SELF-DRIVING CARS BEHAVIOR ESTIMATION

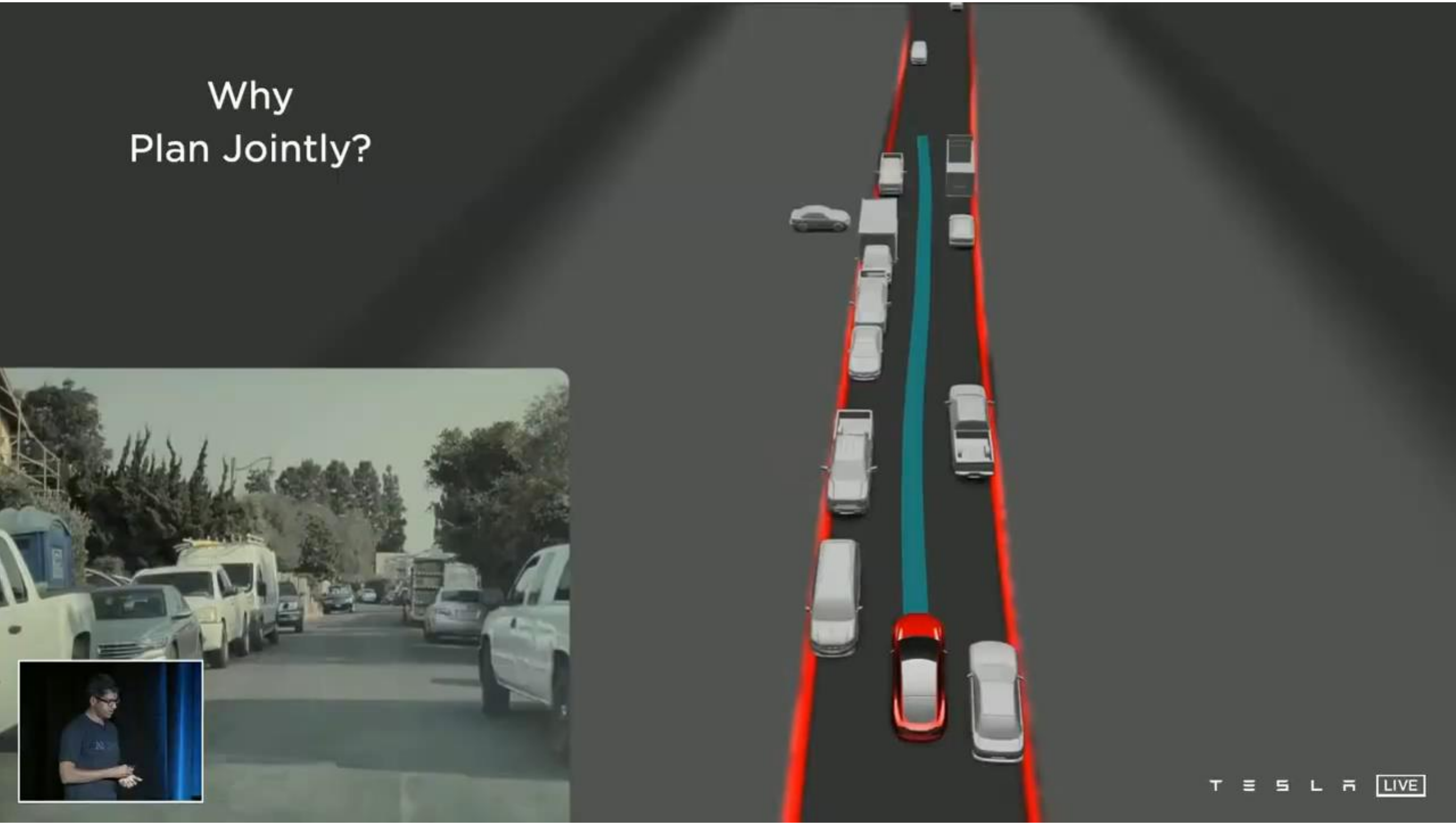
Photogrammetry & Robotics Lab

Behavior Estimation for Self-Driving Cars

Benedikt Mersch

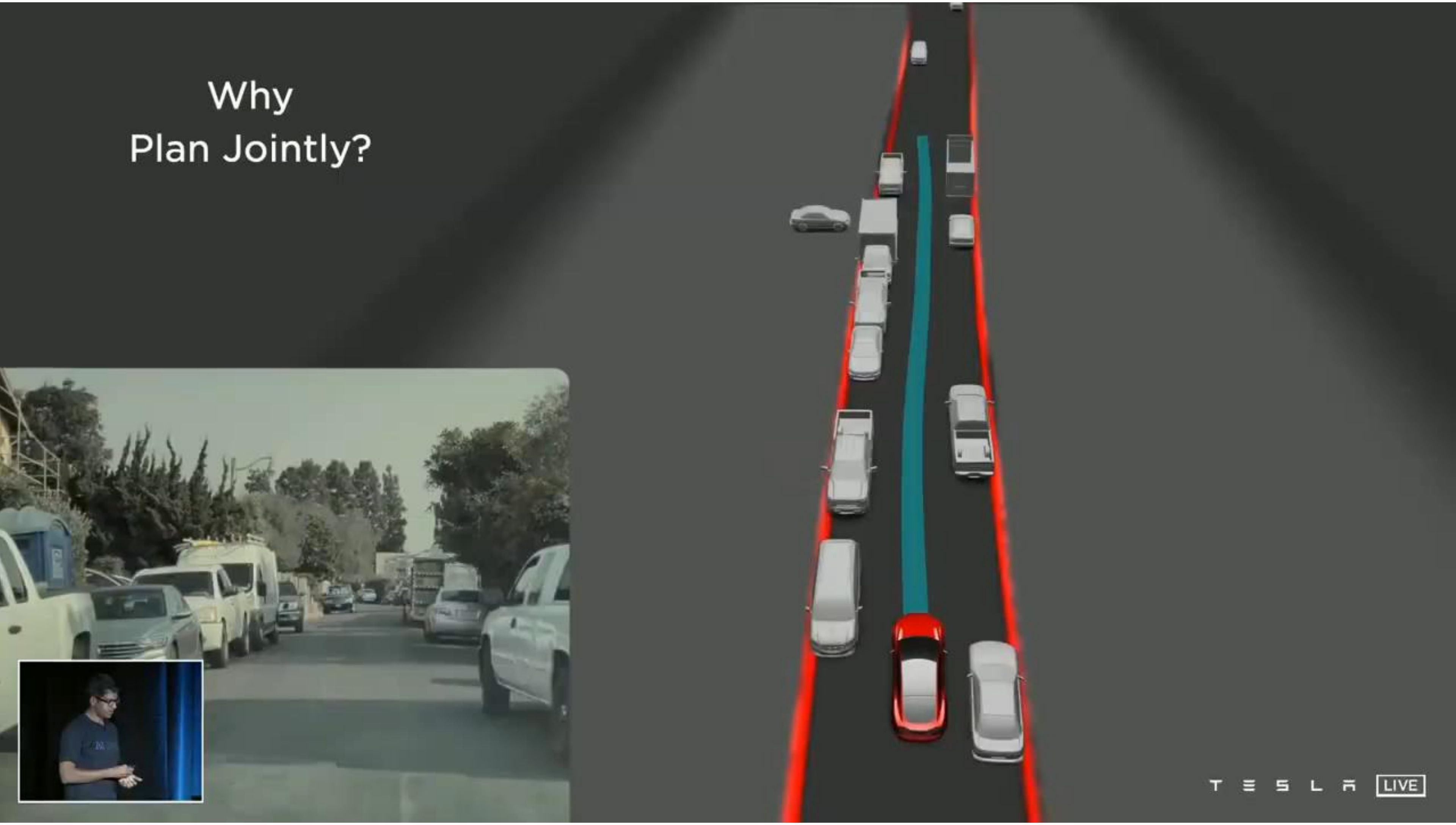
Part of the Course: Techniques for Self-Driving Cars by
C. Stachniss, J. Behley, N. Chebrolu, B. Mersch, I. Bogoslavskyi, L. Peters

Motivation



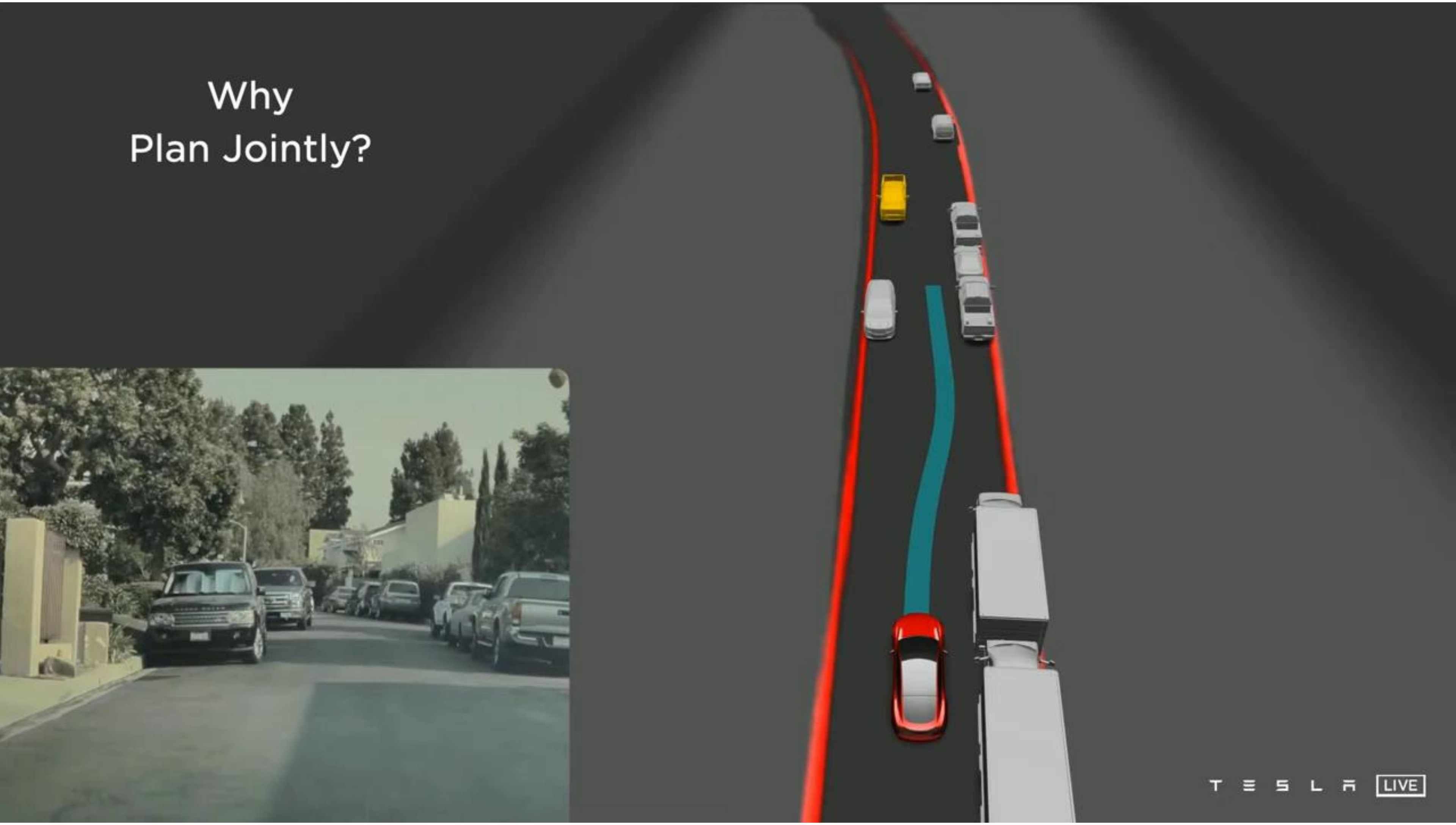
[Courtesy of Tesla]

Motivation



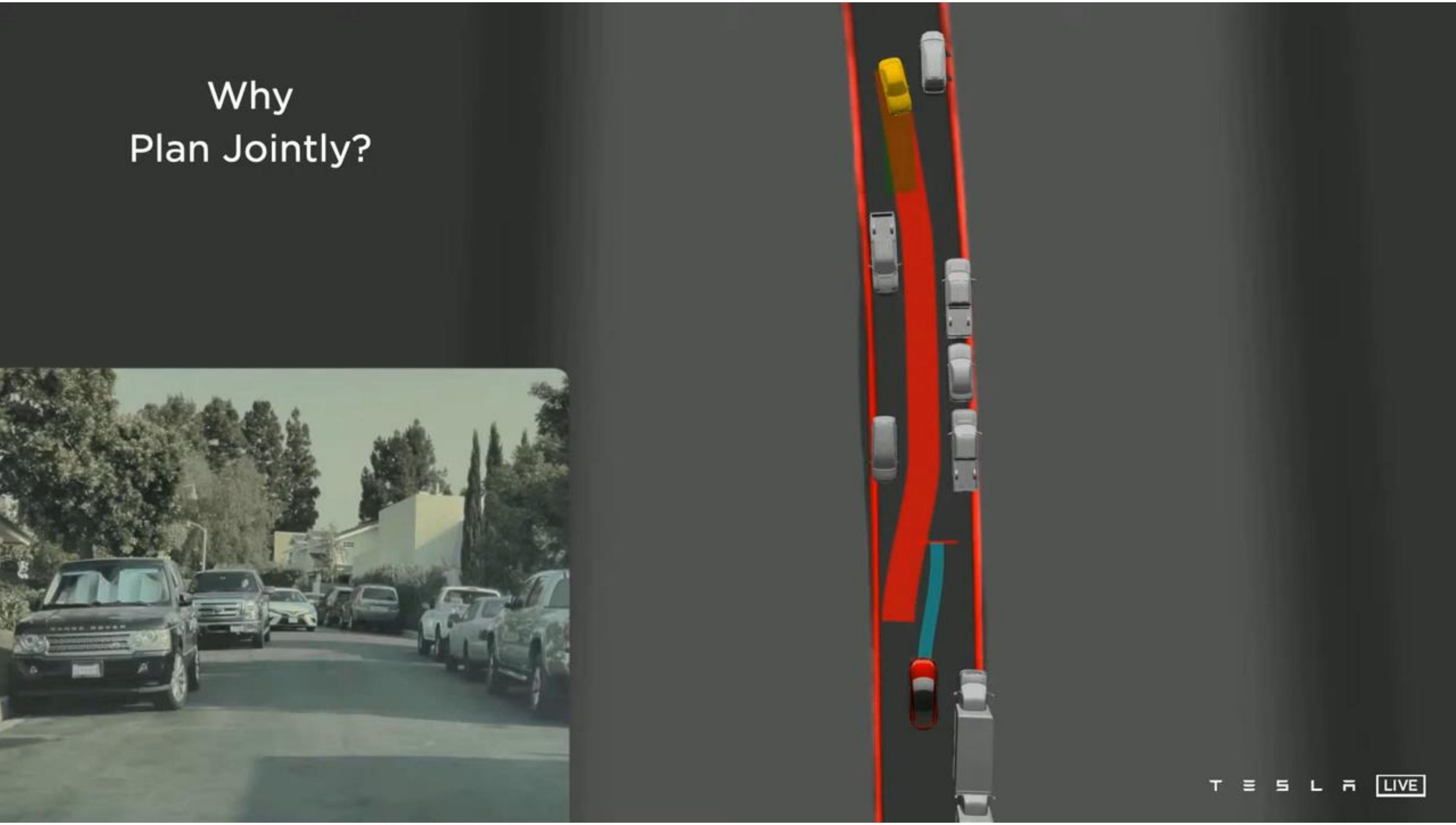
[Courtesy of Tesla]

Motivation



[Courtesy of Tesla]

Motivation



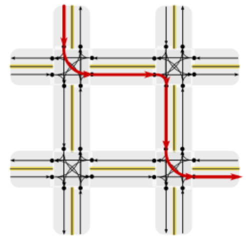
[Courtesy of Tesla]

Planning in General

High abstraction

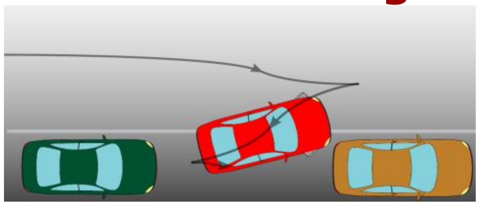
Low frequency

Global Planning



Behavior Planning

Local Planning



Low abstraction

High frequency

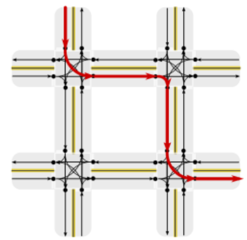
[Paden et al.]

Planning in General

High abstraction → Low abstraction

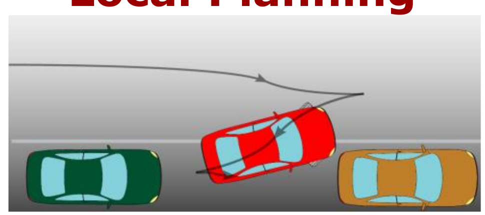
Low frequency → High frequency

Global Planning



Behavior Planning

Local Planning



[Paden et al.]

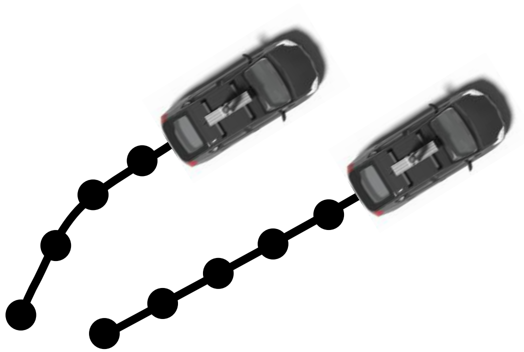
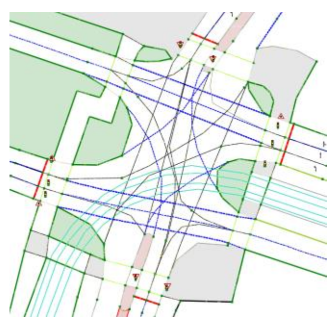
Behavior Planning vs. Estimation

- Plan maneuvers to follow global plan
- Transitions between maneuvers depend on traffic participants
- Behavior estimation can support decision making

[Courtesy of Waymo]

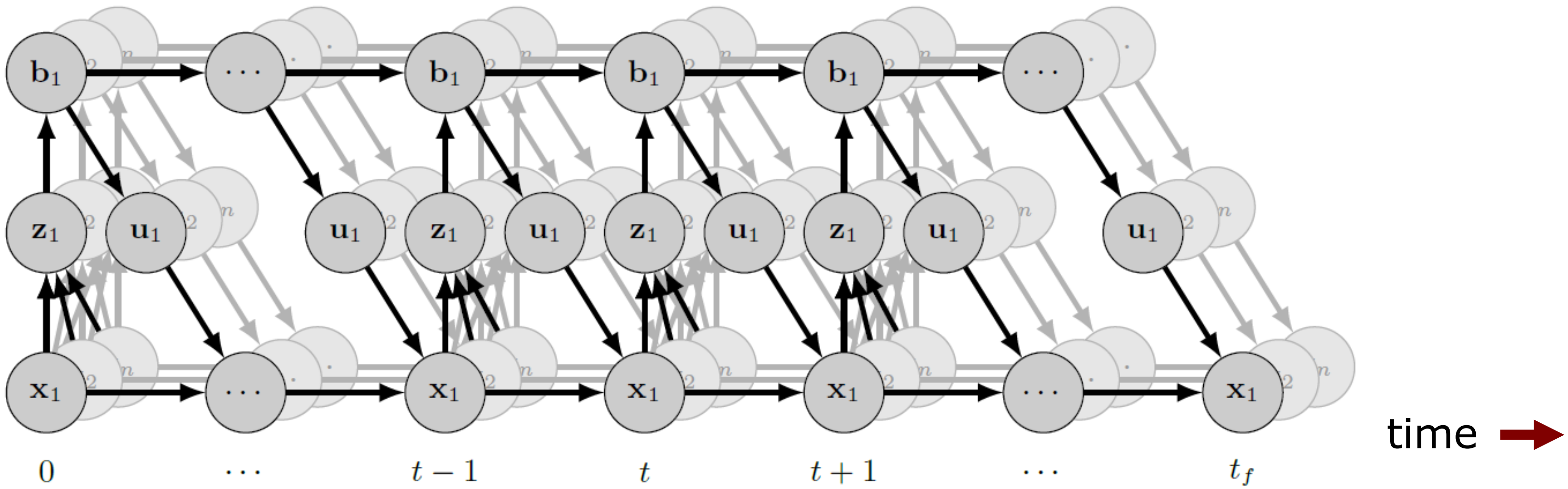
What Information Can We Use?

- Infer behavior of other traffic participants from
  - past states (e.g. position, velocity, acceleration)
  - map information
  - sensor data (e.g. camera, LiDAR, radar)


[Poggenhans et al.]

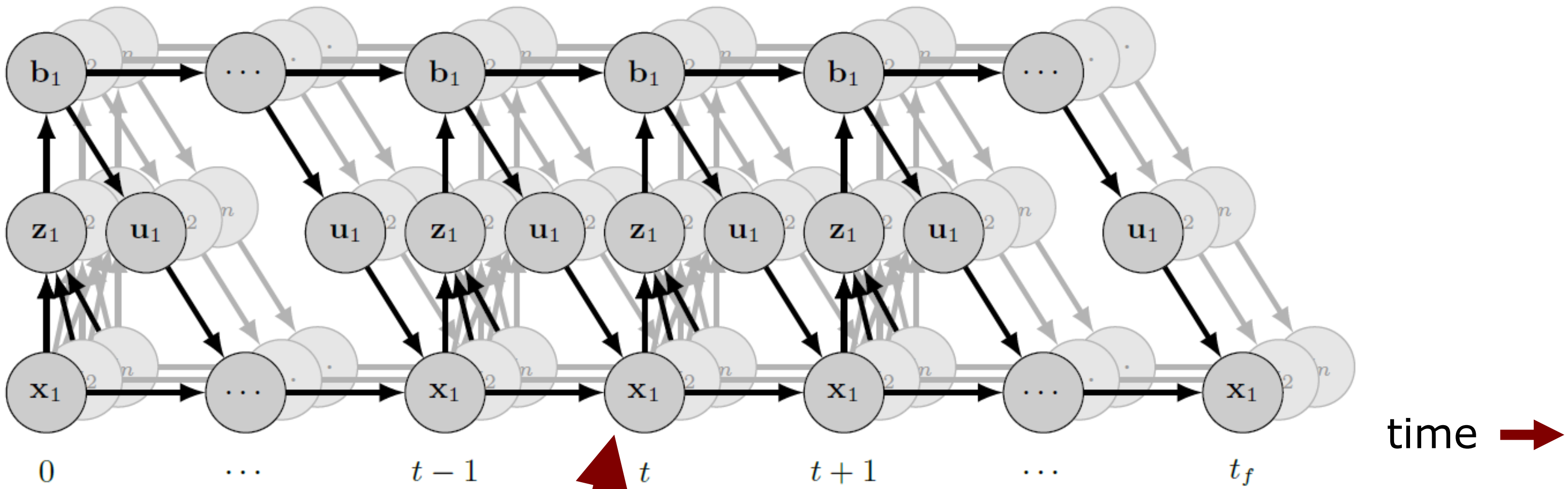
Graphical Model



- n-agent partially observable stochastic game
- Physical state $x_i^{(t)}$, observations $z_i^{(t)}$, internal state $b_i^{(t)}$ and control action $u_i^{(t)}$

[Brown et al.]

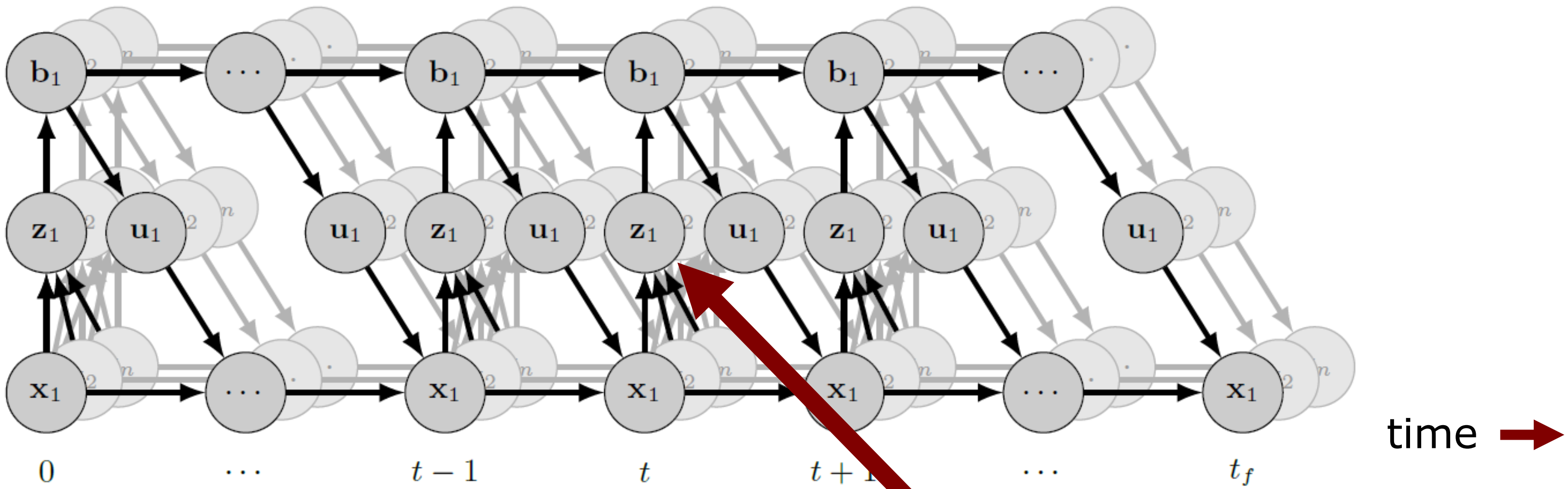
Graphical Model



- n-agent partially observable stochastic game
- Physical state $x_i^{(t)}$, observations $z_i^{(t)}$, internal state $b_i^{(t)}$ and control action $u_i^{(t)}$

[Brown et al.]

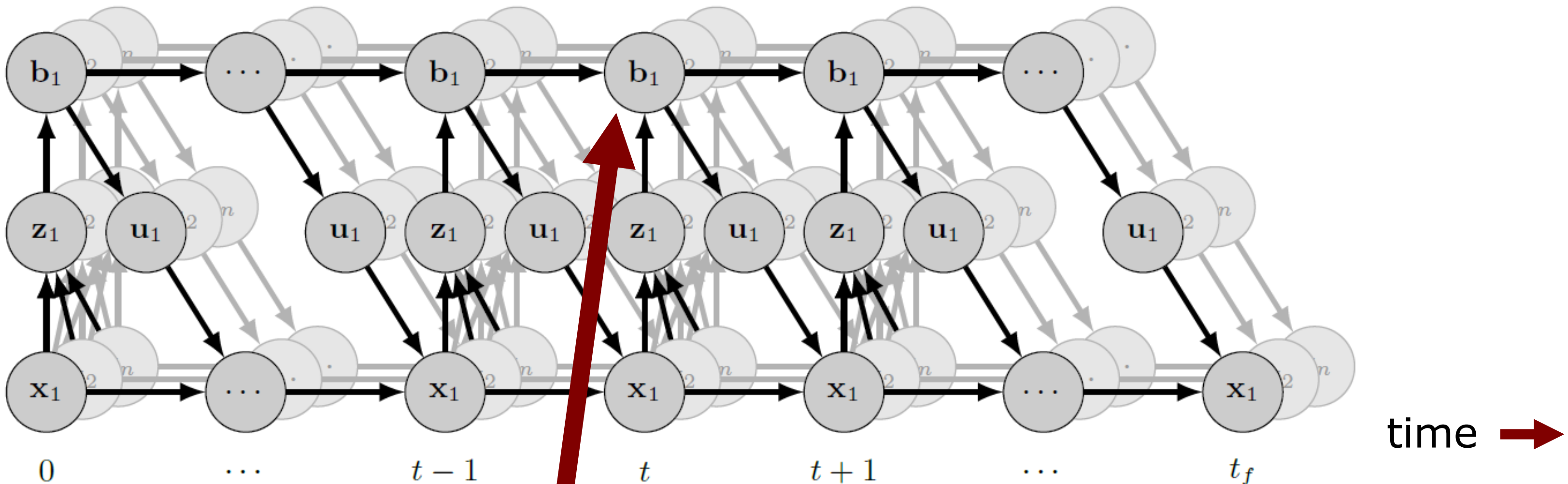
Graphical Model



- n-agent partially observable stochastic game
- Physical state $x_i^{(t)}$, observations $z_i^{(t)}$, internal state $b_i^{(t)}$ and control action $u_i^{(t)}$

[Brown et al.]

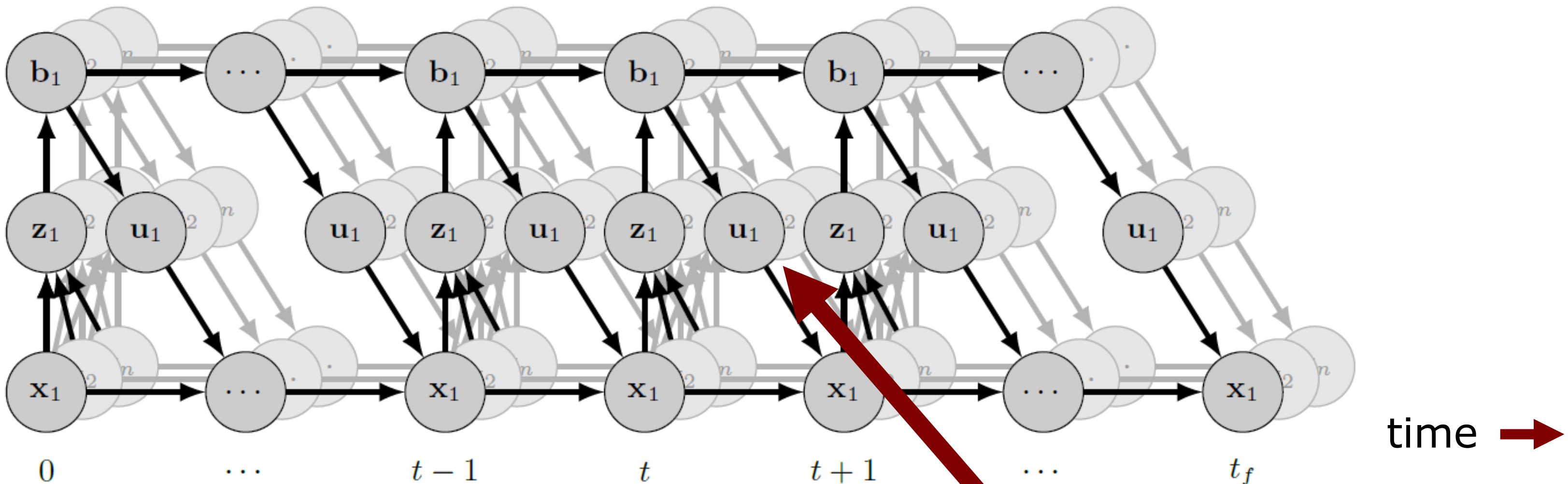
Graphical Model



- n-agent partially observable stochastic game
- Physical state $x_i^{(t)}$, observations $z_i^{(t)}$, internal state $b_i^{(t)}$ and control action $u_i^{(t)}$

[Brown et al.]

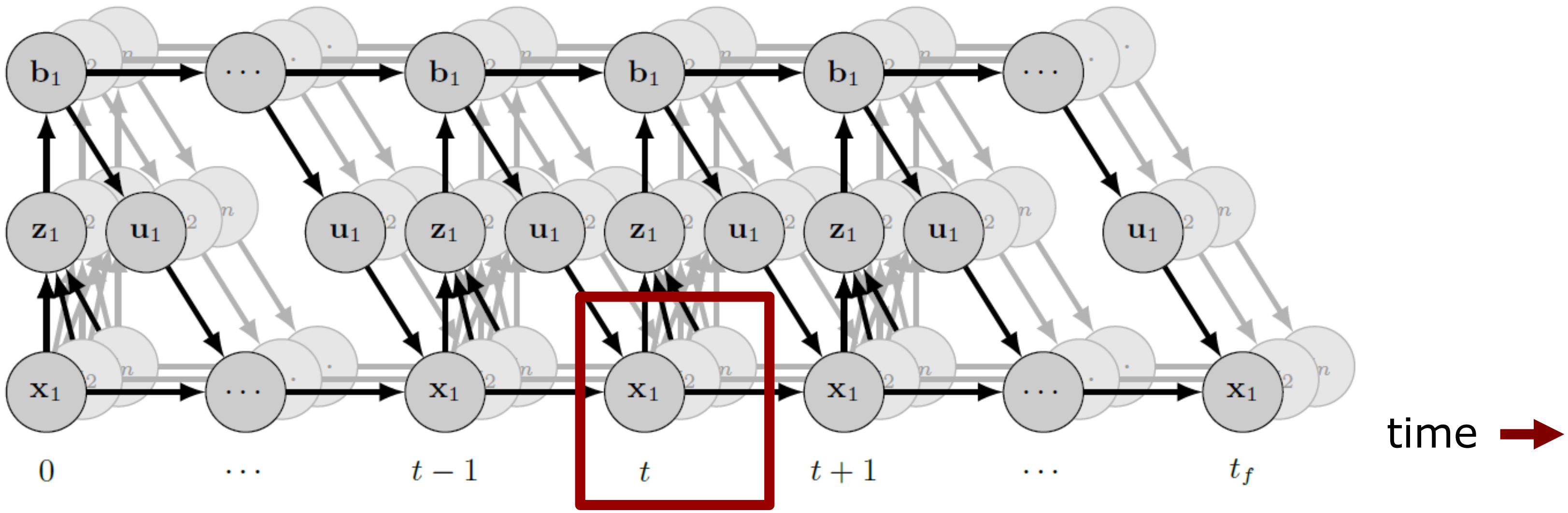
Graphical Model



- n-agent partially observable stochastic game
- Physical state $x_i^{(t)}$, observations $z_i^{(t)}$, internal state $b_i^{(t)}$ and control action $u_i^{(t)}$

[Brown et al.]

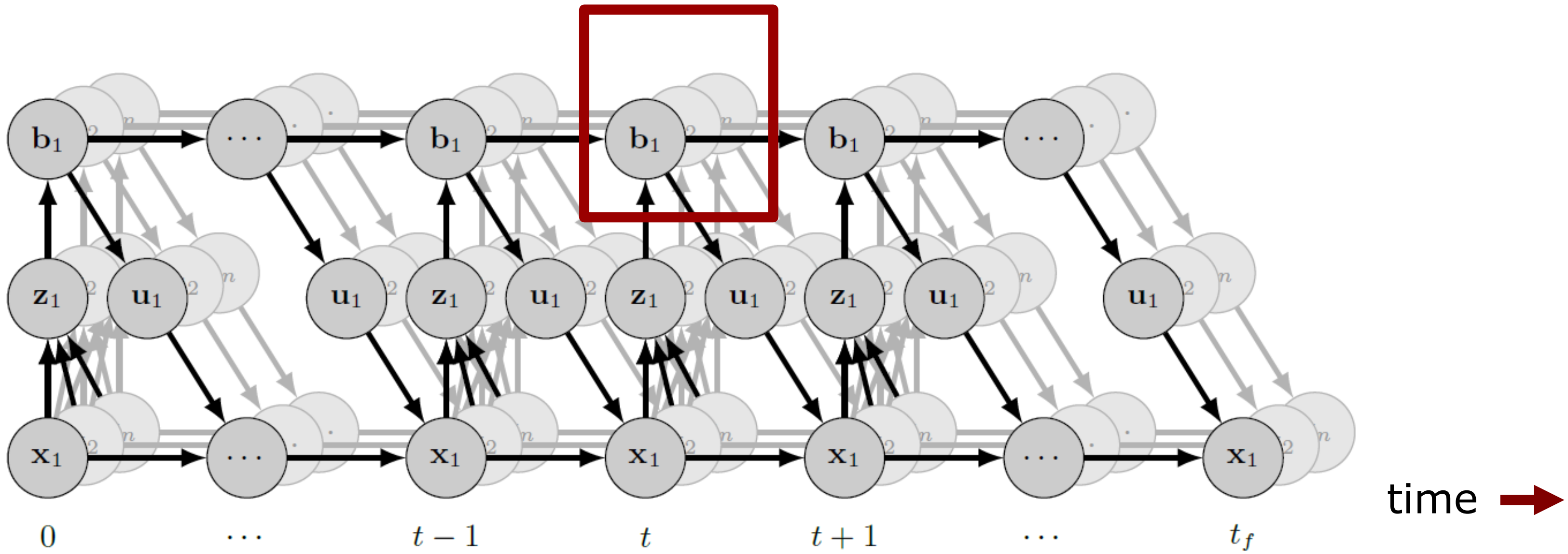
Graphical Model



- State estimation: $x_{1:n}^{(t)}$
- Intention and trait estimation: $b_{1:n}^{(t)}$
- Motion prediction: $x_{1:n}^{(t+1:t_f)}$

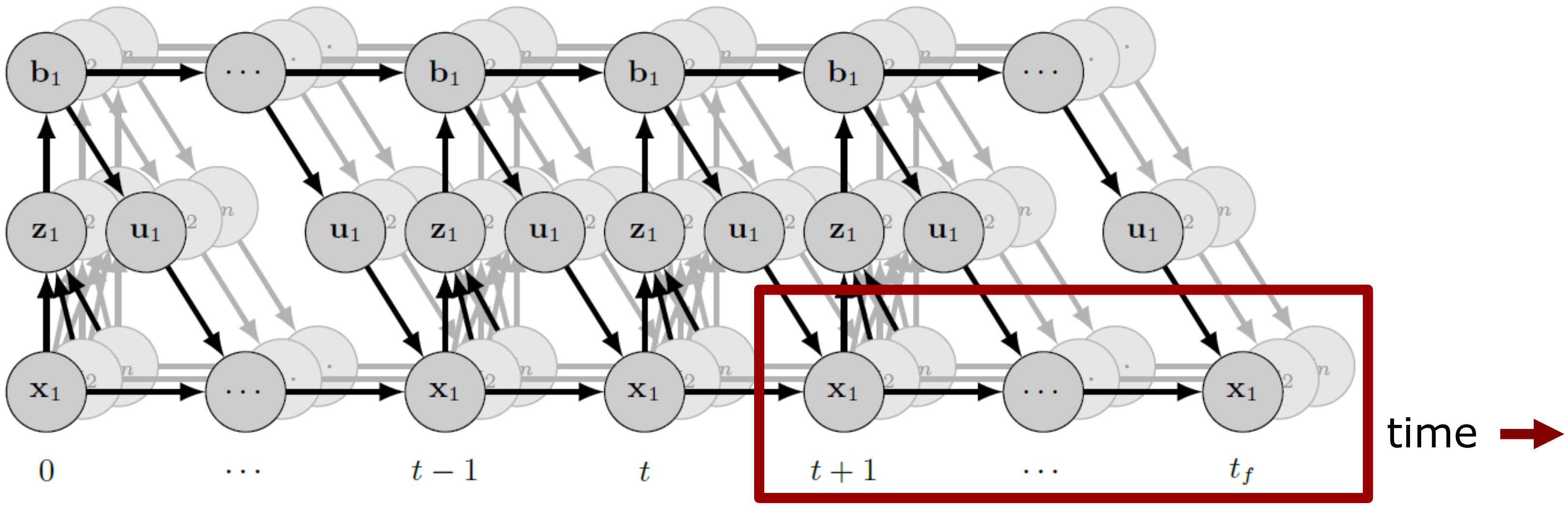
[Brown et al.]

Graphical Model



- State estimation: $x_{1:n}^{(t)}$
- Intention and trait estimation: $b_{1:n}^{(t)}$
- Motion prediction: $x_{1:n}^{(t+1:t_f)}$

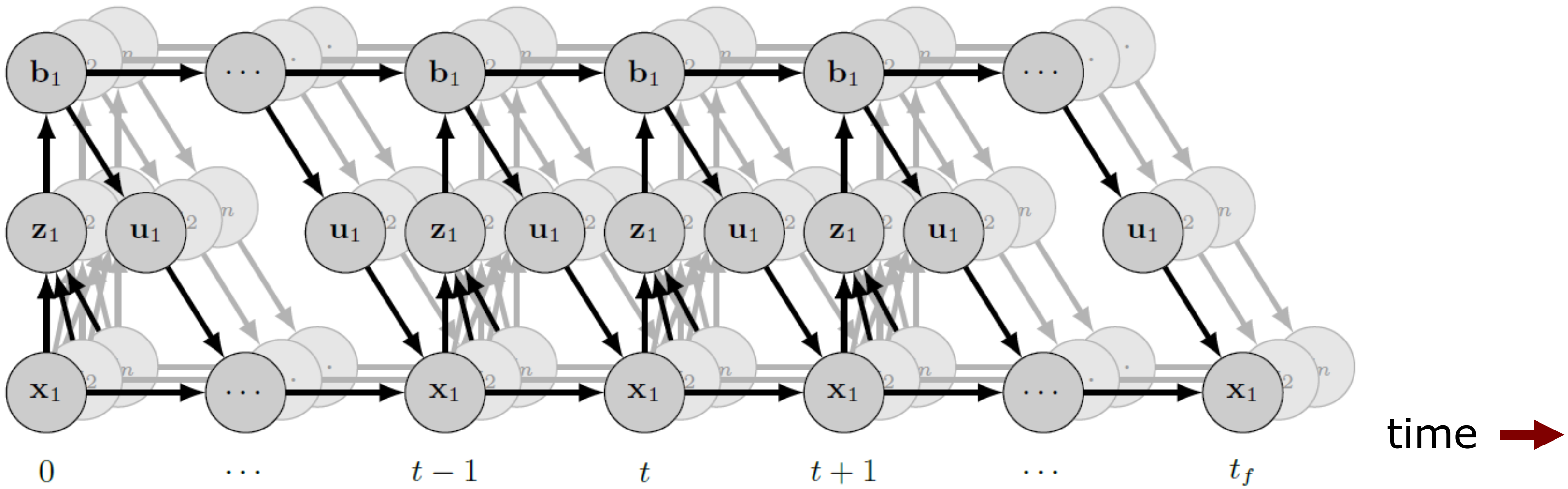
Graphical Model



- State estimation: $x_{1:n}^{(t)}$
- Intention and trait estimation: $b_{1:n}^{(t)}$
- Motion prediction: $x_{1:n}^{(t+1:t_f)}$

[Brown et al.]

Graphical Model



- State estimation: $x_{1:n}^{(t)}$
- Intention and trait estimation: $b_{1:n}^{(t)}$
- Motion prediction: $x_{1:n}^{(t+1:t_f)}$

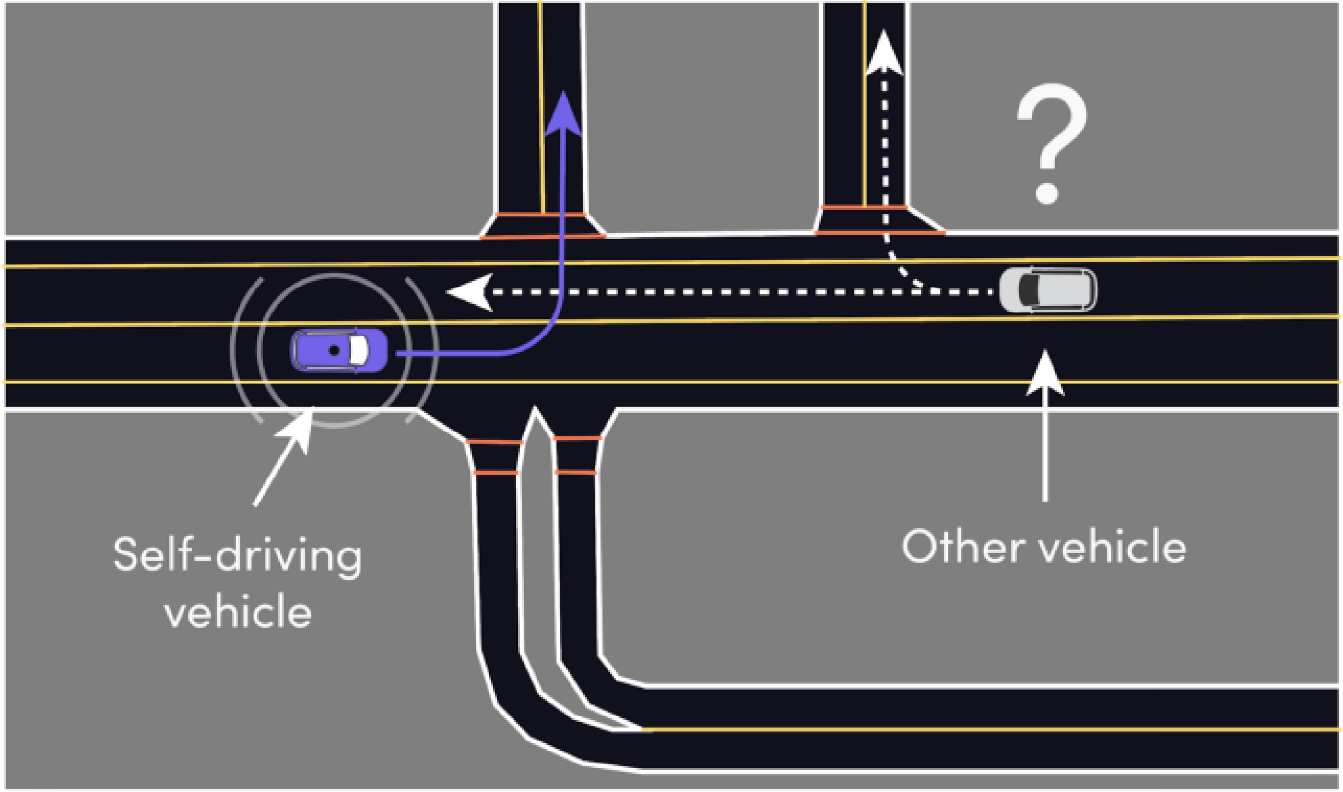
This lecture

[Brown et al.]

Intention Estimation

Intention Estimation

Is the white car turning right or driving straight?



[Courtesy of Lyft]

Intention Estimation

- Infer what other drivers **want to do** in the future
- Often probability distribution over high-level behavior modes (e.g. lane changing, turning, overtaking)
- Motion prediction can be conditioned on estimated intention

Intention Estimation Paradigms

- **Recursive** estimation

$$p\left(b_{1:n}^{(t)}\right) = f\left(p\left(b_{1:n}^{(t-1)}\right), {z_{\text{ego}}}^{(t)}\right)$$

- **Single-shot** estimation

$$p\left(b_{1:n}^{(t)}\right) = f\left({z_{\text{ego}}}^{(t_p:t)}\right)$$

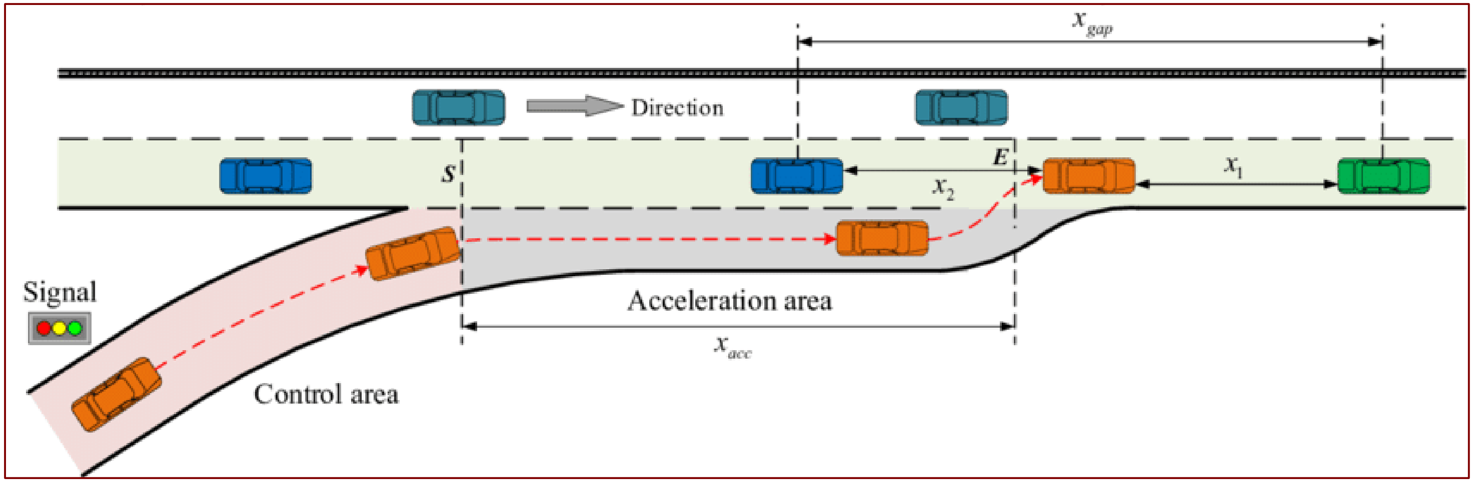
- Bayesian model
- Deep learning methods (black box)
- Game-theory

[Brown et al.]

Trait Estimation

Trait Estimation

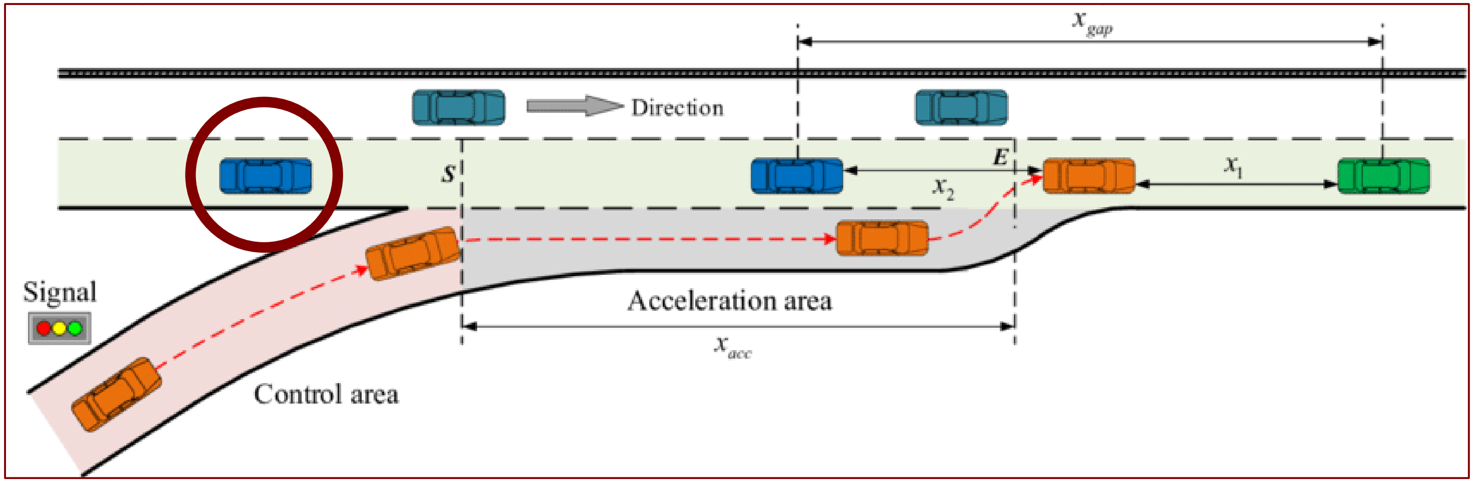
- In contrast to intention, a trait defines **how** the goal should be accomplished
- Traits depend on e.g. driver skill, preferences, aggressiveness



[Courtesy of Li and Dai]

Trait Estimation

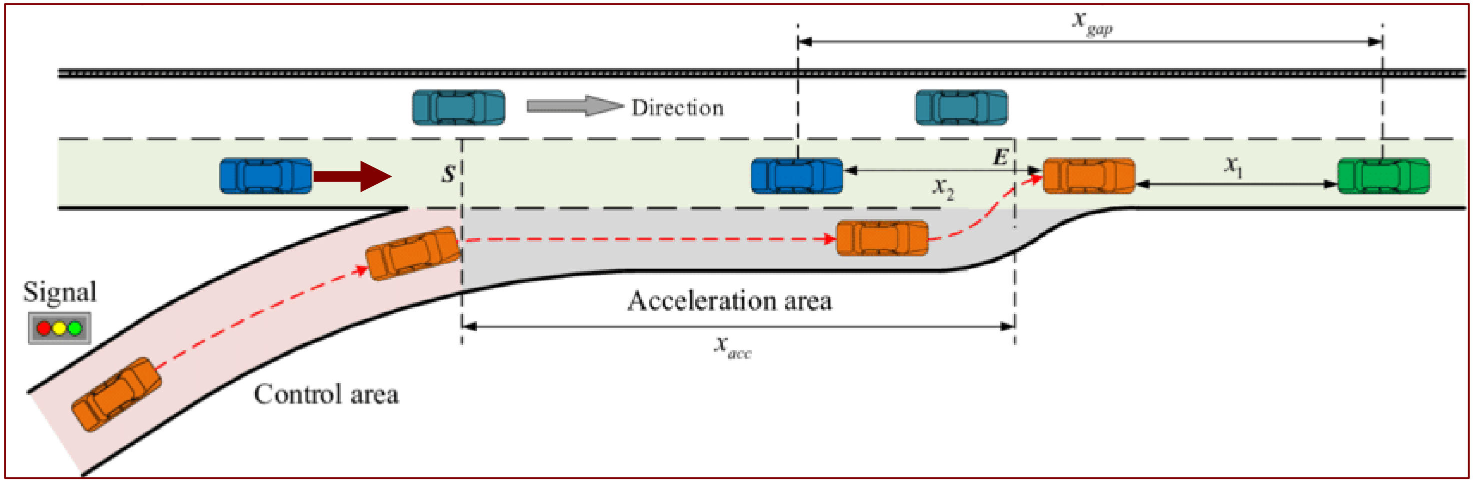
- In contrast to intention, a trait defines **how** the goal should be accomplished
- Traits depend on e.g. driver skill, preferences, aggressiveness



[Courtesy of Li and Dai]

Trait Estimation

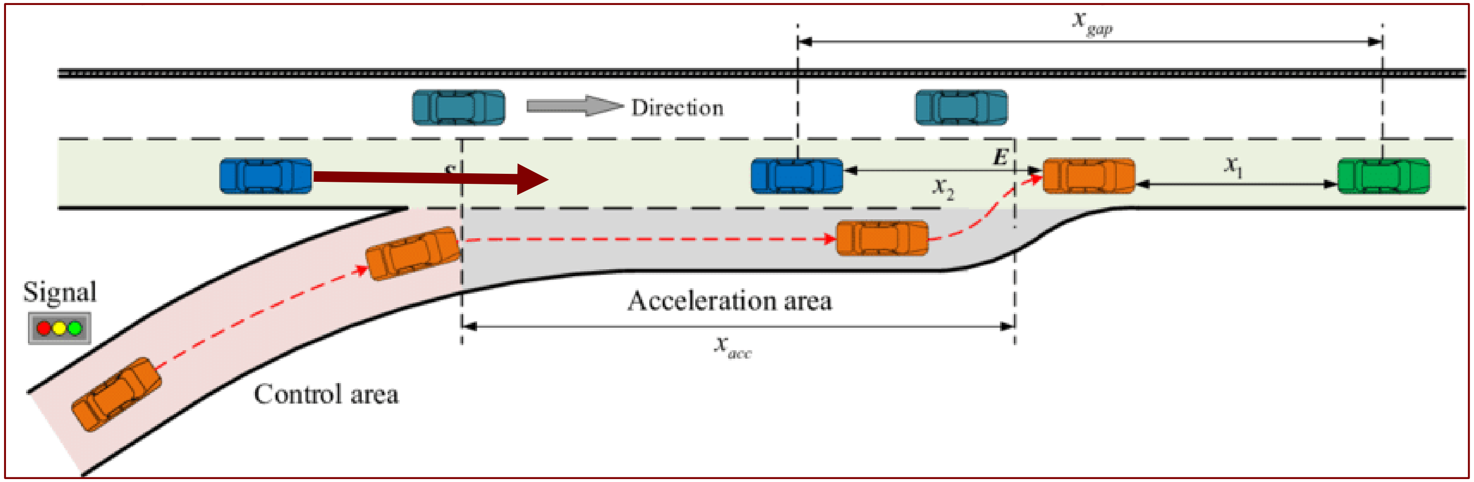
- In contrast to intention, a trait defines **how** the goal should be accomplished
- Traits depend on e.g. driver skill, preferences, aggressiveness



[Courtesy of Li and Dai]

Trait Estimation

- In contrast to intention, a trait defines **how** the goal should be accomplished
- Traits depend on e.g. driver skill, preferences, aggressiveness



[Courtesy of Li and Dai]

Trait Estimation

- Example traits:
  - Policy parameters of a driver model like minimum desired gap, maximum feasible acceleration
  - Parametric cost function that players try to optimize
- **Offline**: Estimate parameters in advance based on observations
- **Online**: Update parameters for previously unobserved drivers

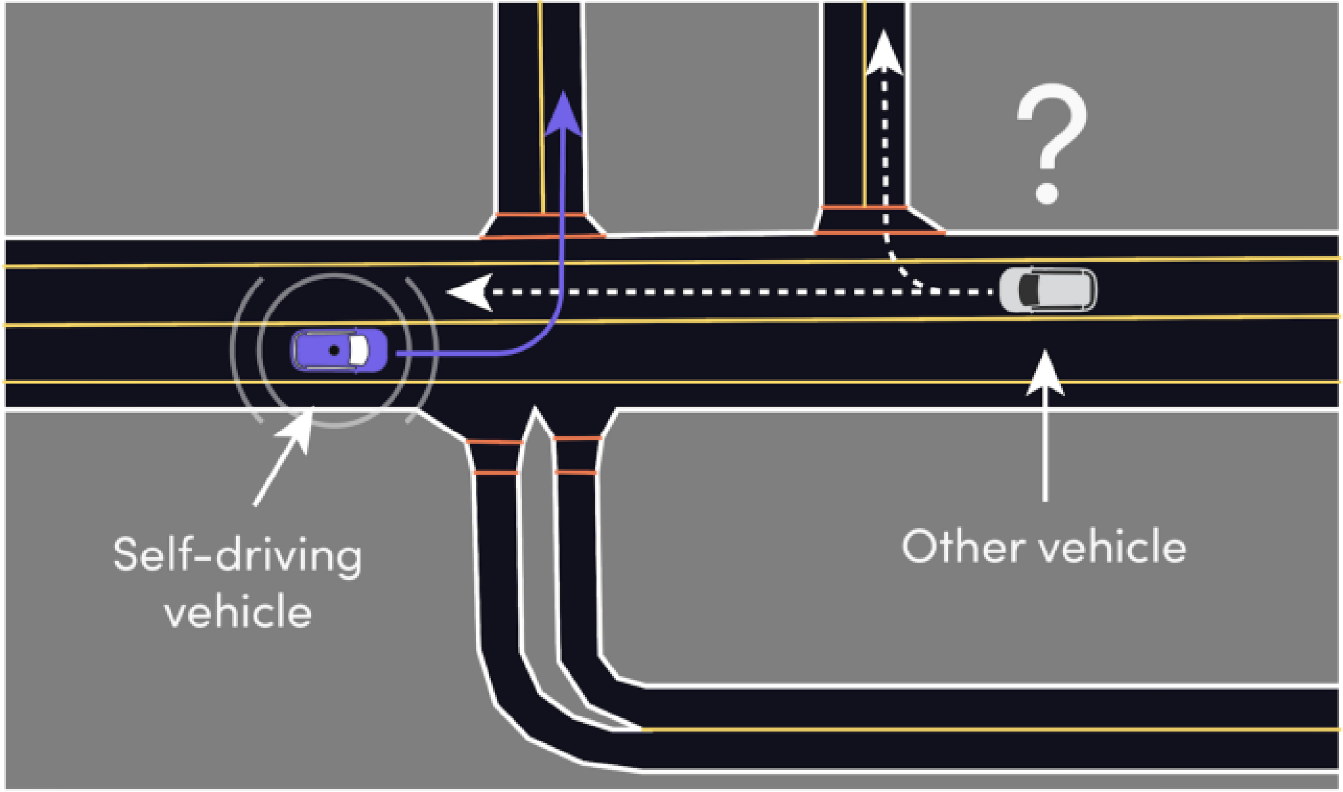
Trait Estimation Paradigms

- Bayesian model
- Optimization e.g. inverse reinforcement learning
- Heuristics (use for example recommended parameters)

Motion Prediction

Motion Prediction

Now we are interested in the **full trajectory**, not the high-level action



[Courtesy of Lyft]

Motion Prediction

- Predict future states $x_{1:n}^{(t+1:t_f)}$ of n traffic participants
- State transition model

$$x_i^{(t+1)} \sim F_i\left(x_i^{(t)}, u_i^{(t)}\right)$$

can be physics-/geometry-based or learned from data

- Future interactions among traffic participants

Motion Hypotheses

- Single trajectories
- Multi-modal trajectories
- 3D/2D Bounding boxes
- Gaussian (mixture) distributions
- Occupancy grid maps
- Forward/backward reachable sets
- Raw sensor data

Motion Prediction Paradigms

- **Closed-loop forward simulation**
  - \+ Interaction aware
  - \- Requires control policy

Algorithm 1 Motion Prediction via Forward Simulation

---

for $\tau \in t, \ldots, t_f - 1$
  for $i \in 1, \ldots, n$
    $\mathbf{z}_i^{(\tau)} \leftarrow \boldsymbol{G}_i(\mathbf{x}_{1:n}^{(\tau)})$ ▷ receive observation
    $\mathbf{b}_i^{(\tau)} \leftarrow \boldsymbol{H}_i(\mathbf{b}_i^{(\tau-1)}, \mathbf{z}_i^{(\tau)})$ ▷ update internal state
    $\mathbf{u}_i^{(\tau)} \leftarrow \pi_i(\mathbf{b}_i^{(\tau)})$ ▷ select action
    $\mathbf{x}_i^{(\tau+1)} \leftarrow \boldsymbol{F}_i(\mathbf{x}_i^{(\tau)}, \mathbf{u}_i^{(\tau)})$ ▷ step forward

---

[Brown et al.]

Motion Prediction Paradigms

- **Closed-loop forward simulation**
  + Interaction aware
  - Requires control policy

Algorithm 1 Motion Prediction via Forward Simulation

---

for $\tau \in t, \ldots, t_f - 1$
  for $i \in 1, \ldots, n$
    $\mathbf{z}_i^{(\tau)} \leftarrow \boxed{\boldsymbol{G}_i(\mathbf{x}_{1:n}^{(\tau)})}$ ▷ receive observation
    $\mathbf{b}_i^{(\tau)} \leftarrow \boldsymbol{H}_i(\mathbf{b}_i^{(\tau-1)}, \mathbf{z}_i^{(\tau)})$ ▷ update internal state
    $\mathbf{u}_i^{(\tau)} \leftarrow \pi_i(\mathbf{b}_i^{(\tau)})$ ▷ select action
    $\mathbf{x}_i^{(\tau+1)} \leftarrow \boldsymbol{F}_i(\mathbf{x}_i^{(\tau)}, \mathbf{u}_i^{(\tau)})$ ▷ step forward

---

[Brown et al.]

Motion Prediction Paradigms

- **Closed-loop forward simulation**

  \+ Interaction aware

  \- Requires control policy

Algorithm 1 Motion Prediction via Forward Simulation

for $\tau \in t, \ldots, t_f - 1$
    for $i \in 1, \ldots, n$
        $\mathbf{z}_i^{(\tau)} \leftarrow \boldsymbol{G}_i(\mathbf{x}_{1:n}^{(\tau)})$ ▷ receive observation
        $\mathbf{b}_i^{(\tau)} \leftarrow \boldsymbol{H}_i(\mathbf{b}_i^{(\tau-1)}, \mathbf{z}_i^{(\tau)})$ ▷ update internal state
        $\mathbf{u}_i^{(\tau)} \leftarrow \boxed{\pi_i(\mathbf{b}_i^{(\tau)})}$ ▷ select action
        $\mathbf{x}_i^{(\tau+1)} \leftarrow \boldsymbol{F}_i(\mathbf{x}_i^{(\tau)}, \mathbf{u}_i^{(\tau)})$ ▷ step forward

[Brown et al.]

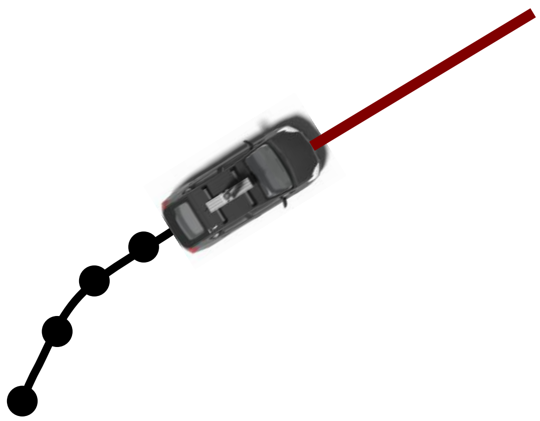
Motion Prediction Paradigms

- **Closed-loop forward simulation**
  + Interaction aware
  - Requires control policy
- **Independent prediction**
  + Fast and parallelizable
  - No interactions
- **Game-theoretic approaches**
  + Accounting for future interactions
  - Not easy to solve with many agents

Motion Prediction Methods

Constant Velocity Model

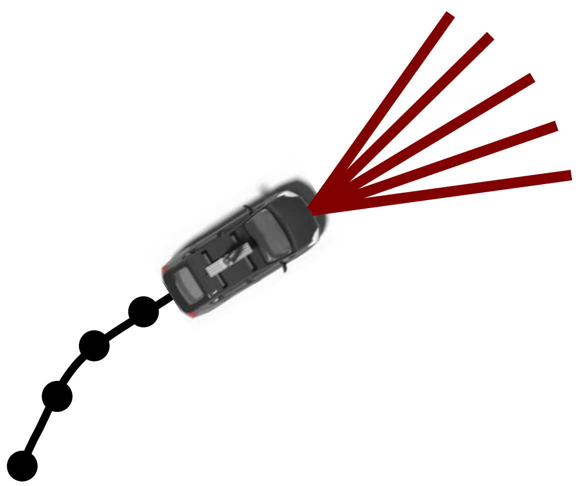
- Idea: Assume that agents move with constant velocity within prediction horizon



Single trajectory

Constant Velocity Model

- Idea: Assume that agents move with constant velocity within prediction horizon

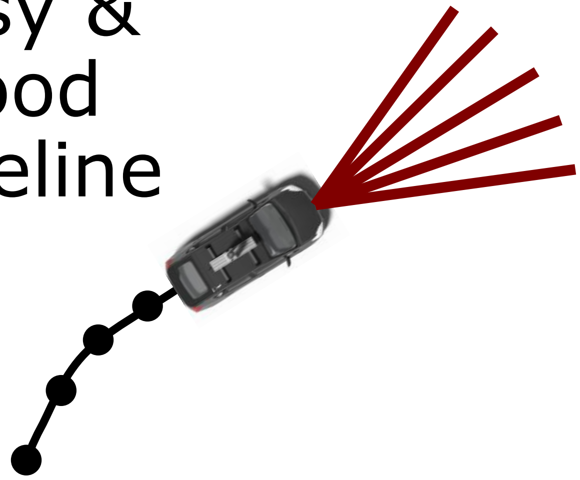


Multiple trajectories

Constant Velocity Model

- Idea: Assume that agents move with constant velocity within prediction horizon

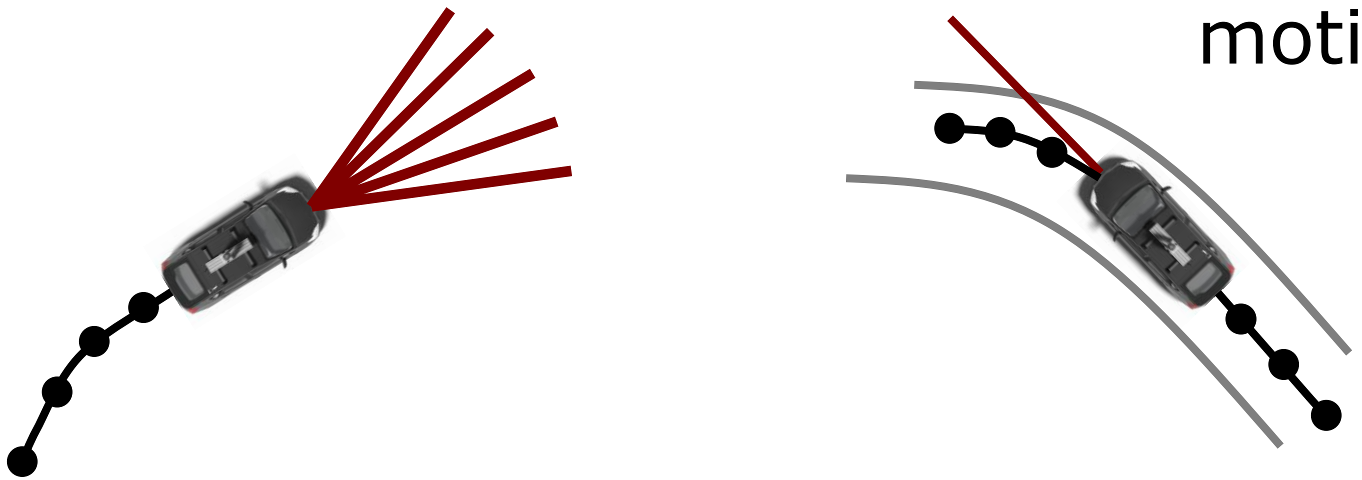
Easy & good baseline



Constant Velocity Model

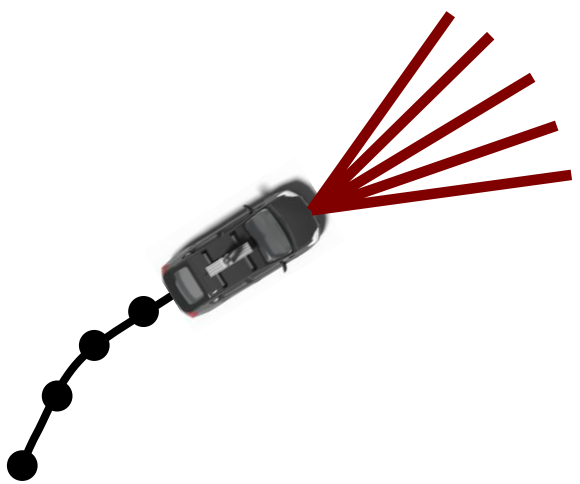
- Idea: Assume that agents move with constant velocity within prediction horizon

But fails for non-linear motion

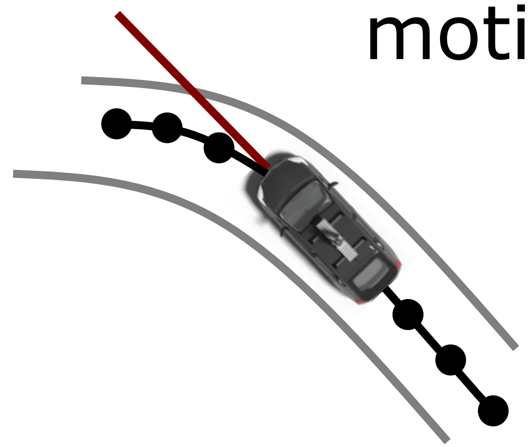


Constant Velocity Model

- Idea: Assume that agents move with constant velocity within prediction horizon



But fails for non-linear motion



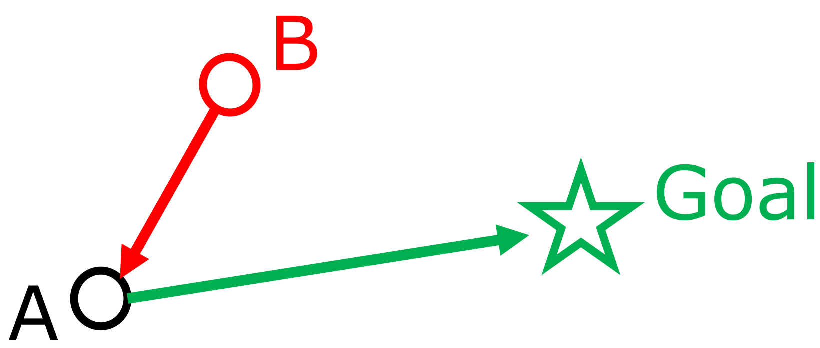
Alternative:
Constant acceleration, yaw rate

Social Forces

- Idea: Agents A, B act in a force field
- Get trajectory from differential equation

$$\ddot{x}(t) = \frac{F(t)}{m}$$

- Force depends on goal, humans and obstacles

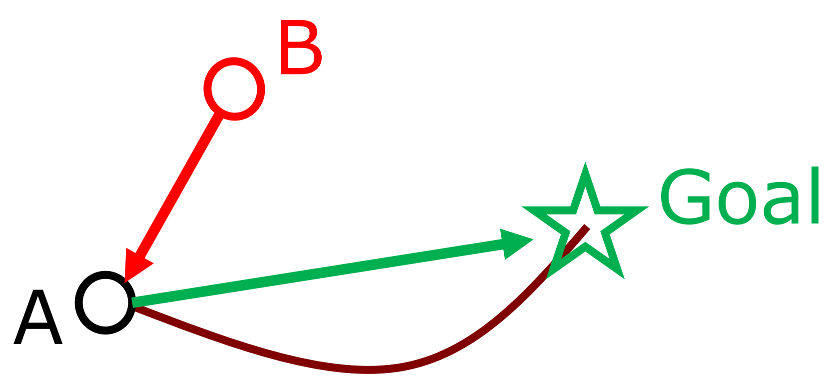


Social Forces

- Idea: Agents A, B act in a force field
- Get trajectory from differential equation

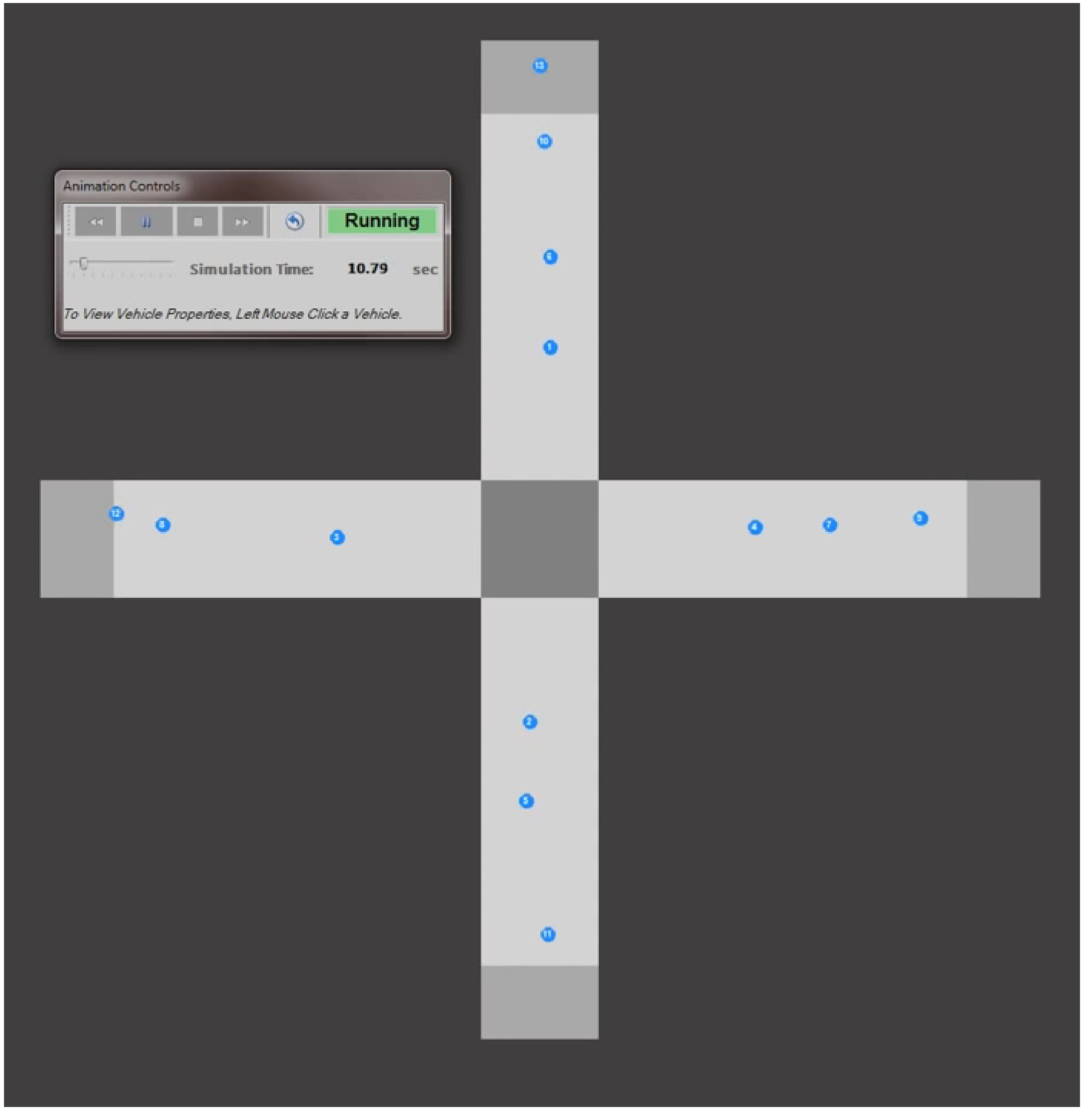
$$\ddot{x}(t) = \frac{F(t)}{m}$$

- Force depends on goal, humans and obstacles



[Helbing et al.]

Social Forces - Simulation



Social Forces

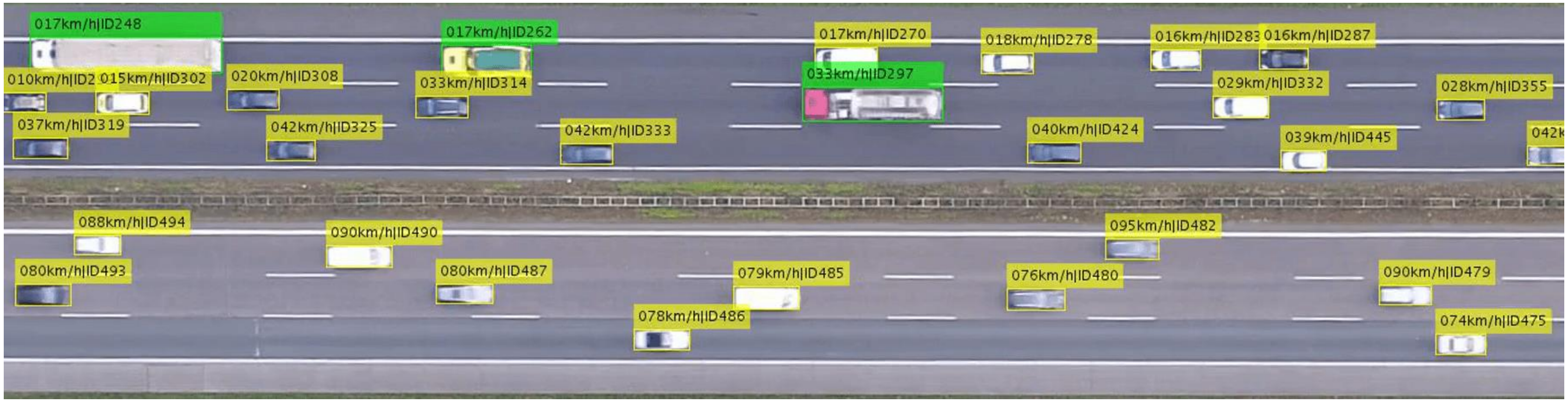
- Need to define and parameterize forces that explain behavior
- Less realistic predictions
- Does not apply for cars that follow road structures

Social Forces

- Need to define and parameterize forces that explain behavior
- Less realistic predictions
- Does not apply for cars that follow road structures

Can we model human driving behavior?

Intelligent Driver Model

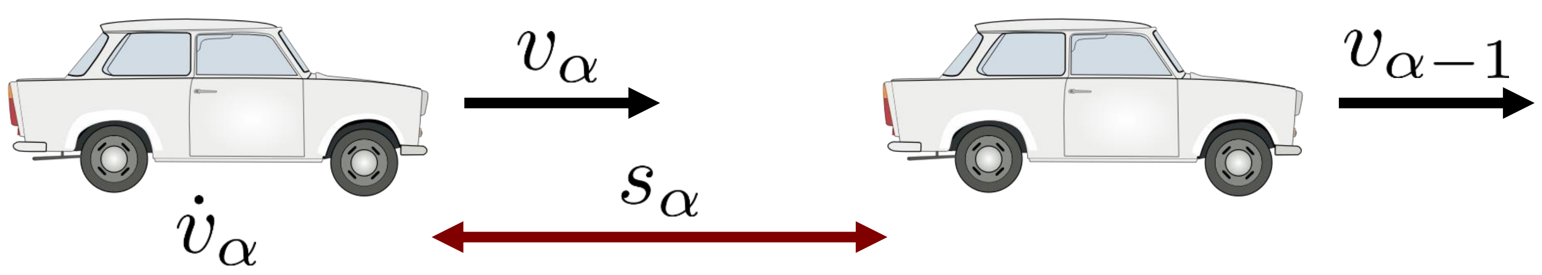


[Courtesy of highD]

Intelligent Driver Model

- Car following model with parameters θ
- Output: Acceleration of ego vehicle

$$\dot{v}_\alpha = f(v_\alpha, v_{\alpha-1}, s_\alpha, \theta)$$



[Treiber et al.]

Intelligent Driver Model

$$\dot{v}_\alpha = a \left(1 - \left(\frac{v_\alpha}{v_0} \right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha} \right)^2 \right)$$

with $\quad s^*(v, \Delta v_\alpha) = s_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}}$

- Maximum vehicle acceleration a
- Desired velocity v_0
- Minimum spacing s_0 in congested traffic
- Desired time headway T
- Comfortable braking deceleration b
- Exponent δ

[Treiber et al.]

Intelligent Driver Model

$$\dot{v}_\alpha = a\left(1 - \left(\frac{v_\alpha}{v_0}\right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2\right)$$

with $\quad s^*(v, \Delta v_\alpha) = s_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}}$

- Maximum vehicle acceleration a
- Desired velocity v_0
- Minimum spacing s_0 in congested traffic
- Desired time headway T
- Comfortable braking deceleration b
- Exponent δ

[Treiber et al.]

Intelligent Driver Model

$$\dot{v}_\alpha = a\left(1 - \left(\frac{v_\alpha}{v_0}\right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2\right)$$

with $\quad s^*(v, \Delta v_\alpha) = s_0 + v_\alpha T + \dfrac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}}$

- Maximum vehicle acceleration a
- Desired velocity v_0
- Minimum spacing s_0 in congested traffic
- Desired time headway T
- Comfortable braking deceleration b
- Exponent δ

[Treiber et al.]

Intelligent Driver Model

$$\dot{v}_\alpha = a\left(1 - \left(\frac{v_\alpha}{v_0}\right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2\right)$$

with $\quad s^*(v, \Delta v_\alpha) = s_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}}$

- Maximum vehicle acceleration a
- Desired velocity v_0
- Minimum spacing s_0 in congested traffic
- Desired time headway T
- Comfortable braking deceleration b
- Exponent δ

[Treiber et al.]

Intelligent Driver Model

$$\dot{v}_\alpha = a\left(1 - \left(\frac{v_\alpha}{v_0}\right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2\right)$$

with $$s^*(v, \Delta v_\alpha) = s_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}}$$

- Maximum vehicle acceleration a
- Desired velocity v_0
- Minimum spacing s_0 in congested traffic
- Desired time headway T
- Comfortable braking deceleration b
- Exponent δ

[Treiber et al.]

Intelligent Driver Model

$$\dot{v}_\alpha = a\left(1 - \left(\frac{v_\alpha}{v_0}\right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2\right)$$

with $s^*(v, \Delta v_\alpha) = s_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}}$

- Maximum vehicle acceleration a
- Desired velocity v_0
- Minimum spacing s_0 in congested traffic
- Desired time headway T
- Comfortable braking deceleration b
- Exponent δ

[Treiber et al.]

Intelligent Driver Model

$$\dot{v}_\alpha = a\left(1 - \left(\frac{v_\alpha}{v_0}\right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2\right)$$

with $$s^*(v, \Delta v_\alpha) = s_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}}$$

- Maximum vehicle acceleration a
- Desired velocity v_0
- Minimum spacing s_0 in congested traffic
- Desired time headway T
- Comfortable braking deceleration b
- Exponent δ

[Treiber et al.]

Intelligent Driver Model

Free road behavior

$$\dot{v}_\alpha = a\left(\boxed{1-\left(\frac{v_\alpha}{v_0}\right)^\delta} - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2\right)$$

with $\quad s^*(v, \Delta v_\alpha) = s_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}}$

- Maximum vehicle acceleration a
- Desired velocity v_0
- Minimum spacing s_0 in congested traffic
- Desired time headway T
- Comfortable braking deceleration b
- Exponent δ

[Treiber et al.]

Intelligent Driver Model

Interaction term

$$\dot{v}_\alpha = a\left(1 - \left(\frac{v_\alpha}{v_0}\right)^\delta \boxed{- \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2}\right)$$

with $\quad s^*(v, \Delta v_\alpha) = s_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}}$

- Maximum vehicle acceleration a
- Desired velocity v_0
- Minimum spacing s_0 in congested traffic
- Desired time headway T
- Comfortable braking deceleration b
- Exponent δ

[Treiber et al.]

Intelligent Driver Model

- **Advantages:**
  - Simple but effective model
  - Parameters are intuitive
- **Disadvantages:**
  - Less realistic in some scnenarios
  - Does not work well for pedestrians

Intelligent Driver Model

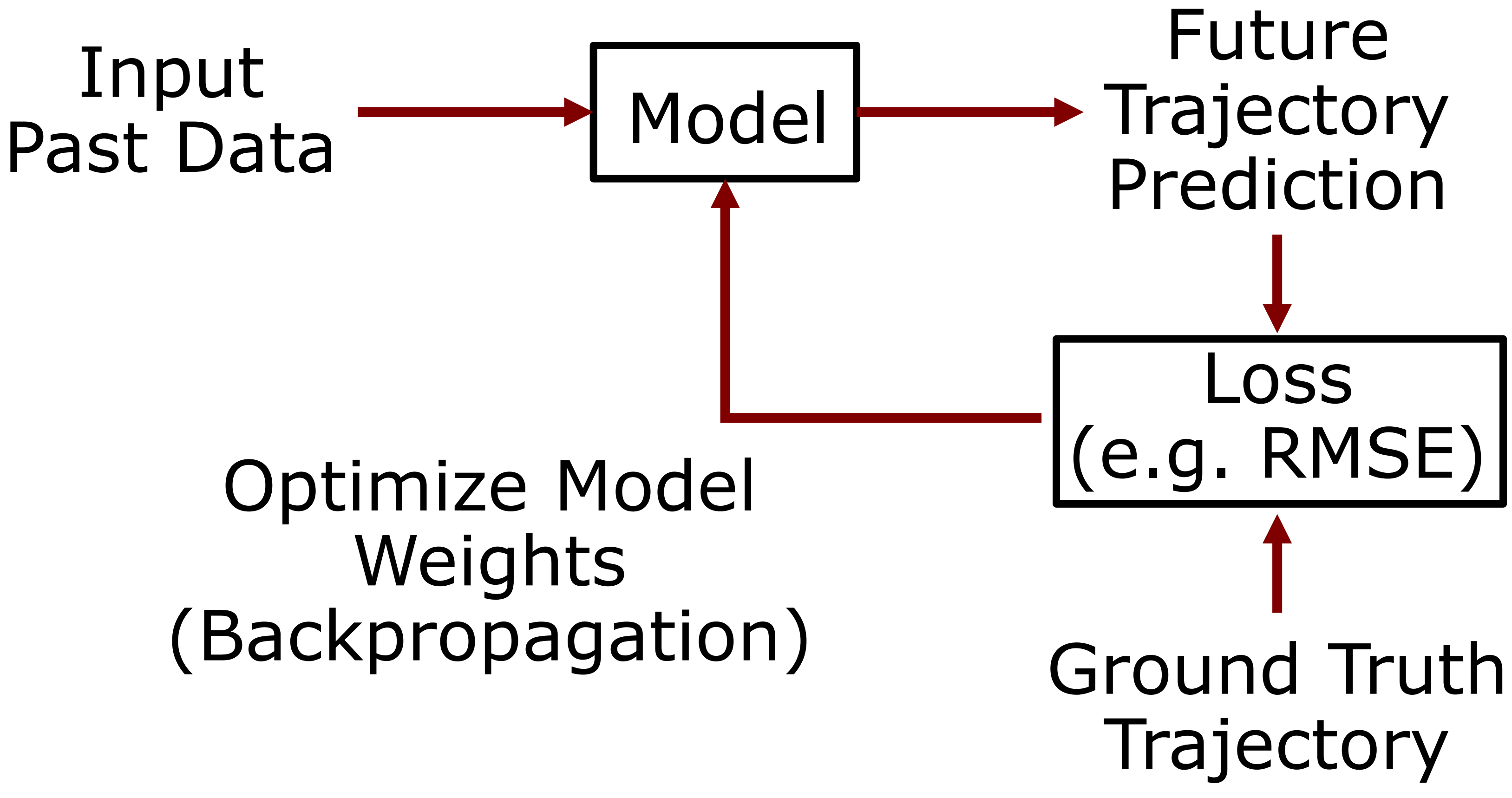
- **Advantages:**
  - Simple but effective model
  - Parameters are intuitive
- **Disadvantages:**
  - Less realistic in some scenarios
  - Does not work well for pedestrians

How can we model more realistic motion?

Deep Learning-based Prediction

- Learn to predict a future trajectory from large real-world datasets
- **Advantages:**
  - Implicit trait modeling
  - High model representation capacity
- **Disadvantages:**
  - Parameters are not interpretable
  - No explicit modeling of interactions
  - Less robust for unseen scenarios

Training Scheme



Deep Learning Paradigms

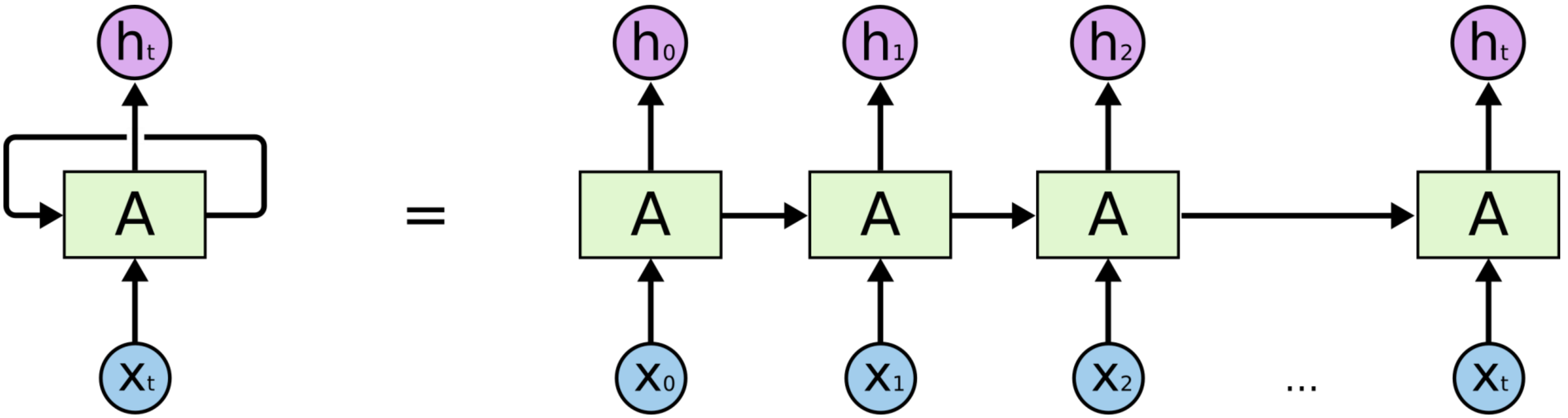
- Sequence-to-sequence prediction with
  - Recurrent Neural Networks (RNN)
  - Convolutional Neural Networks (CNN)
  - Combination of RNN and CNN
  - Graph Neural Networks
  - Transformers
  - Generative Adversarial Networks
- Deterministic vs stochastic models

Deep Learning Paradigms

- Sequence-to-sequence prediction with
  - Recurrent Neural Networks (RNN)
  - Convolutional Neural Networks (CNN)
  - Combination of RNN and CNN
  - Graph Neural Networks
  - Transformers
  - Generative Adversarial Networks
- Deterministic vs stochastic models

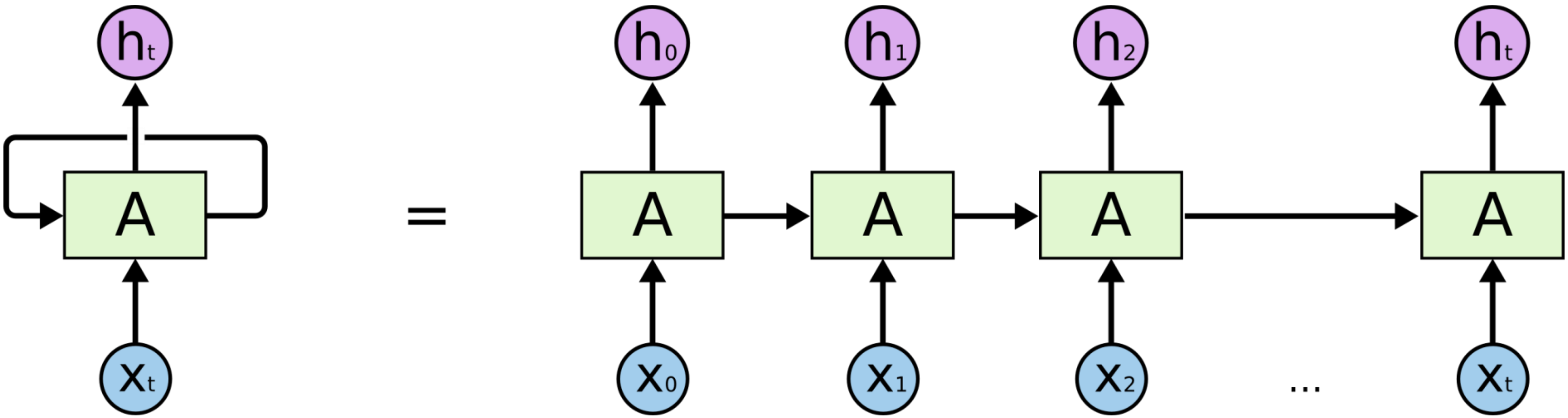
This lecture

Recurrent Neural Networks



[Courtesy of Christopher Olah]

Recurrent Neural Networks

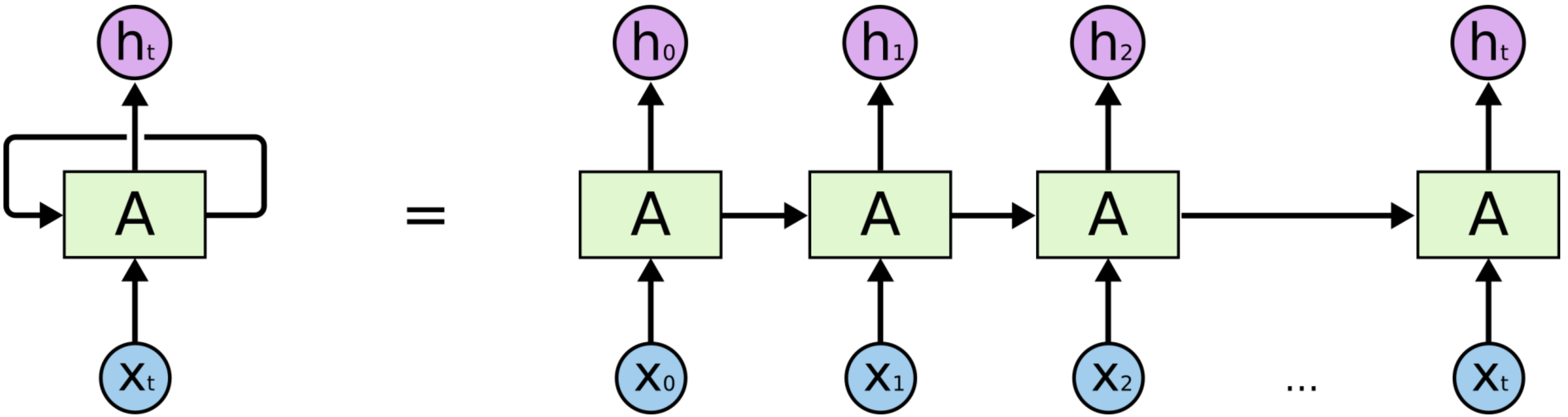


What happens in A?

$$h_t = \sigma\left(W\left[h_{t-1}, x_t\right] + b\right)$$

[Courtesy of Christopher Olah]

Recurrent Neural Networks



What happens in A?

Learnable weights

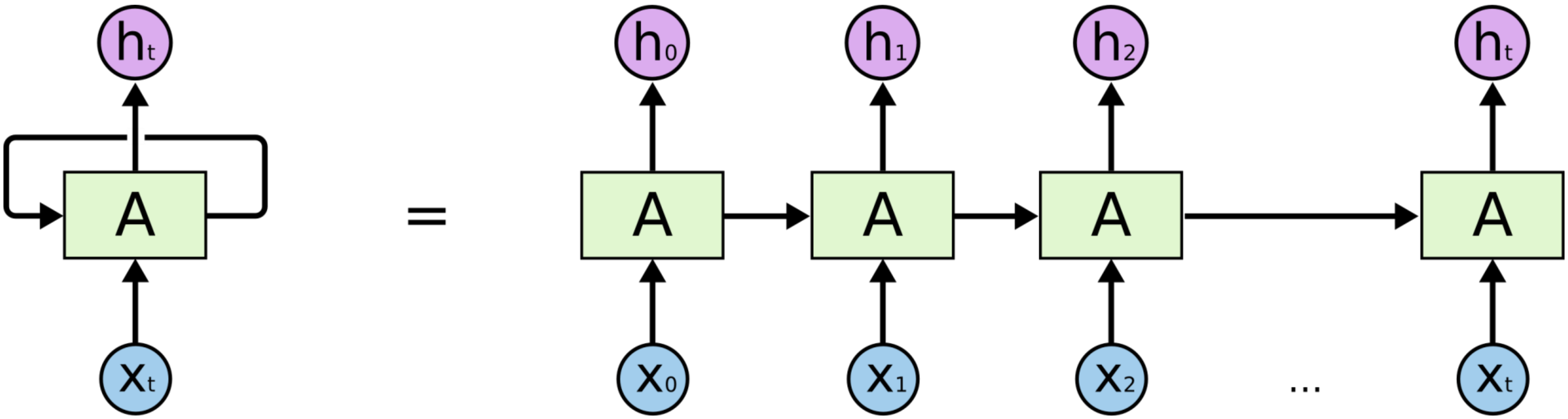
Learnable bias

$$h_t = \sigma \left(W \left[h_{t-1}, x_t \right] + b \right)$$

Activation function

[Courtesy of Christopher Olah]

Recurrent Neural Networks

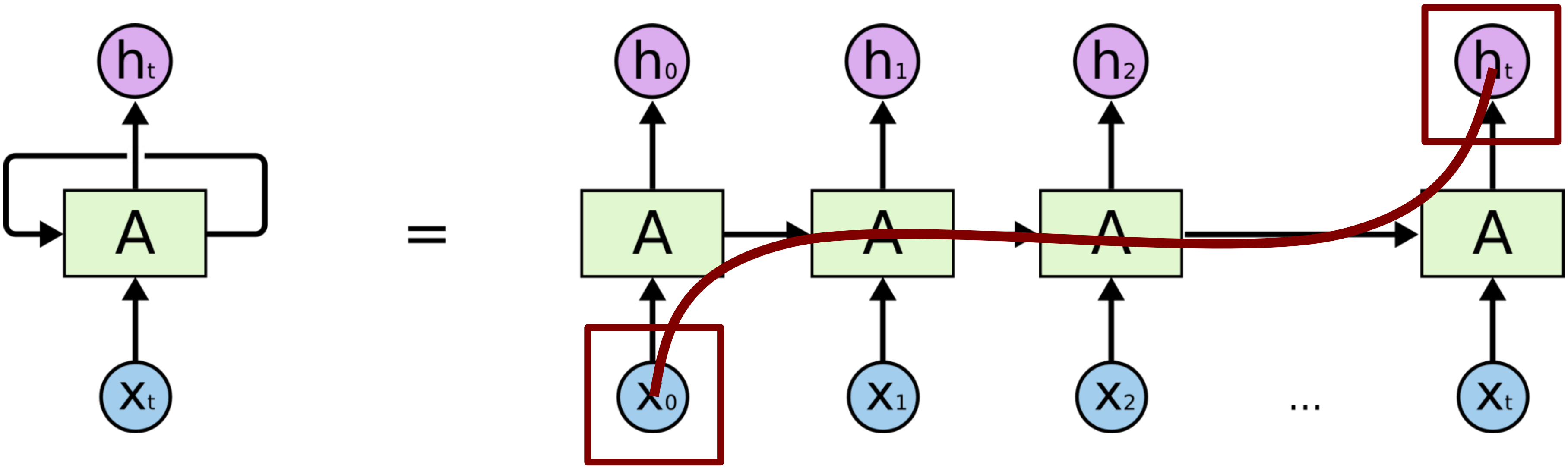


Advantages: Weight sharing, variable sequence length

Disadvantages: Slow prediction, vanishing/exploding gradients

[Courtesy of Christopher Olah]

Recurrent Neural Networks



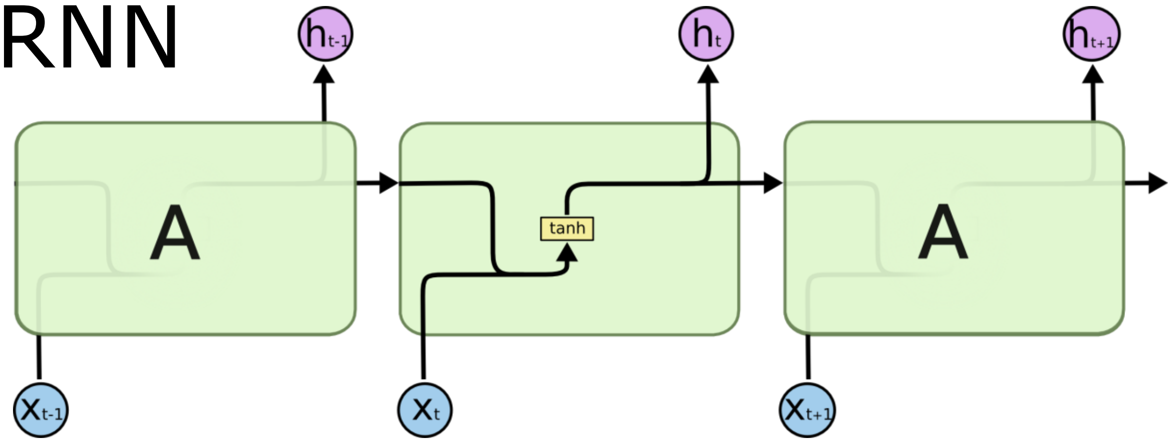
Advantages: Weight sharing, variable sequence length

Disadvantages: Slow prediction, vanishing/exploding gradients

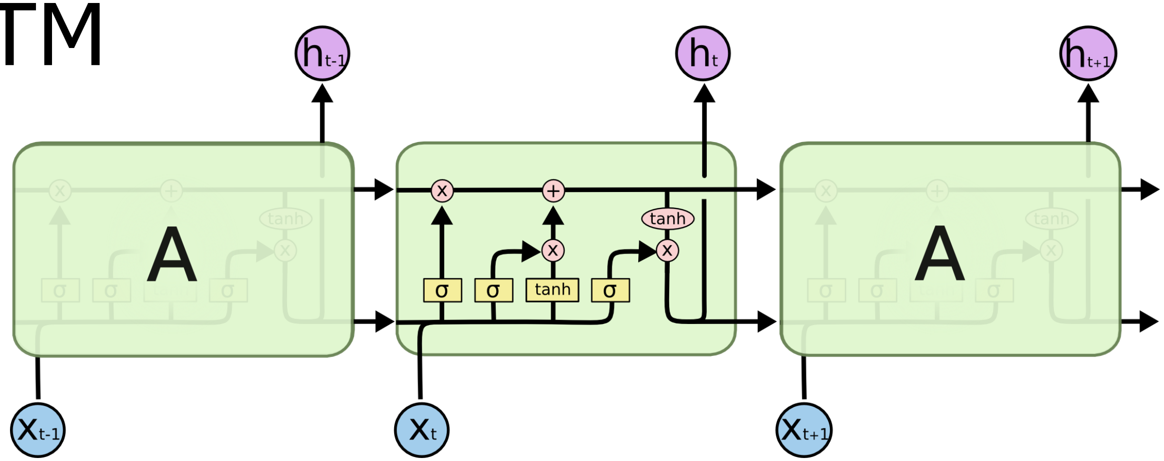
[Courtesy of Christopher Olah]

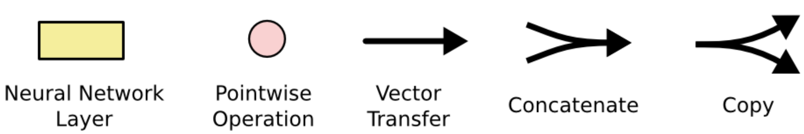
Long Short-Term Memory

From RNN



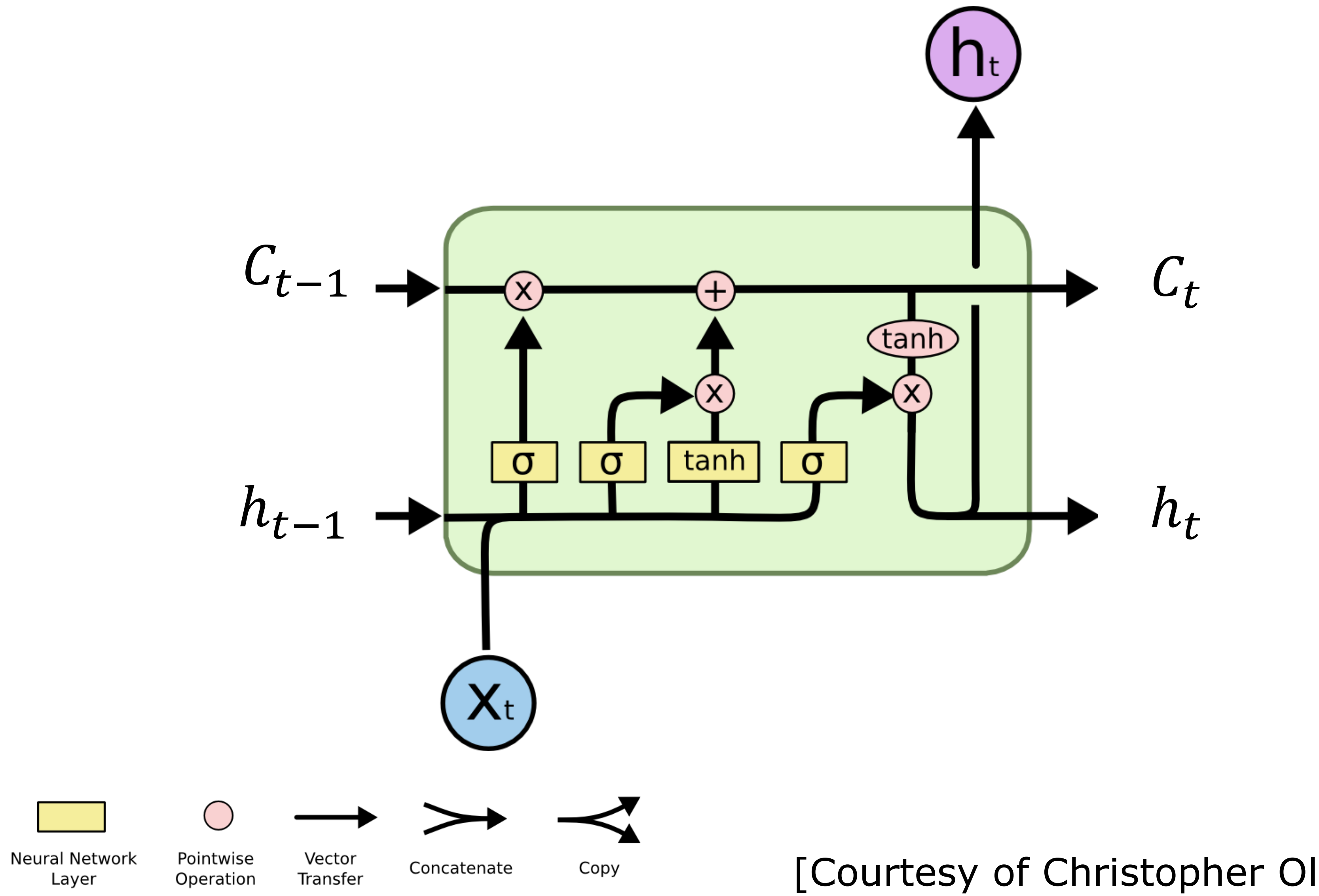
To LSTM



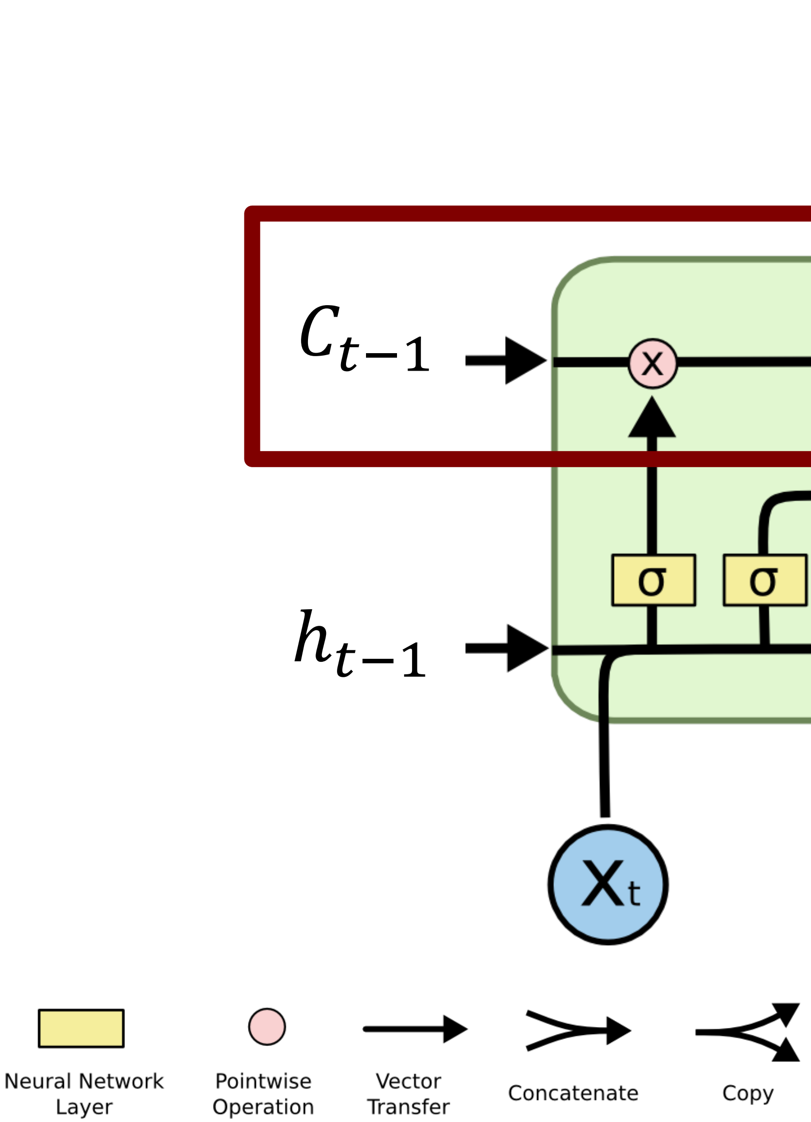


[Hochreiter and Schmidhuber]
[Courtesy of Christopher Olah]

Long Short-Term Memory

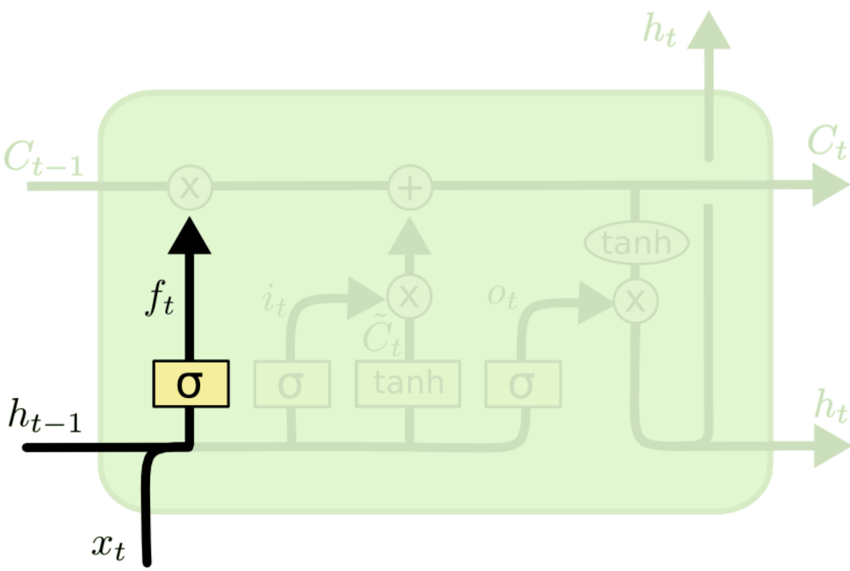


Long Short-Term Memory



[Courtesy of Christopher Olah]

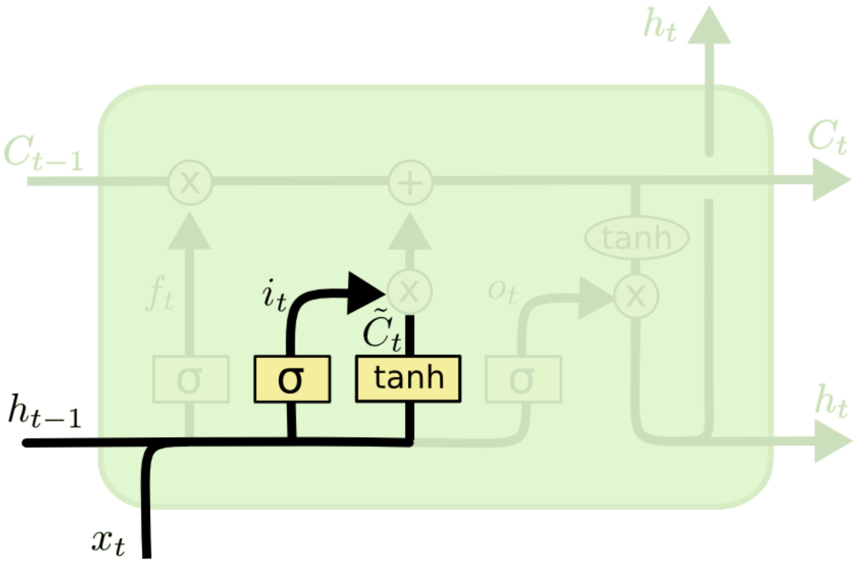
LSTM Forget Gate



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \;+\; b_f\right)$$

[Courtesy of Christopher Olah]

LSTM Input Gate

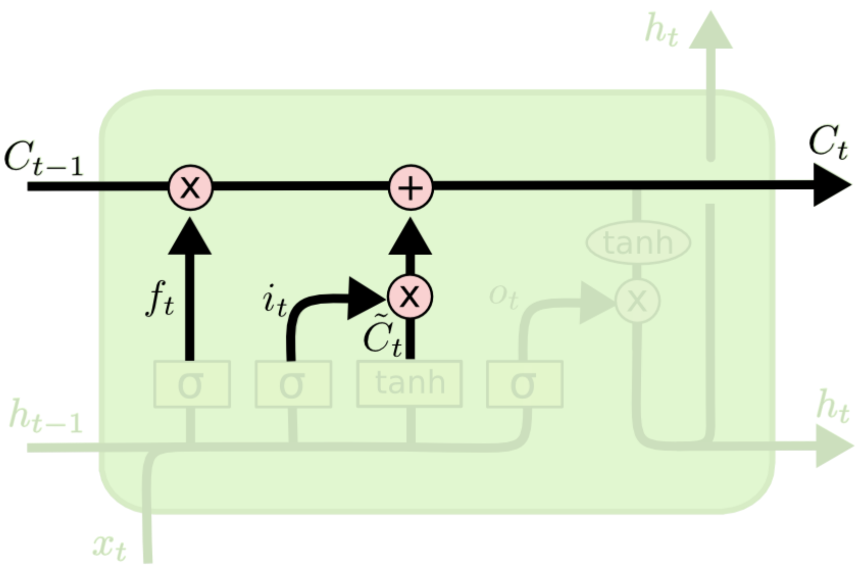


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

[Courtesy of Christopher Olah]

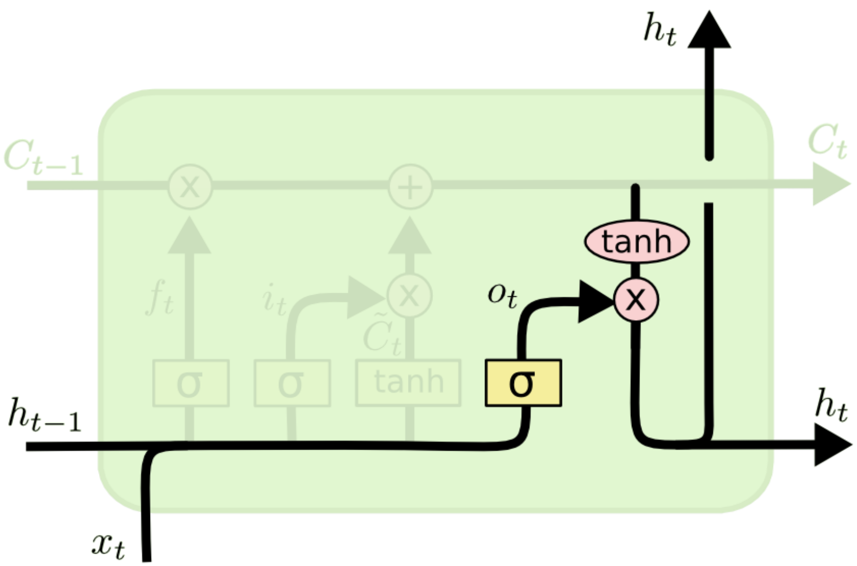
LSTM Cell Update



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

[Courtesy of Christopher Olah]

LSTM Output Gate

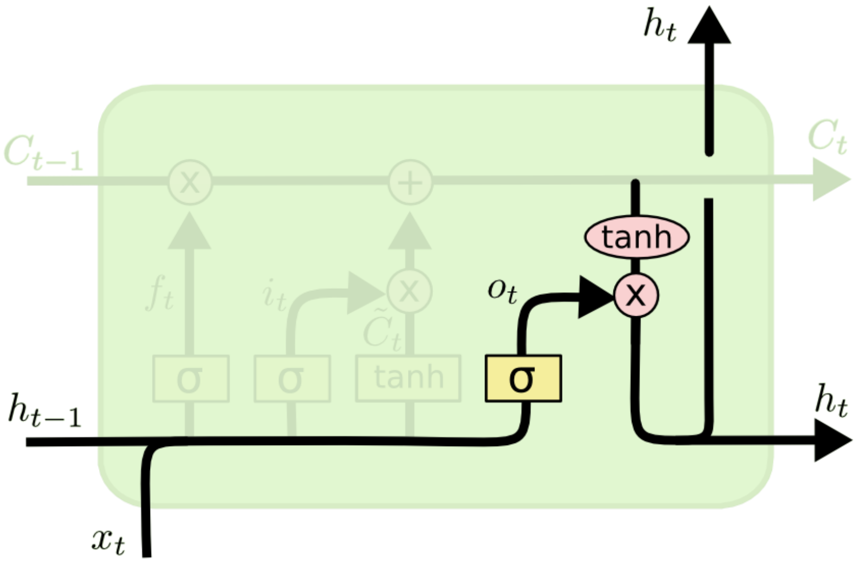


$$o_t = \sigma\left(W_o\ [\,h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

[Courtesy of Christopher Olah]

LSTM Output Gate



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

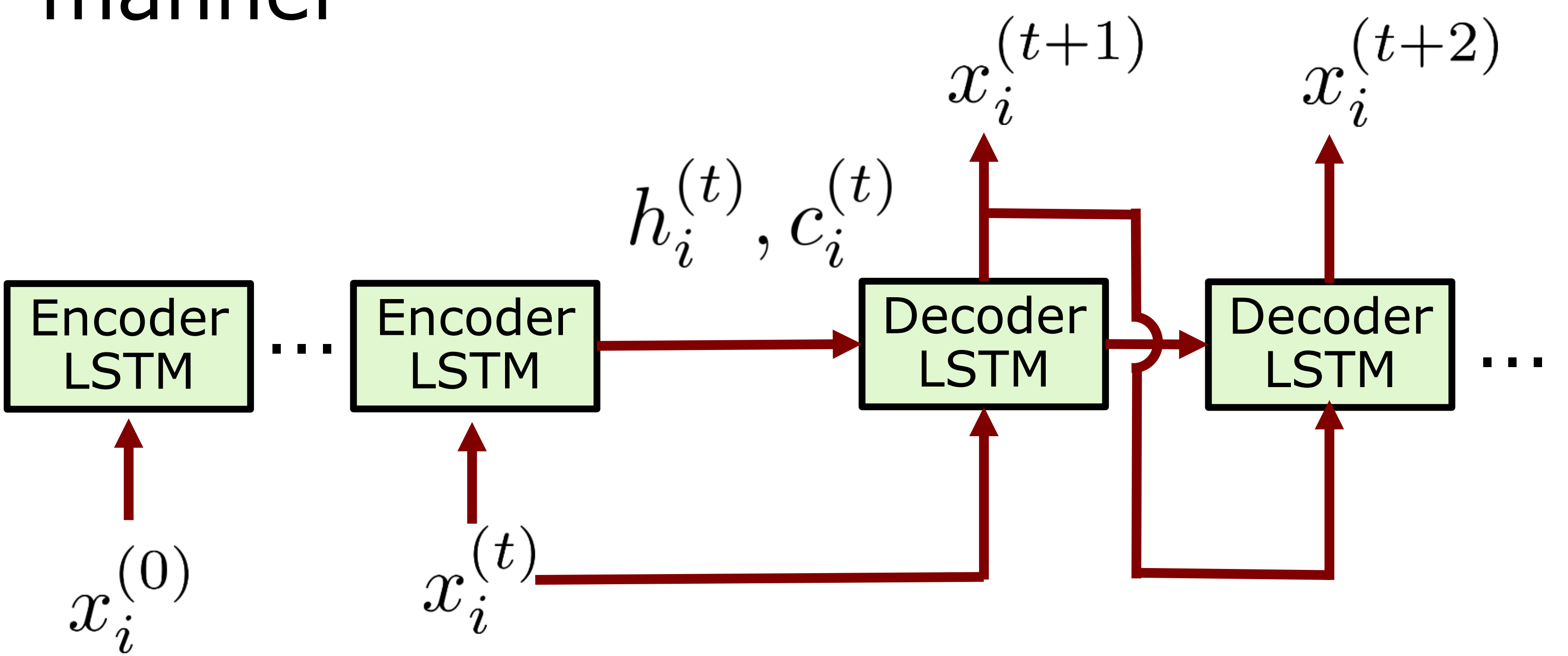
$$h_t = o_t * \tanh\left(C_t\right)$$

How Can We Use LSTMs for Sequence Prediction?

[Courtesy of Christopher Olah]

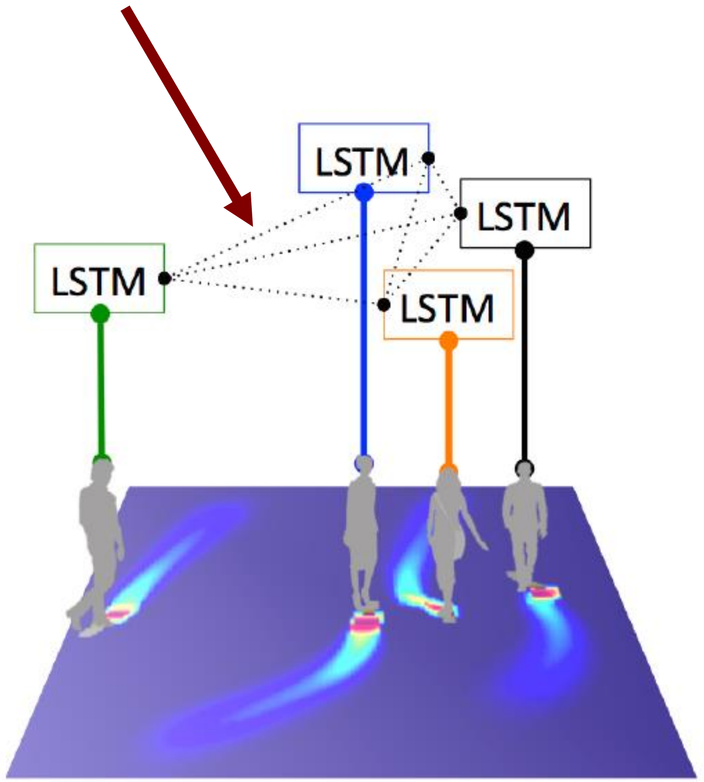
Sequence Prediction

- Encode past states of agent i
- Decode future states in autoregressive manner



Example: SocialLSTM

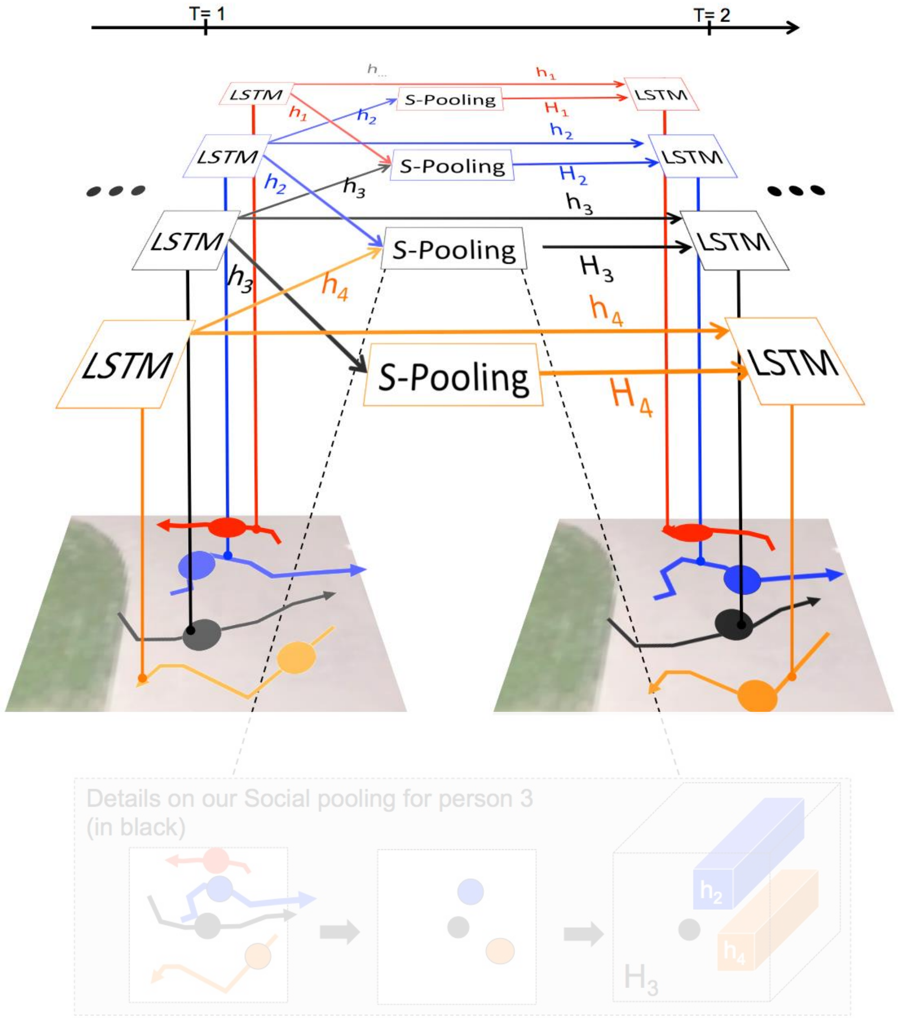
- What about the influence of neighbors?
- How to consider interactions?



[Alahi et al.]

Example: SocialLSTM

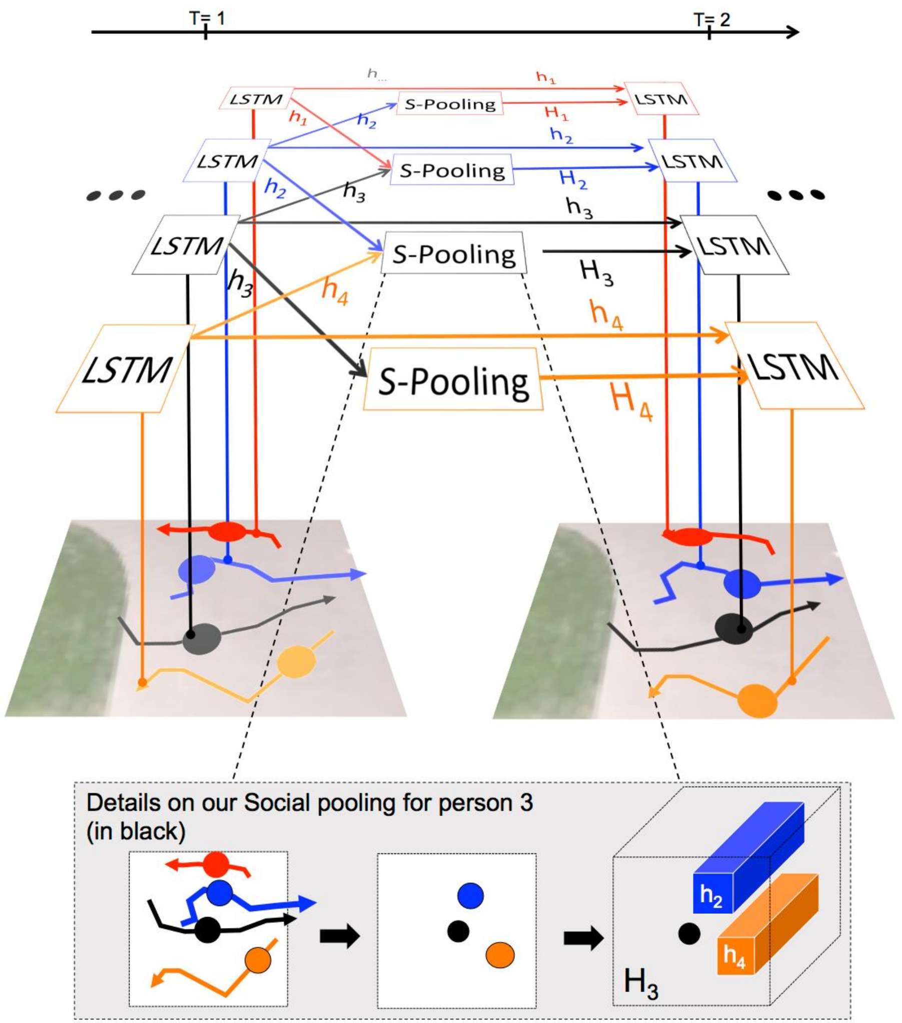
- Assumption: Hidden state contains motion information
- Hidden states of neighbors are shared



[Alahi et al.]

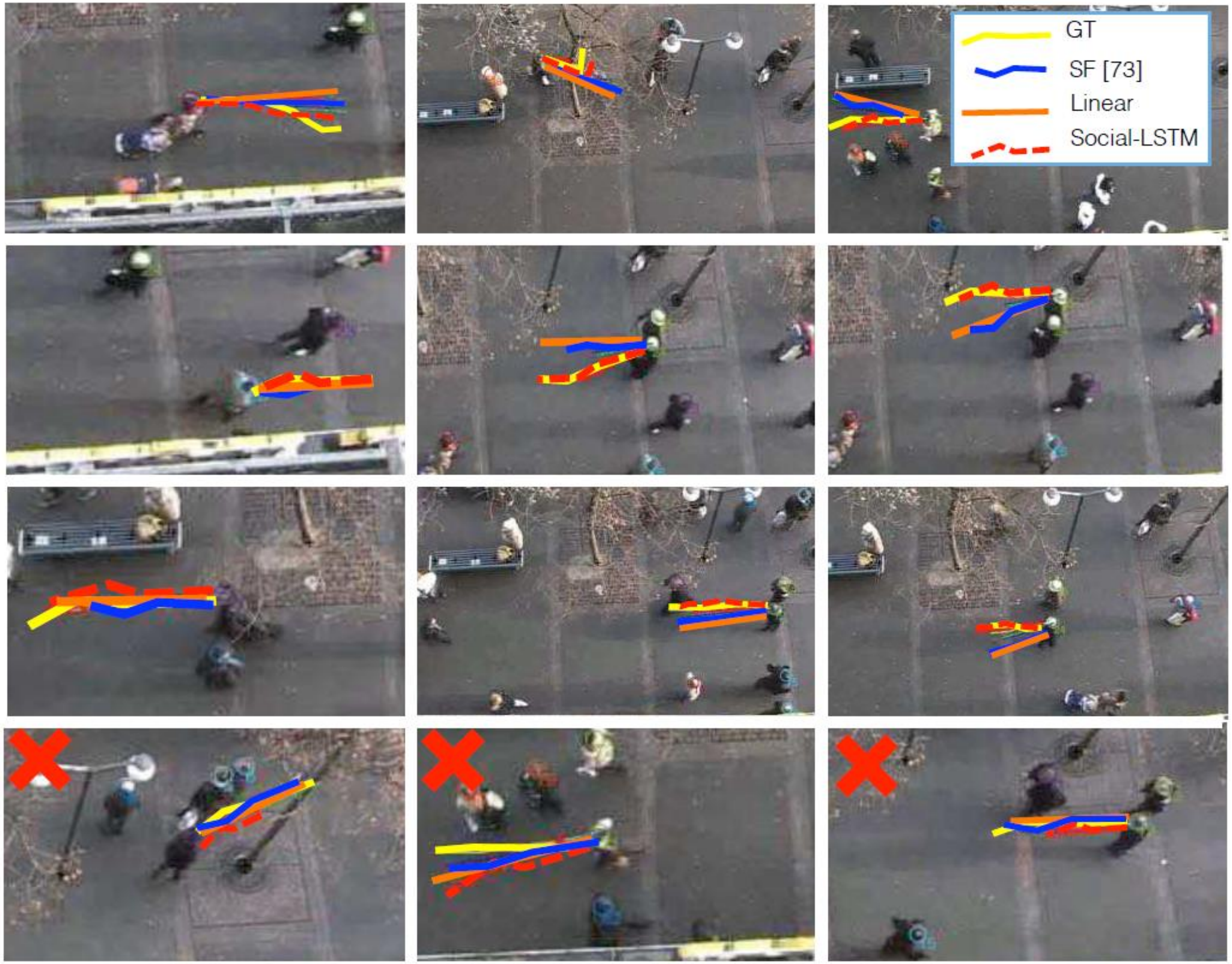
Example: SocialLSTM

- Assumption: Hidden state contains motion information
- Hidden states of neighbors are shared
- Pool hidden states into fixed-sized tensor



[Alahi et al.]

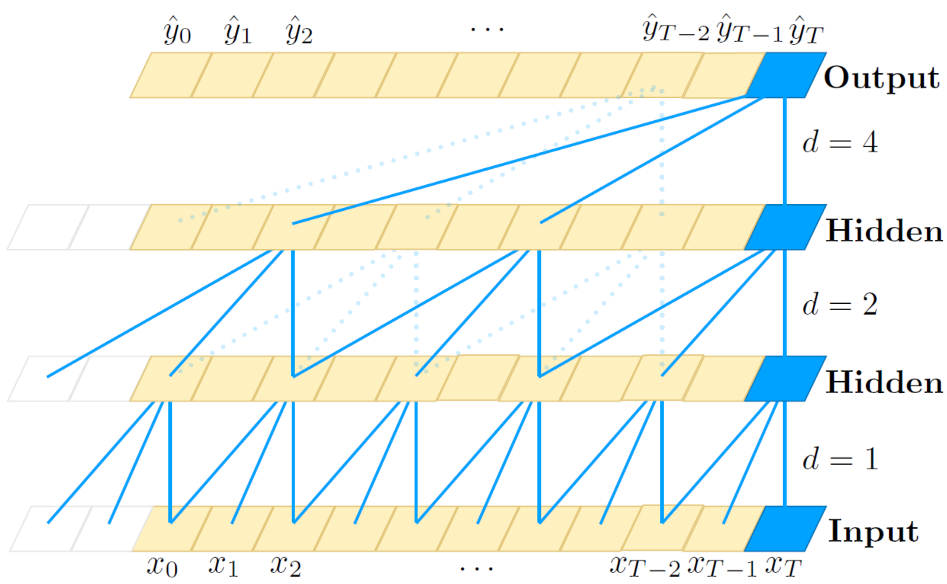
SocialLSTM Comparison



[Alahi et al.]

Temporal Convolutions

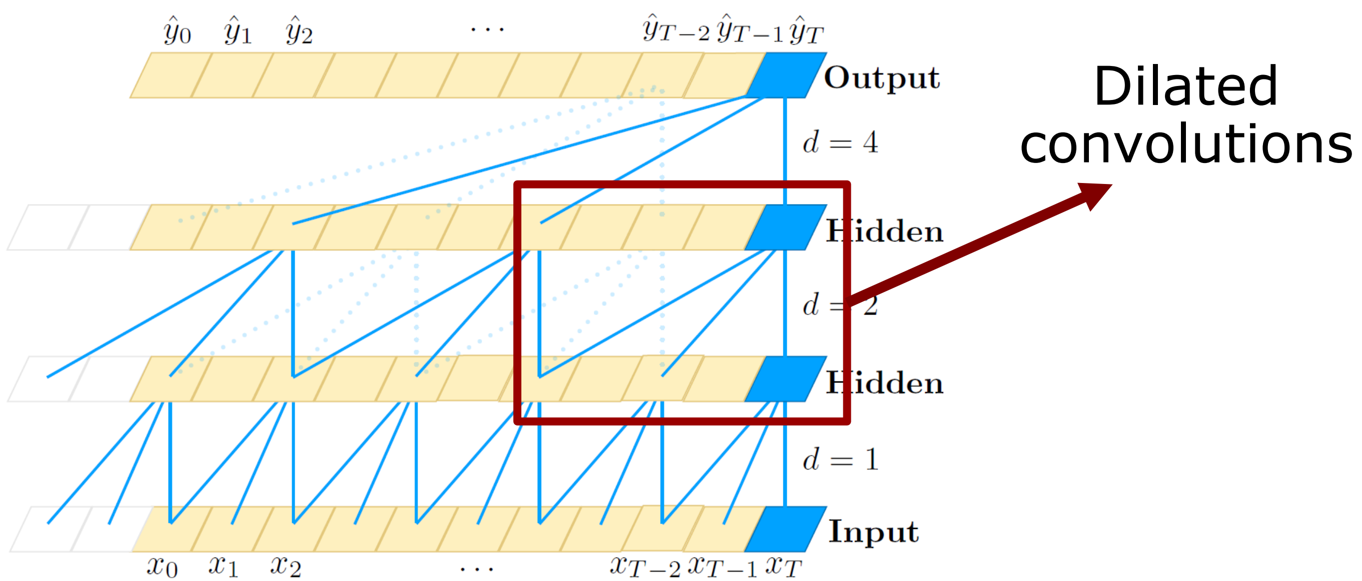
- Apply convolutions across time dimensions
- Easier to train, but needs sufficient receptive field



[Bai et al.]

Temporal Convolutions

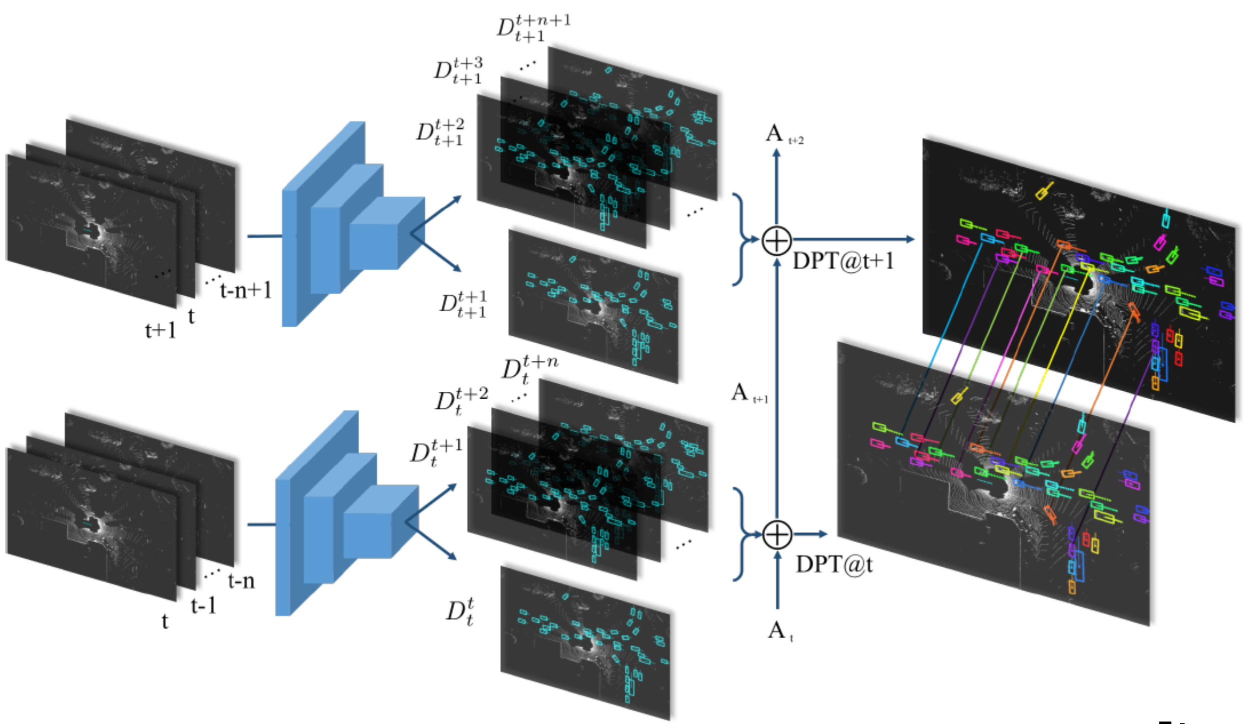
- Apply convolutions across time dimensions
- Easier to train, but needs sufficient receptive field



[Bai et al.]

Example: Fast and Furious

- Voxelize LiDAR into bird‘s eye view
- 3D CNN for end-to-end detection, tracking and motion prediction

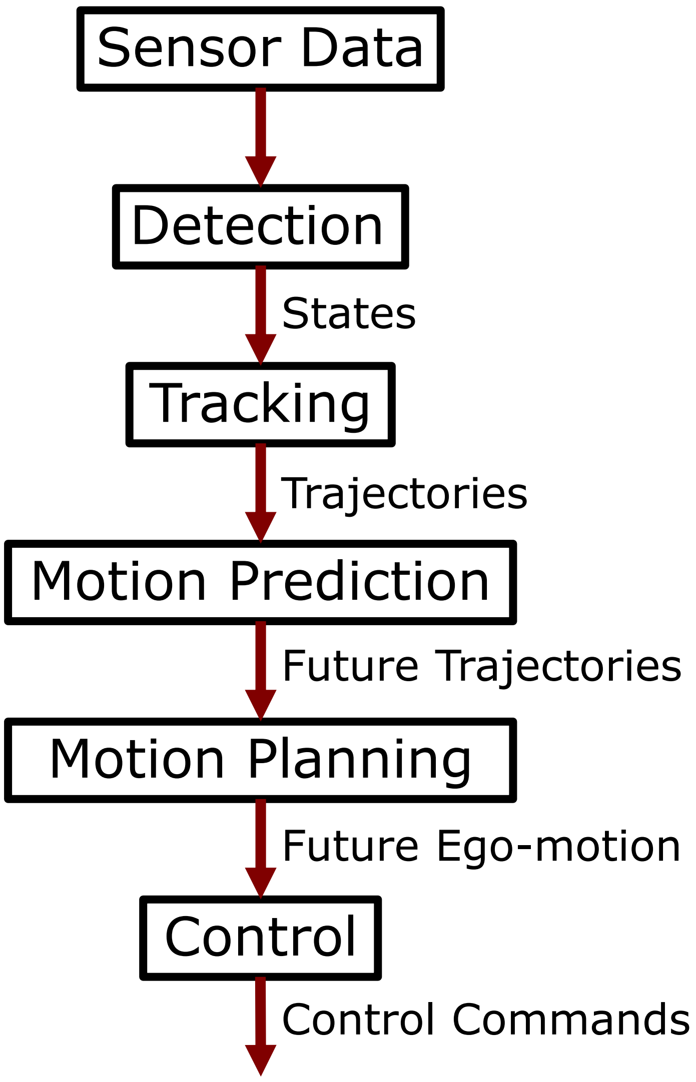


[Luo et al.]

What If Our Perception System Fails?

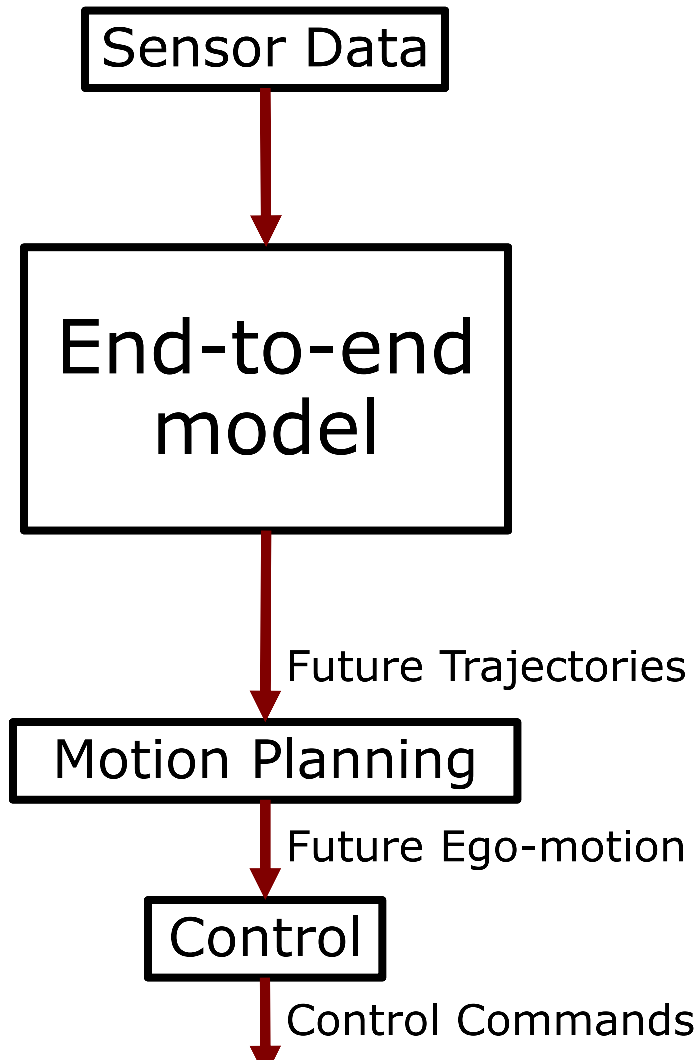
End-to-end Learning

- Most prediction approaches require already tracked past trajectories
- Misperception is not considered
- Idea: End-to-end learning



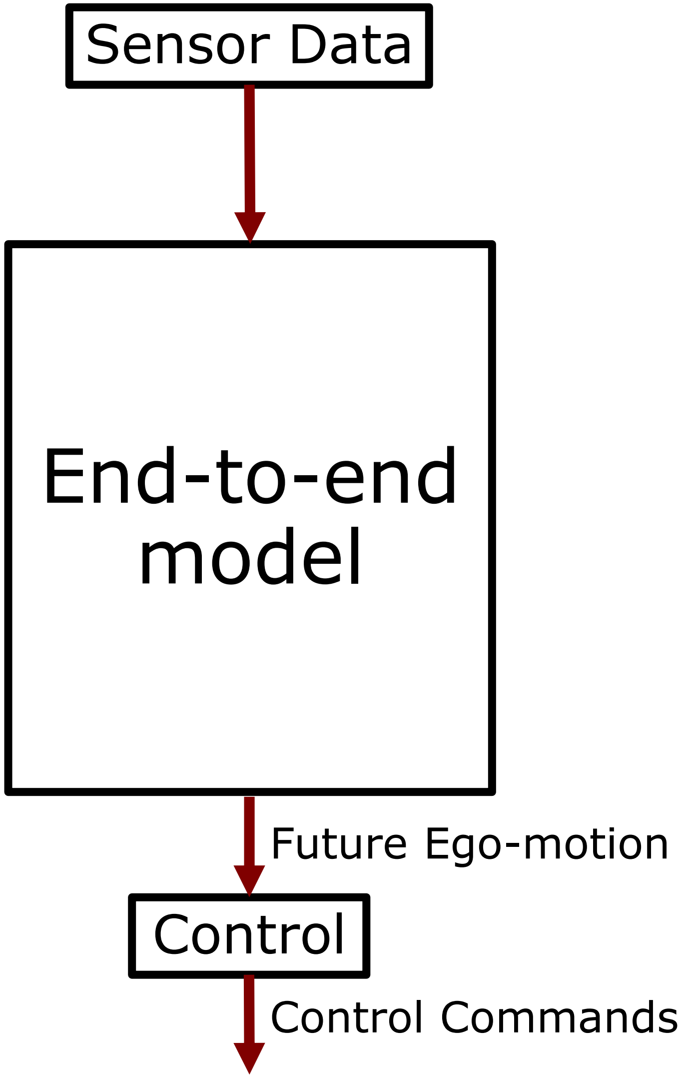
End-to-end Learning

- Most prediction approaches require already tracked past trajectories
- Misperception is not considered
- Idea: End-to-end learning



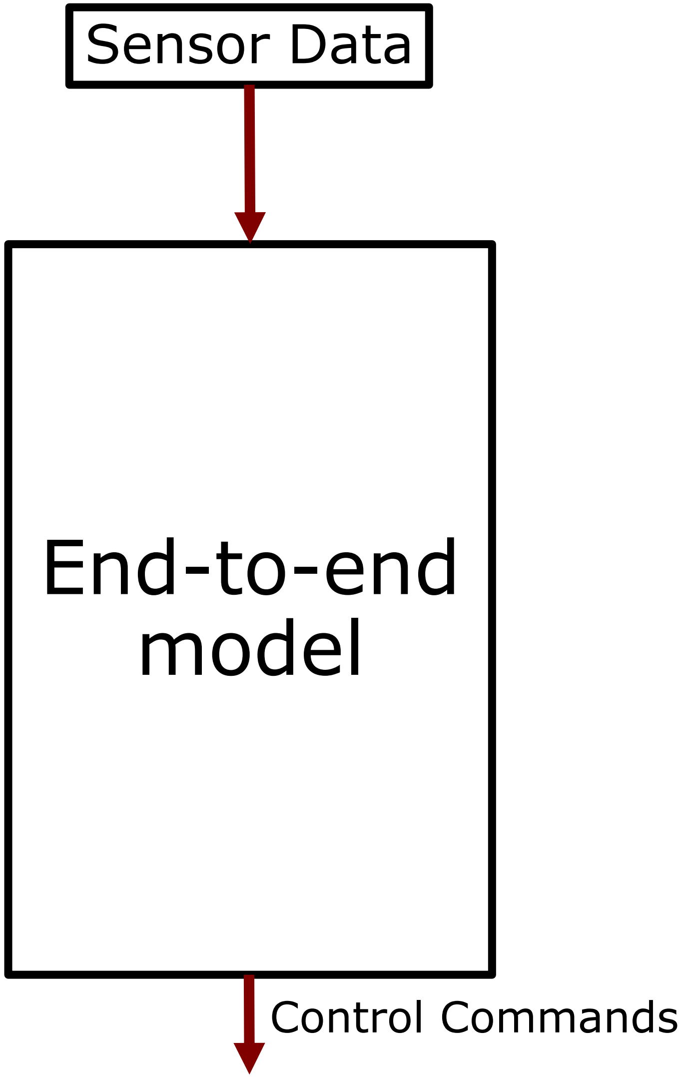
End-to-end Learning

- Most prediction approaches require already tracked past trajectories
- Misperception is not considered
- Idea: End-to-end learning



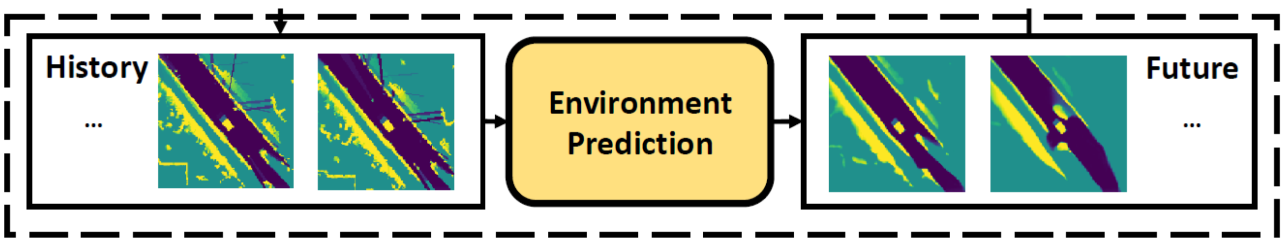
End-to-end Learning

- Most prediction approaches require already tracked past trajectories
- Misperception is not considered
- Idea: End-to-end learning

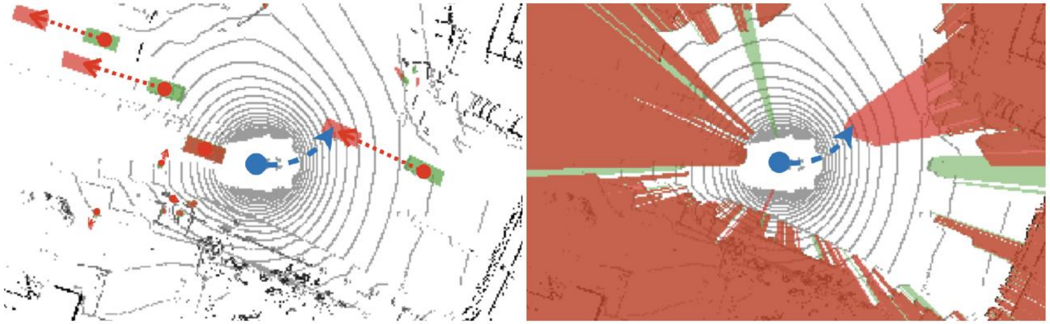


Self-supervised Prediction

- Labeling trajectories or bounding boxes is expensive
- Idea: Predict raw sensor data into the future



[Lange et al.]



[Hu et al.]

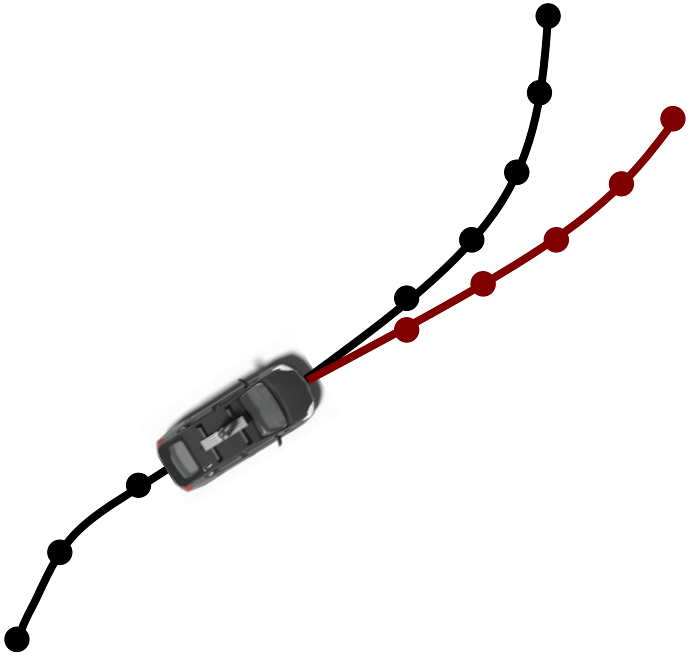
How Do We Evaluate Our Prediction?

Loss & Evaluation Metrics

- Final Displacement Error
- Average Displacement Error

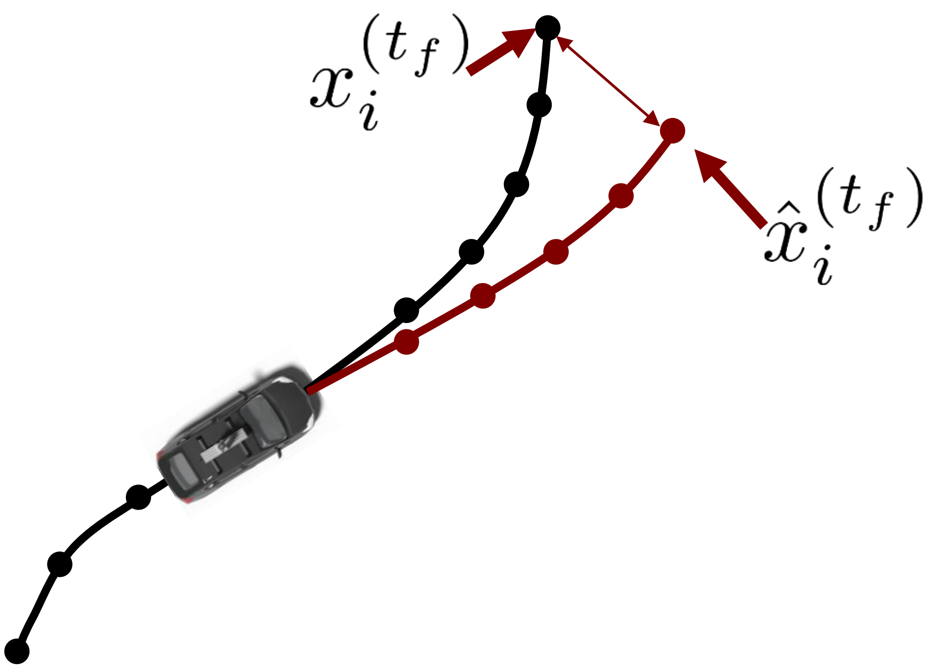
Loss & Evaluation Metrics

- Final Displacement Error
- Average Displacement Error



Loss & Evaluation Metrics

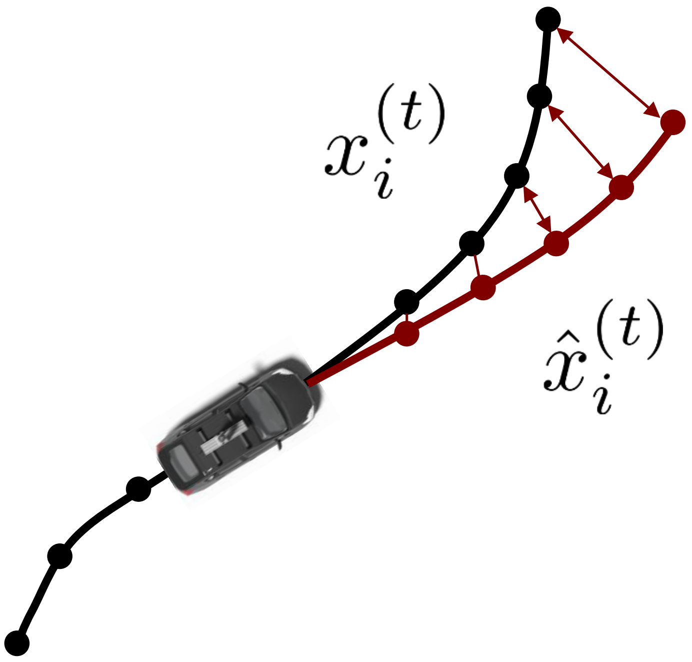
- Final Displacement Error
- Average Displacement Error



$$\mathrm{FDE} = \left\| \hat{x}_i^{(t_f)} - x_i^{(t_f)} \right\|_2$$

Loss & Evaluation Metrics

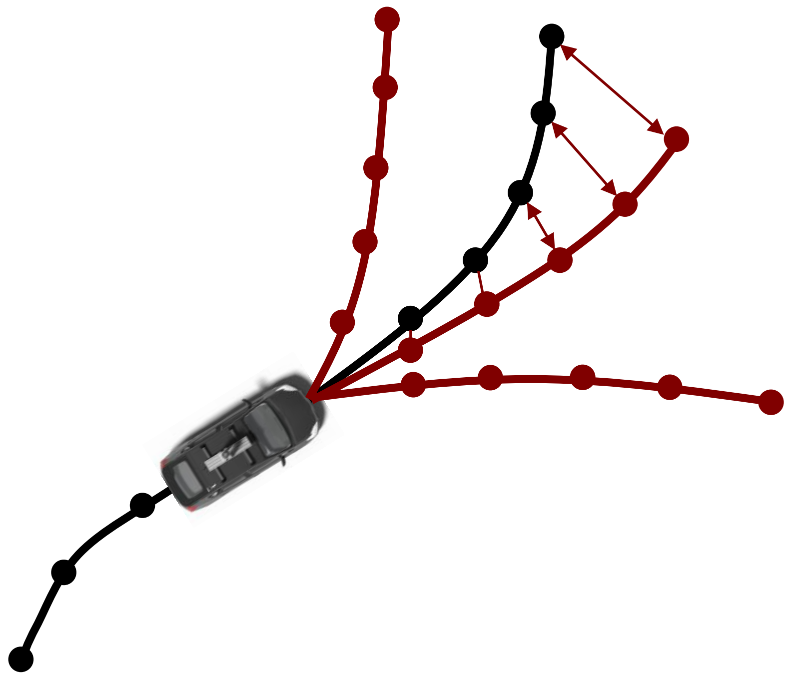
- Final Displacement Error
- Average Displacement Error



$$\mathrm{ADE} = \frac{1}{P}\sum_{t=1}^{P}\left\|\hat{x}_i^{(t)} - x_i^{(t)}\right\|_2$$

Loss & Evaluation Metrics

- Final Displacement Error
- Average Displacement Error



- Unimodal vs multimodal

Datasets and Benchmarks

- KITTI
- SemanticKITTI
- Lyft Level 5
- Waymo Challenge
- nuScenes
- Argoverse
- highD, inD, rounD
- Pedestrian prediction: ETH, UCY
- ...

Summary

- Estimate intention, traits of future trajectory for planning own behavior
- Different solution strategies depending on e.g. model complexity or level of interaction
- Can learn behavior estimation from large real-world datasets with deep learning

Thank you for your attention

References

- B. Paden et al. "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles." IEEE Transactions on Intelligent Vehicles, 2016.
- K. Brown et al. "Modeling and Prediction of Human Driver Behavior: A Survey". arXiv preprint, abs/2006.08832, 2020.
- F. Poggenhans et al. "Lanelet2: A high-definition map framework for the future of automated driving." 2018 21st International Conference on Intelligent Transportation Systems (ITSC) (2018): 1672-1679.
- H. Li, R. Dai. "The merging control method of on-ramp vehicles based on cooperative vehicle infrastructure system for highway entering ramp." IOP Conference Series: Materials Science and Engineering, 2019.
- D. Helbing, P. Molnar. "Social force model for pedestrian dynamics", 1995.
- M. Treiber et al. "Congested Traffic States in Empirical Observations and Microscopic Simulations." Physical Review E. 62. 1805-1824, 2000.
- C. Olah "Understanding LSTM Networks". Retrieved from http://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.
- S. Hochreiter, J. Schmidhuber. "Long Short-term Memory". Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735., 1997.
- A. Alahi et al. "Social LSTM: Human Trajectory Prediction in Crowded Spaces." In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016.
- S. Bai et al. "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.", arXiv preprint, abs/1803.01271, 2018.
- W. Luo et al. "Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net." In Proc. of the IEEE Conf. on Computer Vision and 318 Pattern Recognition (CVPR), 2018.
- P. Hu et al. „Safe Local Motion Planning with Self-Supervised Freespace Forecasting.", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.
- B. Lange et al. „Attention Augmented ConvLSTM for Environment Prediction .", In Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2021.