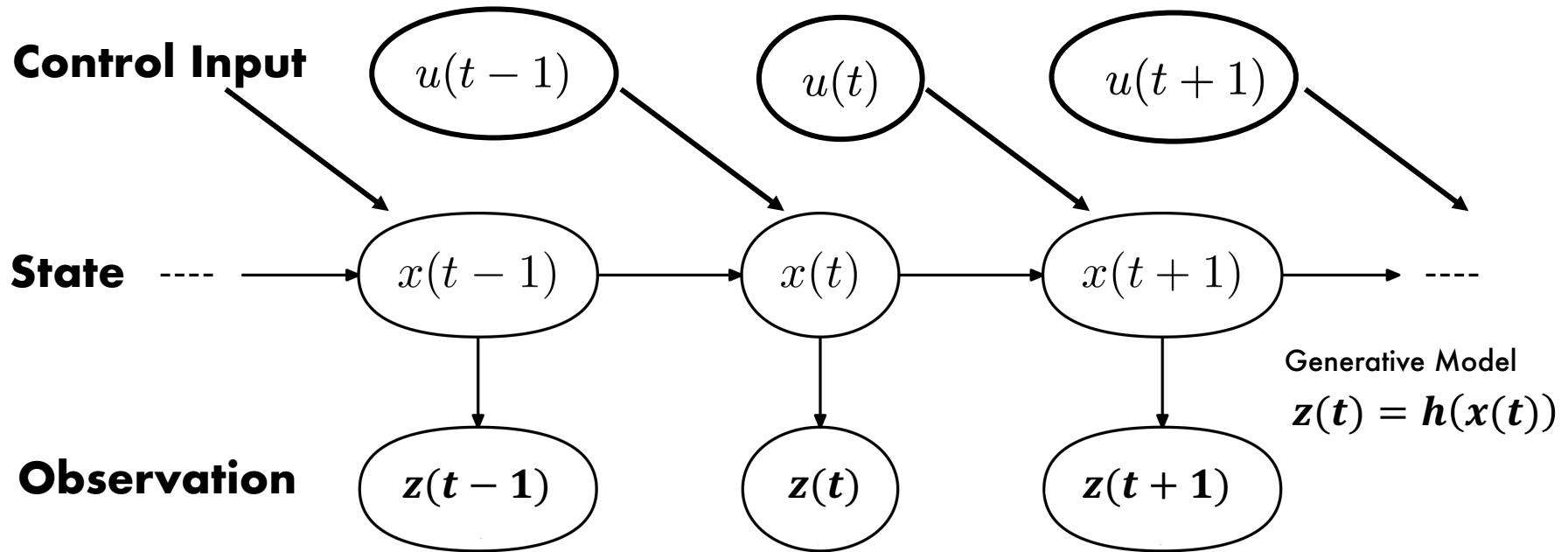
Extended Kalman Filter

Non-Linear Models (linearizable)
Gaussian Distributions

Particle Filter

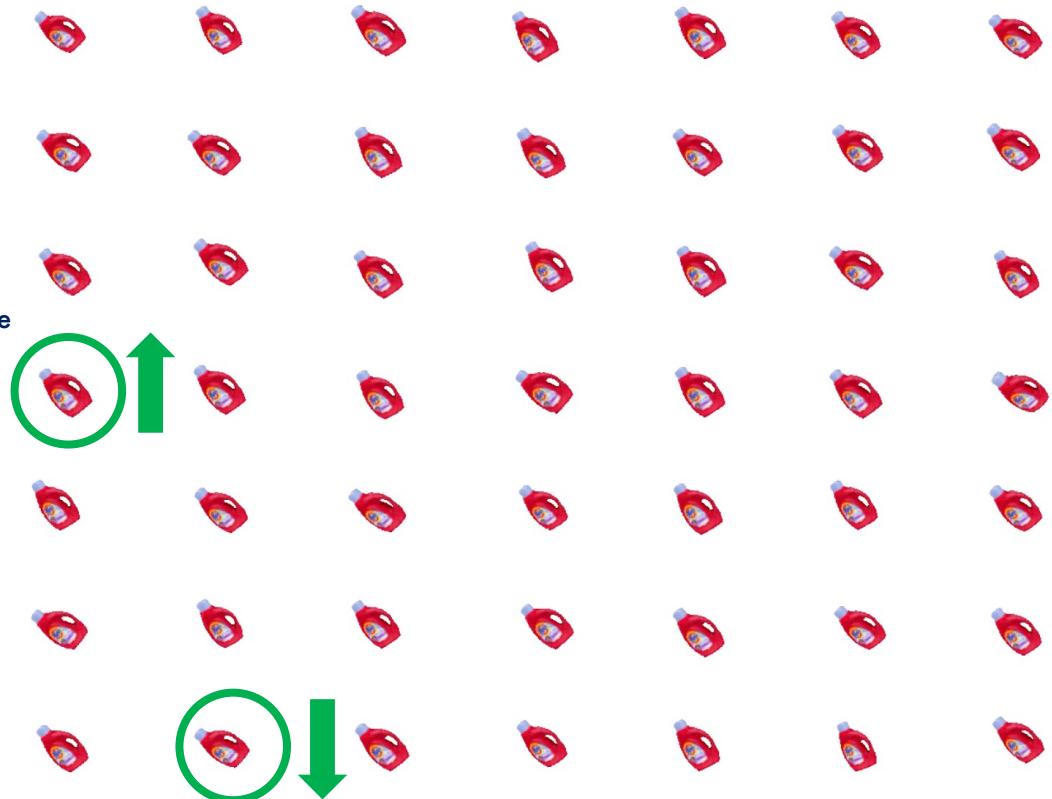
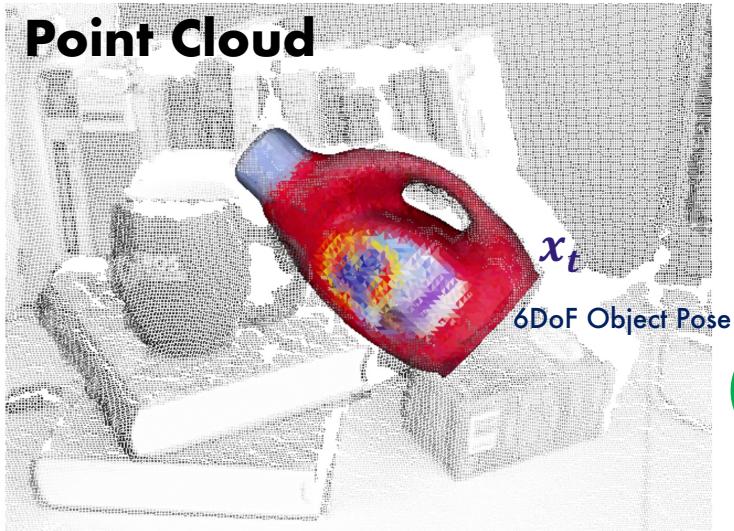
Any Model
Any Distribution
Low Dimensional State Space

Graphical Model of System to Estimate



```
1: Algorithm Bayes_filter( $bel(x_{t-1})$ ,  $u_t$ ,  $z_t$ ):  
2:   for all  $x_t$  do  
3:      $\bar{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx$   
4:      $bel(x_t) = \eta p(z_t | x_t) \bar{bel}(x_t)$   
5:   endfor  
6:   return  $bel(x_t)$ 
```

Example Observation model for 3D object



Algorithm Particle filter($\mathcal{X}_{t-1}, u_t, z_t$):

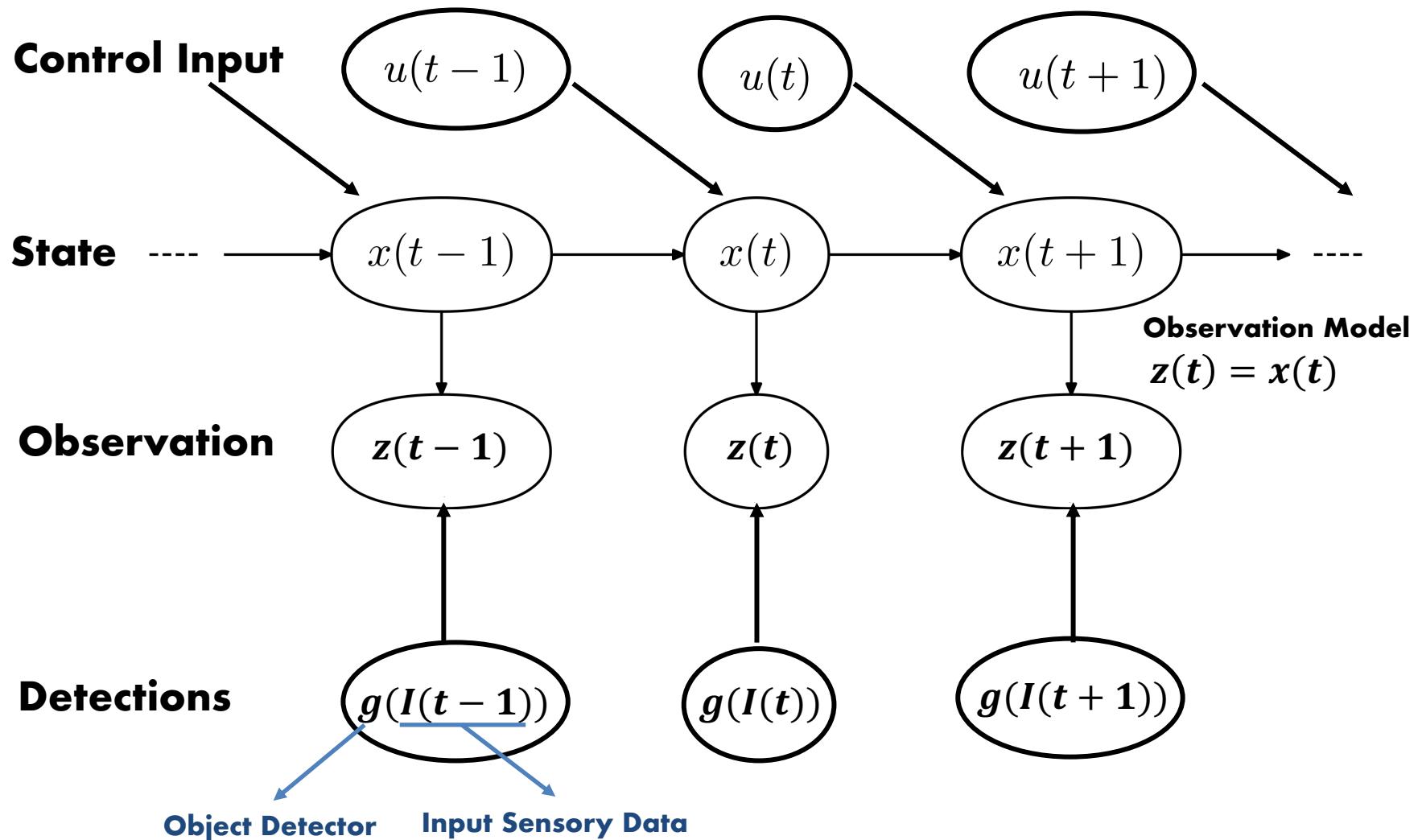
```
 $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$ 
for  $m = 1$  to  $M$  do
    sample  $x_t^{[m]} \sim p(x_t | u_t, x_{t-1}^{[m]})$ 
     $w_t^{[m]} = p(z_t | x_t^{[m]})$ 
     $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$ 
endfor
for  $m = 1$  to  $M$  do
    draw  $i$  with probability  $\propto w_t^{[i]}$ 
    add  $x_t^{[i]}$  to  $\mathcal{X}_t$ 
endfor
return  $\mathcal{X}_t$ 
```

Importance Sampling

Rendered Particles

Changhyun Choi and Henrik I. Christensen. Rgb-d object tracking: A particle filter approach on gpu. In IROS, pages 1084–1091, 2013

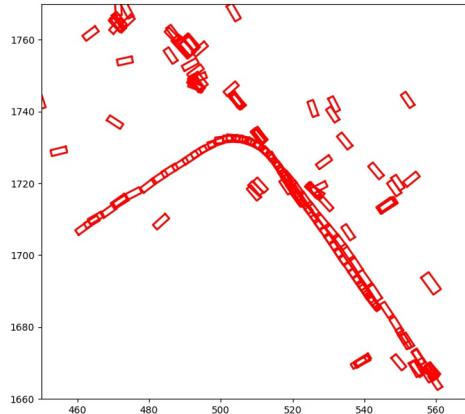
Tracking by Detection



Problem Statement: Input

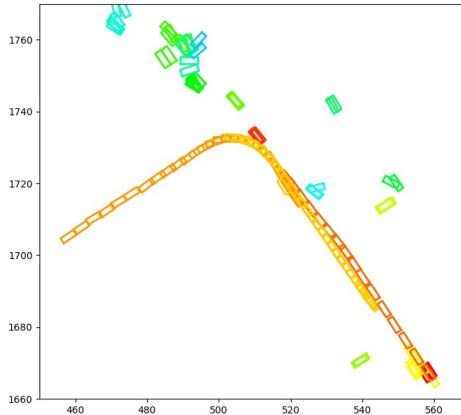
Probabilistic 3d multi-object tracking for autonomous driving. H Chiu, A Prioletti, J Li, J Bohg
arXiv preprint arXiv:2001.05673

- Object detections at each frame in a sequence
- Each detection bounding box is represented by:
 - center position (x, y, z), rotation angle along the z -axis (a), and the scale (l, w, h)
 - category label (car, pedestrian, ...), confidence score (c)



Problem Statement: Output

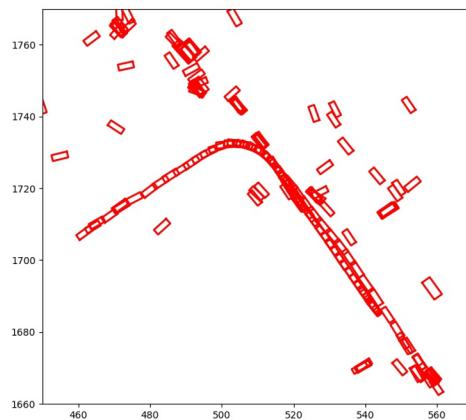
- Tracking object bounding boxes at each frame in a sequence
- Each tracking bounding box is represented by:
 - center position (x, y, z), rotation angle along the z -axis (a), and the scale (l, w, h)
 - category label (car, pedestrian, ...), confidence score (c)
 - tracking id: one unique tracking id for each object instance across frames



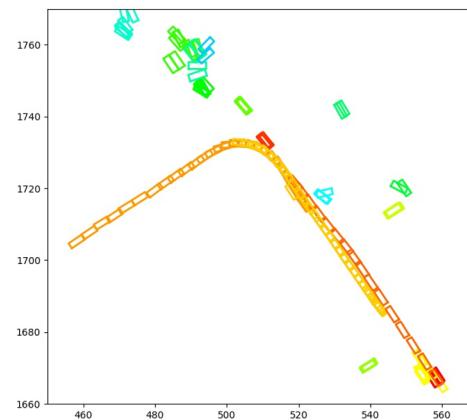
Why Tracking?

- Filter out the out-liners from the detection results
- Continue estimating object states even if occluded
- Forecast the future based on past trajectories and motion patterns
- Make autonomous driving decisions

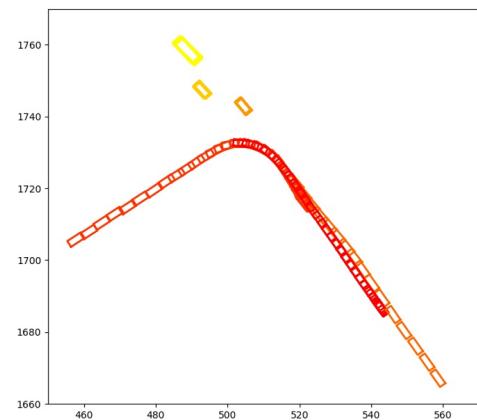
Detection

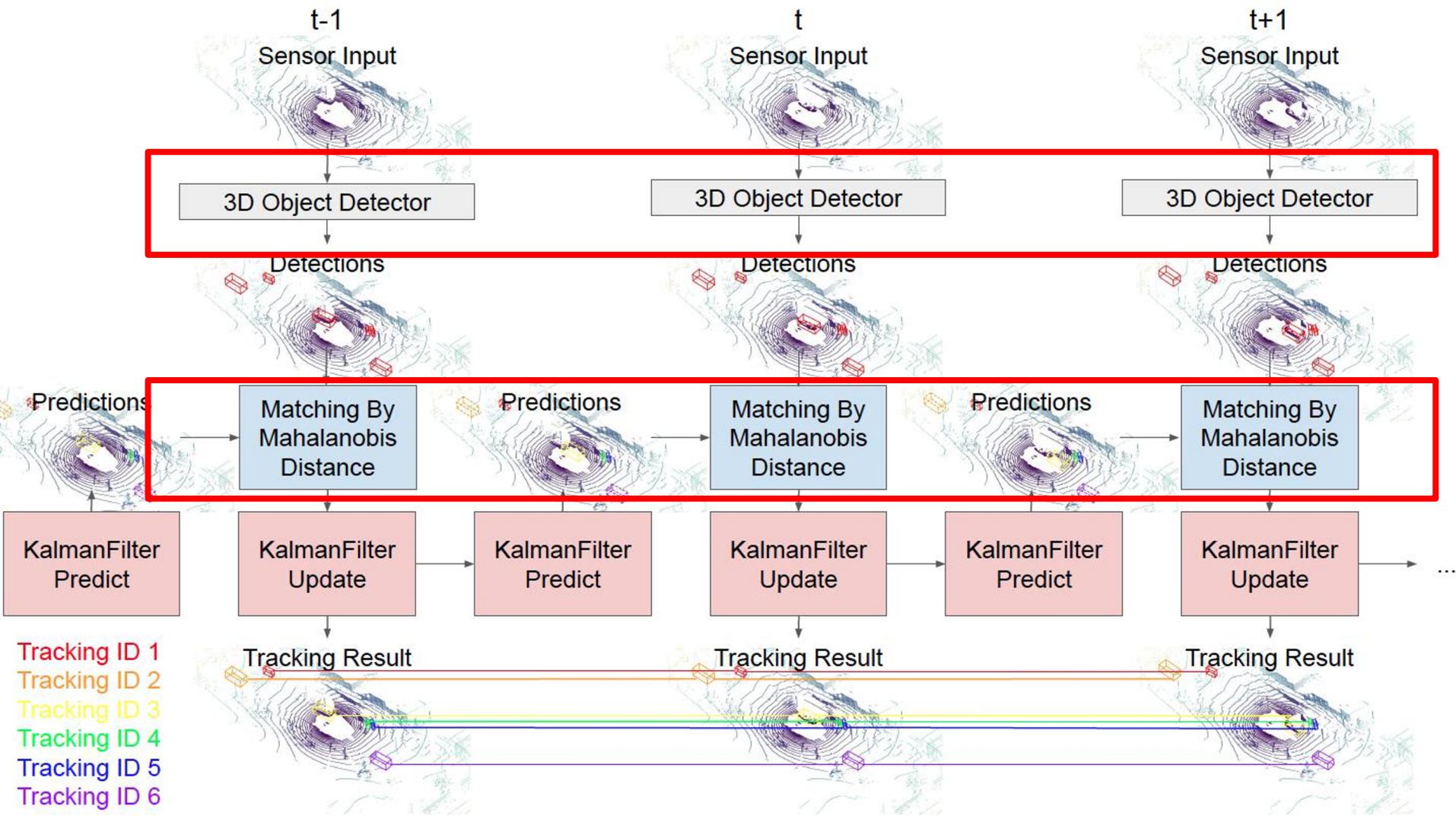


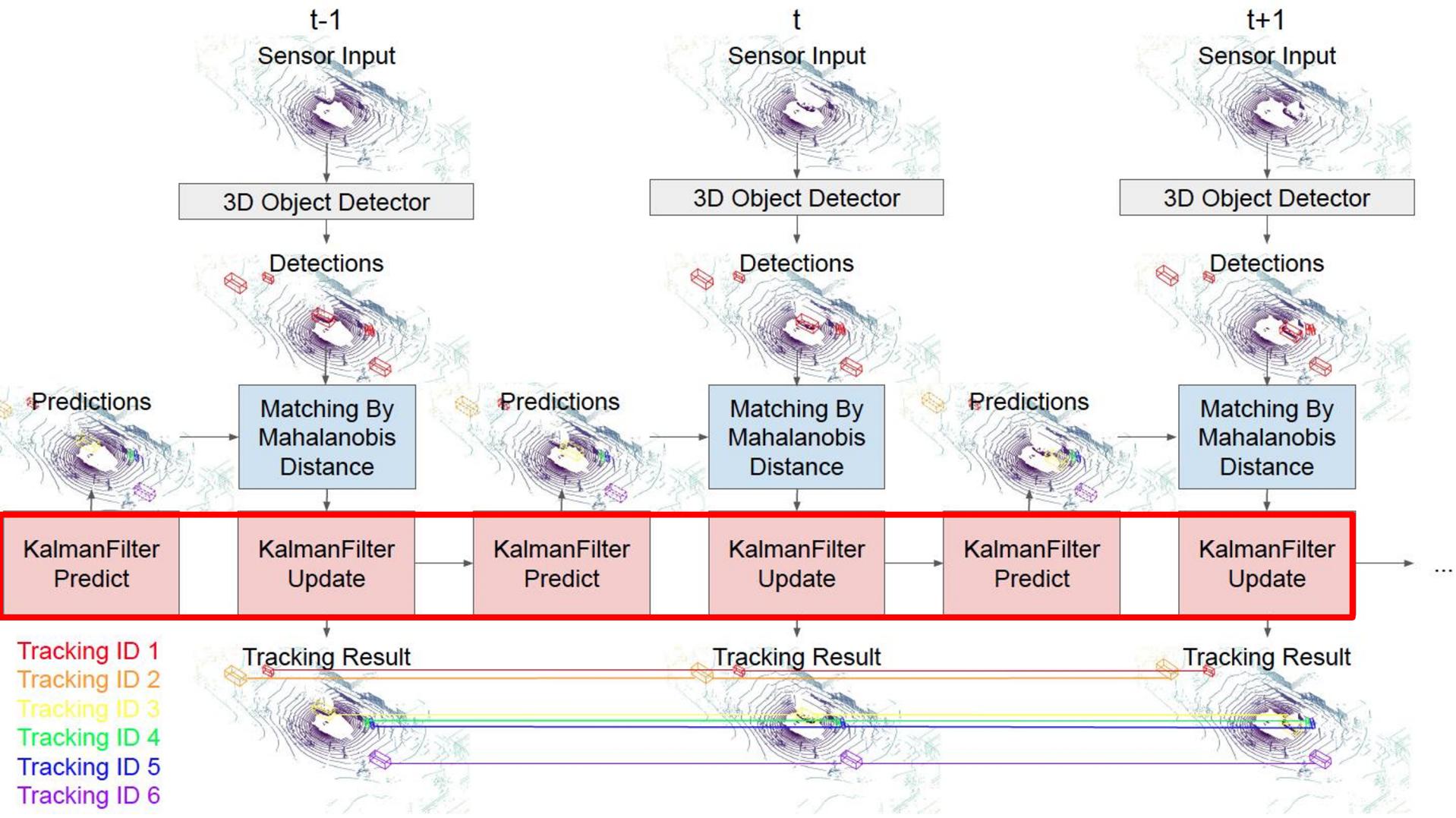
Tracking



Ground-truth







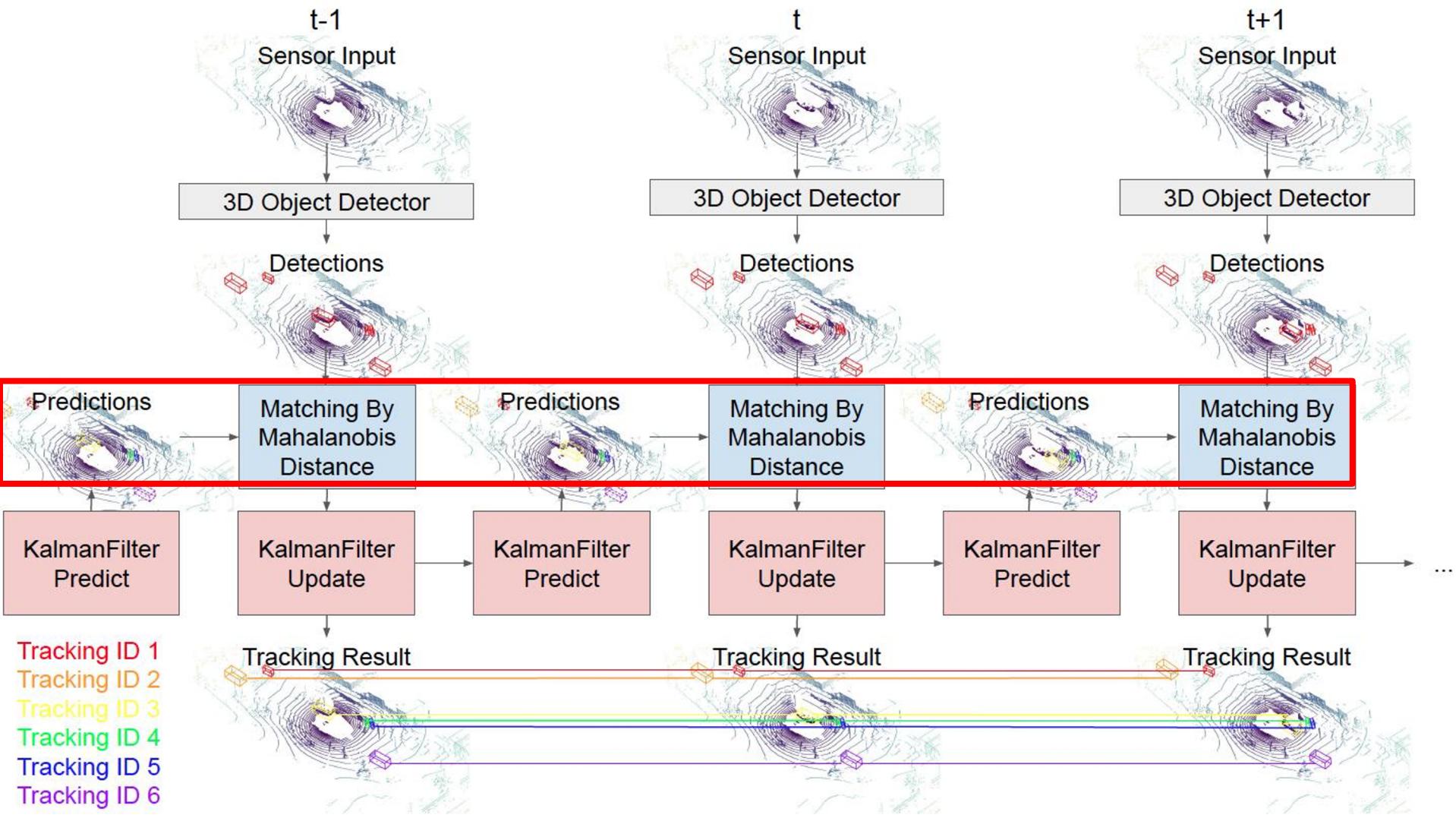
Kalman Filter for Tracking

Define the object **state** using a vector of random variables including the position, the rotation, the scale, linear velocity, and the angular velocity.

$$\mathbf{s}_t = (x, y, z, a, l, w, h, d_x, d_y, d_z, d_a)^T$$

Define the **Process Model** for prediction based on the constant velocity motion:

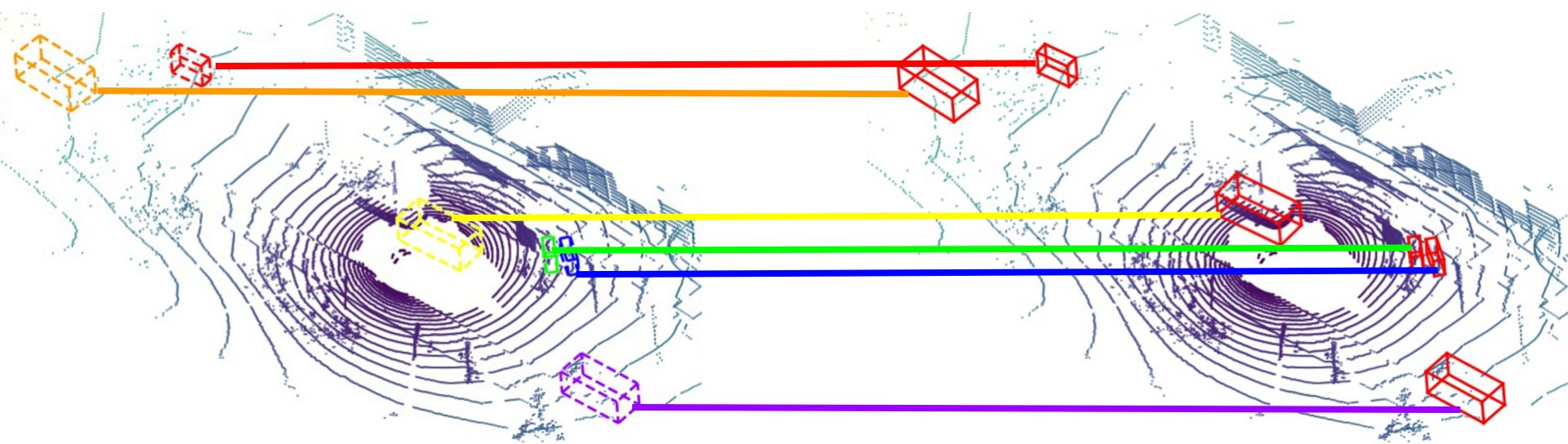
$$\begin{aligned}\hat{x}_{t+1} &= x_t + d_{x_t} + q_{x_t}, & \hat{d}_{x_{t+1}} &= d_{x_t} + q_{d_{x_t}} & \hat{l}_{t+1} &= l_t \\ \hat{y}_{t+1} &= y_t + d_{y_t} + q_{y_t}, & \hat{d}_{y_{t+1}} &= d_{y_t} + q_{d_{y_t}} & \hat{w}_{t+1} &= w_t \\ \hat{z}_{t+1} &= z_t + d_{z_t} + q_{z_t}, & \hat{d}_{z_{t+1}} &= d_{z_t} + q_{d_{z_t}} & \hat{h}_{t+1} &= h_t \\ \hat{a}_{t+1} &= a_t + d_{a_t} + q_{a_t}, & \hat{d}_{a_{t+1}} &= d_{a_t} + q_{d_{a_t}}\end{aligned}$$



Data Association

$$\text{Mahalanobis Distance } m = \sqrt{(z_t - C\bar{\mu}_t)^T S_t^{-1} (z_t - C\bar{\mu}_t)}$$

S = Innovation Covariance
 $z_t - C\bar{\mu}_t$ = innovation



**Kalman Filter
Predictions**

Object Detections

Kalman Filter

```
1: Algorithm Kalman filter( $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$ ):  
2:    $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$   
3:    $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$   
4:    $K_t = \bar{\Sigma}_t C_t^T [C_t \bar{\Sigma}_t C_t^T + Q_t]^{-1} = S_t^{-1}$   
5:    $\mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t)$   
6:    $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$   
7:   return  $\mu_t, \Sigma_t$ 
```

Data Association

$$\text{Mahalanobis Distance } m = \sqrt{(z_t - C\bar{\mu}_t)^T S_t^{-1} (z_t - C\bar{\mu}_t)}$$

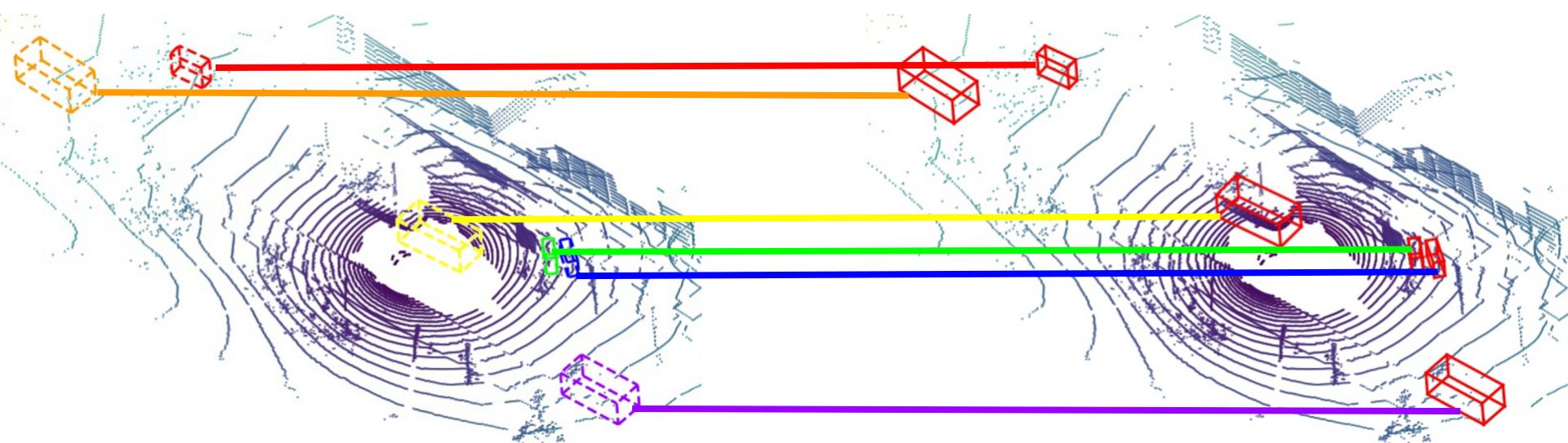
If $m > 3 * \sigma$ then reject as outlier. 99.7% of values lie within 3*standard deviation.

Measuring the distance between the observation and the distribution of the predicted state.

Providing distance measurement **when there is no overlap** between the prediction and detection.

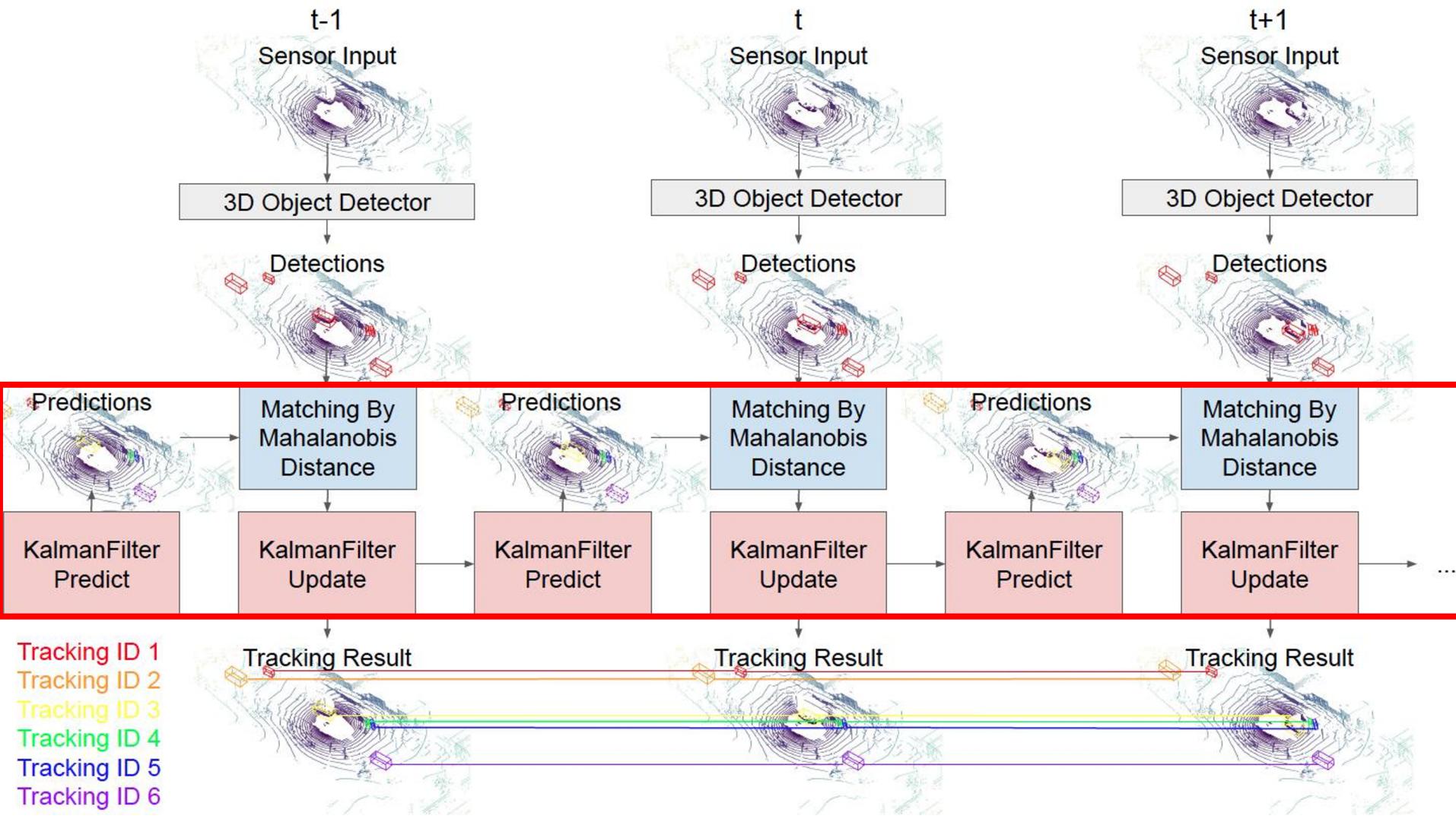
Taking the **uncertainty** information from the prediction into account.

Data Association - Greedy

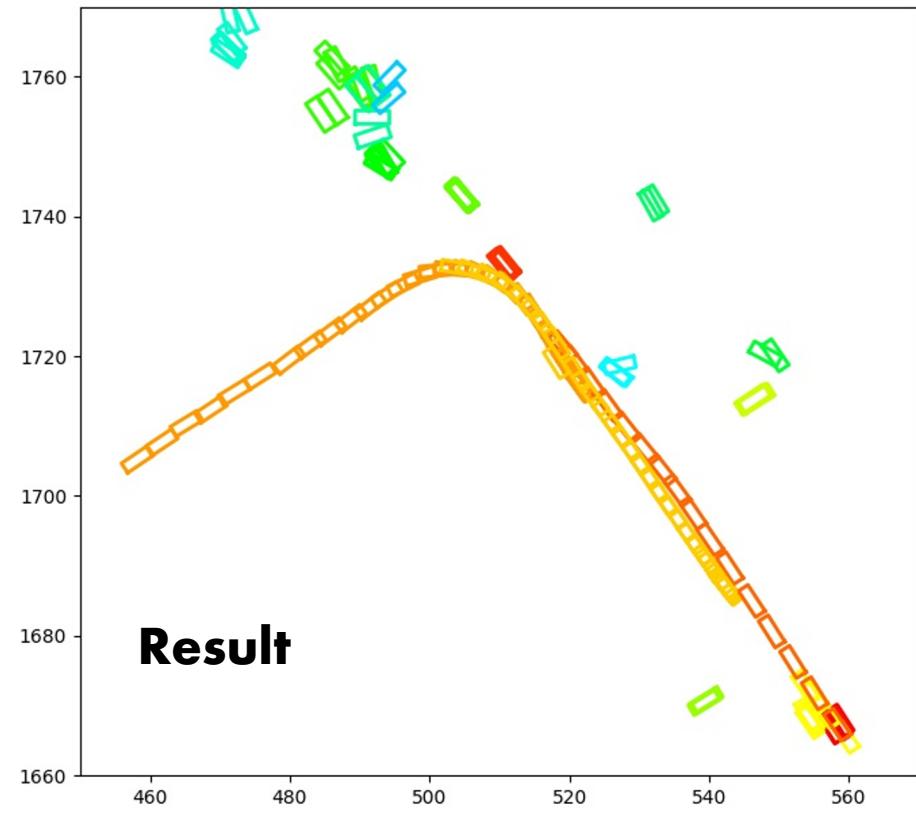
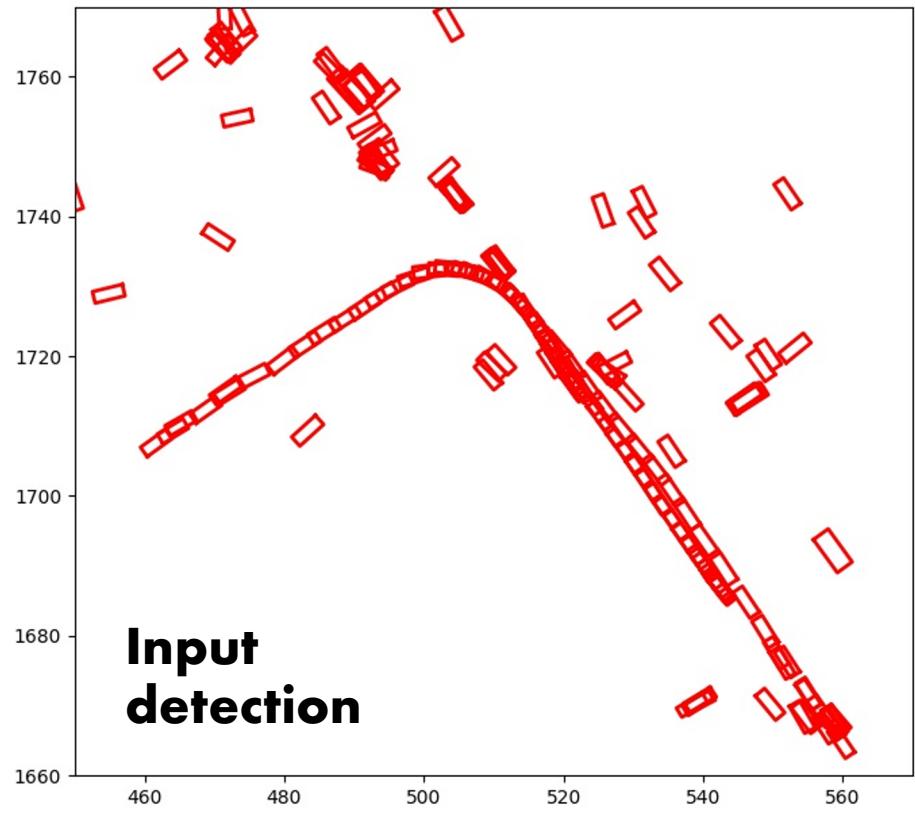


**Kalman Filter
Predictions**

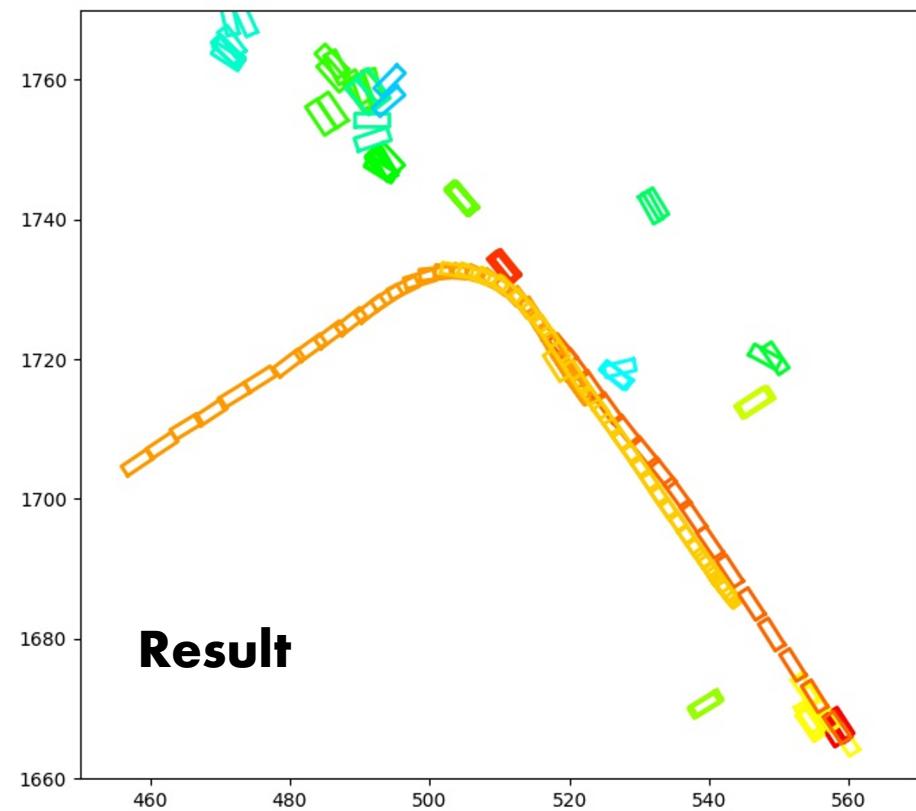
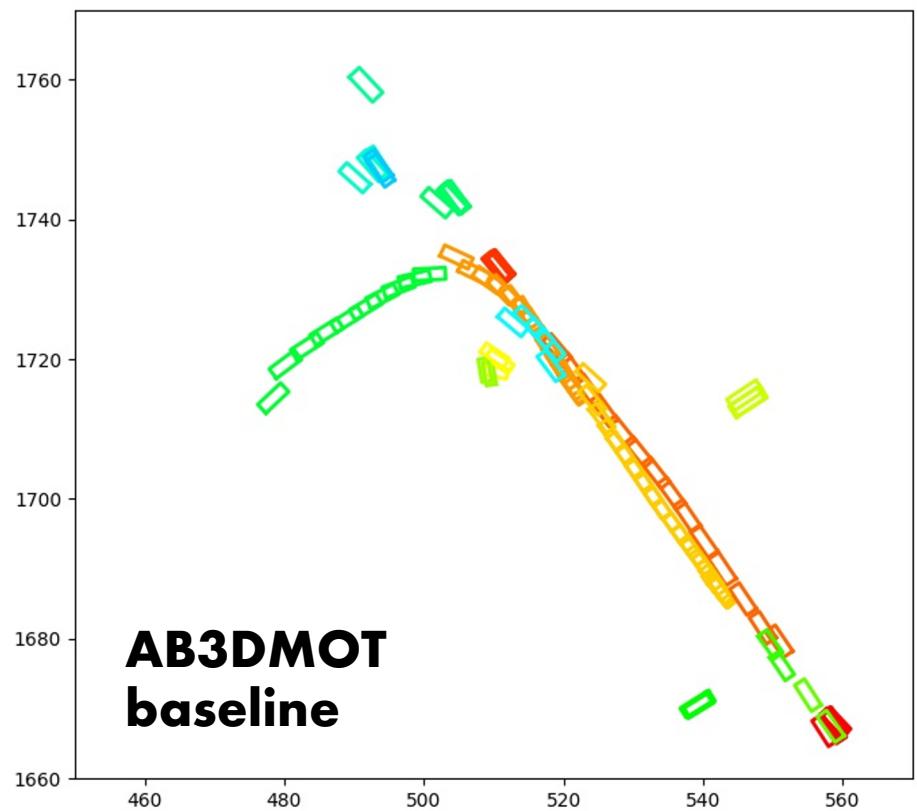
Detections



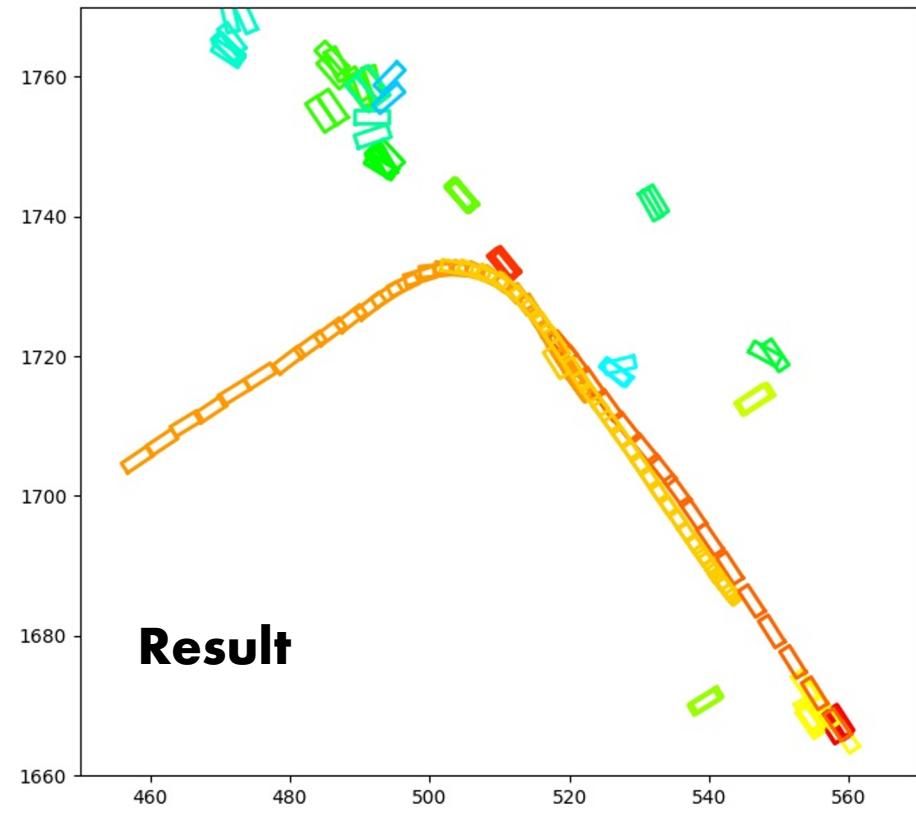
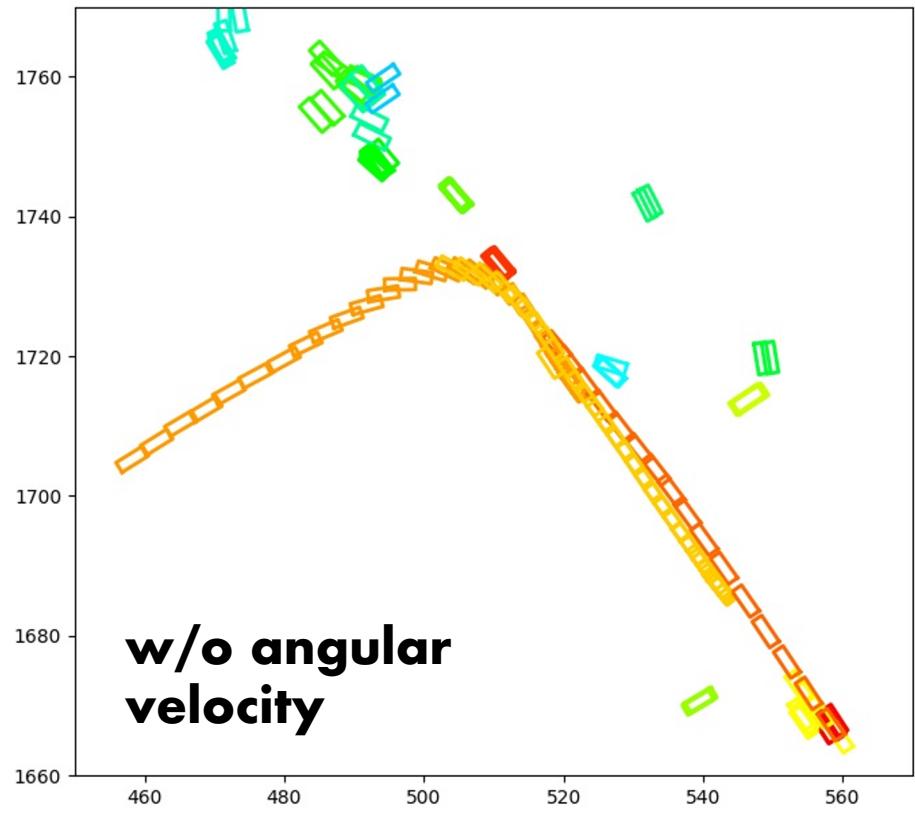
Qualitative Results



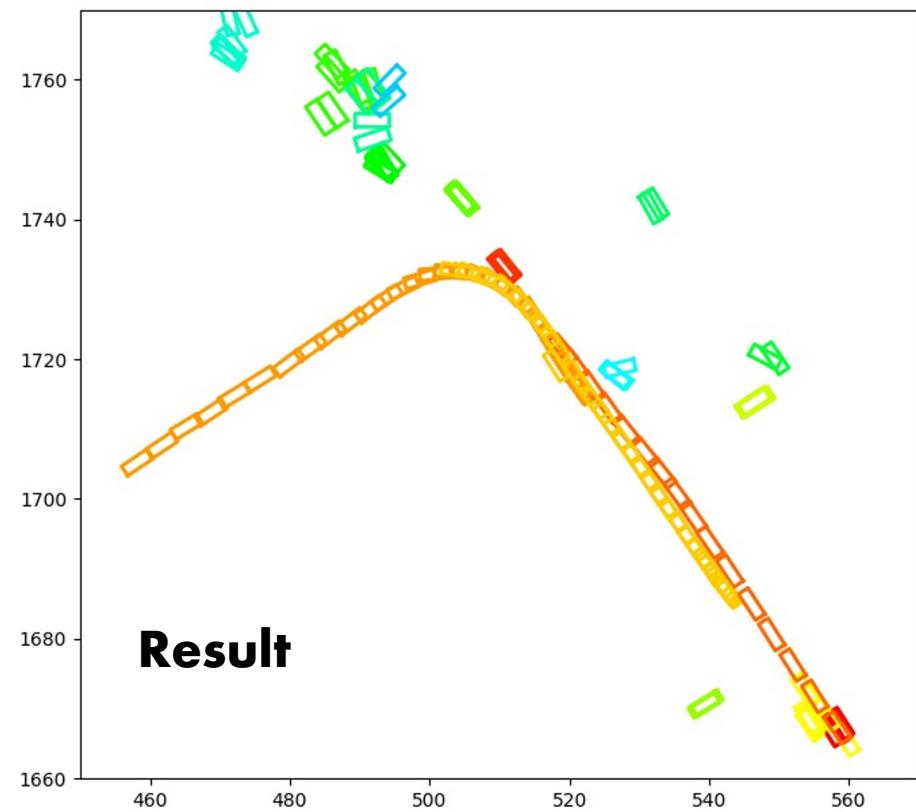
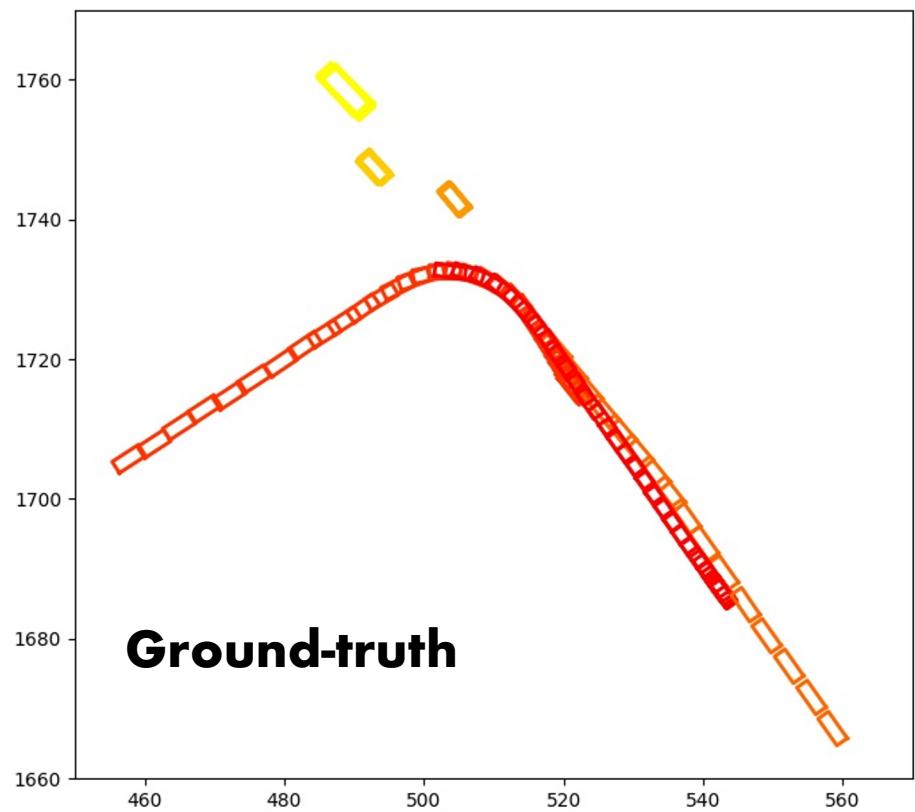
Qualitative Results



Qualitative Results



Qualitative Results



Priors and Hyperparameters

A lot of hardcoded knowledge!

- State Representation
- Models
 - Forward Model
 - State to next state
 - Action to next state
 - Measurement Model
- Probabilistic Properties
 - Process Noise
 - Measurement Noise



Differentiable filters

Can we learn models and hyperparameters from data?

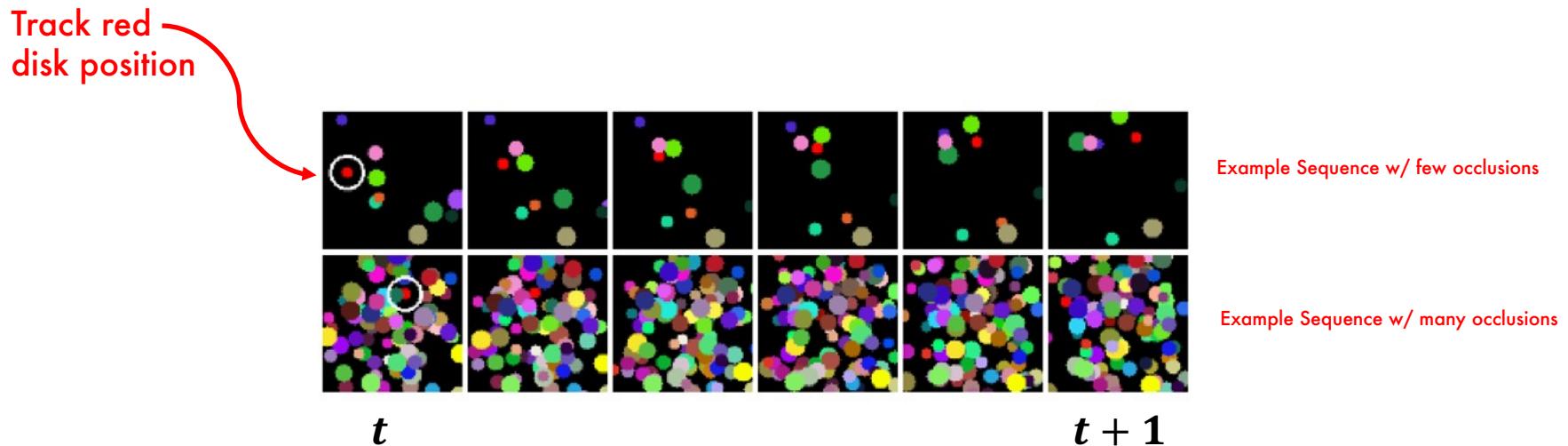
Approach: Embed algorithmic structure of Bayesian Filtering into a recurrent neural network.

- prevents overfitting through regularization
- Avoids manual tuning and modeling

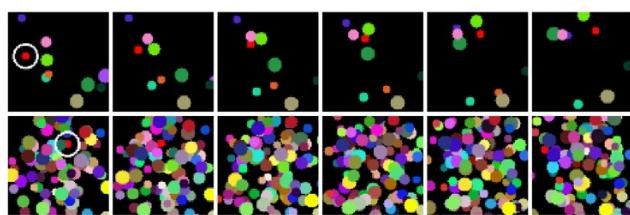
Estimators. Haarnoja et al. NeurIPS 2016

- Differentiable version of the Kalman Filter
- Uses Images as observations; learns a sensors that outputs state directly

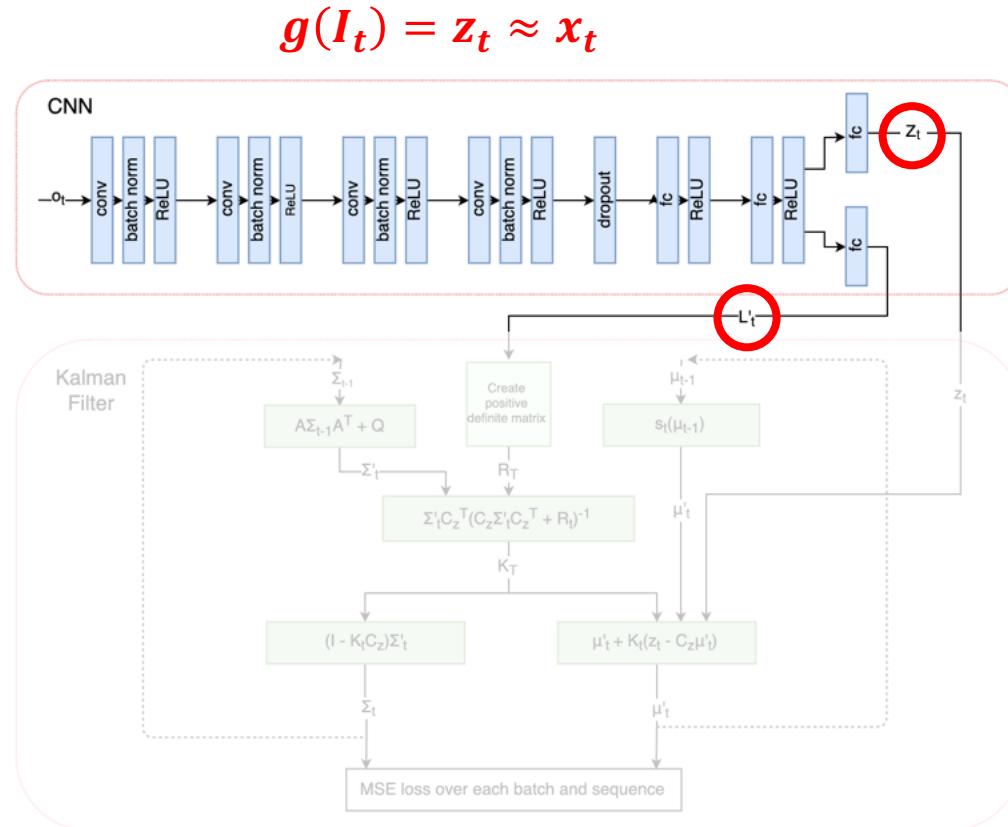
$$g(I_t) = z_t \approx x_t$$



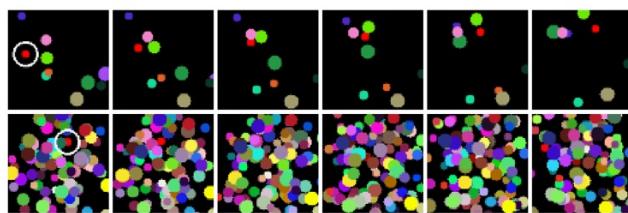
Differentiable Kalman Filter - Structure



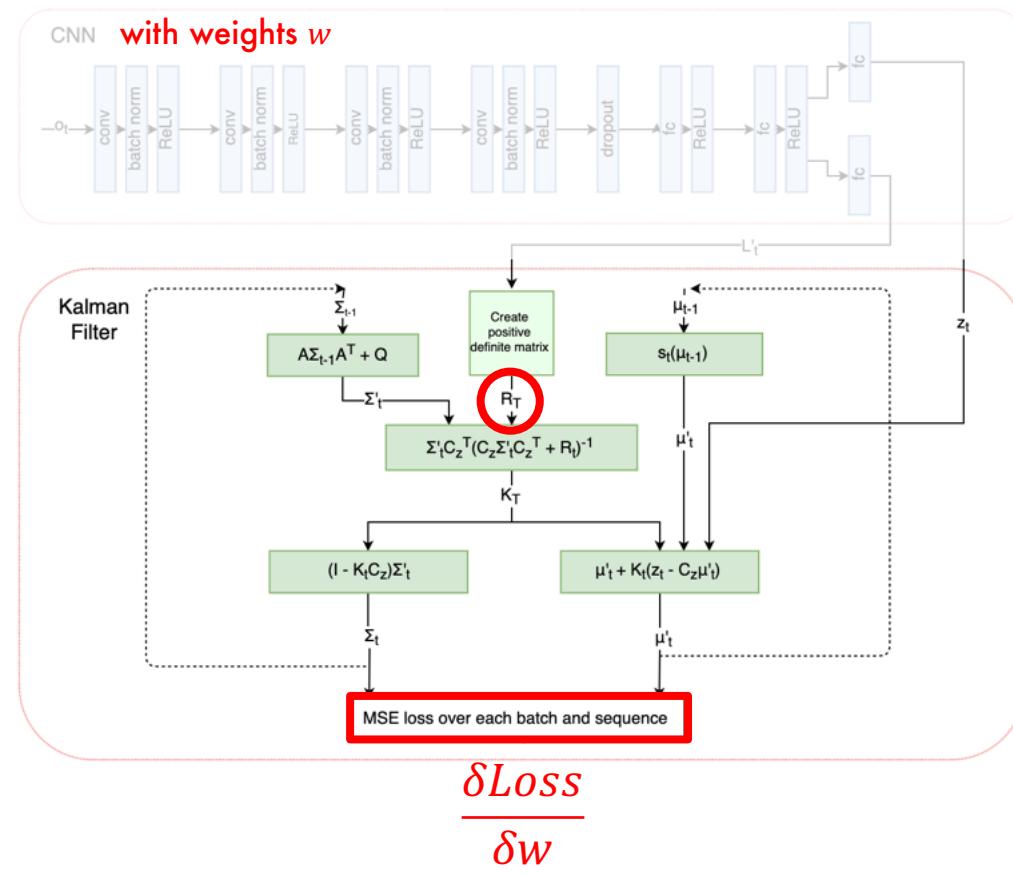
R is high if red disk is
occluded



Differentiable Kalman Filter - Structure



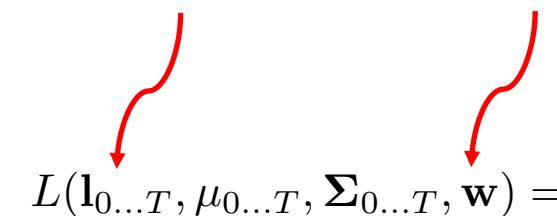
$$L' L^T = R$$



Differentiable Kalman Filter – Loss Function

$$L(\mathbf{l}_{0 \dots T}, \mu_{0 \dots T}, \Sigma_{0 \dots T}, \mathbf{w}) =$$
$$\lambda_1 \sum_{t=0}^T \frac{1}{2} \underbrace{((\mathbf{l}_t - \mu_t)^T \Sigma_t^{-1} (\mathbf{l}_t - \mu_t) + \log(|\Sigma_t|))}_{\text{Negative log likelihood of ground truth given current belief}} + \lambda_2 \sum_{t=0}^T \underbrace{\| (\mathbf{l}_t - \mu_t) \|_2}_{\text{Mean-Squared Error}} + \lambda_3 \underbrace{\| \mathbf{w} \|_2}_{\text{Regularization}}$$

Ground truth state **Network weights**



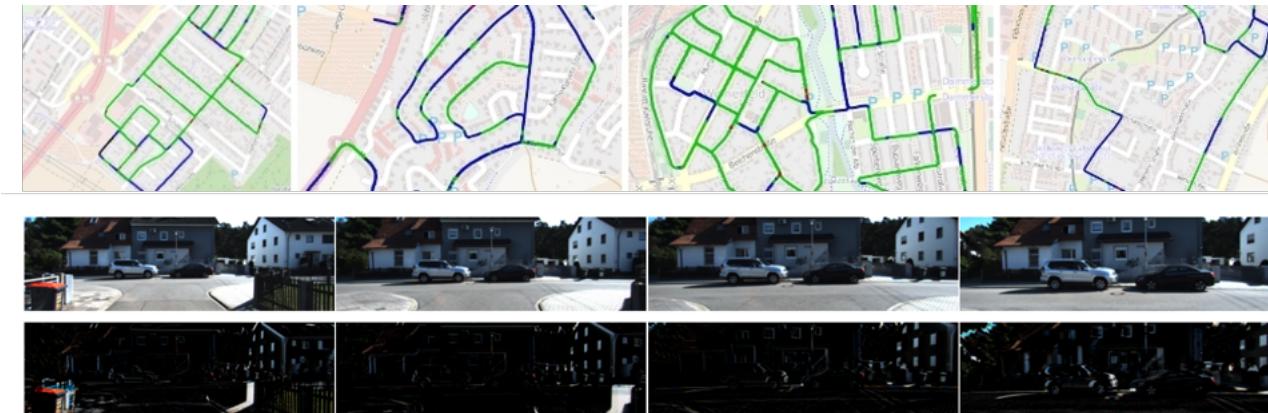
Differentiable Kalman Filter – Experiments and Baselines

Table 1: Benchmark Results

State Estimation Model	# Parameters	RMS test error $\pm \sigma$
feedforward model	7394	0.2322 ± 0.1316
piecewise KF	7397	0.1160 ± 0.0330
LSTM model (64 units)	33506	0.1407 ± 0.1154
LSTM model (128 units)	92450	0.1423 ± 0.1352
BKF (ours)	7493	0.0537 ± 0.1235

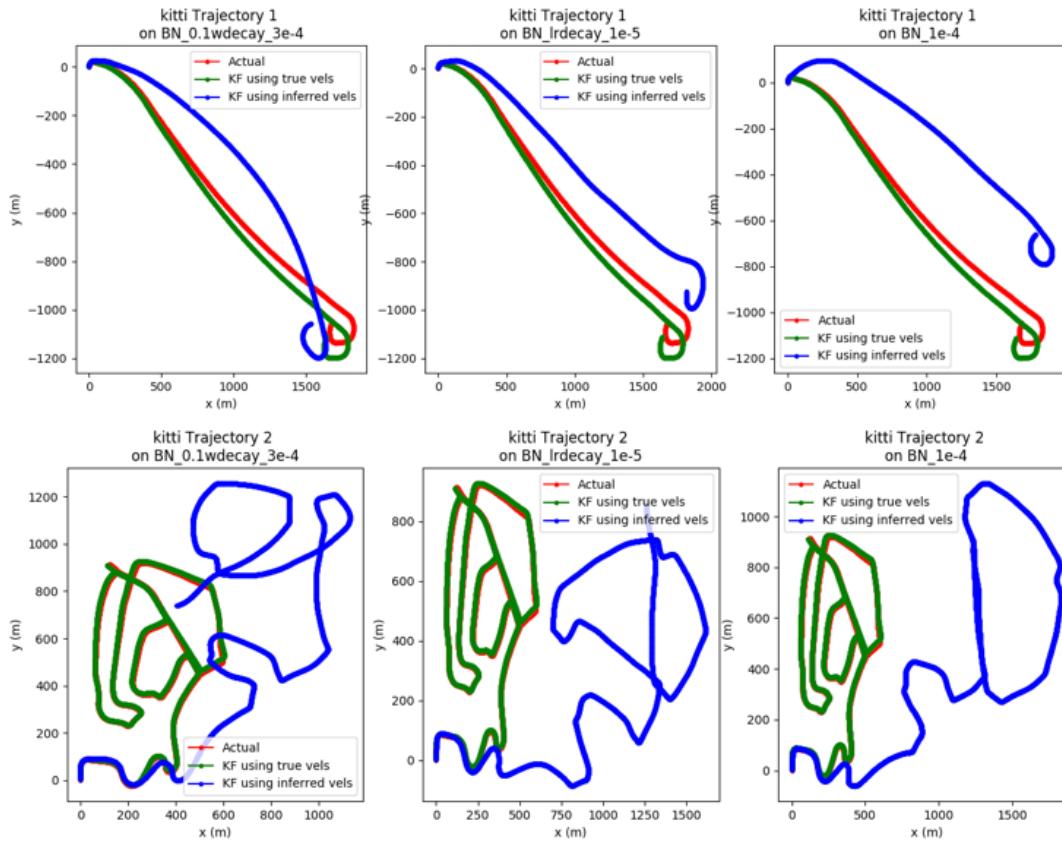
Differentiable Kalman Filter – Experiments and Baselines

- Kitti – Visual Odometry Dataset
- 22 stereo sequences with LIDAR
 - 11 sequences with ground truth (GPS/IMU data)
 - 11 sequences without ground truth (for evaluation)



Differentiable Kalman Filter – Experiments and Baselines

Results reproduced by Claire Chen



Blue – Result of BackpropKF

Red – Ground truth

Green – w/ Ground truth velocities

CS231

Introduction to Computer Vision

1891

Next lecture:

Neural Radiance Fields for Novel View
Synthesis

