

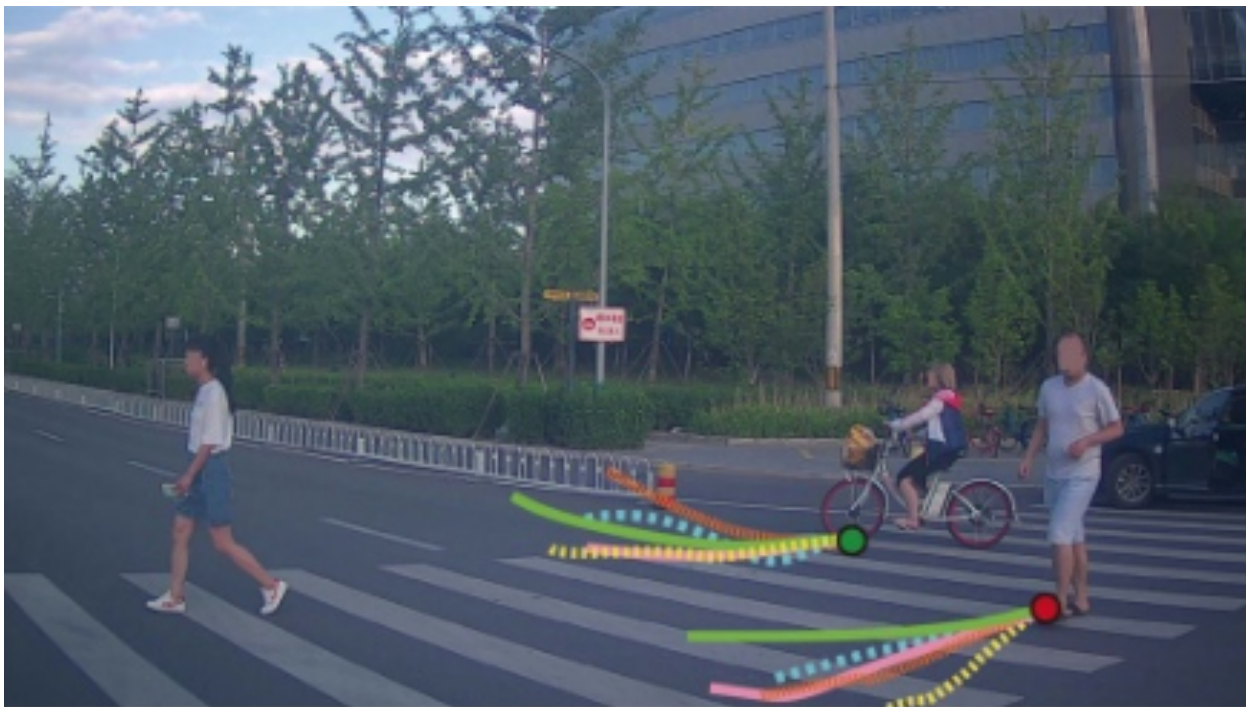
# Pedestrian Trajectory Prediction

# Overview

- What is pedestrian trajectory prediction good for?
- What is the task of pedestrian trajectory prediction?
- What makes trajectory prediction challenging?
- What kind models are used for trajectory prediction?
  - Deterministic models
  - Stochastic Models
- How to model social interactions?
- How to model agent-scene interactions?

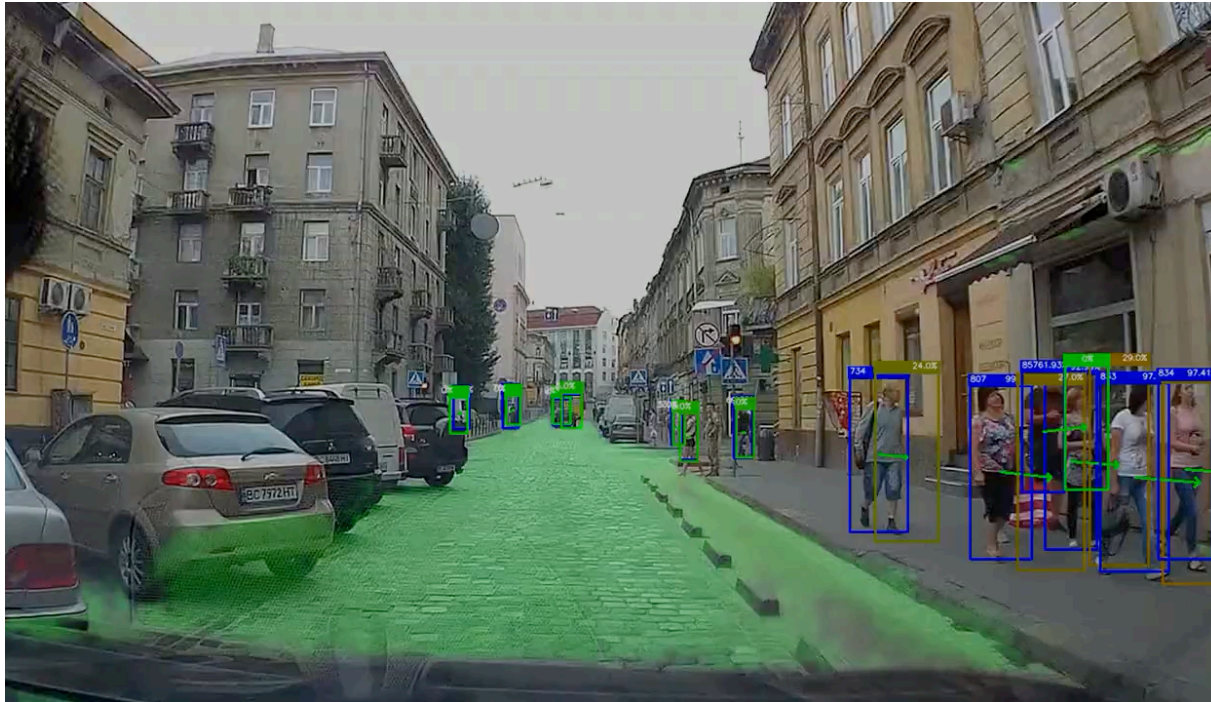
# Introduction

- Autonomous vehicles must predict the future movements of pedestrians in order to avoid fatal collisions



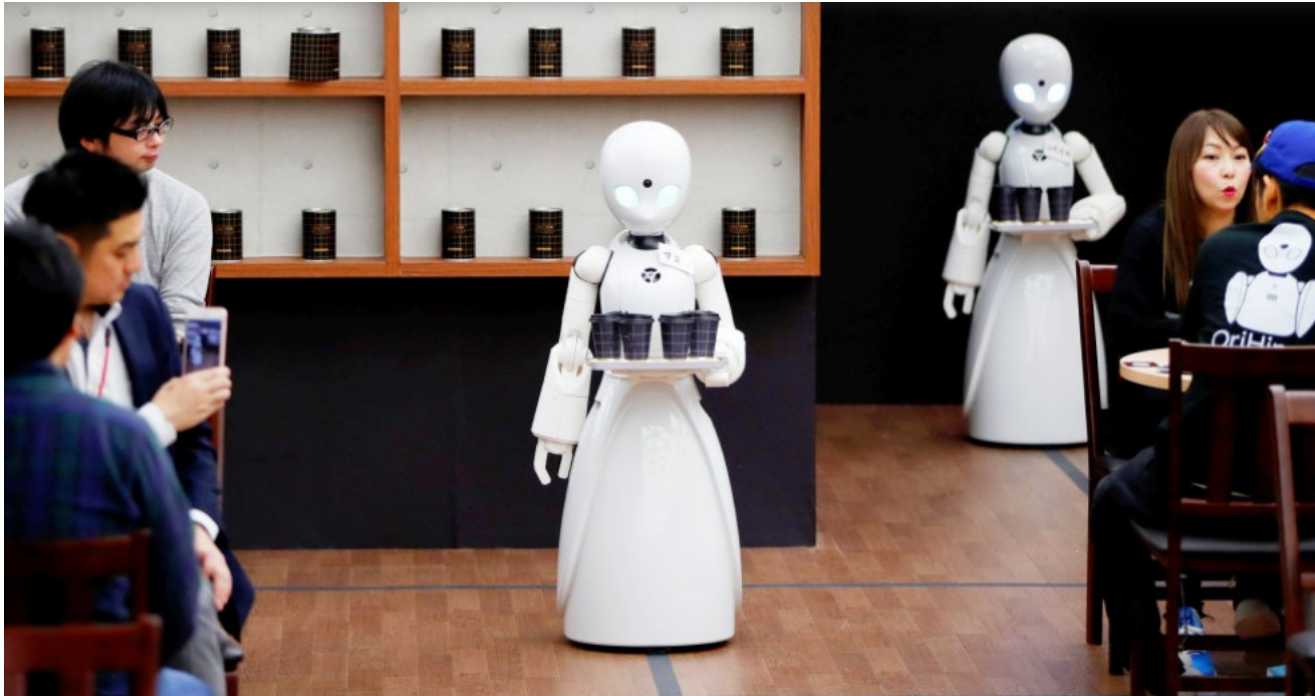
# Introduction

- Autonomous vehicles must predict the future movements of pedestrians in order to avoid fatal collisions



# Introduction

- Social robots that move autonomously through crowded scenes and interact with moving humans



# Introduction

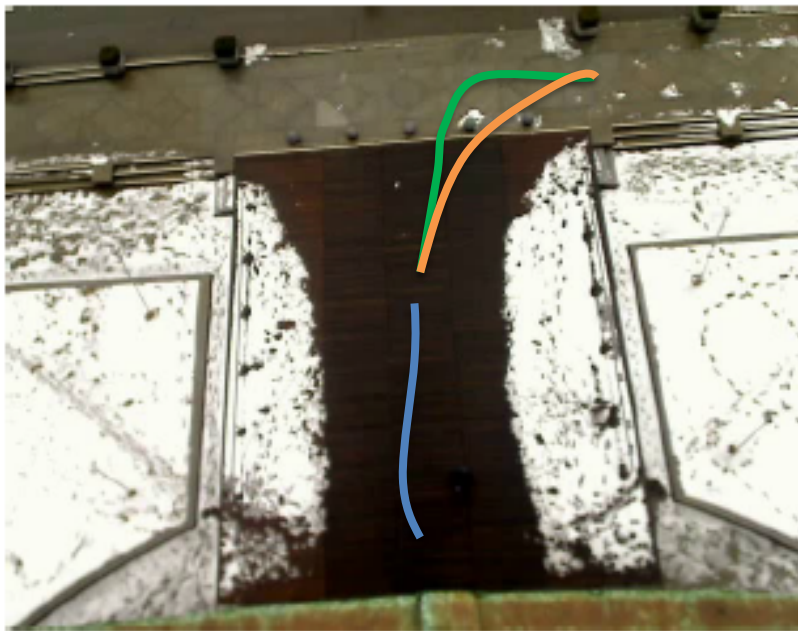
- Motion models improve the performance of Multi-Object Trackers



[www.motchallenge.net](http://www.motchallenge.net)

# Task of pedestrian trajectory prediction

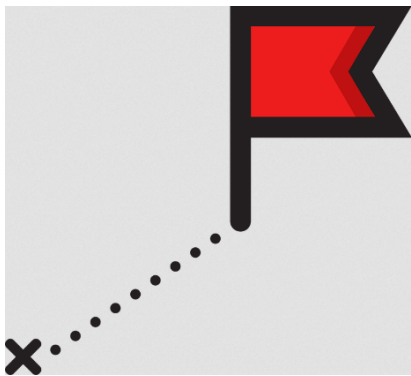
Scene



- Sequence of observations:  
 $X_i^t = (x_i^t, y_i^t)$  for  $t = 1, \dots, T^{Obs}$
- Ground Truth:  
 $Y_i^t = (x_i^t, y_i^t)$   
for  $t = T^{Obs+1}, \dots, T^{Pred}$
- Prediction:  
 $\hat{Y}_i^t = f_W(X_i) = (x_i^t, y_i^t)$   
for  $t = T^{Obs+1}, \dots, T^{Pred}$

# Introduction

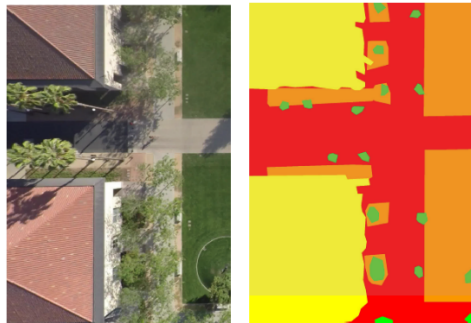
- Human motion behavior is influenced by a variety of different factors:



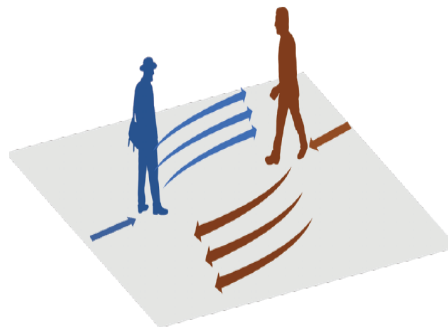
1. Destination



2. Personal Preferences



3. Human – Space Interactions



4. Human – Human Interactions

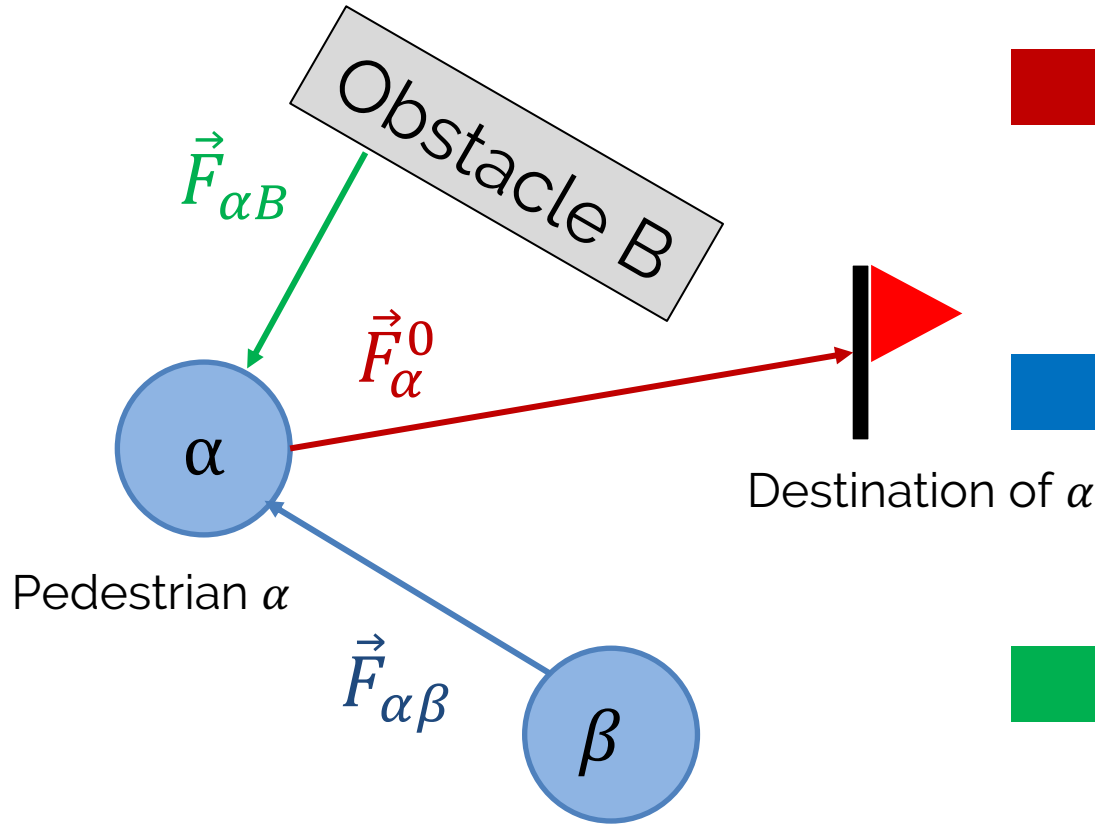


# Social Force Model

Contribution: Mathematical model of dynamics

- **Idea:** Pedestrian act in force field  $\mathbf{F}$  like particles e.g. in an electric field
- Second Newton's law:  $\ddot{\mathbf{x}} = \frac{d\dot{\mathbf{x}}}{dt} = \frac{\mathbf{F}(t)}{m}$
- Trajectory  $\mathbf{x}(t)$  is solution of differential equation

# Social Force Model



## Destination and personal

desired velocity      direction of destination

$$\vec{F}_{\alpha}^0 = \vec{F}_{\alpha}^0(\vec{v}_{\alpha}(t), v_{\alpha}^0 \vec{e}_{\alpha})$$

## Human - Human interactions

distance betw. peds

$$\sum_{\beta} \vec{F}_{\alpha\beta}(\vec{e}_{\alpha}, \vec{r}_{\alpha\beta})$$

## Human - Space interactions

$$\sum_B \vec{F}_{\alpha B}(\vec{e}_{\alpha}, \vec{r}_B^{\alpha})$$

# Social Force Model

- Force resultant determines motion of pedestrian  $\alpha$

$$\vec{F}_\alpha(t) := \underbrace{\vec{F}_\alpha^0(\vec{v}_\alpha, v_\alpha^0 \vec{e}_\alpha)}_{\text{acceleration term towards dest.}} + \underbrace{\sum_\beta \vec{F}_{\alpha\beta}(\vec{e}_\alpha, \vec{r}_{\alpha\beta})}_{\text{force between pedestrians}} + \underbrace{\sum_B \vec{F}_{\alpha B}(\vec{e}_\alpha, \vec{r}_B^\alpha)}_{\text{force between ped. and obstacles}}$$

- The respective trajectory  $\mathbf{x}(t)$  is the solution of a differential equation:

$$\ddot{\mathbf{x}}(t) = \frac{d\mathbf{x}^2}{dt^2} = F_\alpha(t)$$

# Social Force Model

- The force between pedestrians is described by the gradient of a repulsive potential  $V_{\alpha\beta}$ :

$$\vec{F}_{\alpha\beta}(\vec{r}_{\alpha\beta}) = -\nabla_{\vec{r}_{\alpha\beta}} V_{\alpha\beta}(\vec{r}_{\alpha\beta}) \text{ with } V_{\alpha\beta}(\vec{r}_{\alpha\beta}) = V^0 e^{-\frac{\|\vec{r}_{\alpha\beta}\|}{\sigma}}$$

- Parameters  $V^0$  and  $\sigma$  determine the shape of this potential
- Parameters effect strength of interaction between pedestrians

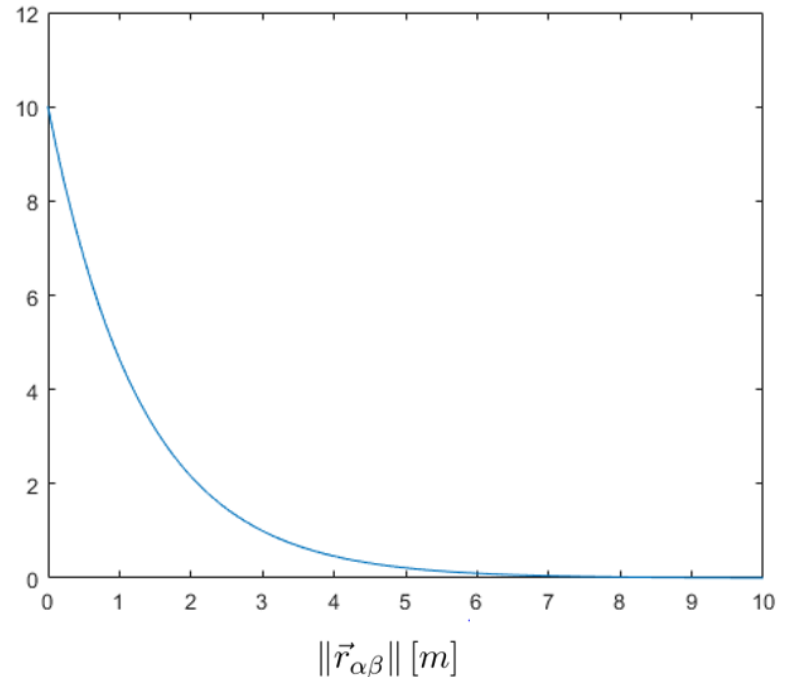
# Social Force Model

- Exponential potential shaped by  $V^0$  and  $\sigma$ :

$$V_{\alpha\beta}(\vec{r}_{\alpha\beta})$$

- Interaction Potential:

- $V_{\alpha\beta}(\vec{r}_{\alpha\beta}) = V^0 e^{-\frac{\|\vec{r}_{\alpha\beta}\|}{\sigma}}$

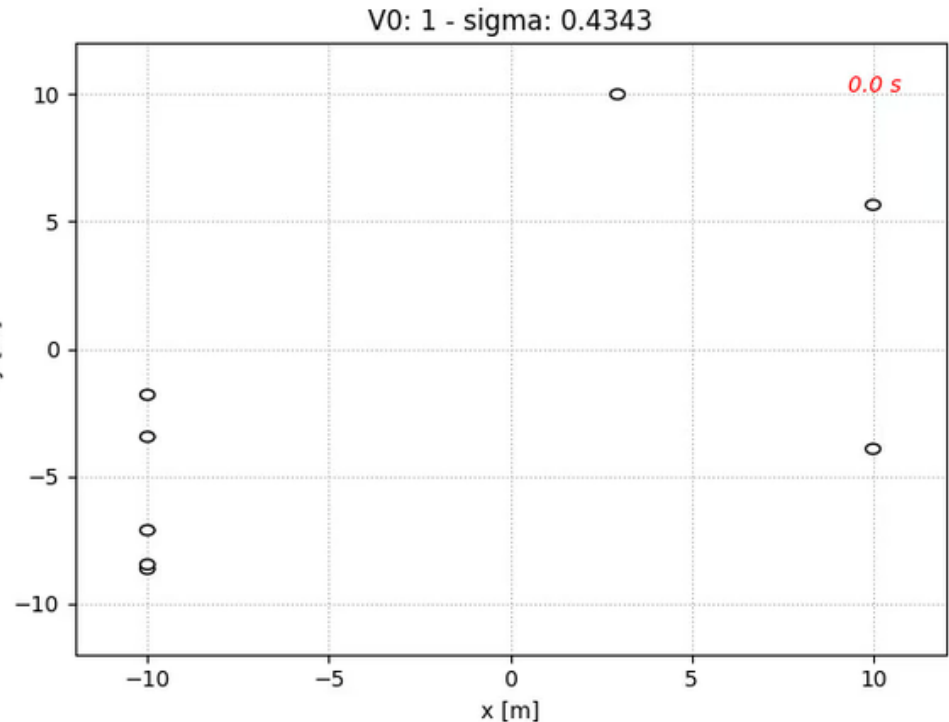


# Social Force Model

- Exponential potential shaped by  $V^0$  and  $\sigma$ :

- Interaction Potential:

- $$V_{\alpha\beta}(\vec{r}_{\alpha\beta}) = V^0 e^{-\frac{\|\vec{r}_{\alpha\beta}\|_y}{\sigma}}$$

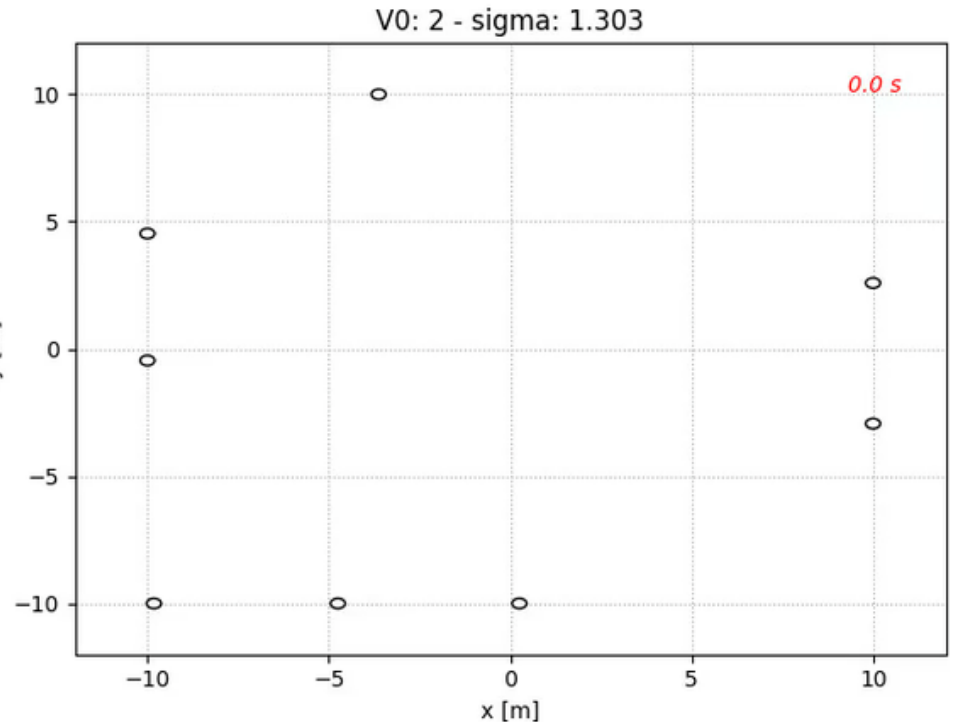


# Social Force Model

- Exponential potential shaped by  $V^0$  and  $\sigma$ :

- Interaction Potential:

- $V_{\alpha\beta}(\vec{r}_{\alpha\beta}) = V^0 e^{-\frac{\|\vec{r}_{\alpha\beta}\|_y \text{ [m]}}{\sigma}}$

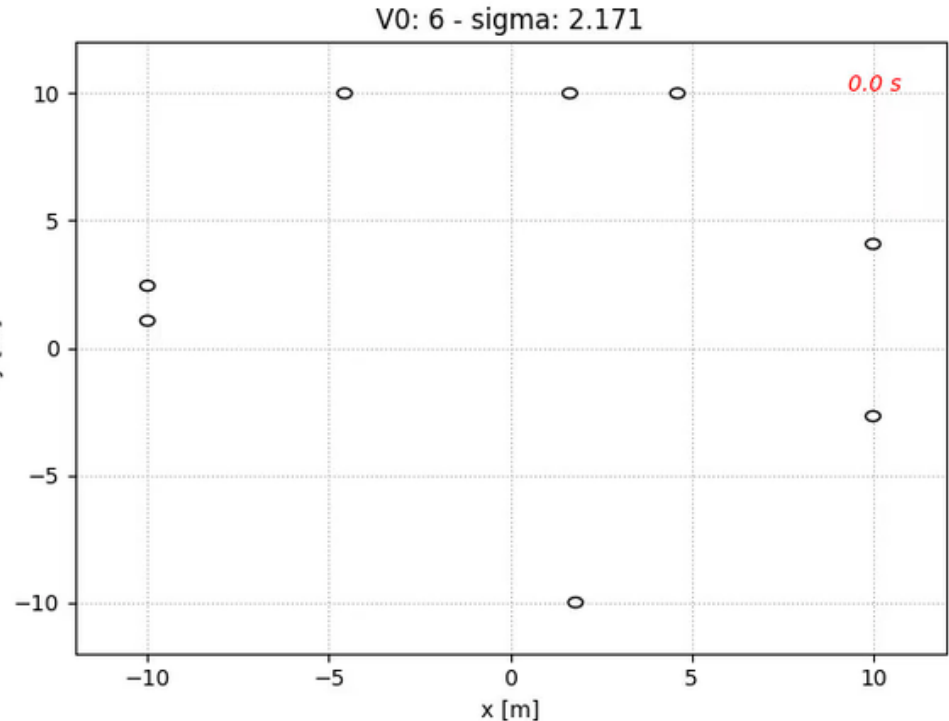


# Social Force Model

- Exponential potential shaped by  $V^0$  and  $\sigma$ :

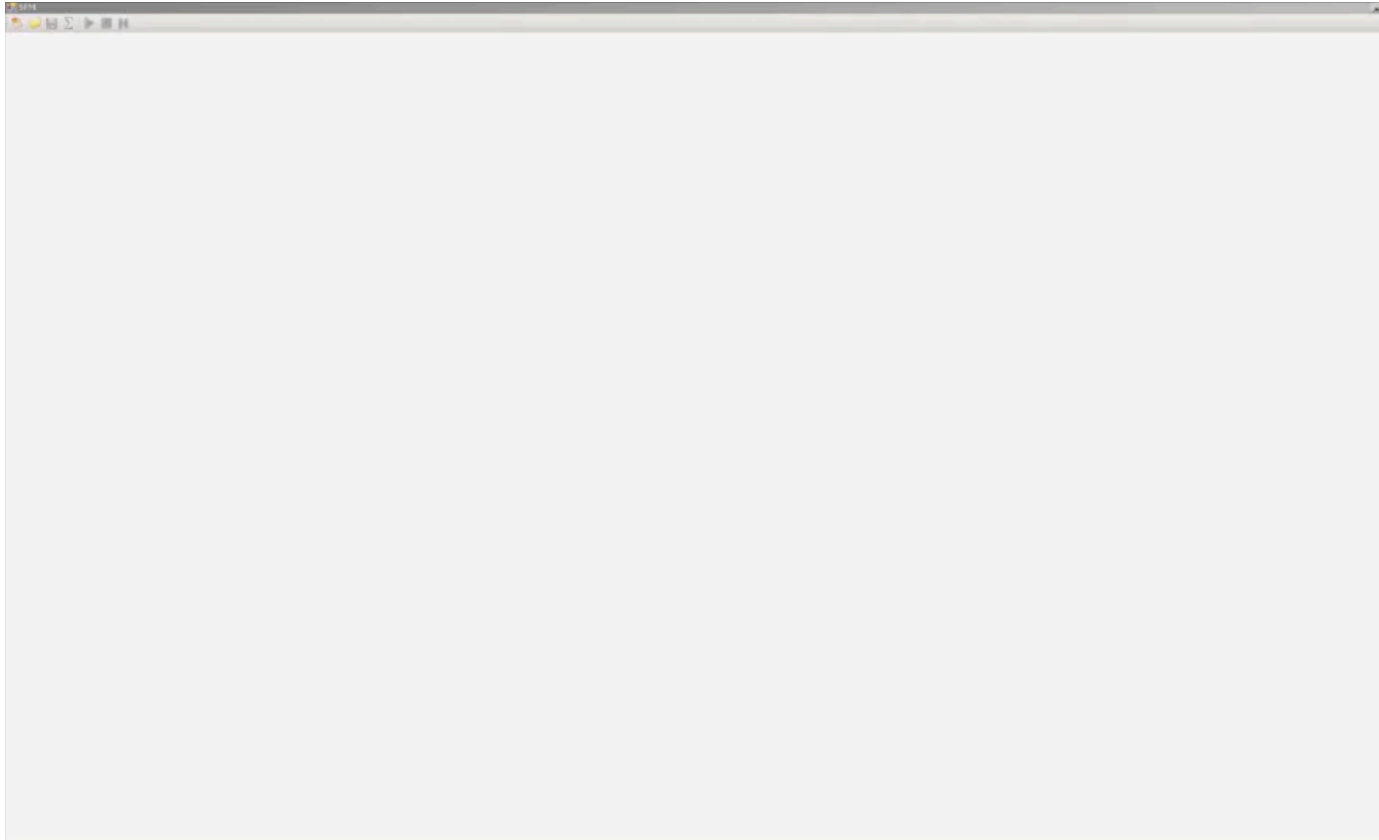
- Interaction Potential:

- $$V_{\alpha\beta}(\vec{r}_{\alpha\beta}) = V^0 e^{-\frac{\|\vec{r}_{\alpha\beta}\|_y \text{ [m]}}{\sigma}}$$





# Social Force Model Simulation



# Drawback of Social Force Model

- Model requires many parameters for each agent and all agent-agent and agent-environment pairs
- Shape of interaction functions are handcrafted

# Drawback of Social Force Model

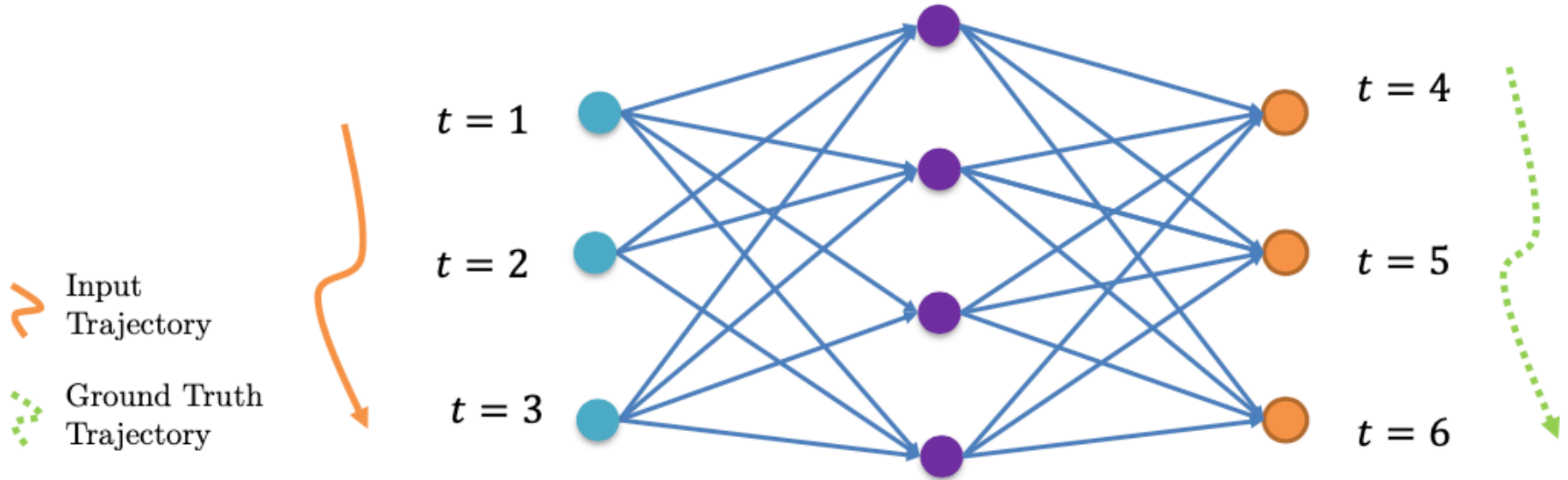
Handcrafted  
Functions



Learned  
Functions

# Neural Network for Trajectory Prediction

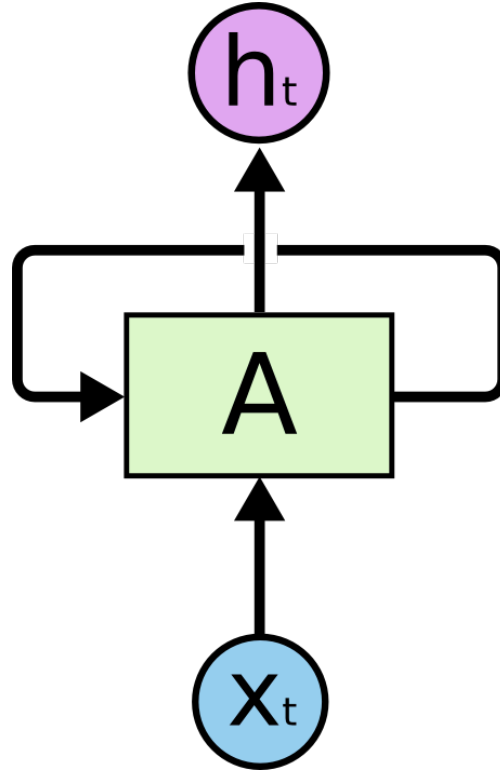
- Building a naïve prediction model with FC layers



- FC Layers do not account for sequential and temporal behavior of trajectories

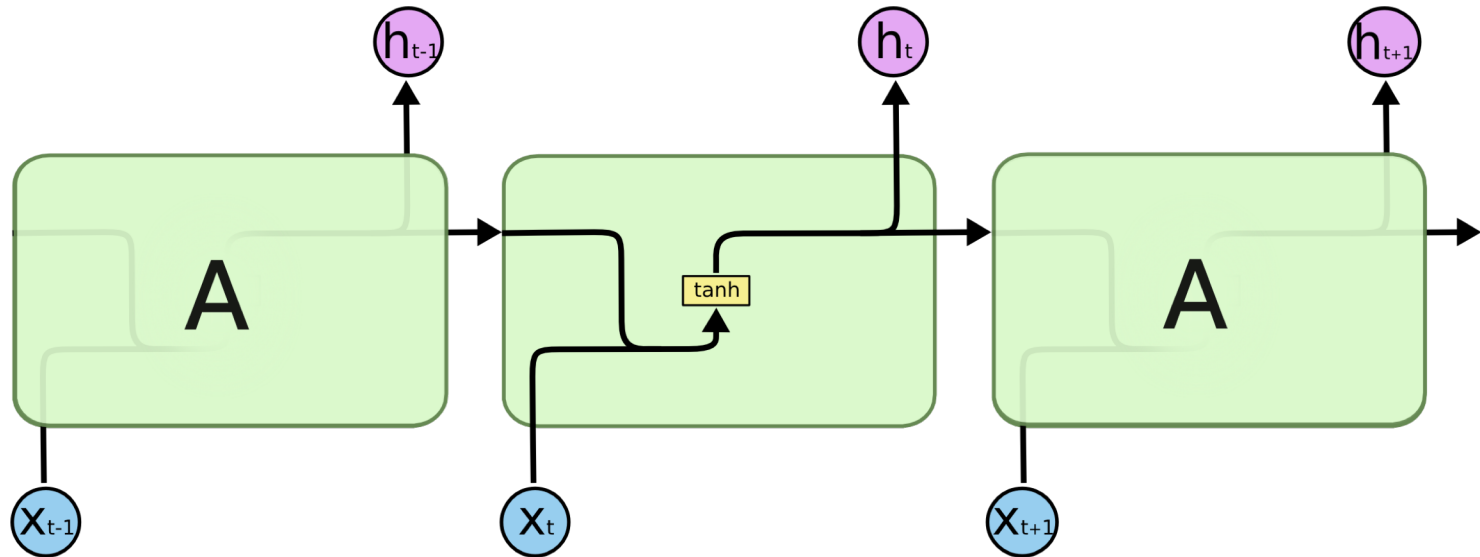
➡ Recurrent Neural Networks

# Recurrent Neural Networks

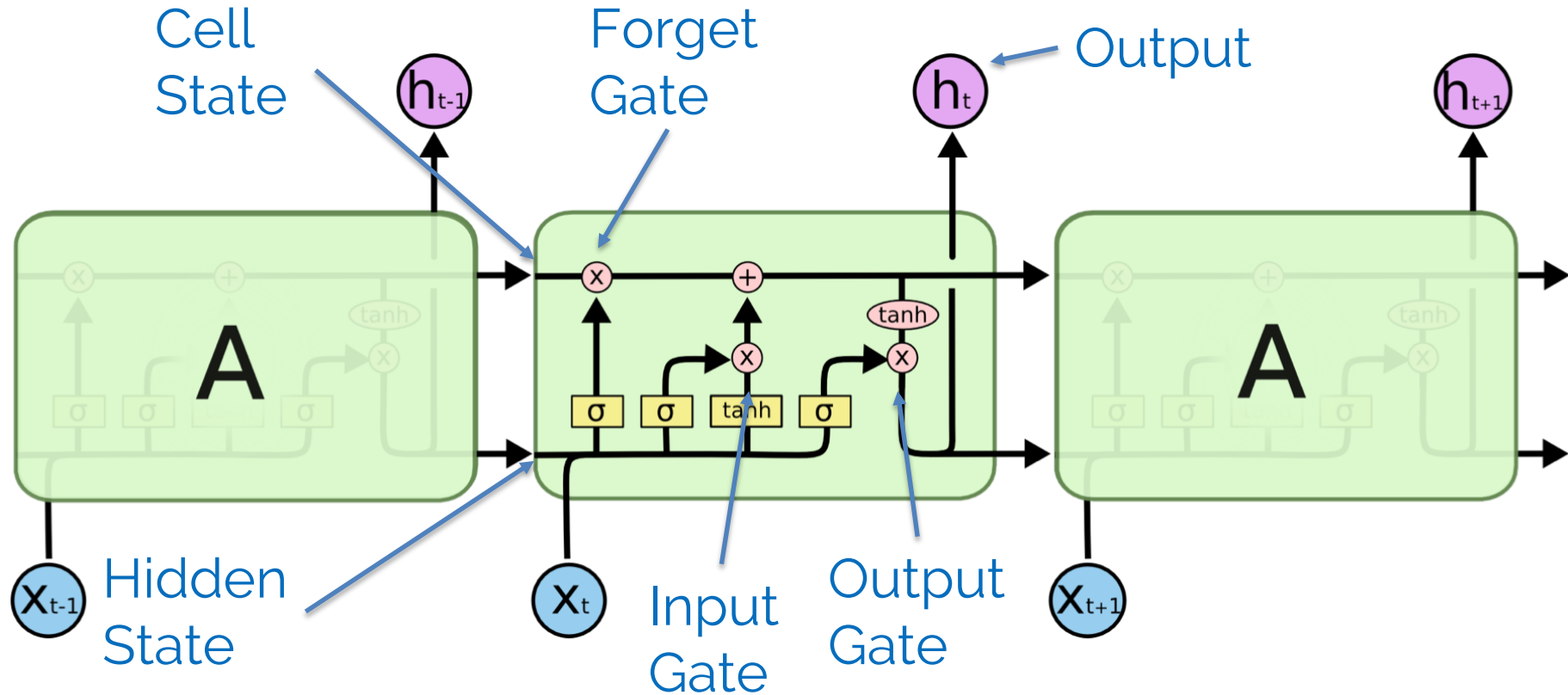


# Simple RNN

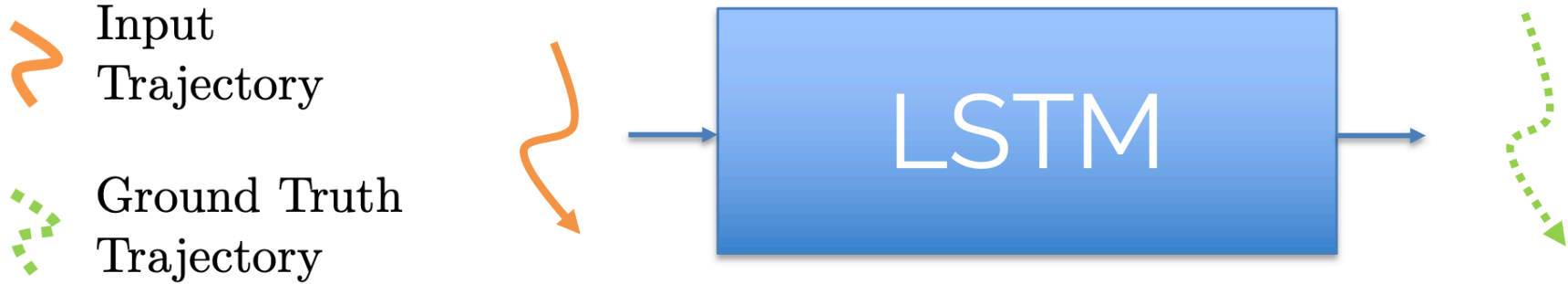
$$h_t = \tanh(W \cdot h_{t-1} + V \cdot x_t + b)$$



# LSTM (Long Short-term Memory)



# LSTM for Trajectory Prediction



- Using a single LSTM does not work well

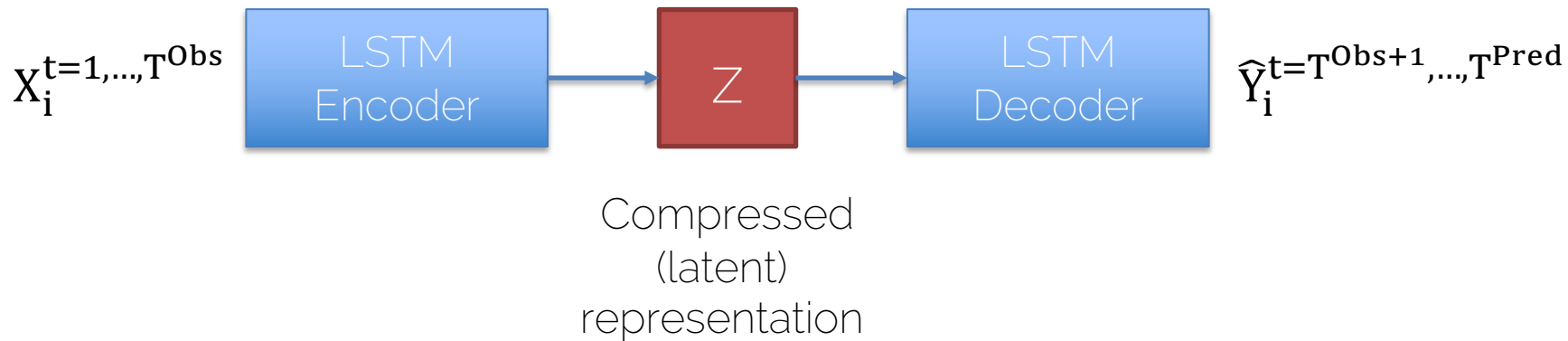
 Encoder-Decoder Architecture



# LSTM Encoder - Decoder Architecture

Input

Output



Training Objective:  $\mathcal{L} = \|\hat{Y} - Y\|_2$

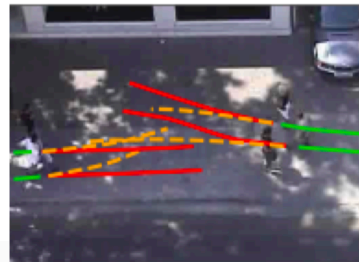
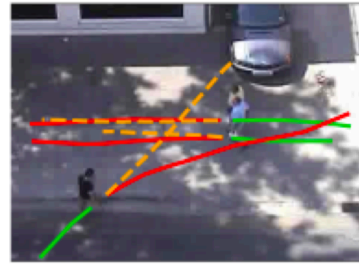
# Vanilla LSTM

— Observation

— Groundtruth

- - - V-LSTM

- - - V-LSTM



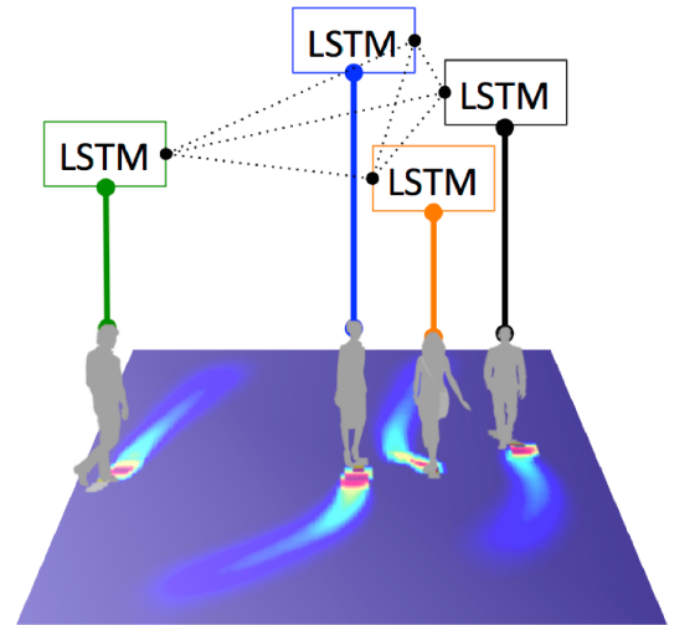
- Vanilla LSTM is not able to predict trajectories of interacting pedestrians

[Zhang et al. 19] SR-LSTM

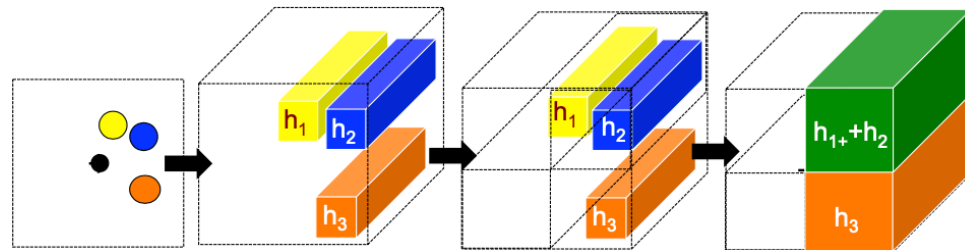
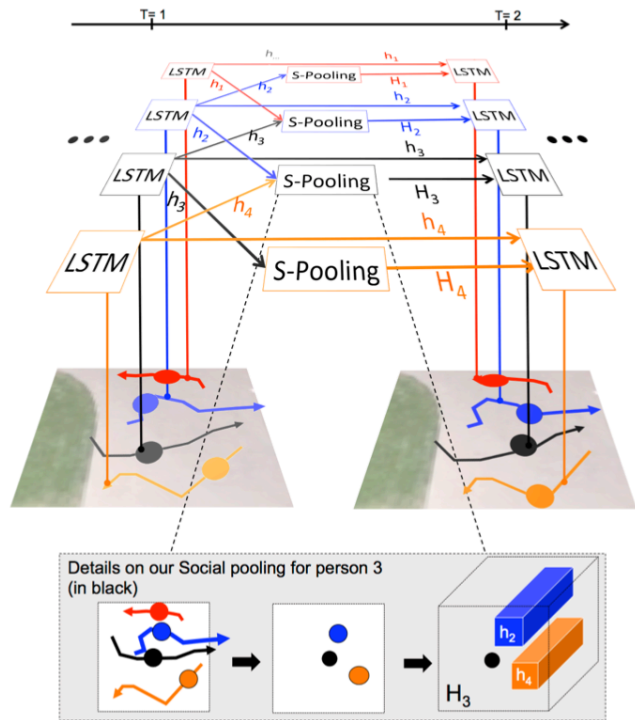
# Social LSTM

Contribution: Modeling social interactions

- LSTM encoder-decoder architectures
- Social pooling between neighboring pedestrians in each time step



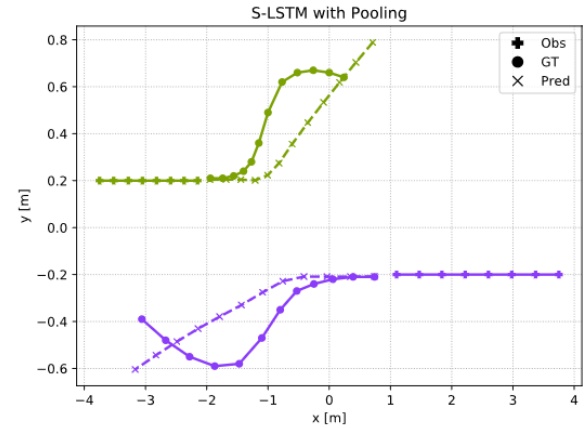
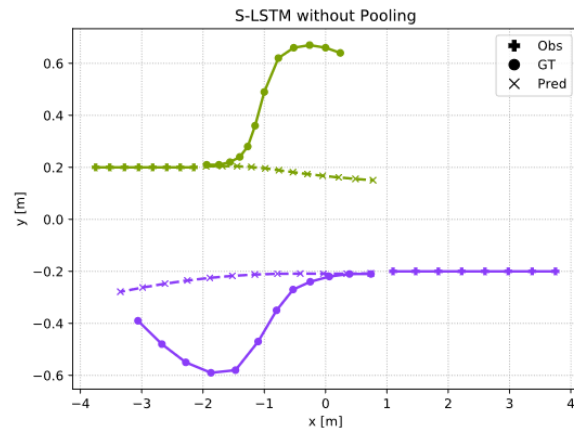
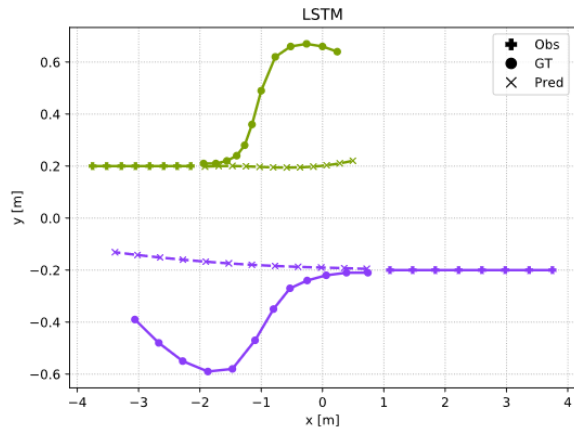
# Social LSTM – Pooling Module



- Interaction Module pools hidden states of LSTM of pedestrian in vicinity
- Pooled hidden states are passed to decoder for next step prediction

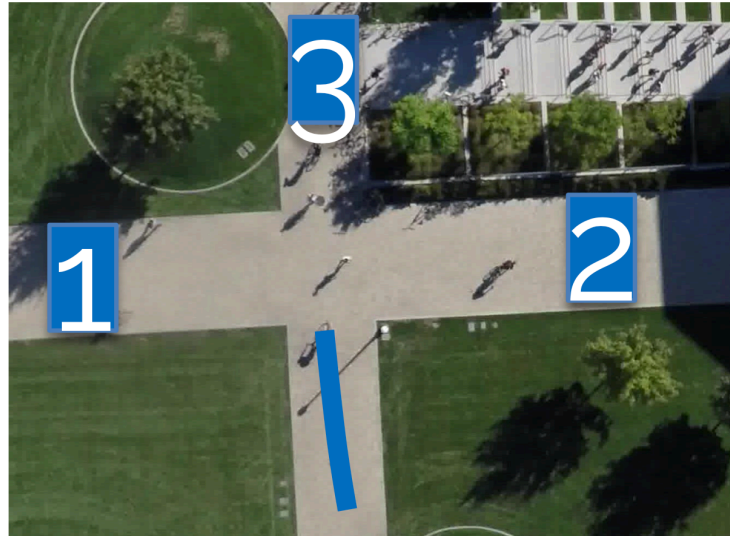
# Social LSTM – Result

Comparison of Models with and without social pooling



Social Pooling can resolve social interactions

# Pedestrian Trajectories are multimodal



➔ Same past trajectory can have multiple realistic future trajectories

# Deterministic vs. Stochastic Models

Deterministic

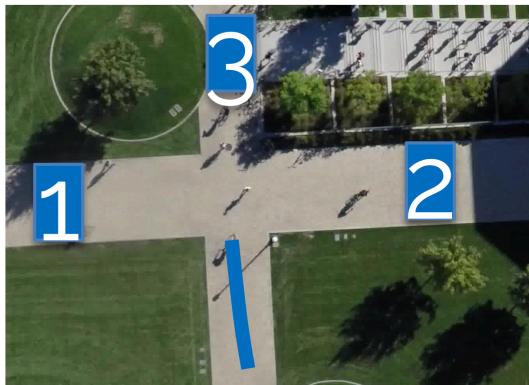
One-to-one mapping



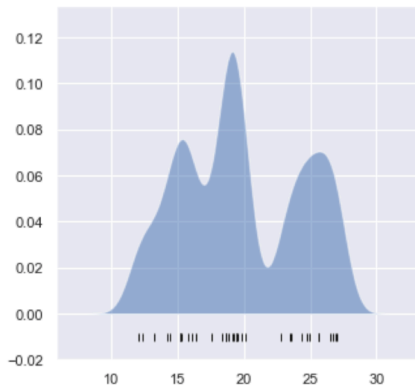
Learning distribution of future trajectories instead of deterministic mapping

Stochastic

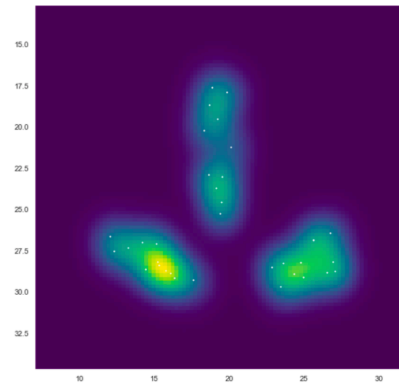
One-to-many mapping



Scenario



Distribution of final end



2d distribution of final end

# Towards Generative Models

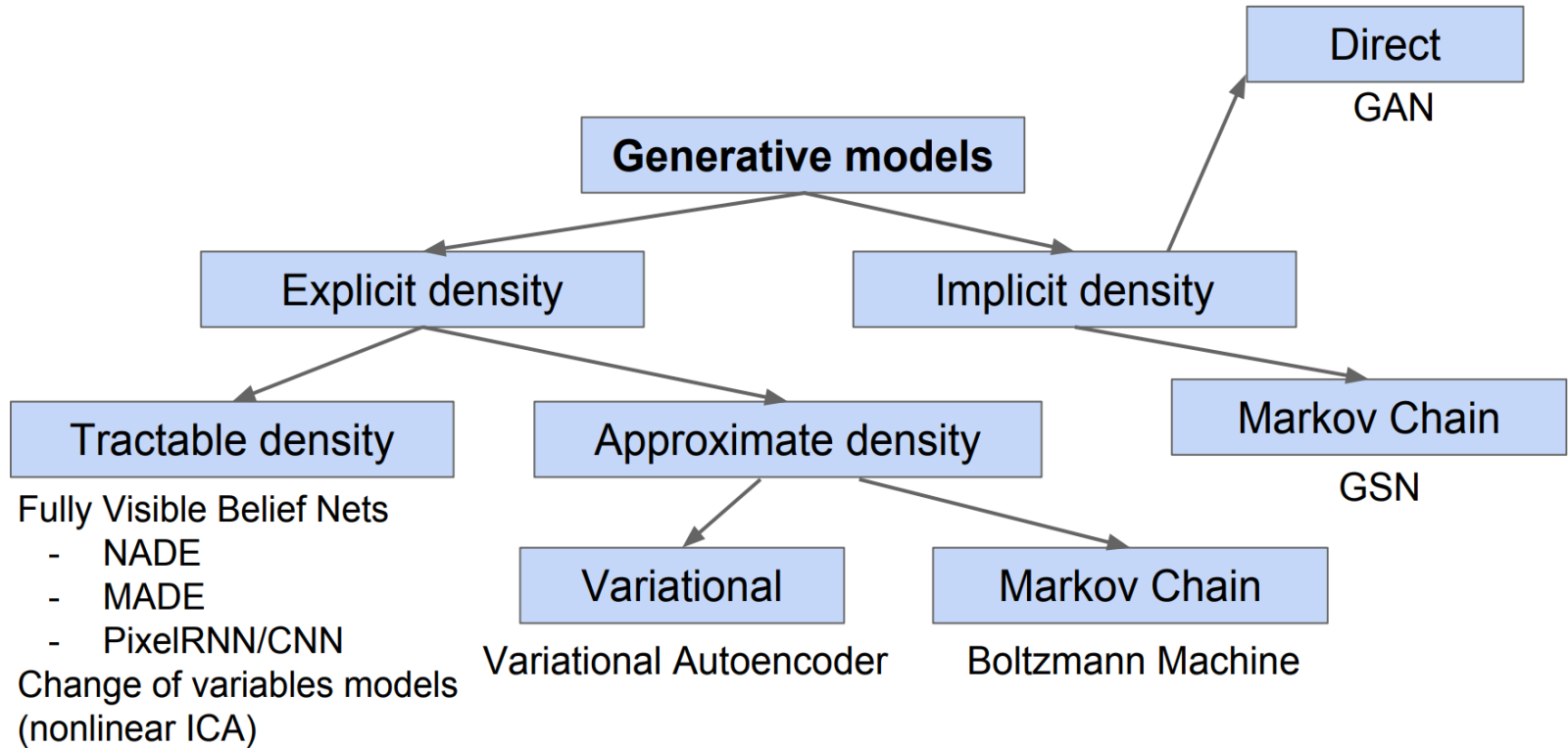
Deterministic  
Models



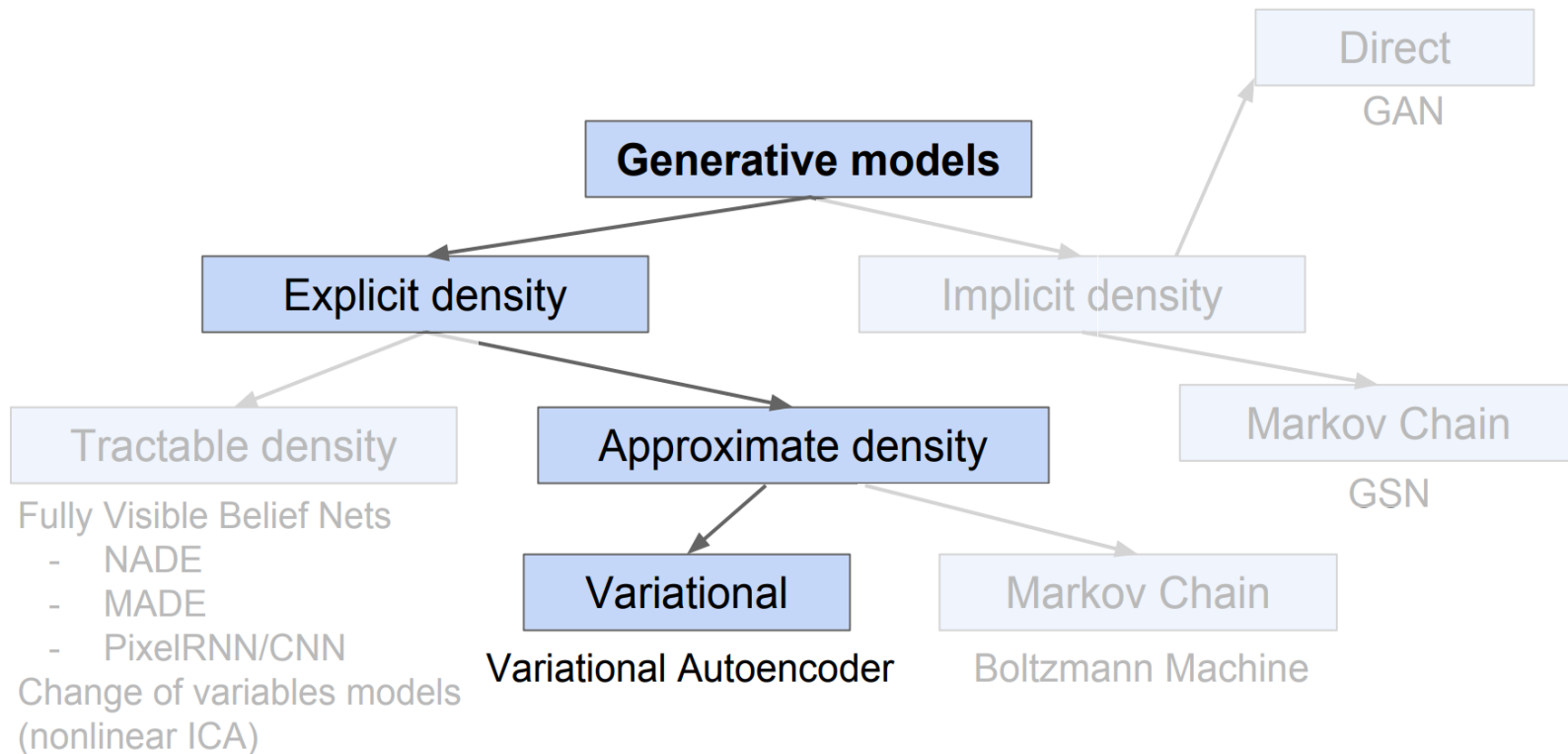
Generative  
Models



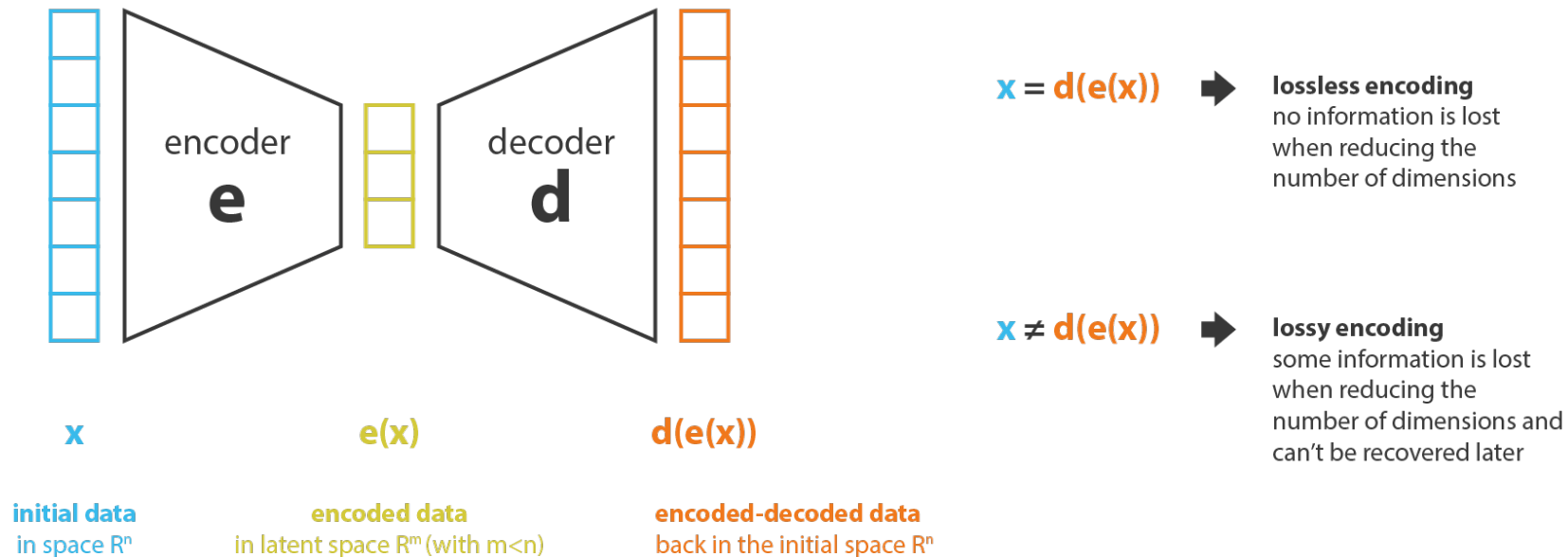
# Generative Models



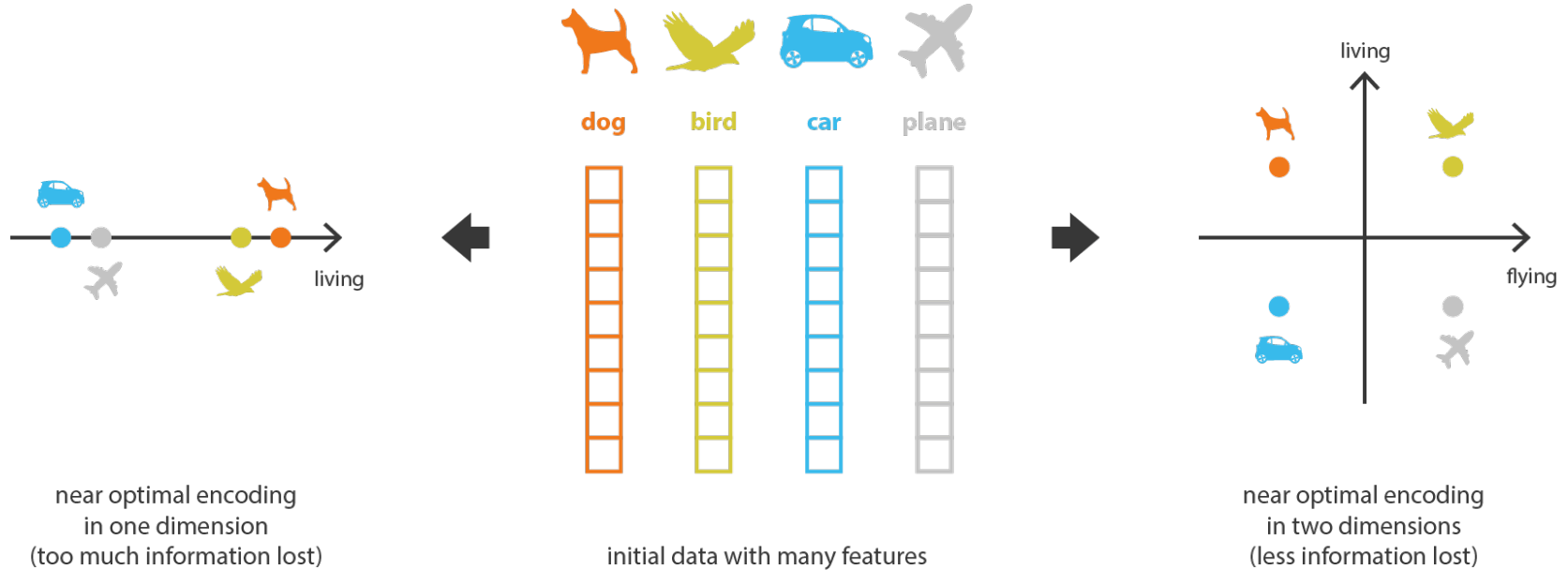
# Recap: Generative Models



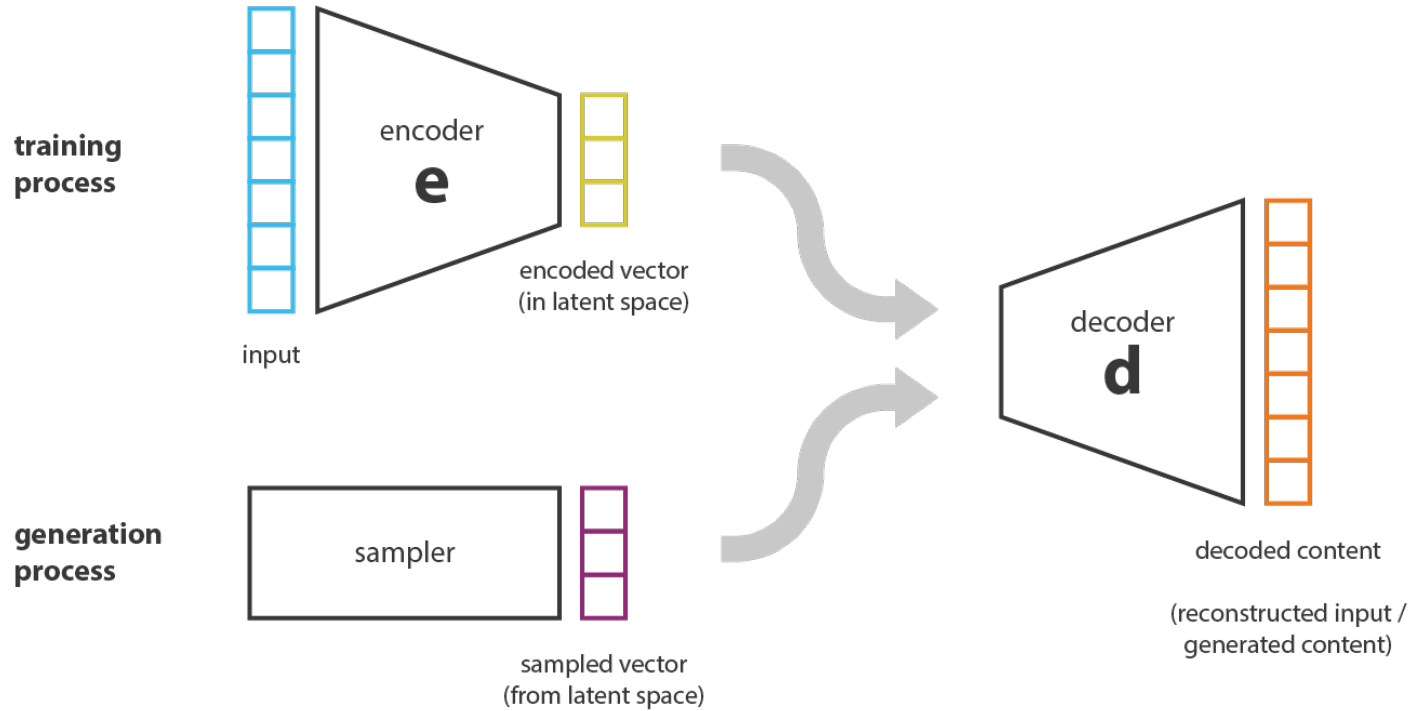
# Variational Autoencoder



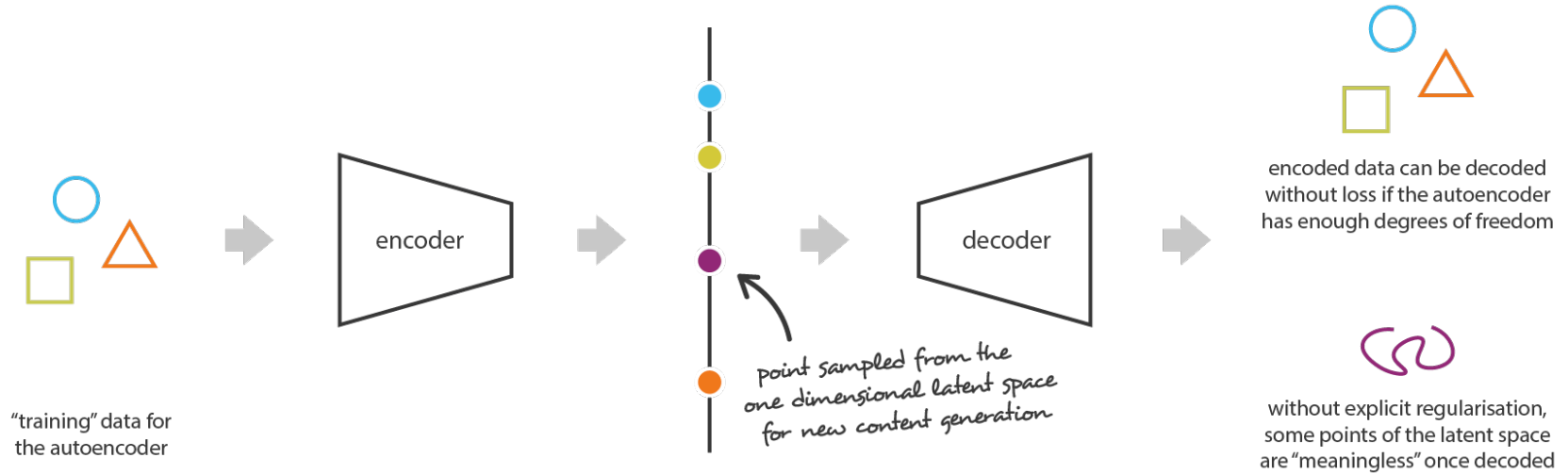
# Variational Autoencoder



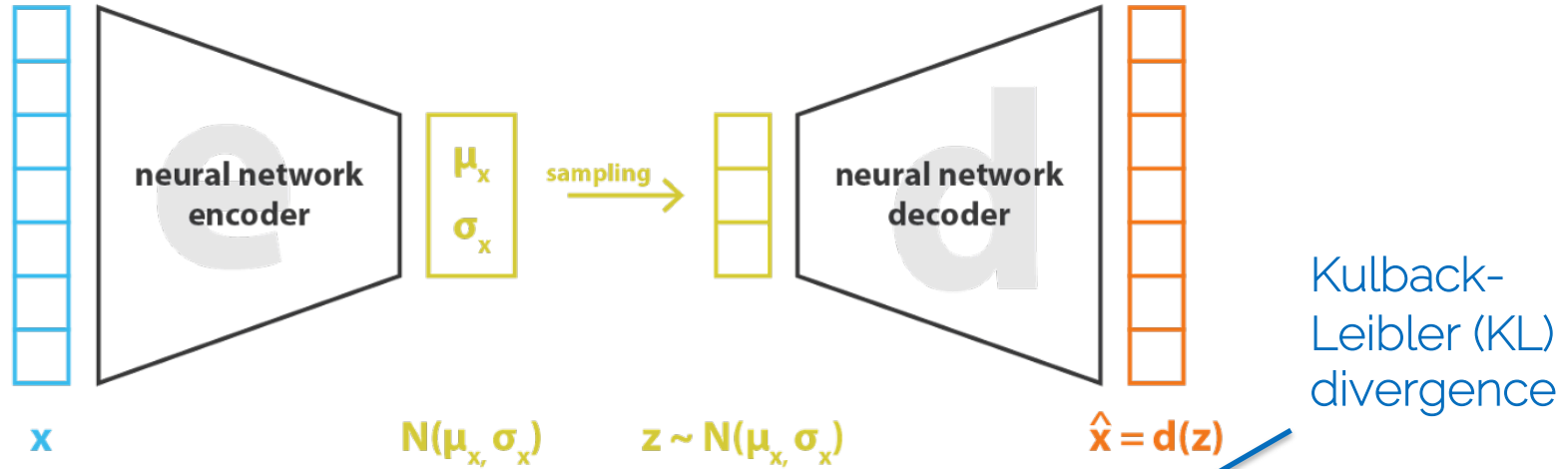
# Variational Autoencoder



# Variational Autoencoder



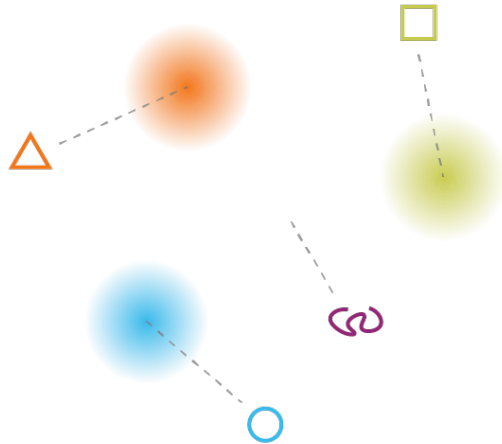
# Variational Autoencoder



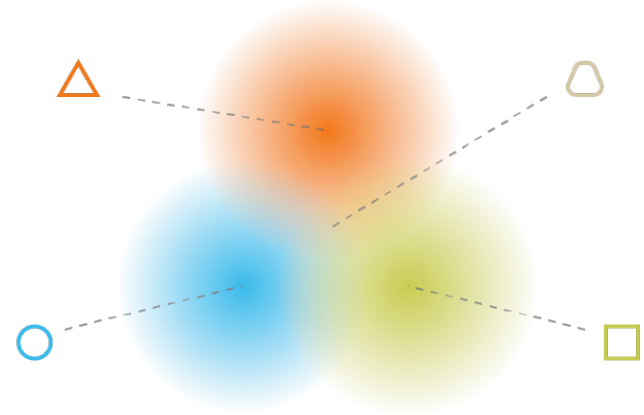
---

$$\text{loss} = \|x - \hat{x}\|^2 + \text{KL}[N(\mu_x, \sigma_x), N(0, I)] = \|x - d(z)\|^2 + \text{KL}[N(\mu_x, \sigma_x), N(0, I)]$$

# Variational Autoencoder



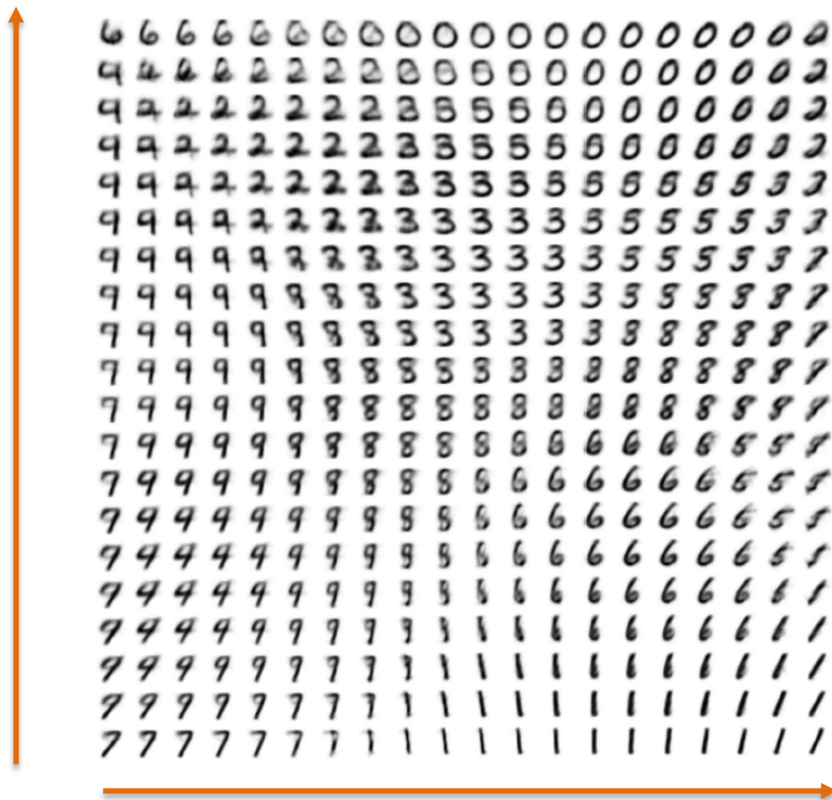
what can happen without regularisation



what we want to obtain with regularisation



# Variational Autoencoder



Each element of  $z$  encodes a different feature

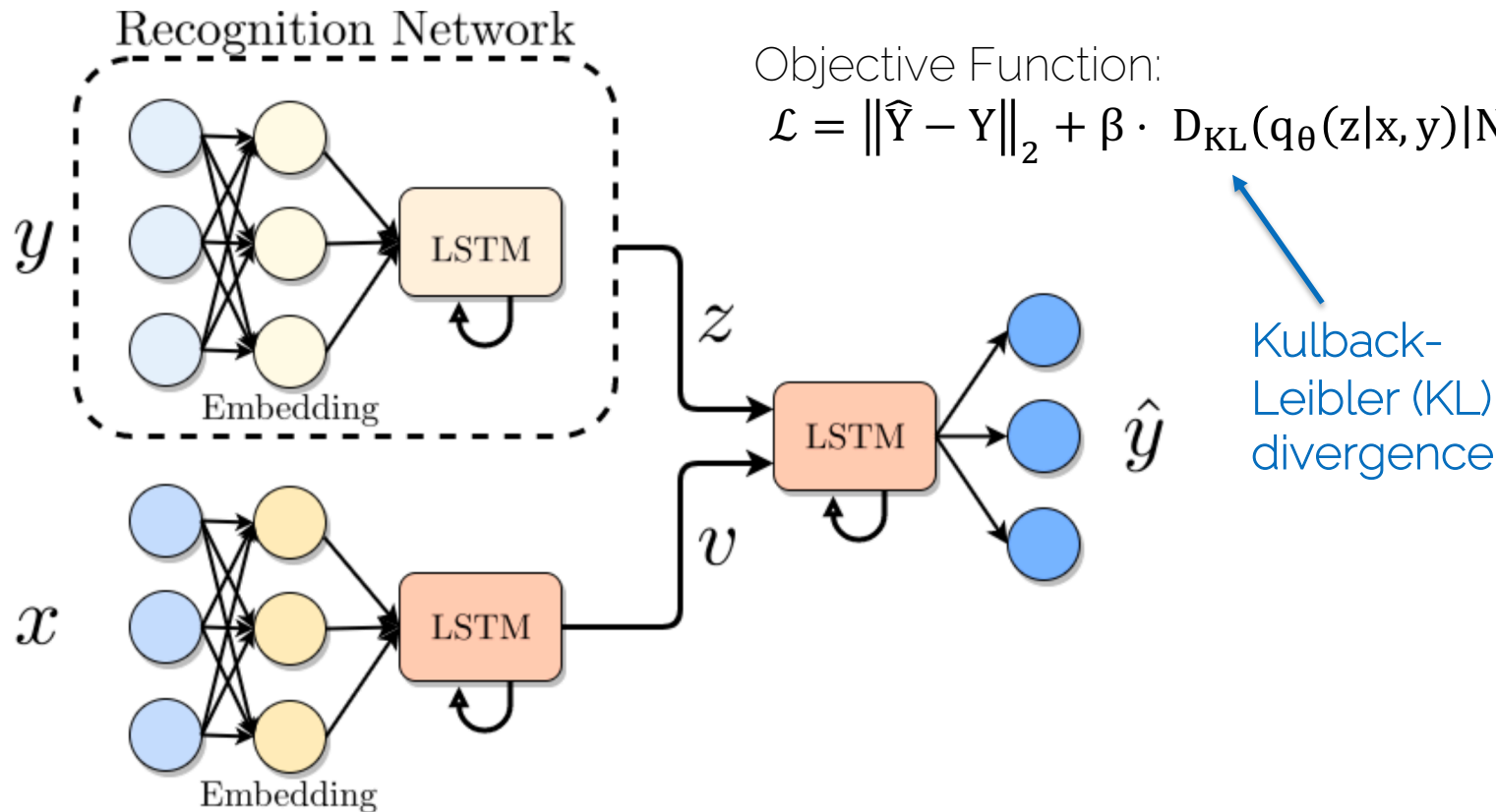
# Variational Autoencoder

Degree of smile



Head pose

# Conditional Variational Autoencoder

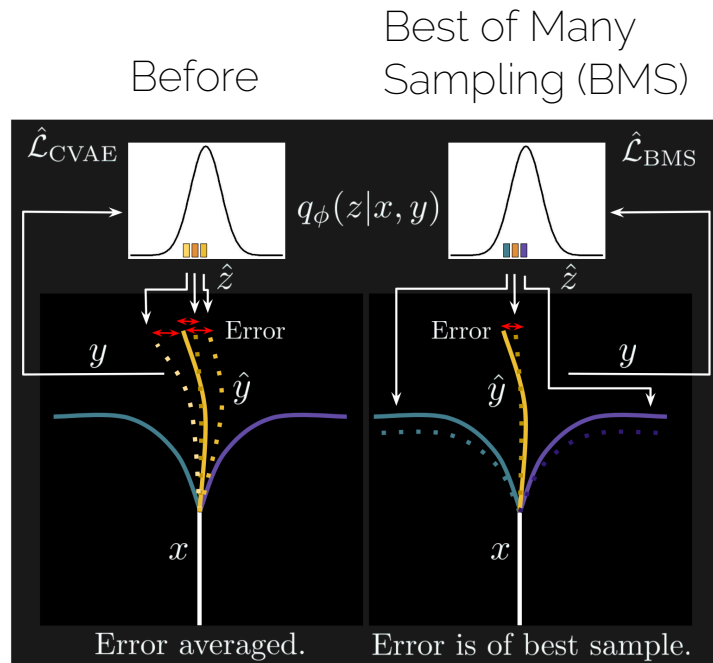


# Best-of-many-sampling

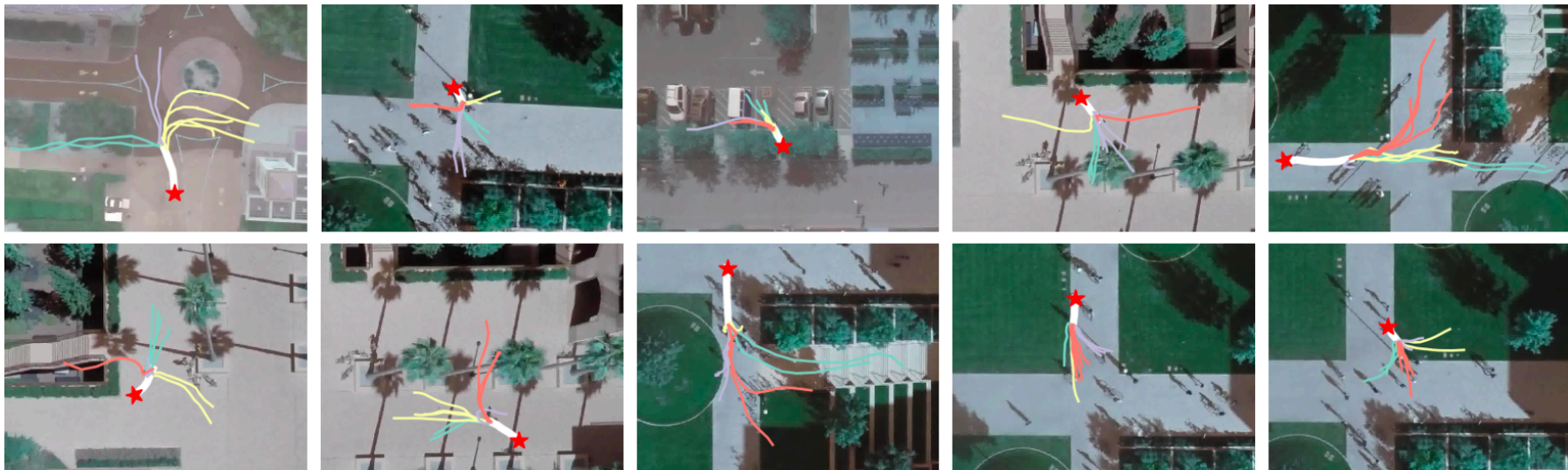
- Sample multiple trajectories
- Backpropagate sample with minimal error

Objective Function:

$$\mathcal{L} = \min_j \|\hat{Y} - Y\|_2 + \beta \cdot D_{\text{KL}}(q_{\theta}(z|x, y) | N(0,1))$$

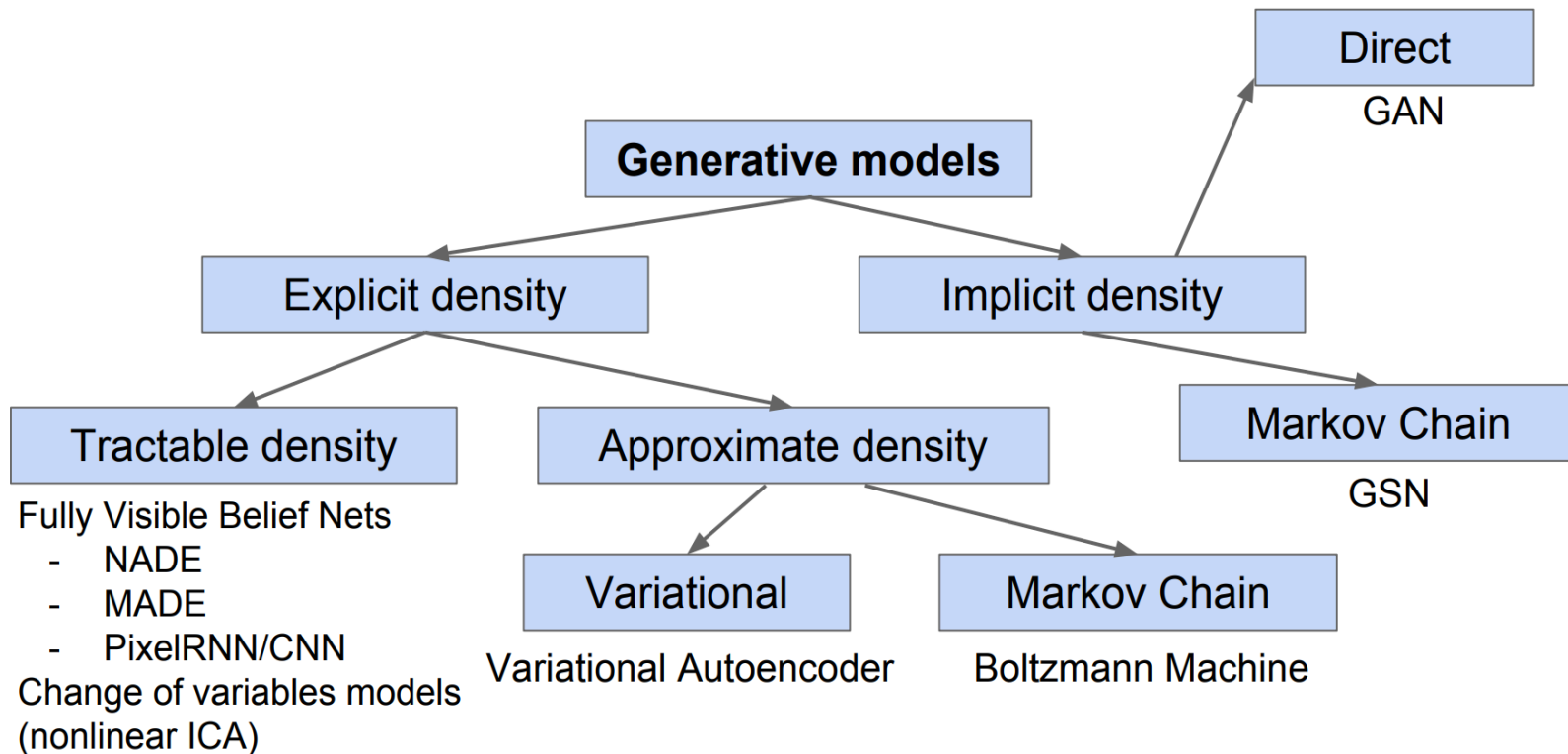


# Visual results cVAE

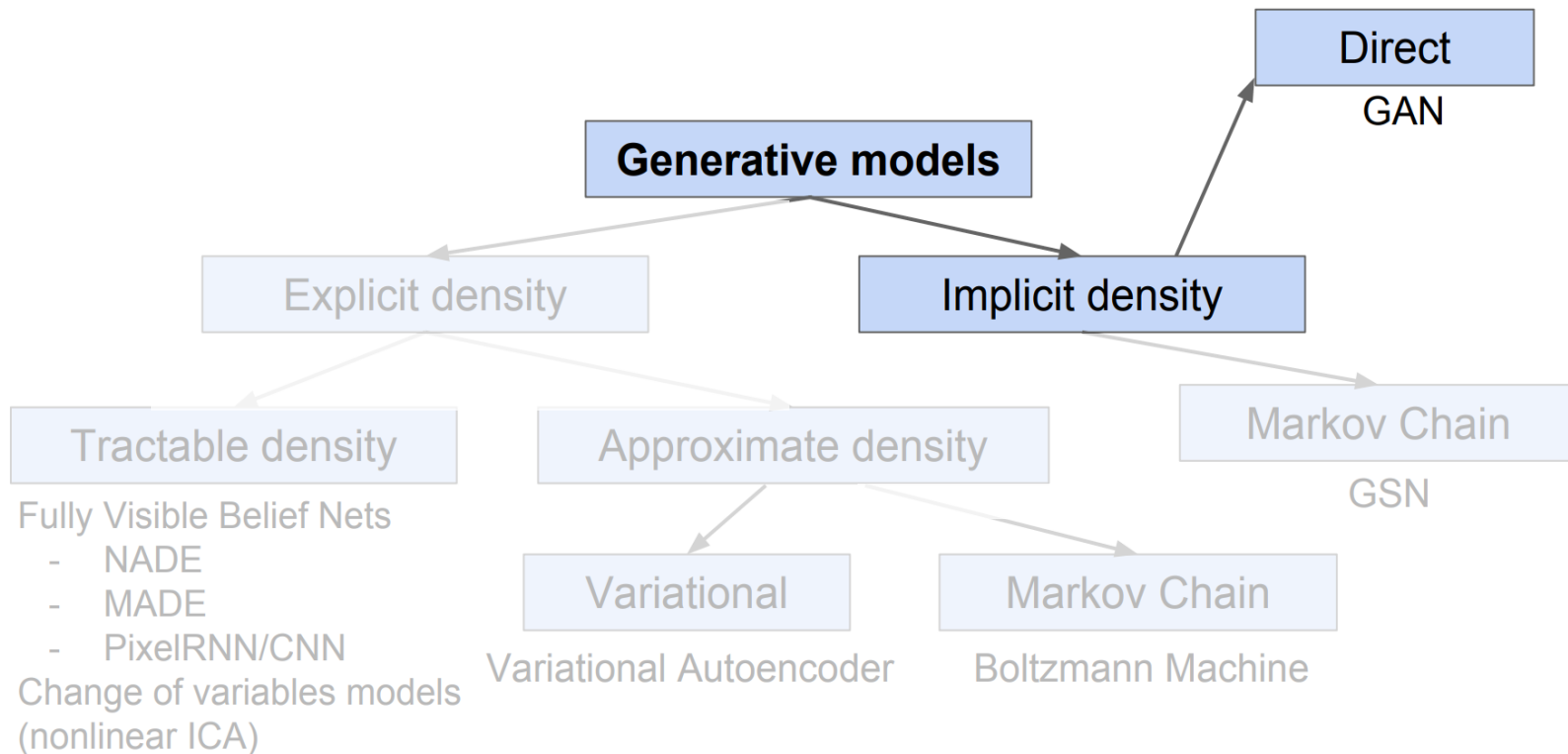


K-mean clustered trajectories

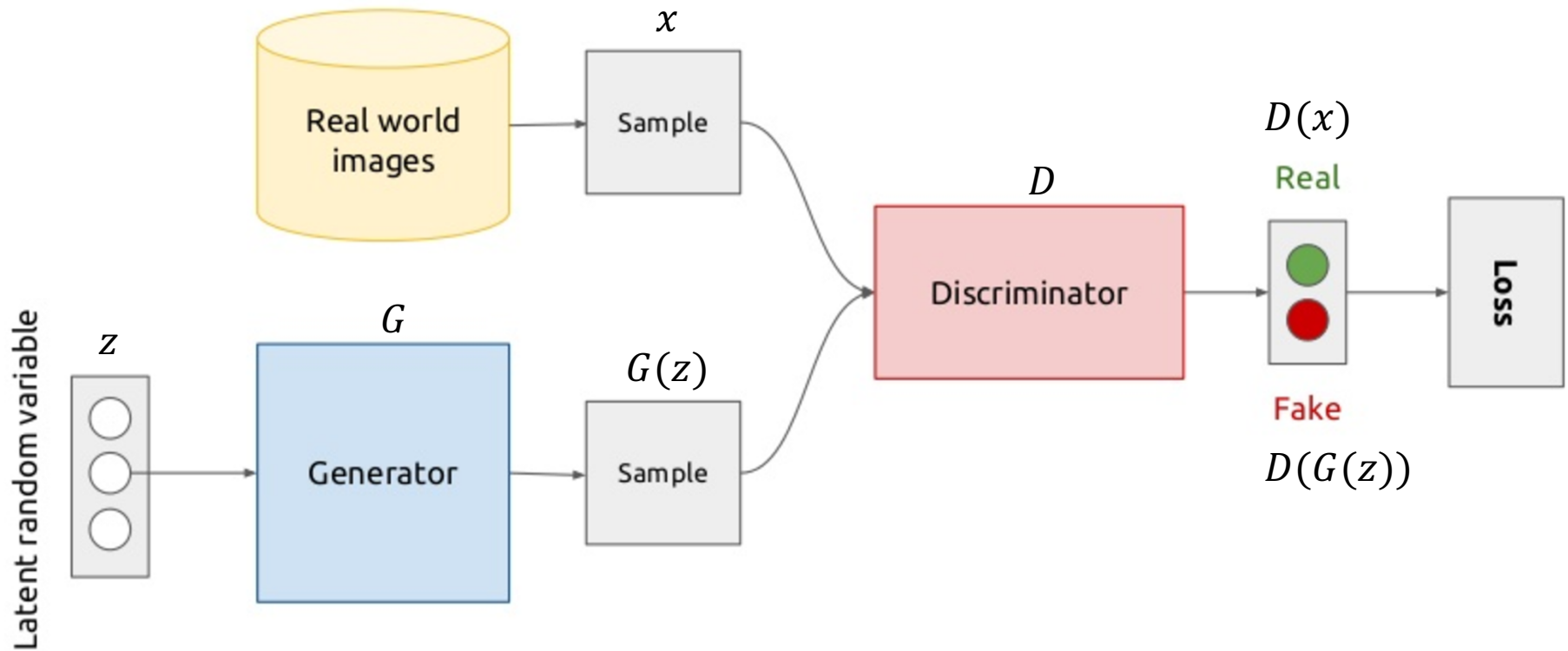
# Recap: Generative Models



# Recap: Generative Models

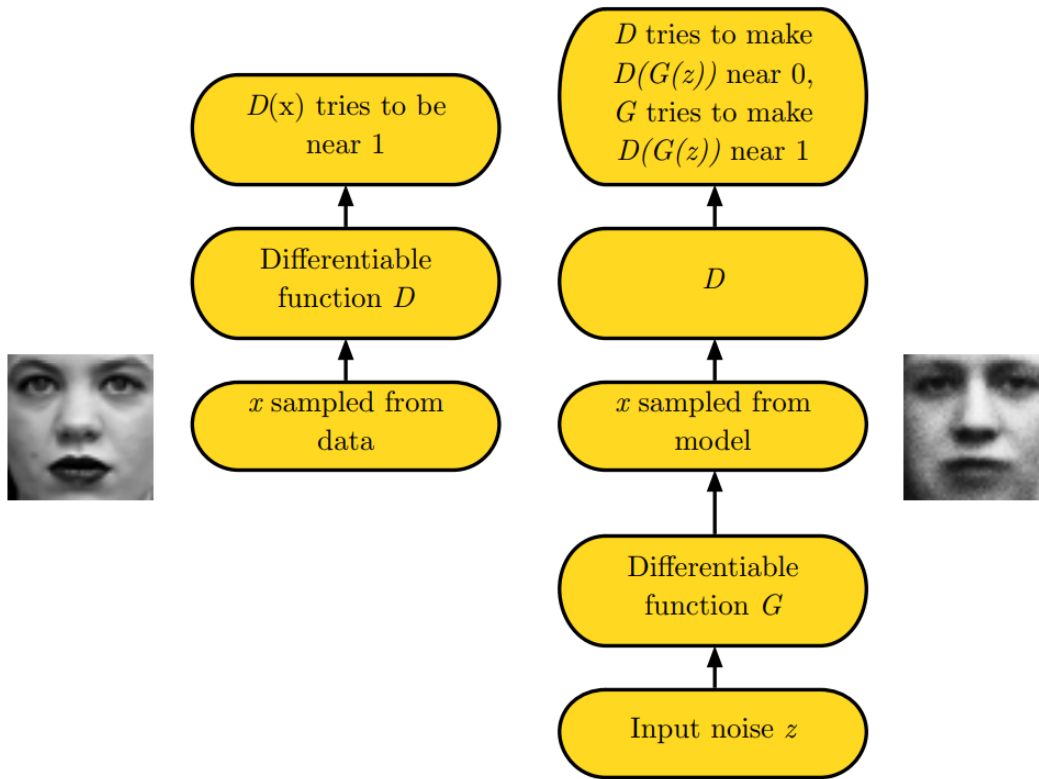


# Generative Adversarial Networks (GANs)





# Generative Adversarial Networks (GANs)



# GANs: Loss Functions

Discriminator loss

$$J^{(D)} = \underbrace{-\frac{1}{2} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \log D(\mathbf{x}) - \frac{1}{2} \mathbb{E}_{\mathbf{z}} \log (1 - D(G(\mathbf{z})))}_{\text{binary cross entropy}}$$

Generator loss

$$J^{(G)} = -J^{(D)}$$

- Minimax Game:
    - G minimizes probability that D is correct
    - Equilibrium is saddle point of discriminator loss
- > D provides supervision (i.e., gradients) for G

# GANs: Loss Functions

Discriminator loss

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\log D(\mathbf{x}) - \frac{1}{2}\mathbb{E}_{\mathbf{z}}\log(1 - D(G(\mathbf{z})))$$

Generator loss

$$J^{(G)} = -\frac{1}{2}\mathbb{E}_{\mathbf{z}}\log D(G(\mathbf{z}))$$

- Heuristic Method (often used in practice)
  - G maximizes the log-probability of D being mistaken
  - G can still learn even when D rejects all generator samples

# Vanilla GAN

**for** number of training iterations **do**

**for**  $k$  steps **do**

- Sample minibatch of  $m$  noise samples  $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$  from noise prior  $p_g(\mathbf{z})$ .
- Sample minibatch of  $m$  examples  $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$  from data generating distribution  $p_{\text{data}}(\mathbf{x})$ .
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[ \log D(\mathbf{x}^{(i)}) + \log (1 - D(G(\mathbf{z}^{(i)}))) \right].$$

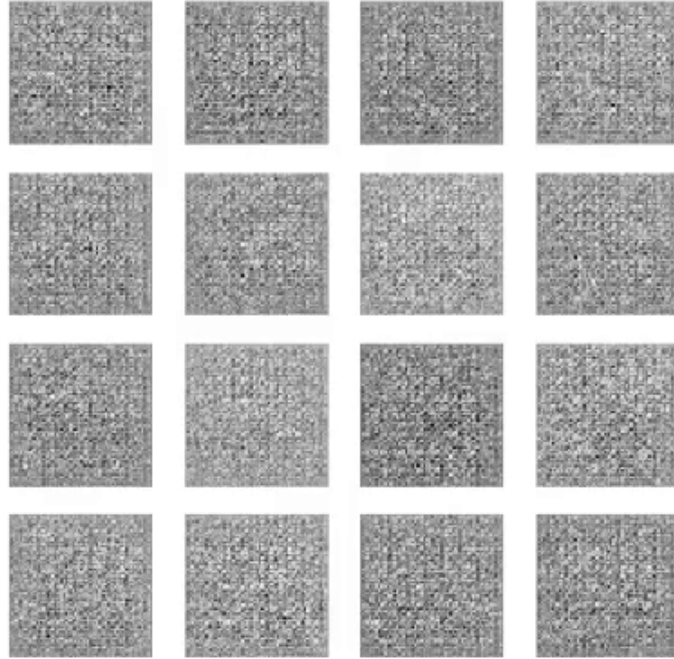
**end for**

- Sample minibatch of  $m$  noise samples  $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$  from noise prior  $p_g(\mathbf{z})$ .
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log (1 - D(G(\mathbf{z}^{(i)}))).$$

**end for**

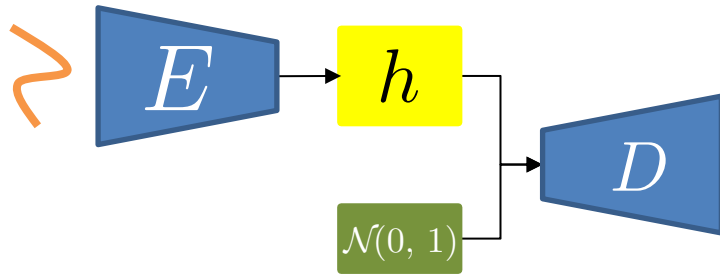
# Training GAN



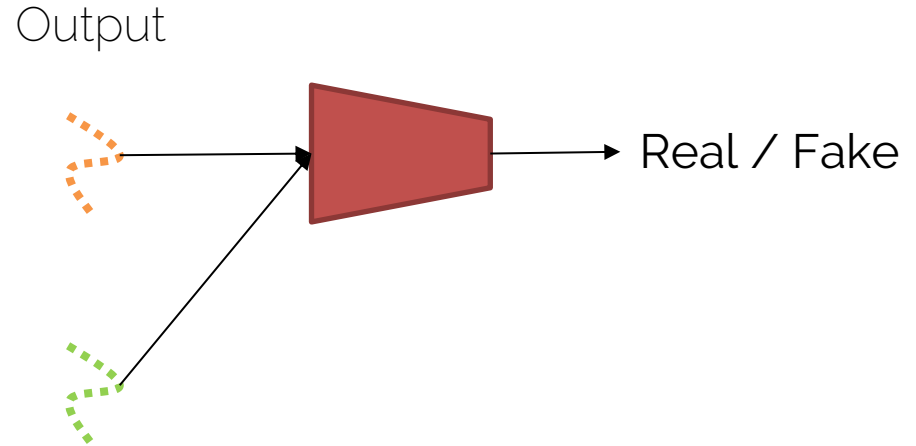
<https://medium.com/ai-society/gans-from-scratch-1-a-deep-introduction-with-code-in-pytorch-and-tensorflow-cb03cdcdba0f>

# GANs for Trajectory Prediction

Generator



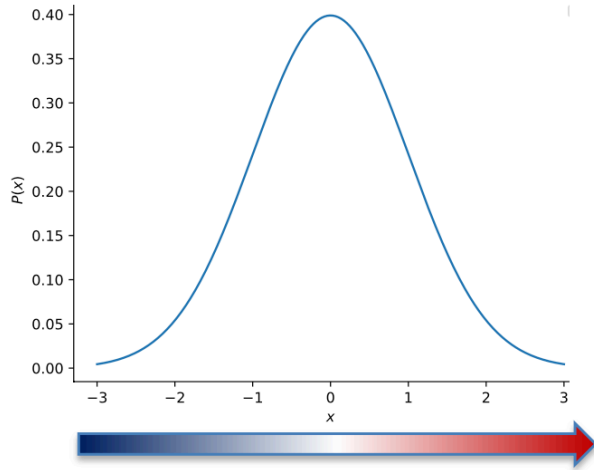
Discriminator



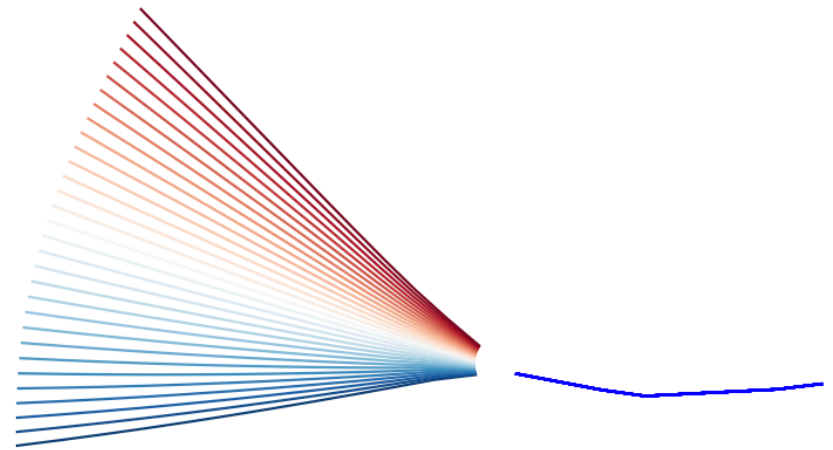
# Latent space in GANs

- Different latent space samples result into different real space output

Latent Space

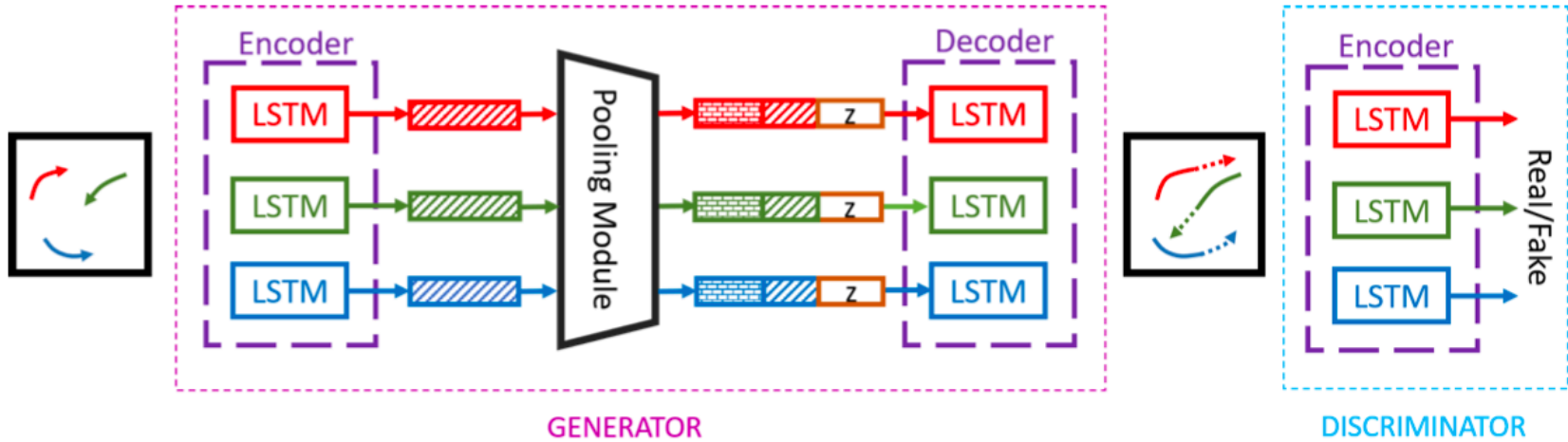


Real Space



# Social GAN

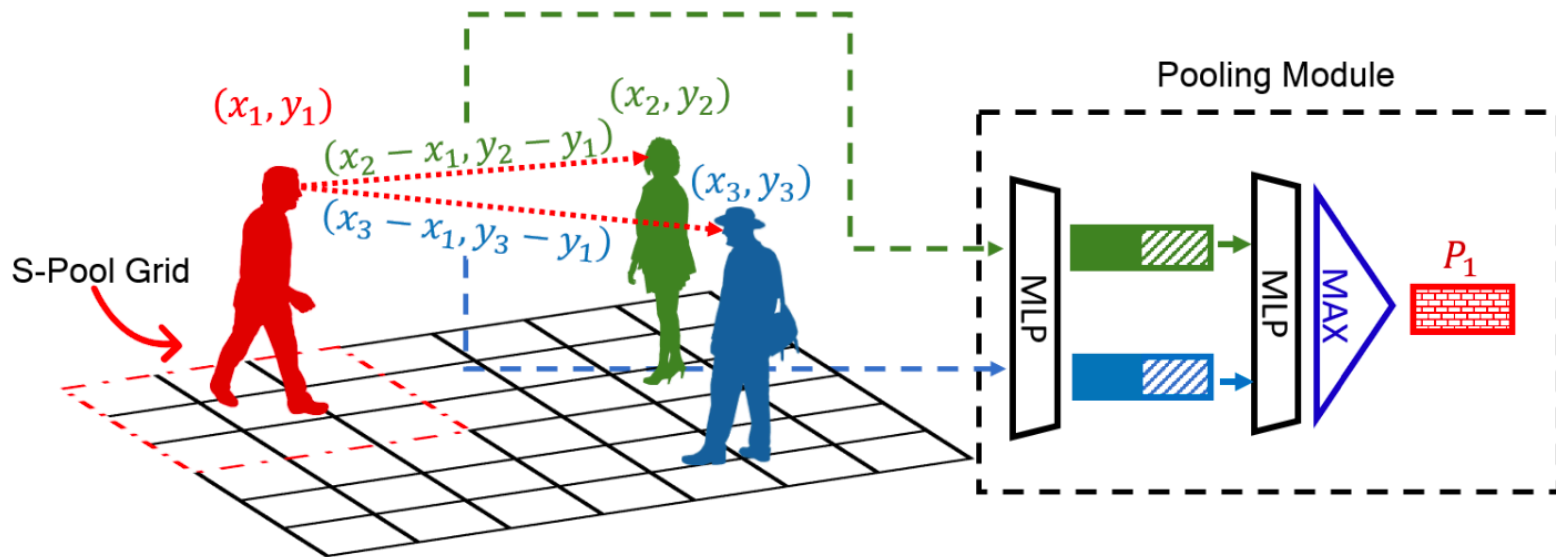
Contribution: GAN + Social Interactions





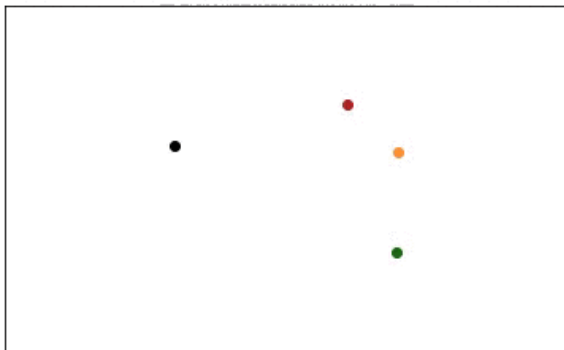
# SGAN Pooling Module

- Pooling Vector:  $\max_j \text{MLP}([x_j - x_i, y_j - y_i, h_j^{t-1}])$
- max – operation is symmetric (initial order does not matter)
- Pooling is not restricted to grid (S-LSTM)

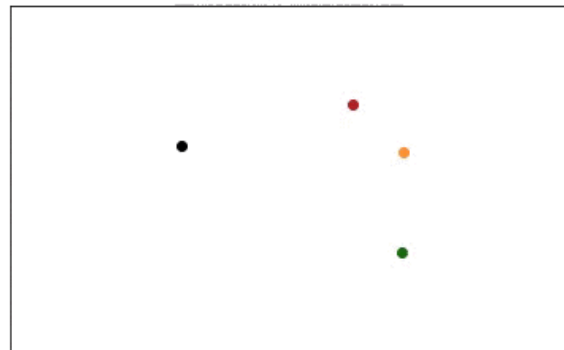


# SGAN Results

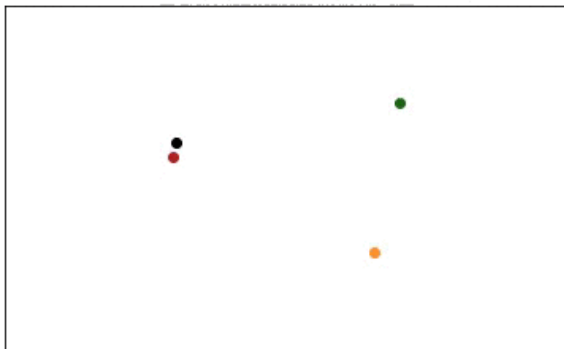
Ground Truth Observed



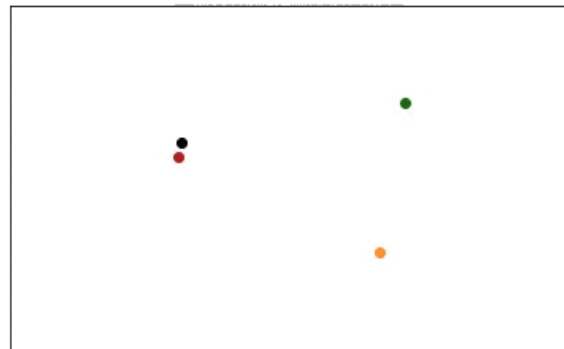
Our Model Observed



Ground Truth Observed



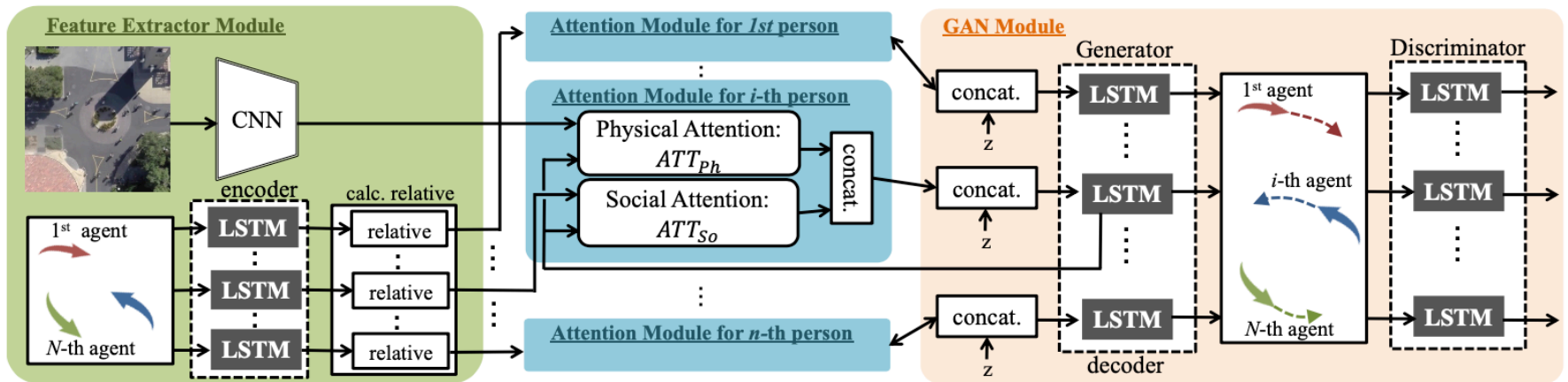
Our Model Observed



# SoPhie

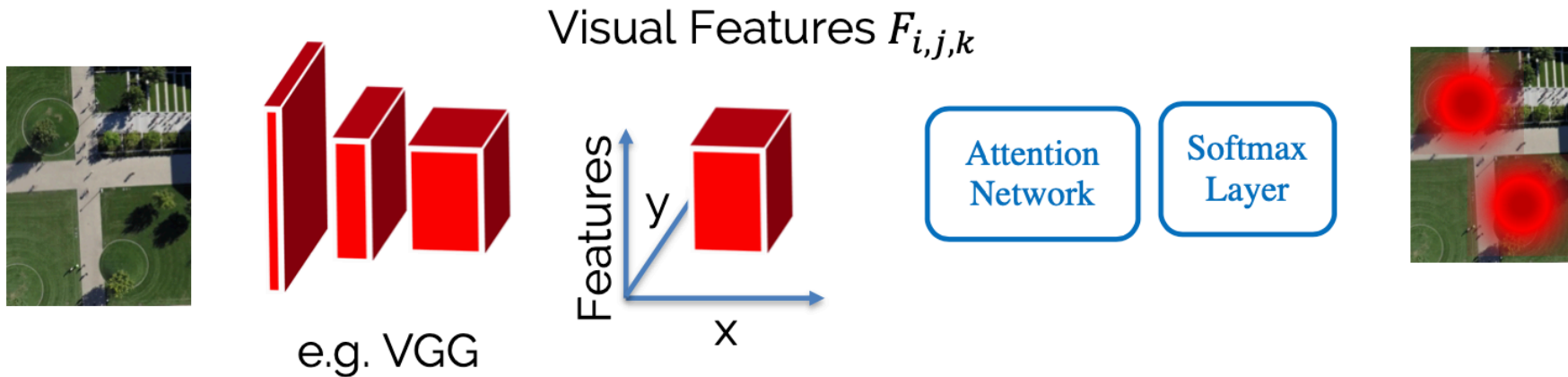
Contribution: Combining physical and social interactions

- SoPhie uses soft attention instead of max-pooling



# Visual Attention

- Spatial Attention selects relevant features for each region in scene



- Attention values weight visual features for each spatial position

$$\alpha_{ijk} = \text{Softmax}(\text{MLP}(F_{ij}))$$

$$A_{ij} = \sum_k \alpha_{ijk} \cdot F_{ijk}$$

[Sadeghian et al. 18] SoPhie

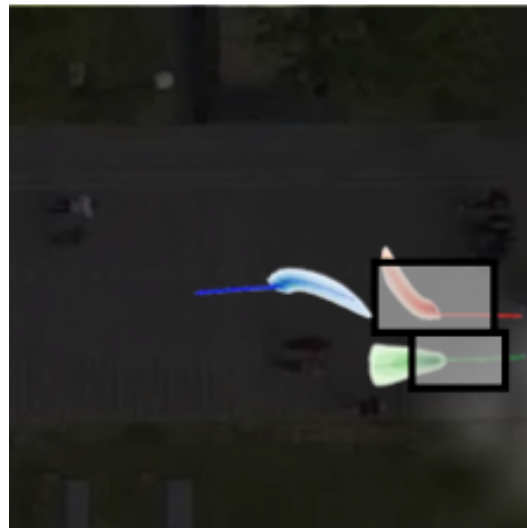
[Xu et. al. 15] Show, Attend and Tell 60

# SoPhie Results

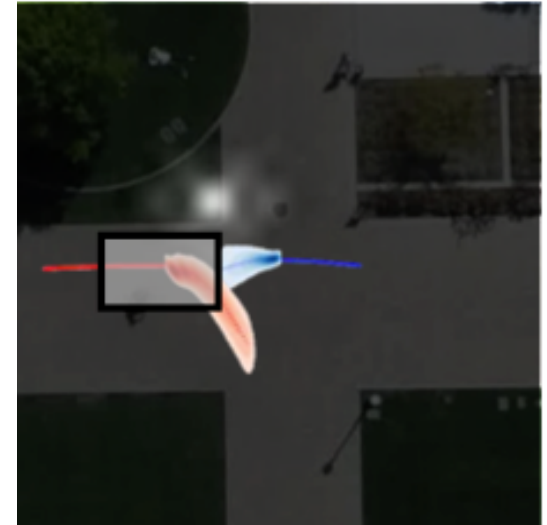
Scene  
Interaction



Social  
Interaction

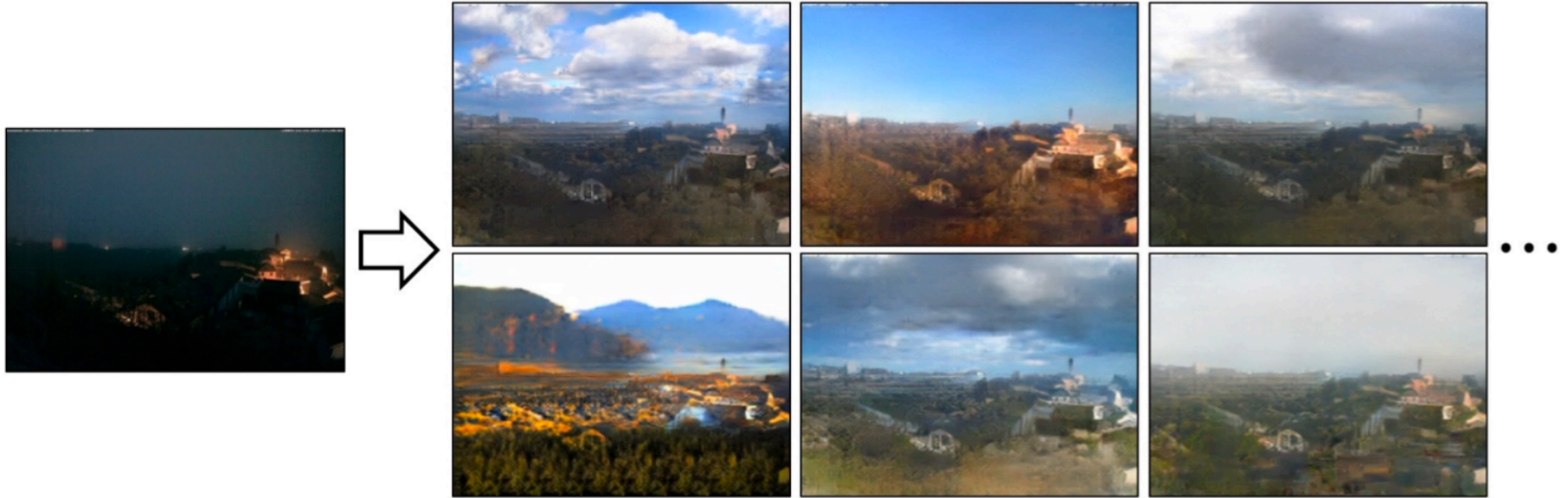


Scene + Social  
Interaction



# Multimodal Image-to-Image Translation

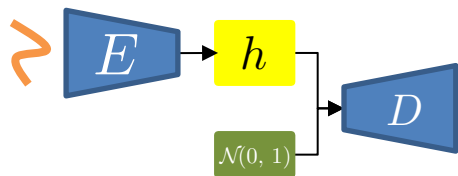
- Multimodality in image-to-image translation



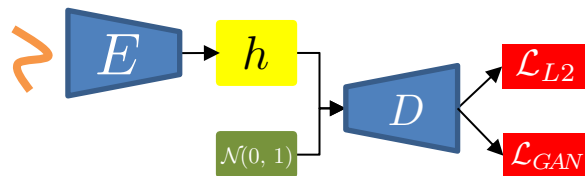
➡ Bicycle GAN training

# Bicycle GAN

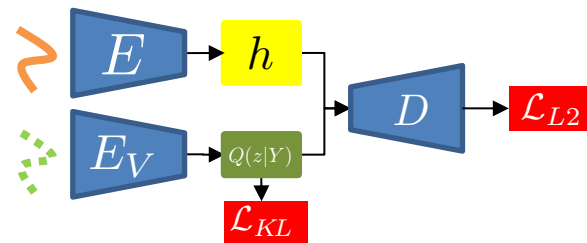
**0** Testing mode for all models



**1** GAN Baseline



**2** Conditional Variational Autoencoder (cVAE)



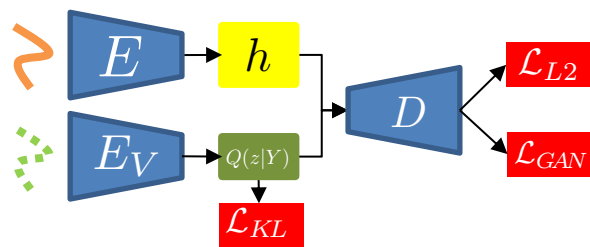
■ Loss

■ Neural Network

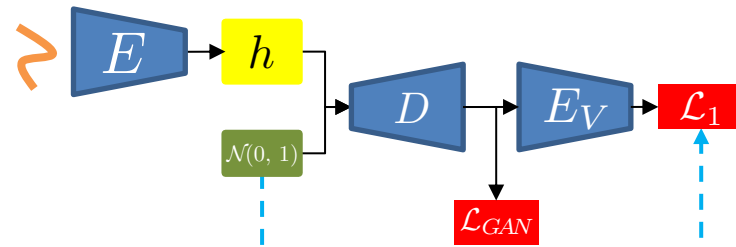
~ Input Trajectory

~ Ground Truth Trajectory

**3** Conditional Variational Autoencoder GAN (cVAE-GAN)



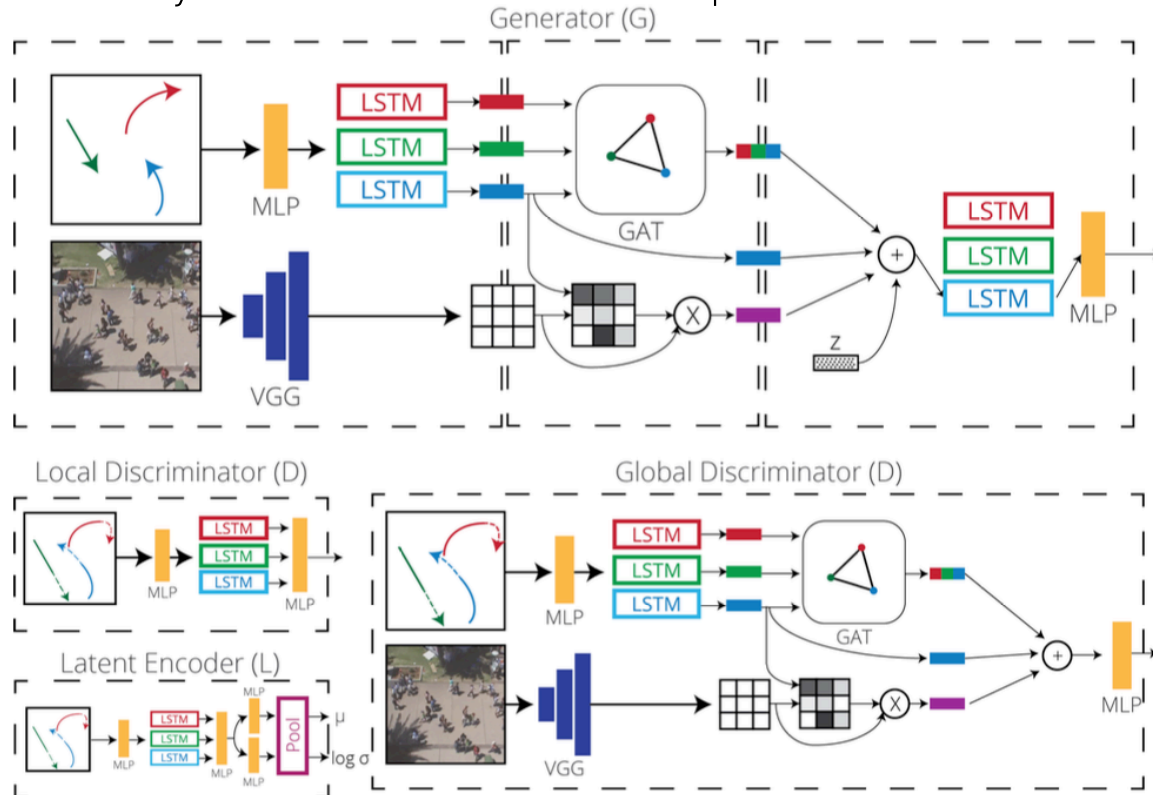
**4** Conditional Latent Regressor GAN (cLR-GAN)



**5** Bicycle Training

# Social BiGAT

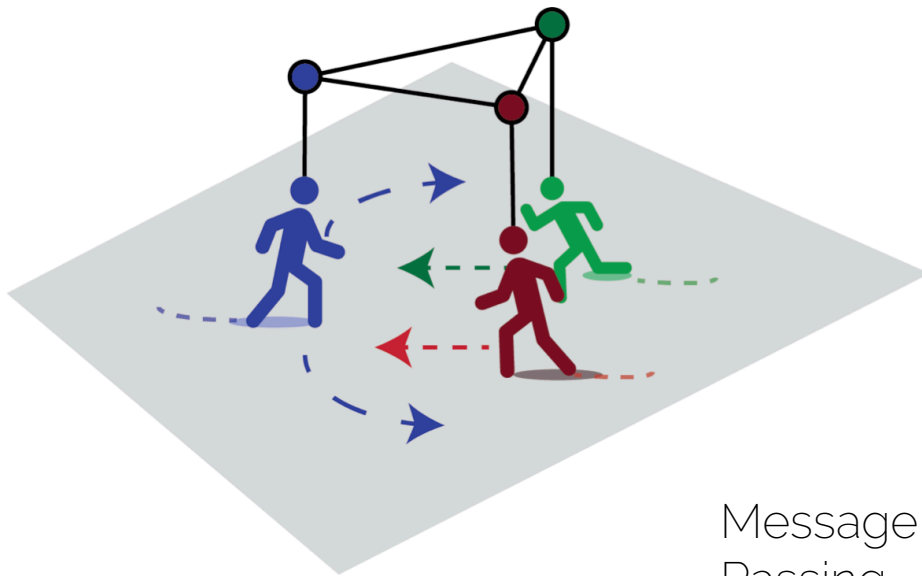
Contribution: Bicycle GAN (Bi) + Graph Attention Network (GAT)





# Social BiGAT: Graph Attention Network

- Modeling social interactions with graph attention network



Attention network

Encoder features

$$e_{ij} = a \left( W_{gat} V_s(i), W_{gat} V_s(j) \right)$$

Parameters

$$a_{ij} = \text{Softmax}_j(e_{ij})$$

Message Passing Steps 1

$$e_s^l(i) = \sum_j \alpha_{ij} W_{gat} V_s(j)$$

# Summary

- Pedestrian Trajectory Prediction is relevant for numerous problems, e.g. autonomous driving and tracking
- Multimodal nature of trajectories requires generative models (VAE and GAN)
- Most methods use LSTM encoder-decoder architecture
- Interactions are modelled with attention modules or graph networks

Thank you for your  
attention!

Interested in working on Pedestrian Trajectory Prediction?  
Contact me for MA or guided research: [Patrick.dendorfer@tum.de](mailto:Patrick.dendorfer@tum.de)