# Pedestrian Trajectory Prediction

# Overview

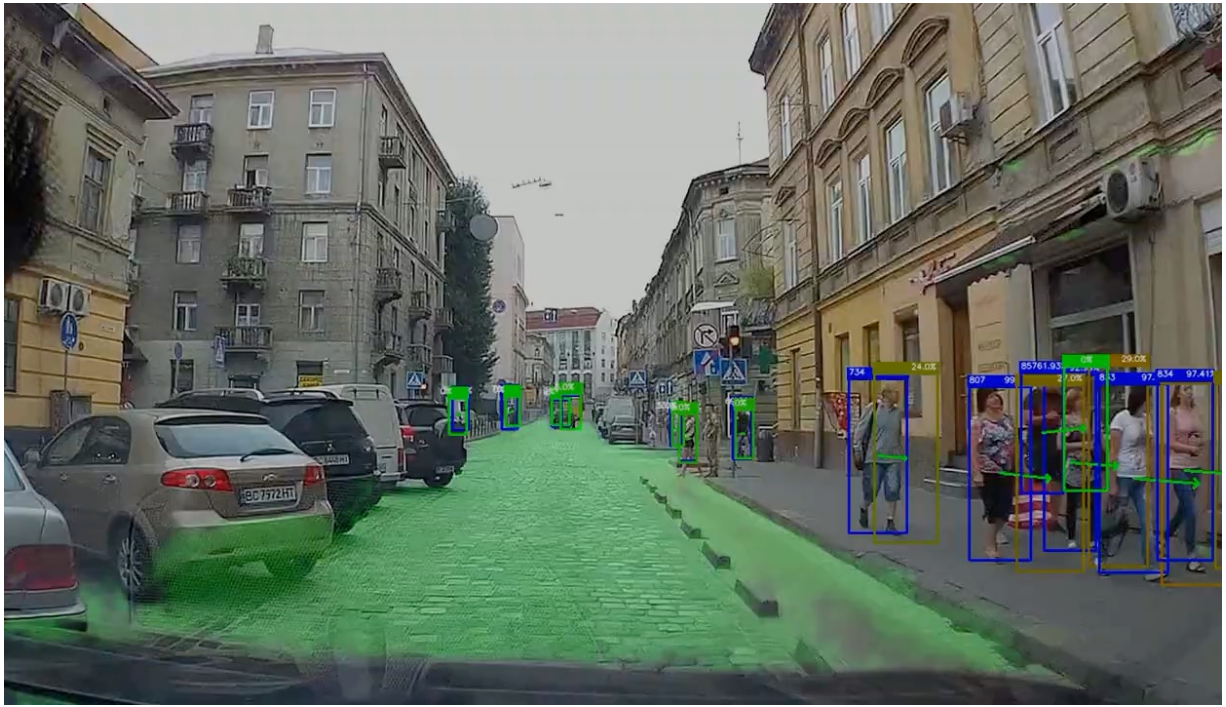
- What is pedestrian trajectory prediction good for?
- What is the task of pedestrian trajectory prediction?
- What makes trajectory prediction challenging?
- What kind models are used for trajectory prediction?
  - Deterministic models
  - Stochastic Models
- How to model social interactions?
- How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

# Introduction

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

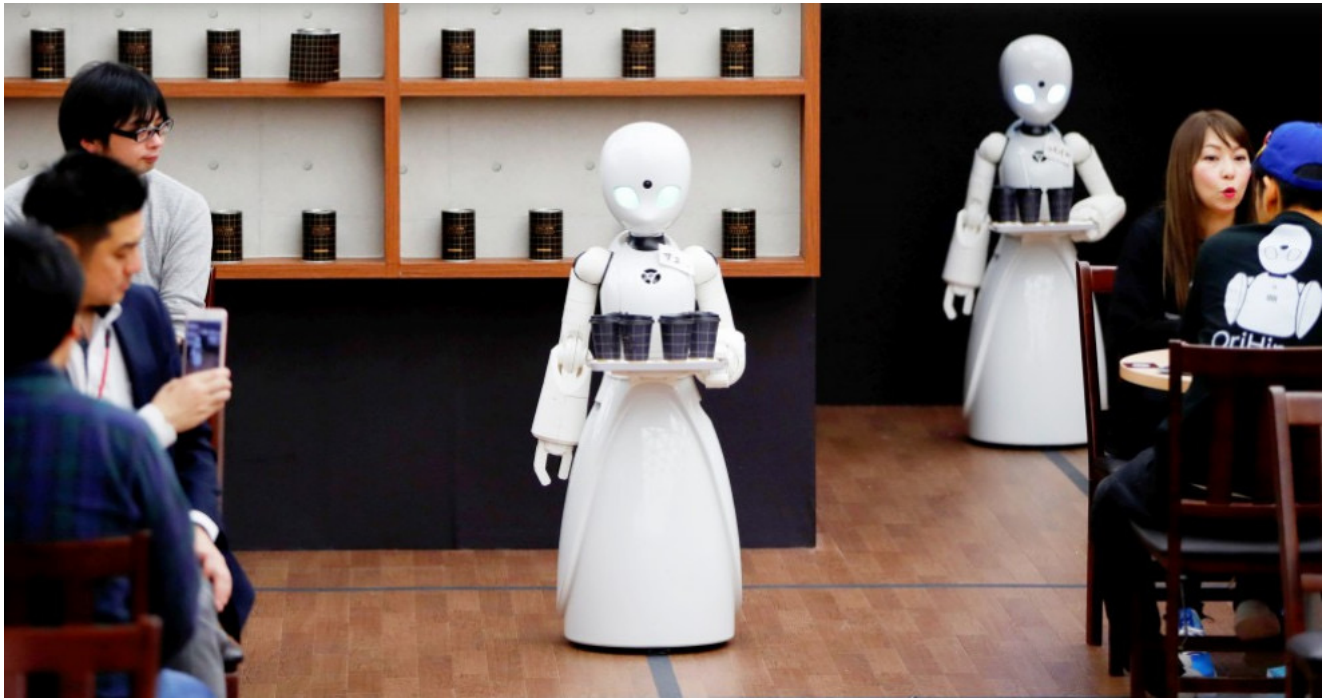
# Introduction

- 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



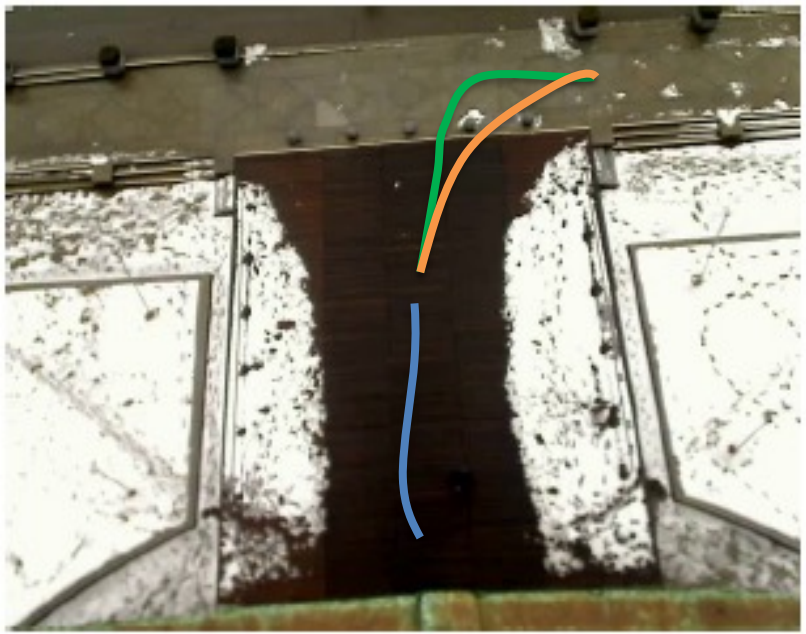
# Introduction

- Motion models improve the performance of Multi-Object Trackers

www.motchallenge.net

# 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 $t = T^{Obs+1}, \dots, T^{Pred}$

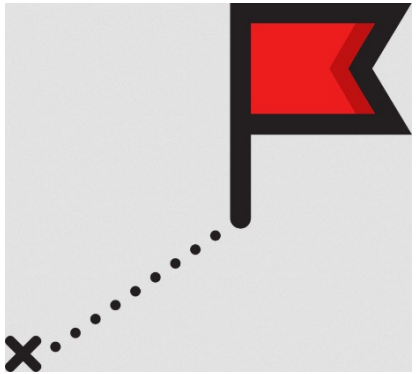
- Prediction:

$\hat{Y}_i^t = f_W(X_i) = (x_i^t, y_i^t)$

for $t = T^{Obs+1}, \dots, T^{Pred}$

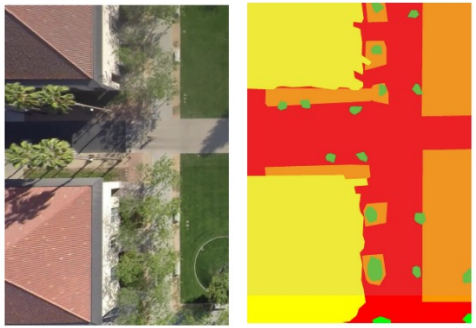
# Introduction

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

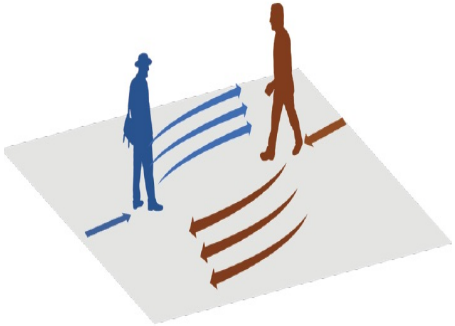


1. Destination

2. Personal Preferences



3. Human – Space Interactions



4. Human – Human Interactions

# Social Force Model

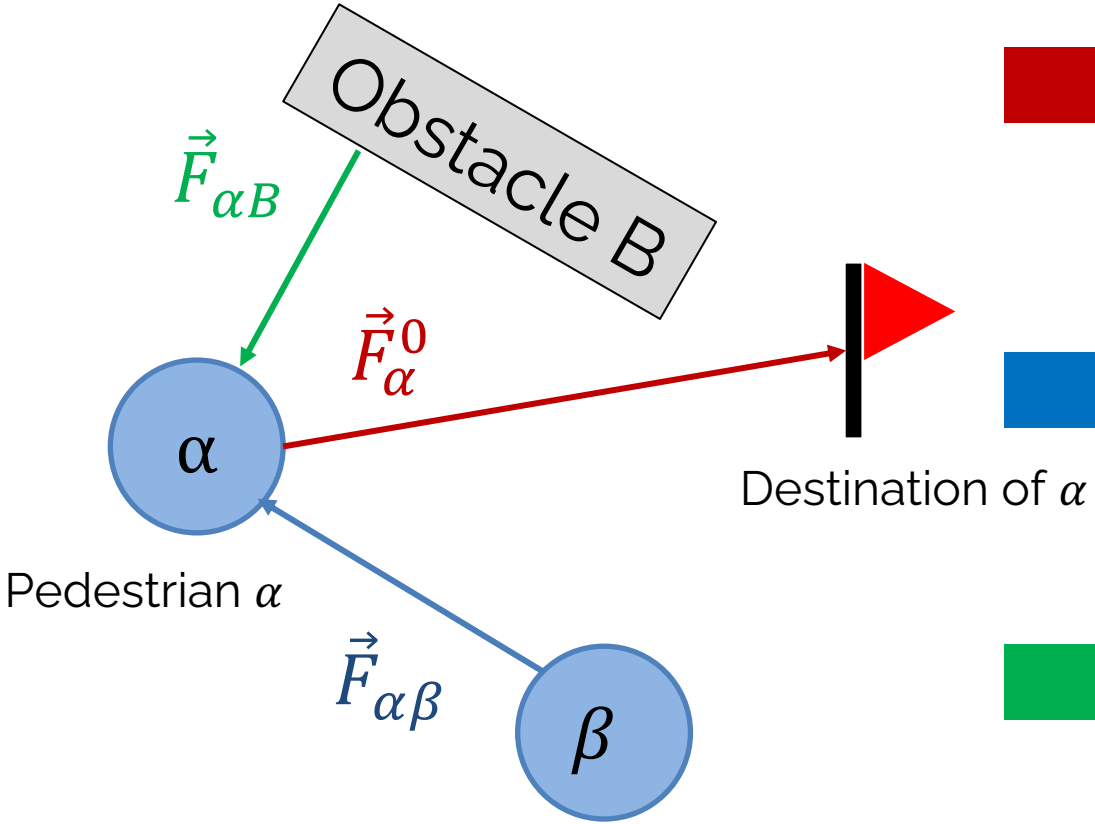
Contribution: Mathematical model of dynamics

- Idea: Pedestrian act in force field $\boldsymbol{F}$ like particles e.g. in an electric field
- Second Newton's law: $\ddot{\boldsymbol{x}} = \frac{d\boldsymbol{x}^2}{dt^2} = \frac{\boldsymbol{F}(t)}{m}$

- Trajectory $\mathbf{x}(t)$ is solution of differential equation

[Helbing et al. 98] *Social Force Model*

# 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 $\alpha$

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

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

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

# Social Force Model

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

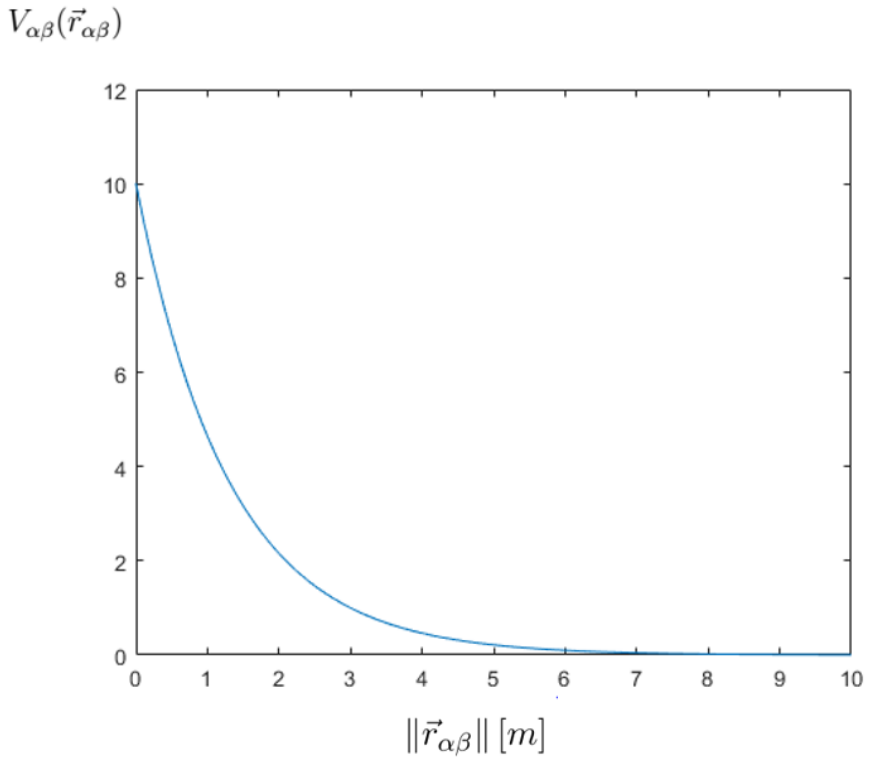
$$\vec{F}_{\alpha\beta}(\vec{r}_{\alpha\beta}) = -\nabla_{\vec{r}_{\alpha\beta}} V_{\alpha\beta}(\vec{r}_{\alpha\beta}) \text{ with } V_{\alpha\beta}(\vec{r}_{\alpha\beta}) = V^0 \, e^{-\frac{\|\vec{r}_{\alpha\beta}\|}{\sigma}}$$

- 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 $\sigma$:

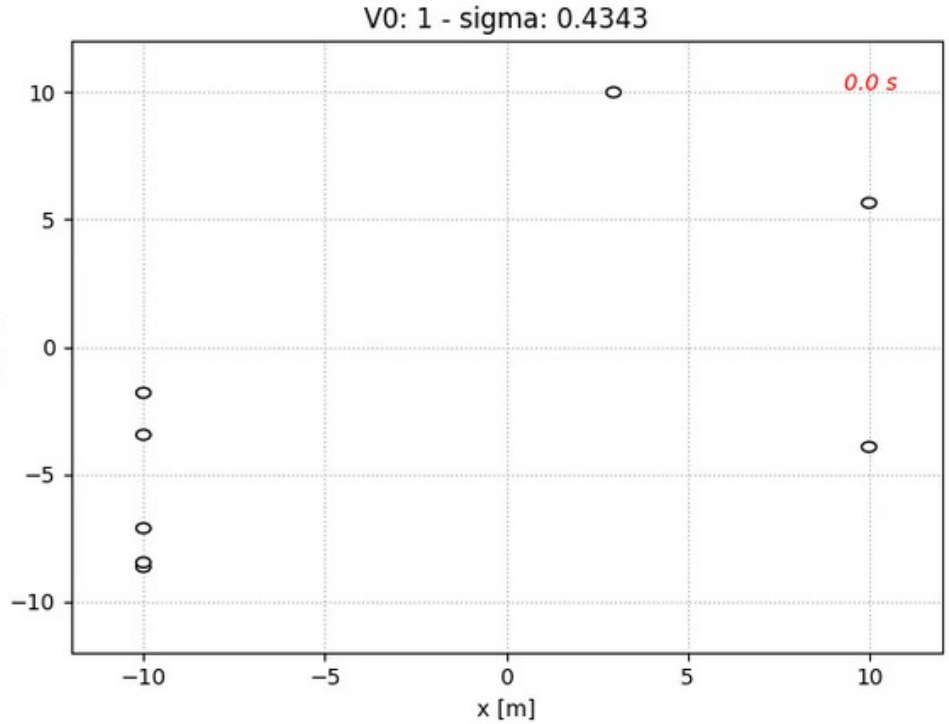
- Interaction Potential:
  - $V_{\alpha\beta}(\vec{r}_{\alpha\beta}) = V^0 \, e^{-\frac{\|\vec{r}_{\alpha\beta}\|}{\sigma}}$



# Social Force Model

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

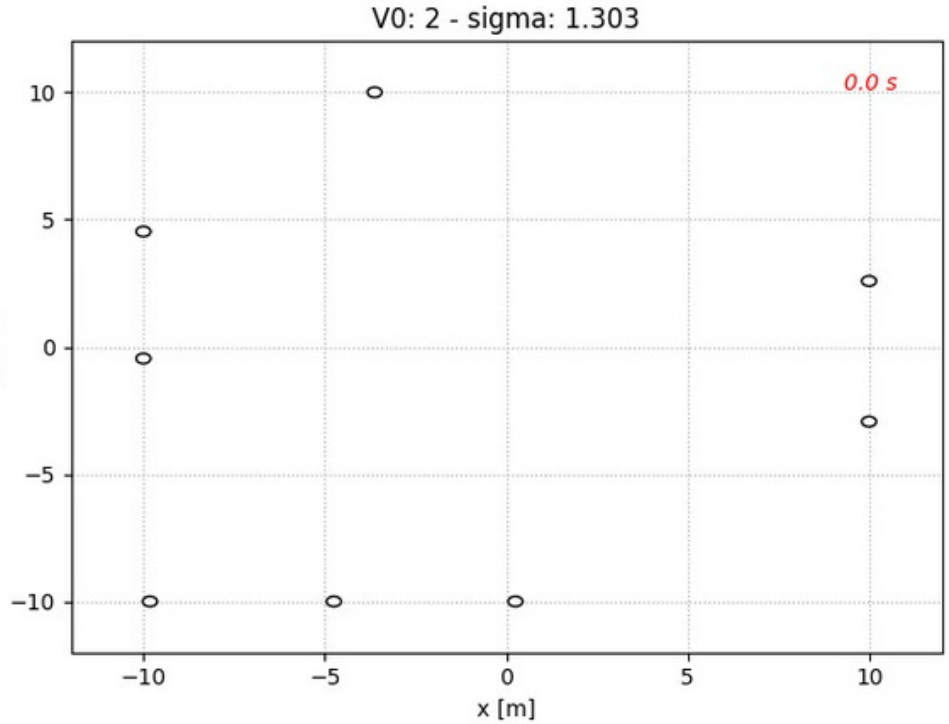
- Interaction Potential:
  - $V_{\alpha\beta}(\vec{r}_{\alpha\beta}) = V^0 \, e^{-\frac{\|\vec{r}_{\alpha\beta}\|}{\sigma}}$



# Social Force Model

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

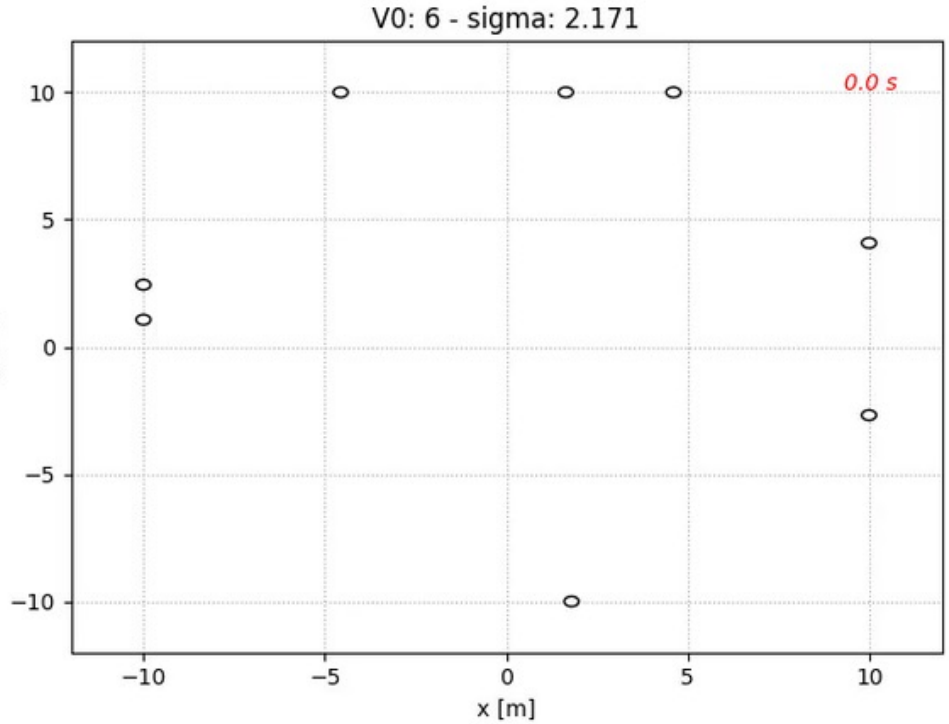
- Interaction Potential:
  - $V_{\alpha\beta}(\vec{r}_{\alpha\beta}) = V^0 \, e^{-\frac{\|\vec{r}_{\alpha\beta}\|}{\sigma}}$



# Social Force Model

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

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



# 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

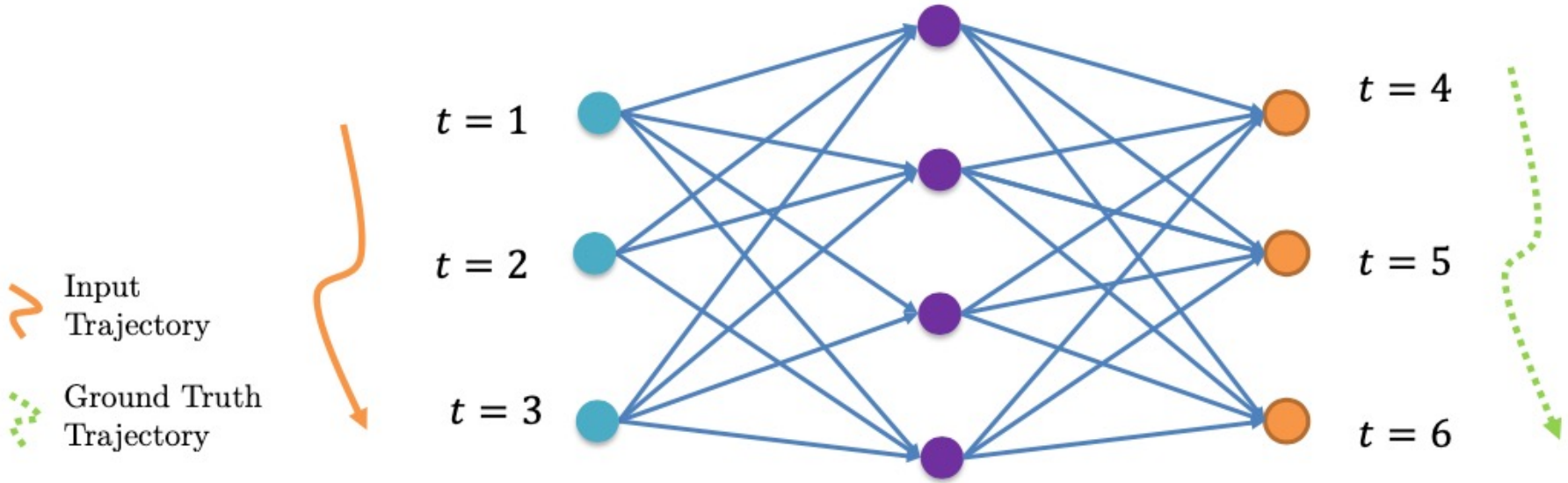
# Drawback of Social Force Model

Handcrafted Functions

Learned Functions

# Neural Network for Trajectory Prediction

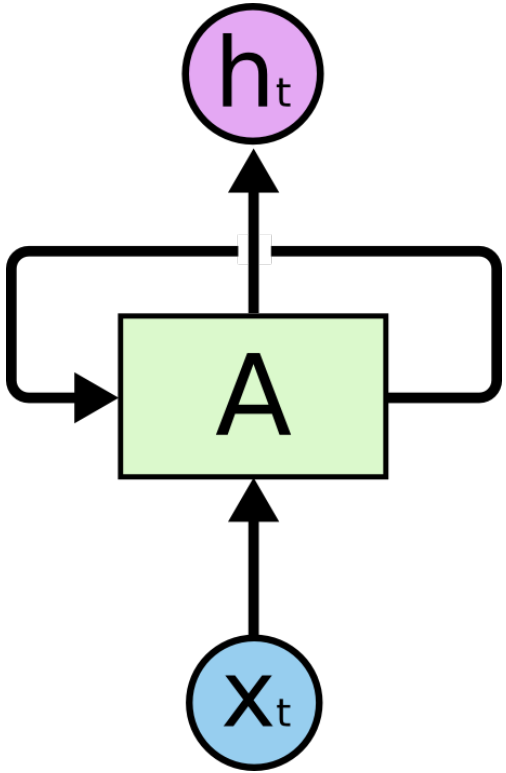
- Building a naïve prediction model with FC layers




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

⇒ Recurrent Neural Networks

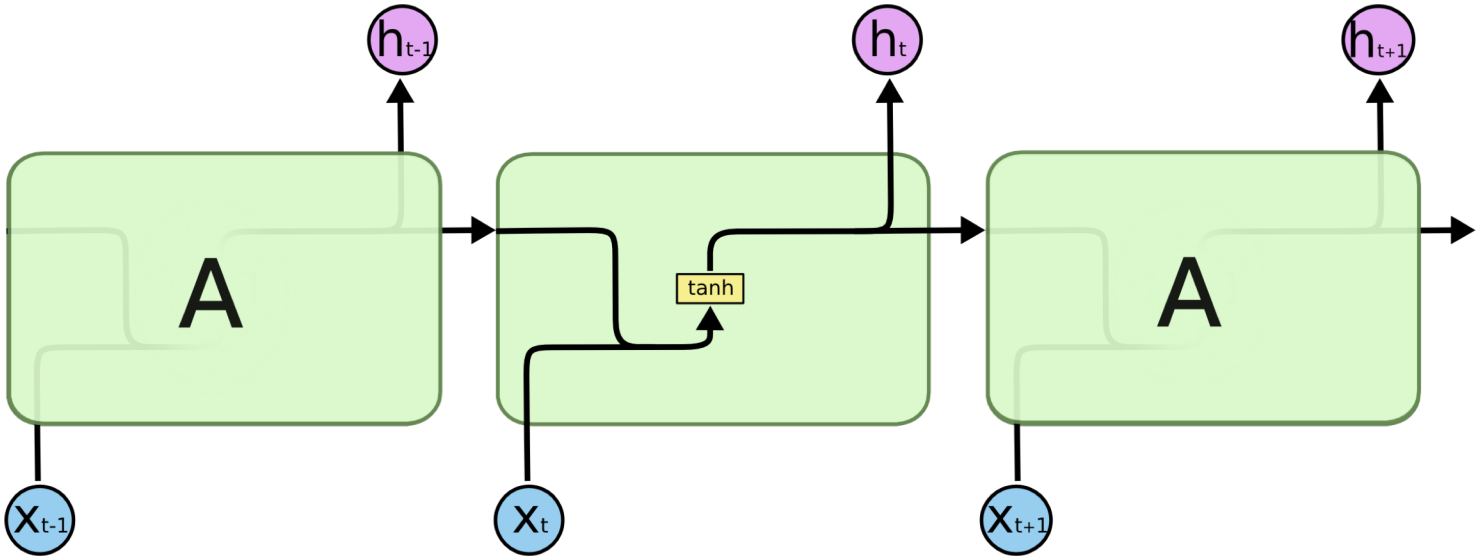
# Recurrent Neural Networks



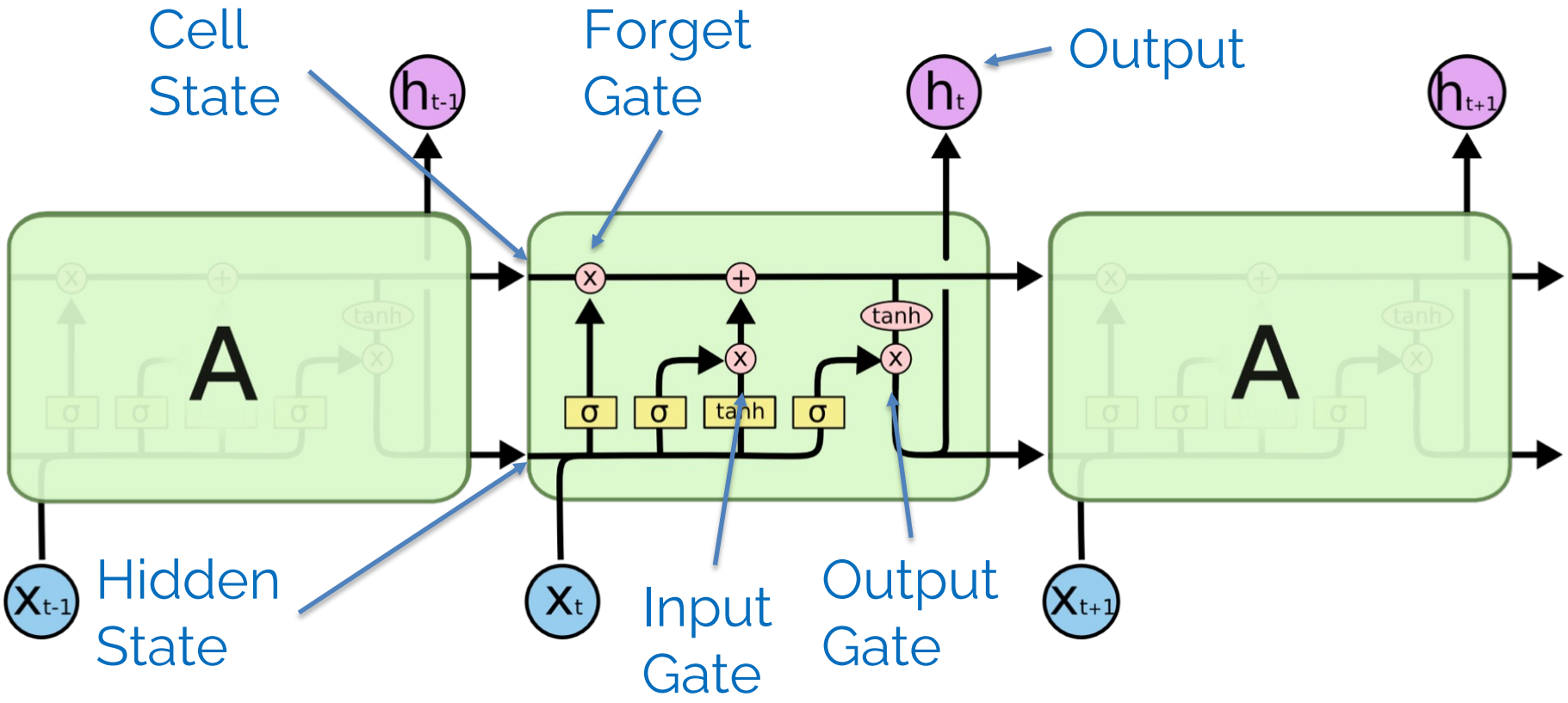

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

# Simple RNN

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

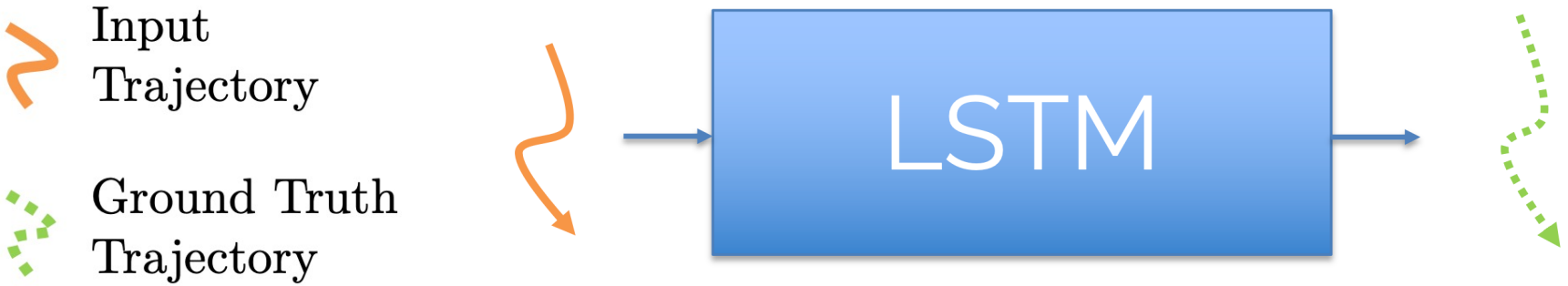


# LSTM (Long Short-term Memory)




[Hochreiter et al. 97] *Long Short-term Memory*

# LSTM for Trajectory Prediction



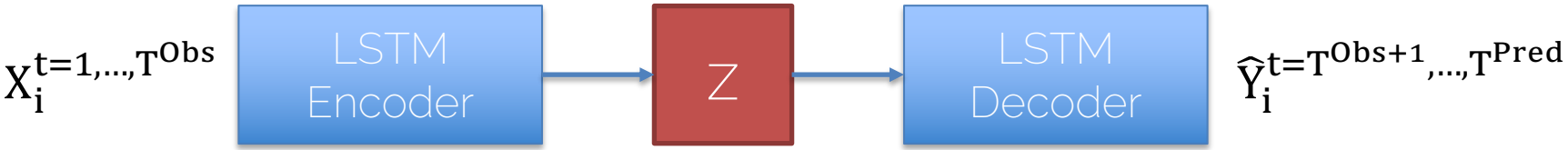

- Using a sinlge LSTM does not work well

Encoder-Decoder Architecture

# LSTM Encoder –Decoder Architecture

Input

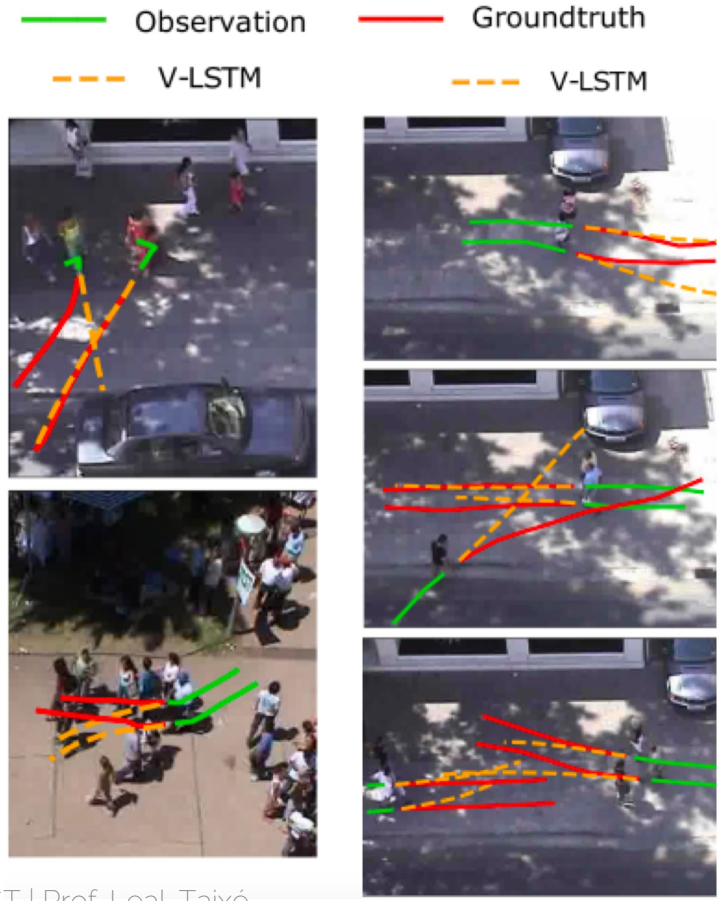
Output




Compressed (latent) representation

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

# Vanilla LSTM



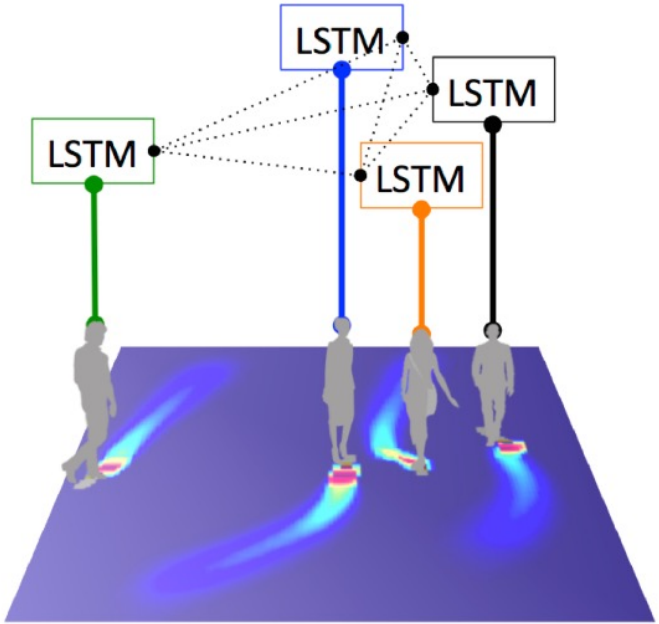

- Vanilla LSTM is not able to predict trajectories of interacting pedestrians

[Zhang et al. 19] SR-LSTM

# Social LSTM

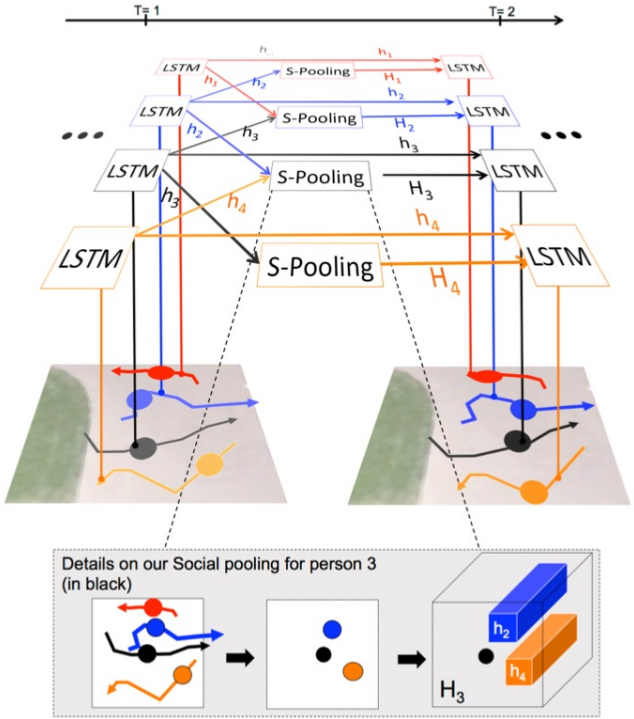
## Contribution: Modeling social interactions

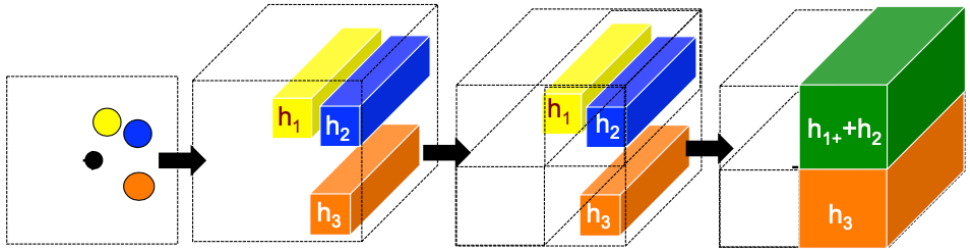
- LSTM encoder-decoder architectures
- Social pooling between neighboring pedestrians in each time step




[Alahi et al. 16] Social LSTM

# Social LSTM – Pooling Module



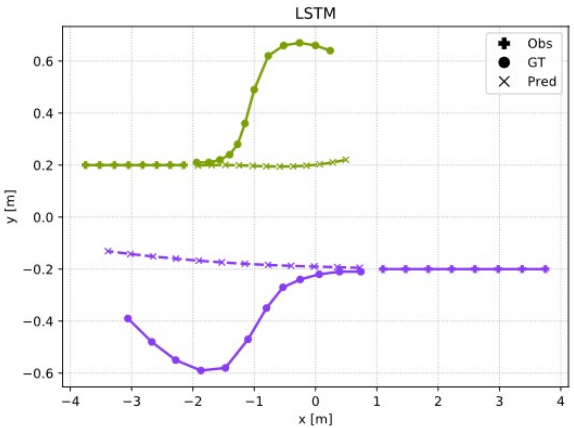



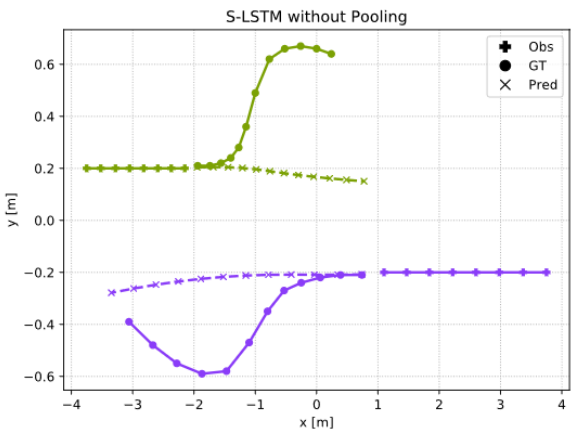

- Interaction Module pools hidden states of LSTM of pedestrian in vicinity
- Pooled hidden states are passed to decoder for next step prediction

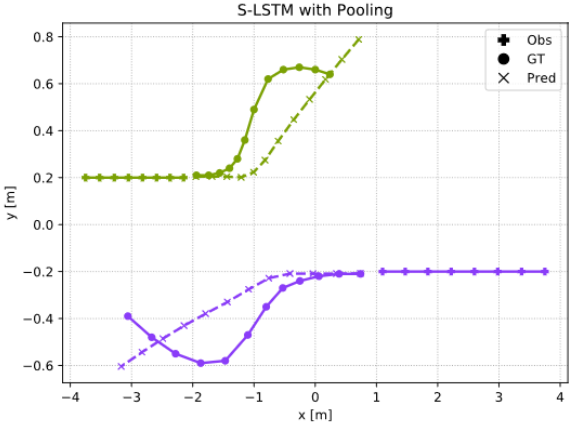
[Alahi et al. 16] Social LSTM

# Social LSTM – Result

Comparison of Models with and without social pooling



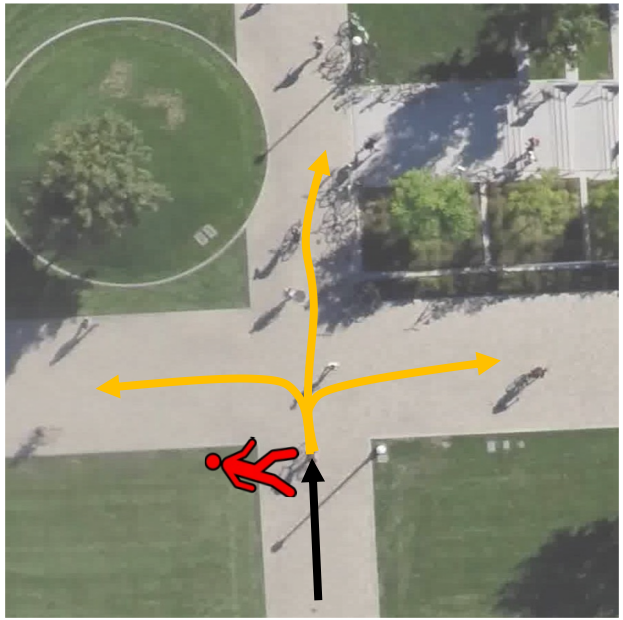




Social Pooling can resolve social interactions

Plots by Philipp Mondorf 20, BA

# Pedestrian Trajectories are multimodal

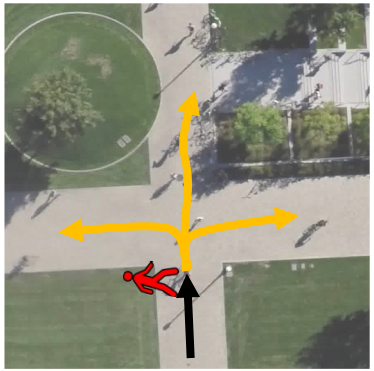


Same past trajectory can have multiple realistic future trajectories

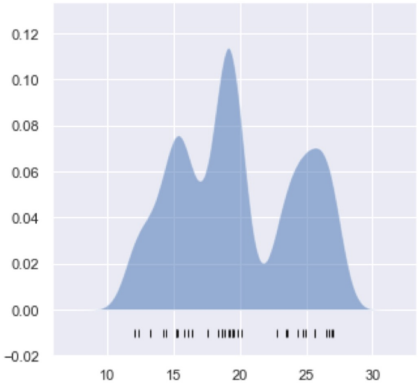
# Deterministic vs. Stochastic Models

| Deterministic | Stochastic |
| --- | --- |
| One-to-one mapping | One-to-many mapping |

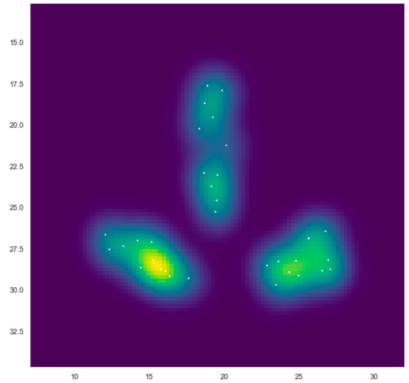
Learning distribution of future trajectories instead of deterministic mapping



Scenario




Distribution of final end



2d distribution of final end

# Towards Generative Models

Deterministic Models

Generative Models

# Recap: Generative Models

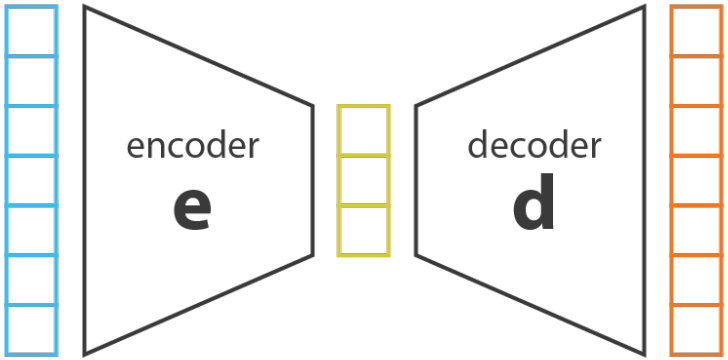
Explicit Density

1. Variational Autoencoder

Implicit Density

2. Generative Adversarial Network

# Variational Autoencoder




$x$ — **initial data** in space $R^n$

$e(x)$ — **encoded data** in latent space $R^m$ (with $m<n$)

$d(e(x))$ — **encoded-decoded data** back in the initial space $R^n$

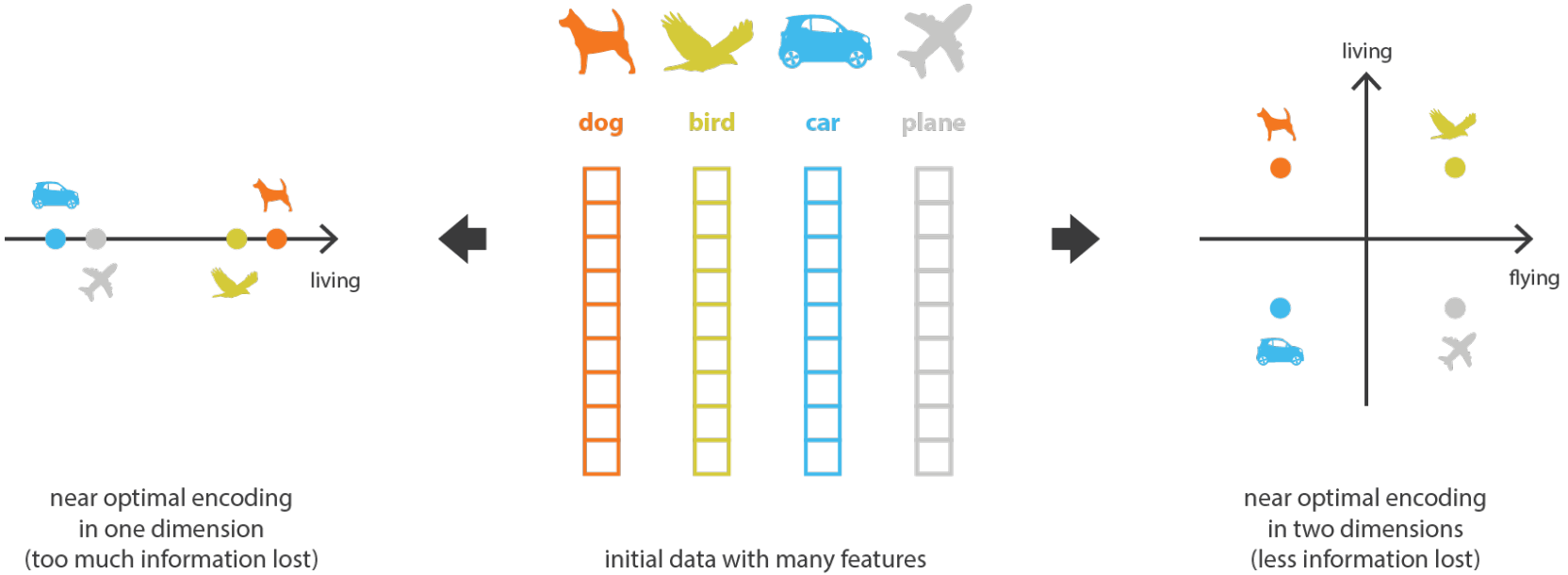
$x = d(e(x))$ ➡ **lossless encoding**
no information is lost when reducing the number of dimensions

$x \neq d(e(x))$ ➡ **lossy encoding**
some information is lost when reducing the number of dimensions and can't be recovered later

https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

[Kingma et al. 16] Variational Bayes

# Variational Autoencoder

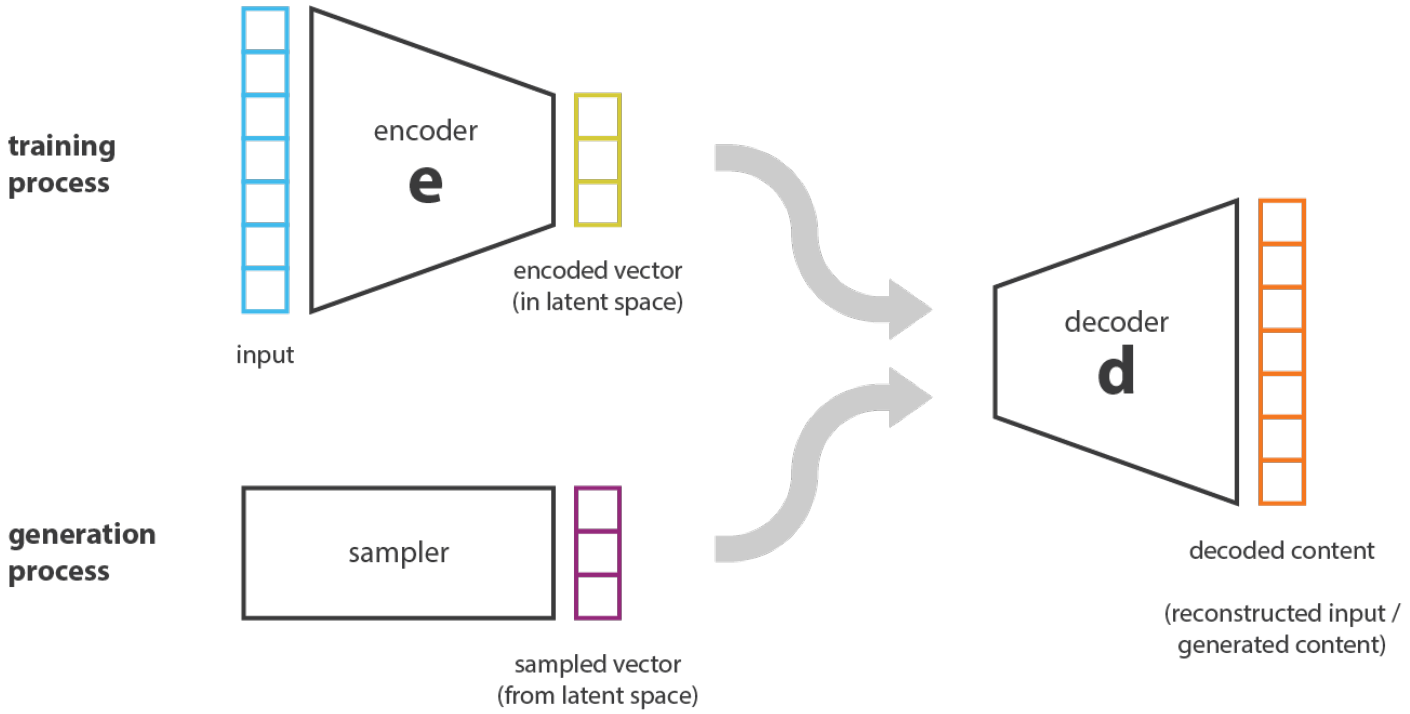





https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

[Kingma et al. 16] Variational Bayes

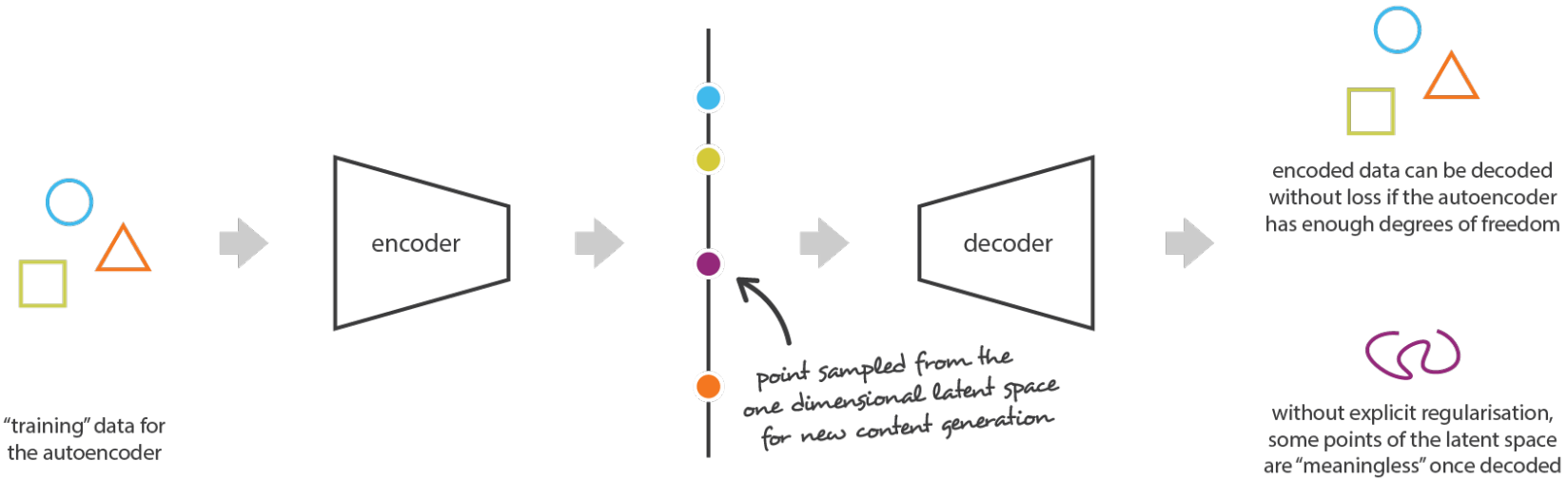
# Variational Autoencoder




https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

[Kingma et al. 16] Variational Bayes

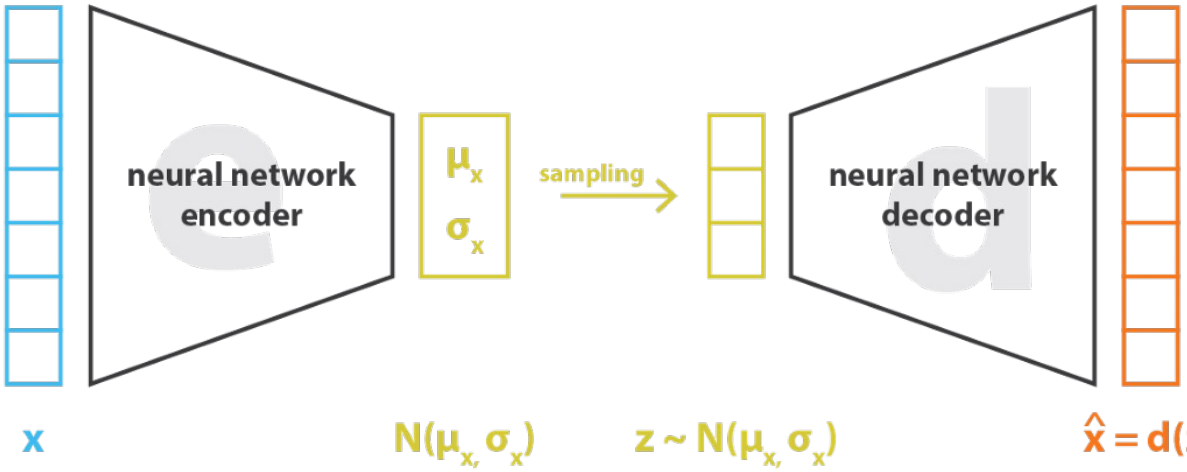
# Variational Autoencoder




https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

[Kingma et al. 16] Variational Bayes

# Variational Autoencoder



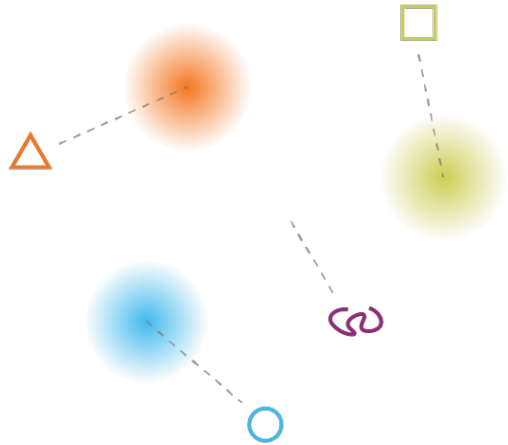

Kulback-Leibler (KL) divergence

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

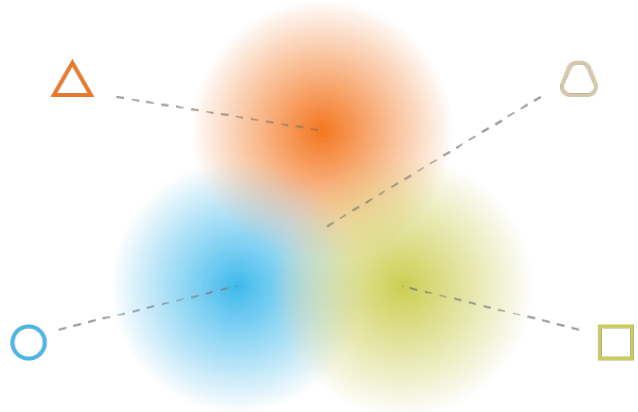
https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

[Kingma et al. 16] Variational Bayes

# Variational Autoencoder



what can happen without regularisation

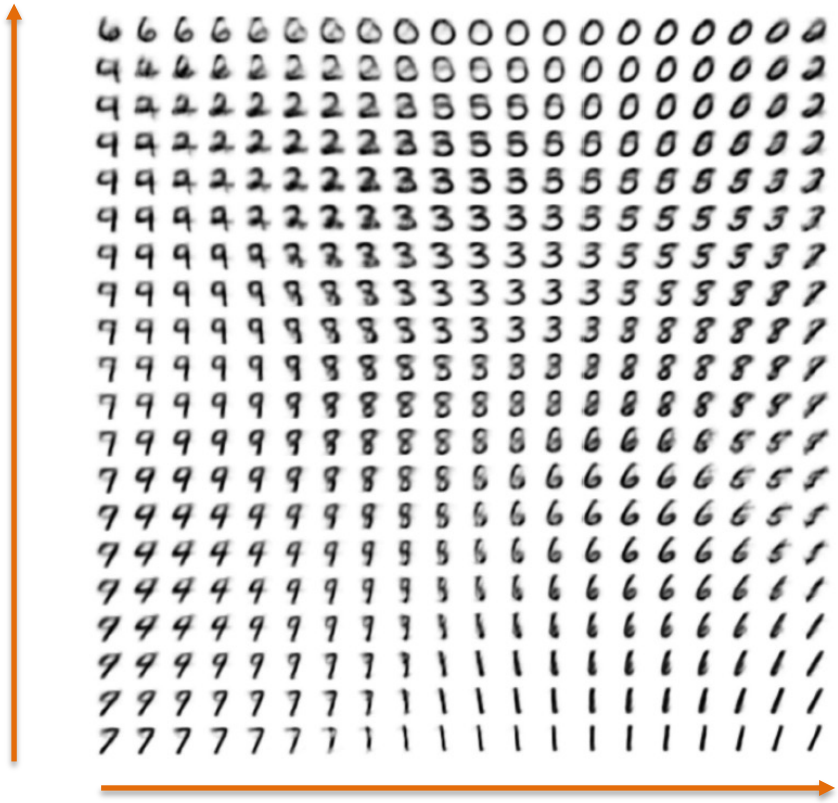


what we want to obtain with regularisation

https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

[Kingma et al. 16] Variational Bayes

# Variational Autoencoder



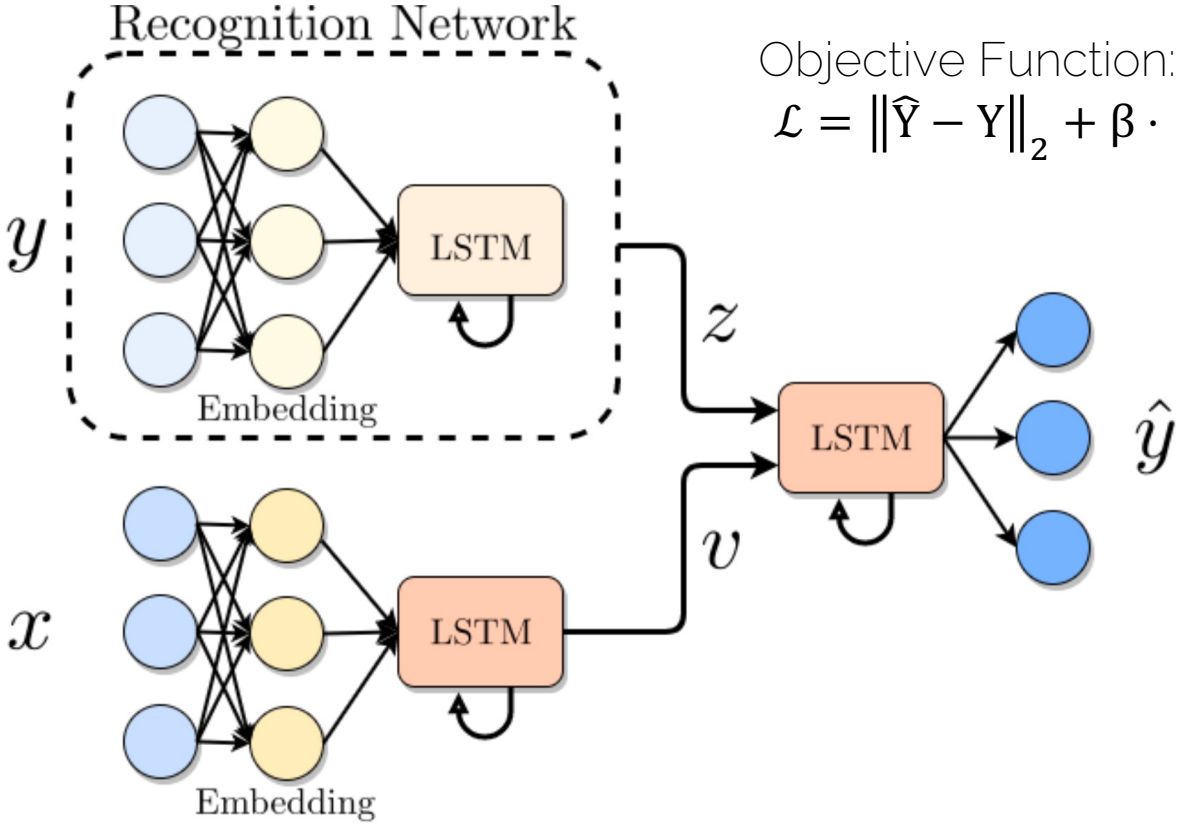
Each element of z encodes a different feature

# Variational Autoencoder

Degree of smile

Head pose

# Conditional Variational Autoencoder




Objective Function:

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

Kulback-Leibler (KL) divergence

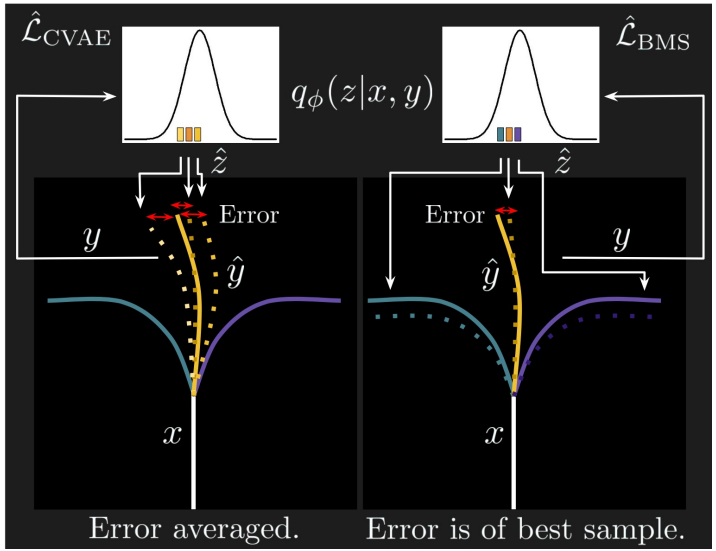
[Bhattacharyya et al. 18] Best of Many

# Best-of-many-sampling

- Sample multiple trajectories
- Backpropagate sample with minimal error

Before

Best of Many Sampling (BMS)




Objective Function:

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



[Bhattacharyya et al. 18] Best of Many

# Visual results cVAE

K-mean clustered trajectories

[Bhattacharyya et al. 18] Best of Many

# Recap: Generative Models

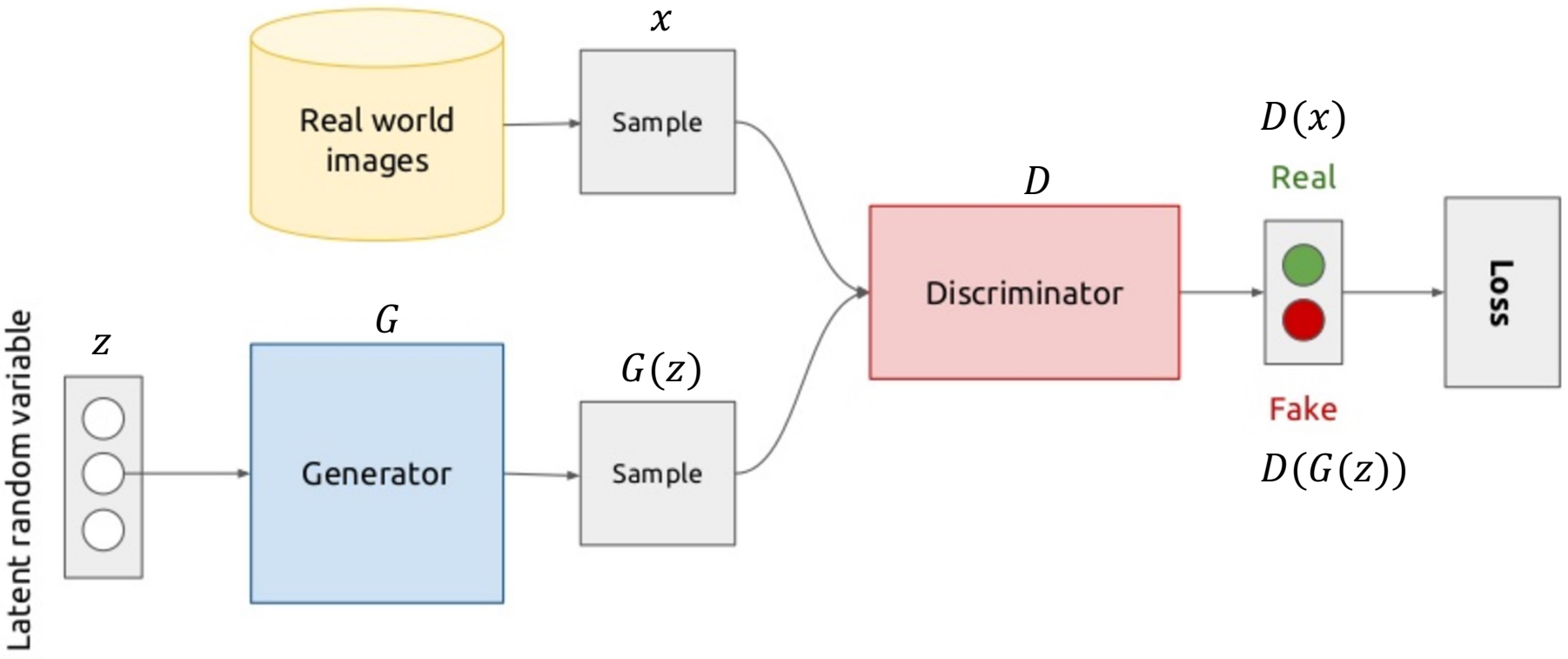
Explicit Density

1. Variational Autoencoder

Implicit Density

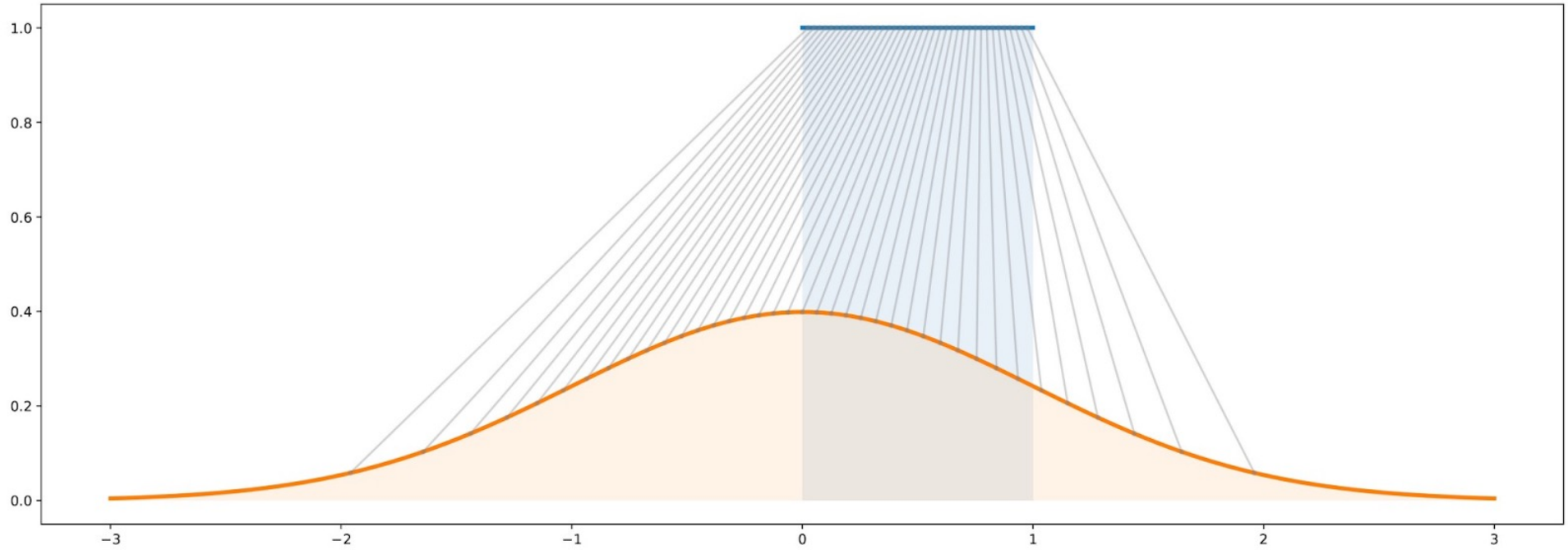
2. Generative Adversarial Network

# Generative Adversarial Networks (GANs)




[Goodfellow et al. 14] GANs (slide McGuinness)

# Generative Adversarial Networks (GANs)




- GAN maps known distribution to target distribution



https://towardsdatascience.com/understanding-generative-adversarial-networks-gans-cd6e4651a29

# GANs: Loss Functions

Discriminator loss

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

Generator loss

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

- Heuristic Method
  - G maximizes the log-probability of D being mistaken
  - G can still learn even when D rejects all generator samples

[Goodfellow et al. 14] GANs (slide McGuinness)

# Vanilla GAN

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

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

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

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

**end for**

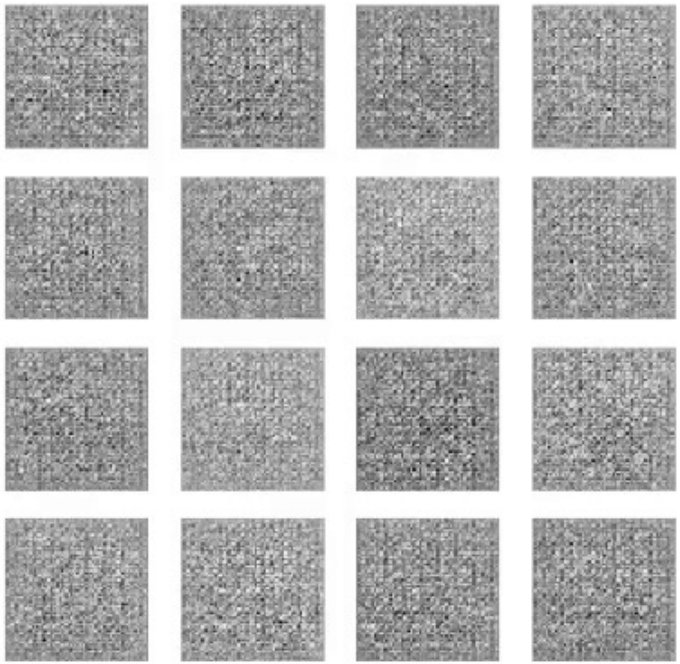
- Sample minibatch of $m$ noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

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

**end for**

[Goodfellow et al. 14] GANs (slide McGuinness)

# Training GAN

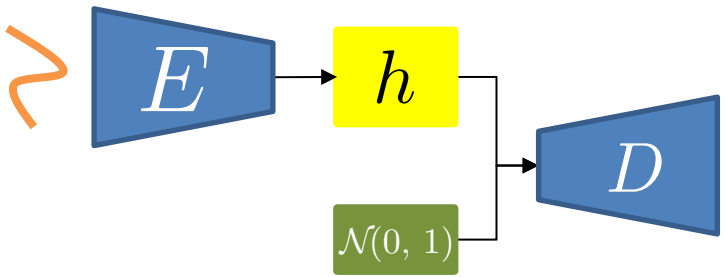


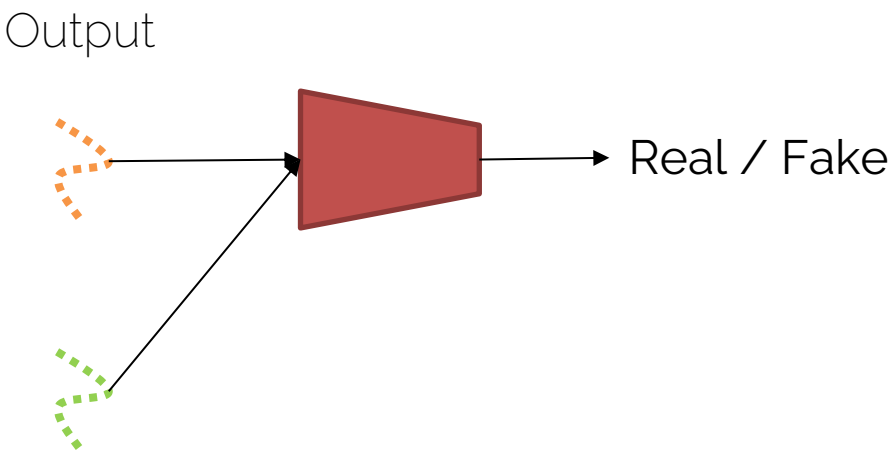
https://medium.com/ai-society/gans-from-scratch-1-a-deep-introduction-with-code-in-pytorch-and-tensorflow-cb03cdcdba0f

# GANs for Trajectory Prediction

Generator

Discriminator

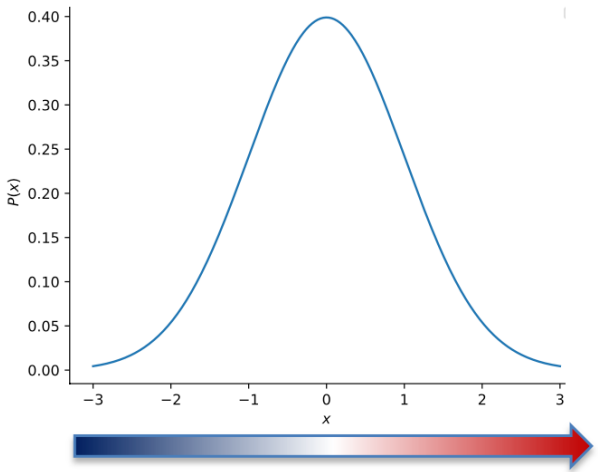


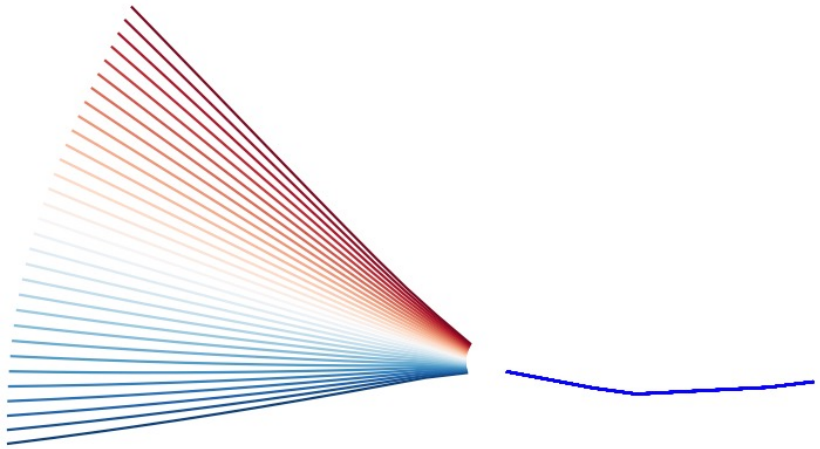
# Latent space in GANs

- Different latent space samples result into different real space output

Latent Space



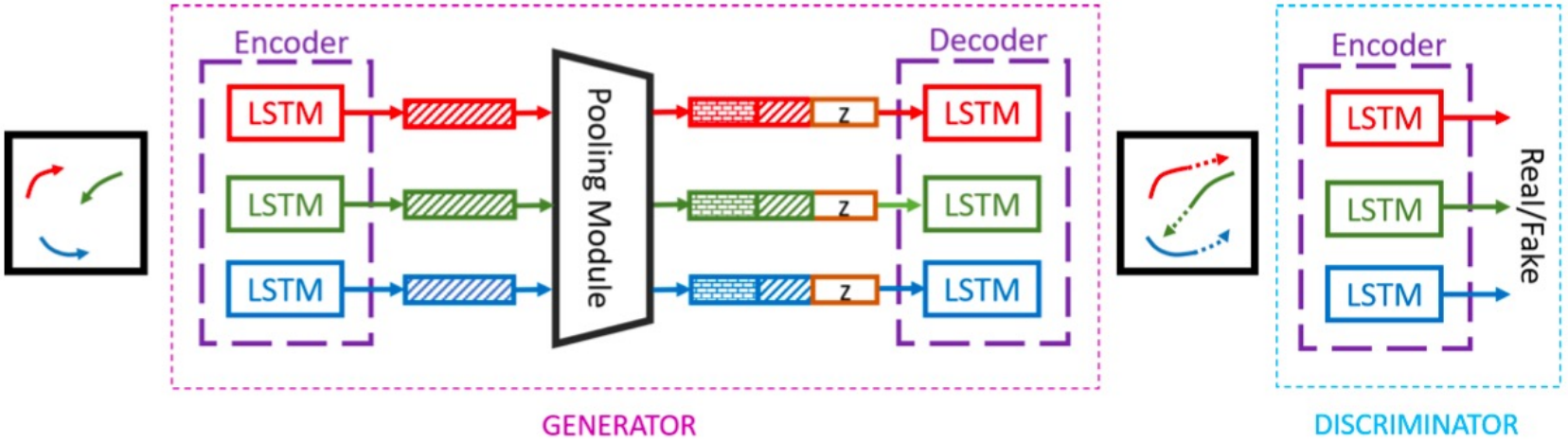

Real Space



[Gupta et al. 18] Social GAN

# Social GAN

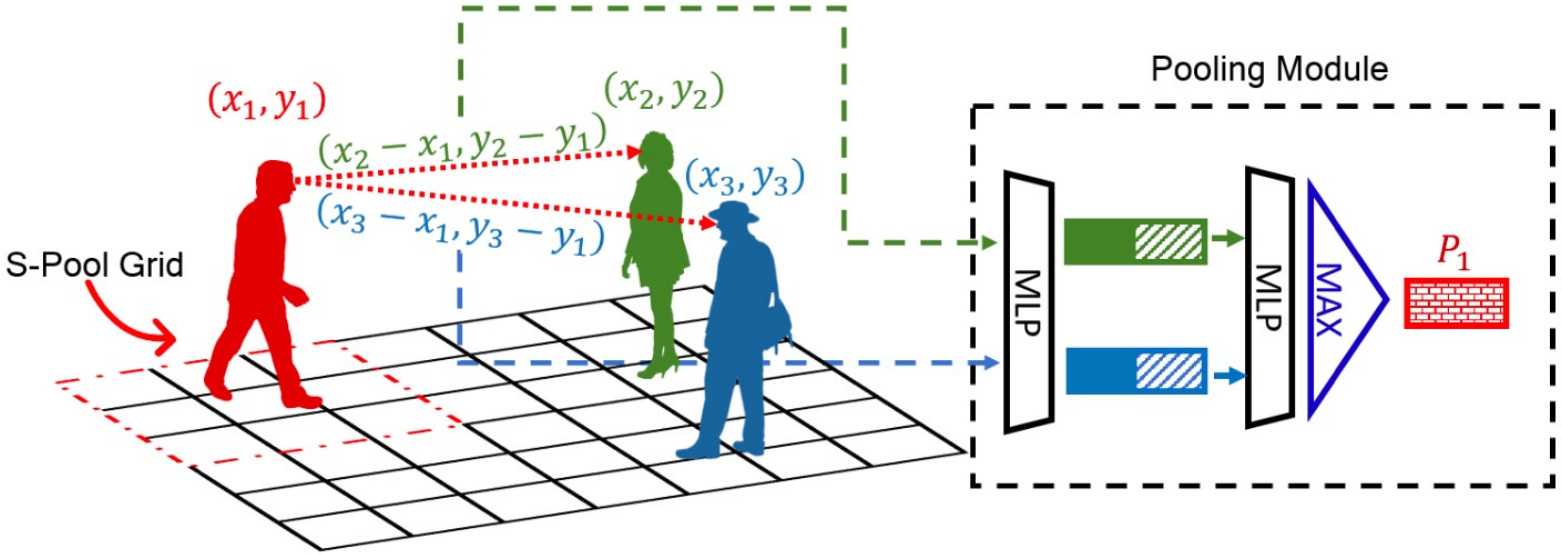
## Contribution: GAN + Social Interactions




[Gupta et al. 18] Social GAN

# SGAN Pooling Module

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




[Gupta et al. 18] Social GAN

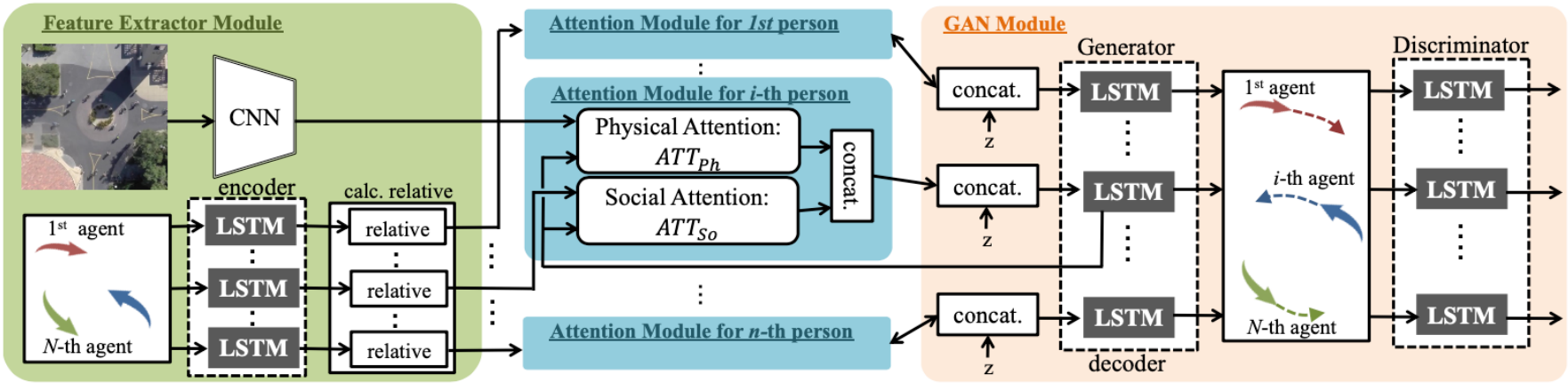
# SGAN Results


[Gupta et al. 18] Social GAN

# SoPhie

## Contribution: Combining physical and social interactions

- SoPhie uses soft attention istead of max-pooling

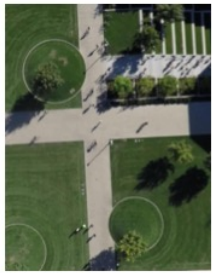


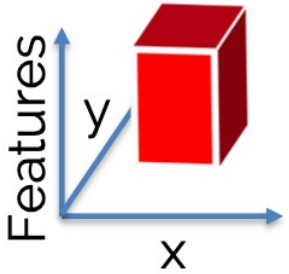


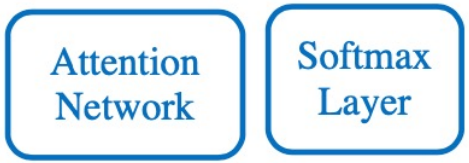

[Sadeghian et al. 19] SoPhie

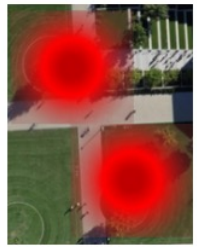
# Visual Attention

- Spatial Attention selects relevant features for each region in scene



- Attention values weight visual features for each spatial position

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

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

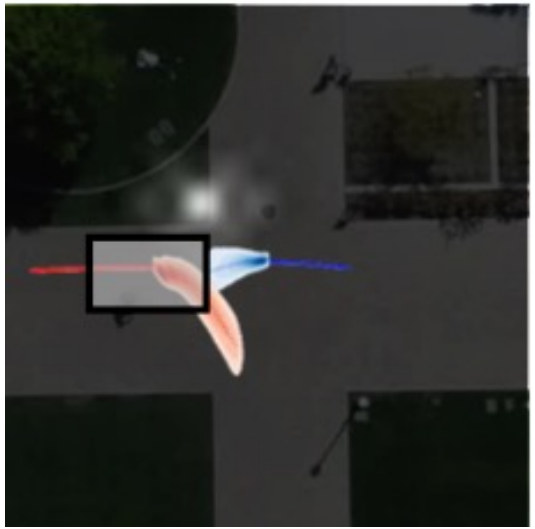
[Sadeghian et al. 18] SoPhie
[Xu et. al. 15] Show, Attend and Tell

# SoPhie Results

Scene Interaction

Social Interaction

Scene + Social Interaction

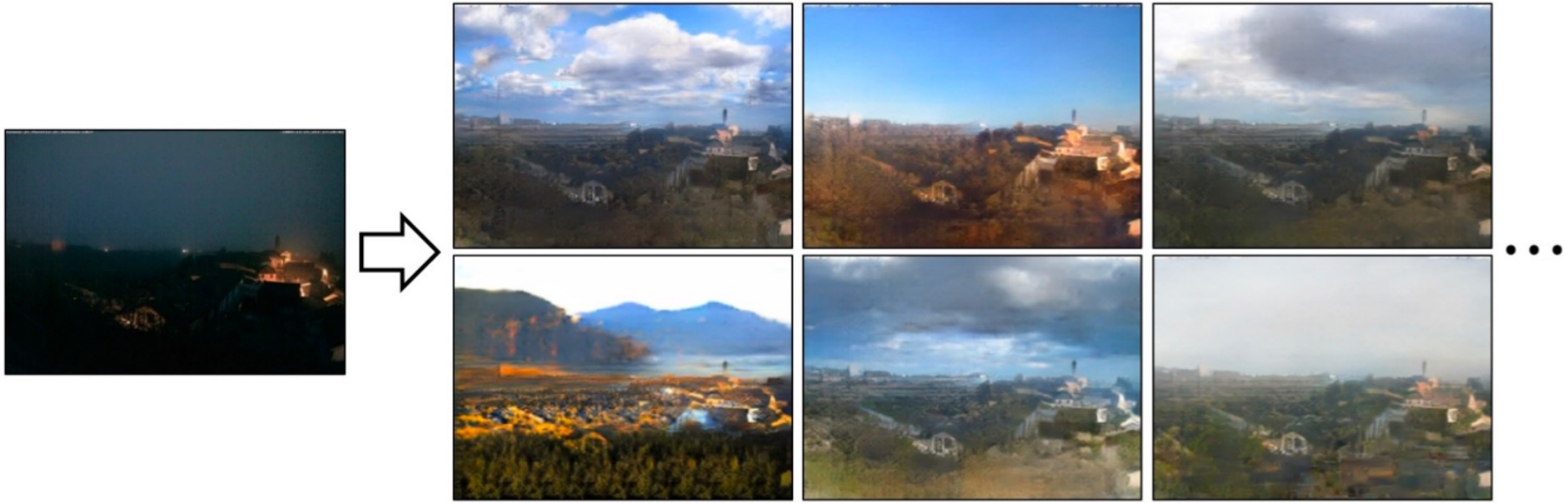




[Sadeghian et al. 19] SoPhie

# Multimodal Image-to-Image Translation

- Multimodality in image-to-image translation

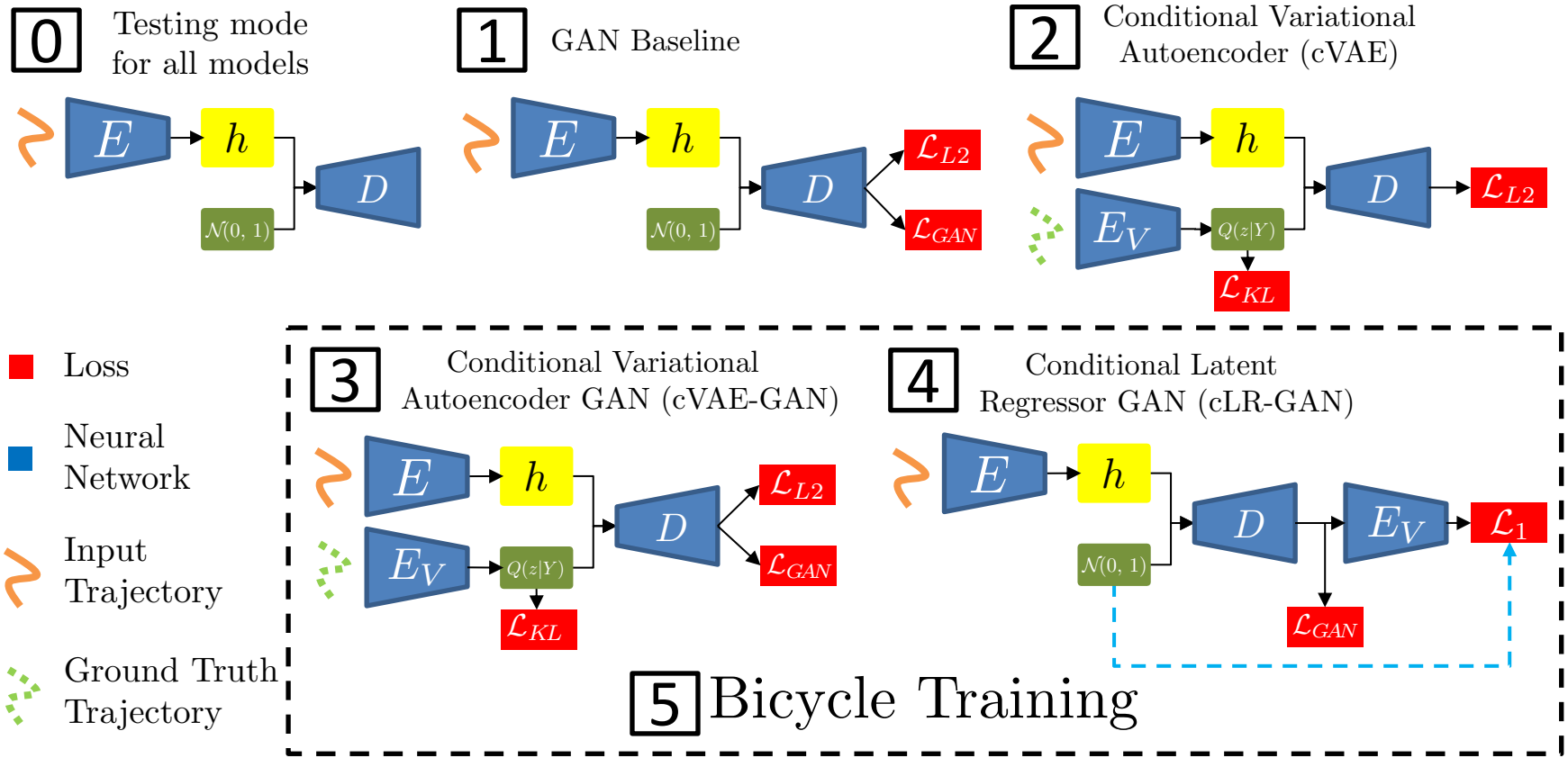


Bicycle GAN training



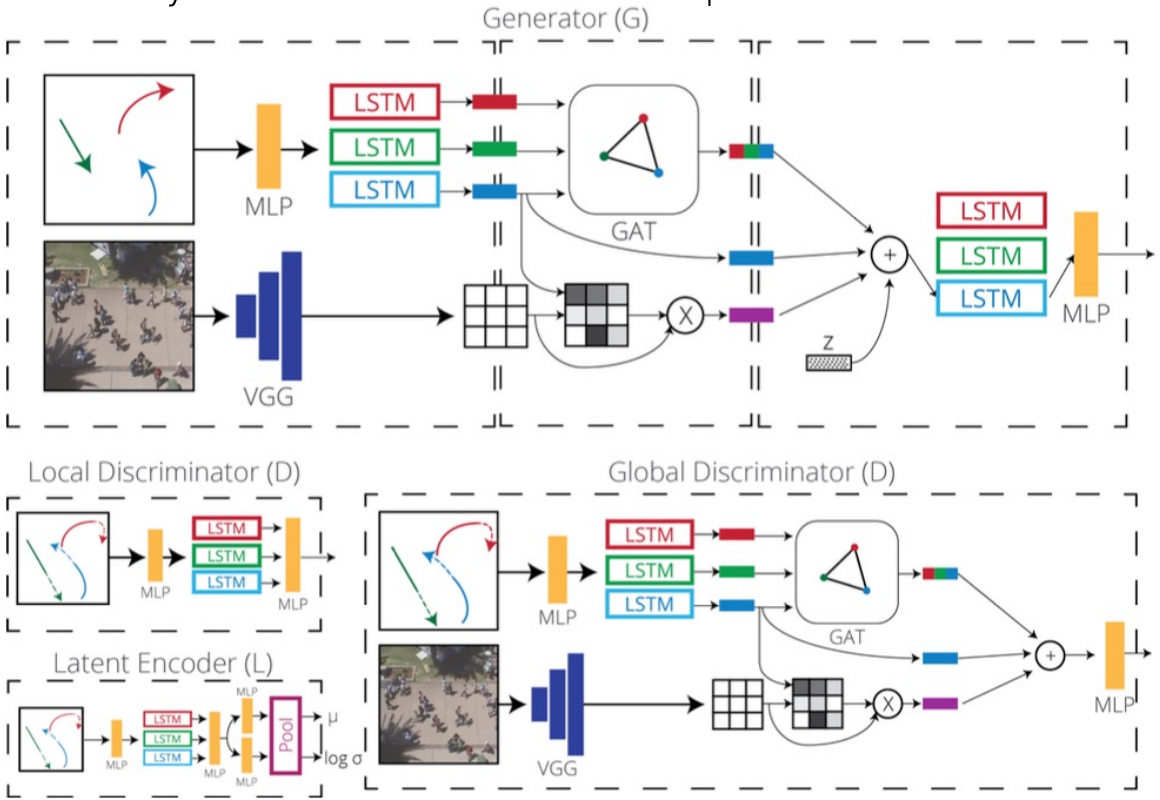
[Zhu et al. 17] Image-to-image translation

# Bicycle GAN


0
Testing mode for all models
E
h
D
$\mathcal{N}(0, 1)$
1
GAN Baseline
$\mathcal{L}_{L2}$
$\mathcal{L}_{GAN}$
2
Conditional Variational Autoencoder (cVAE)
$E_V$
$Q(z|Y)$
$\mathcal{L}_{KL}$
Loss
Neural Network
Input Trajectory
Ground Truth Trajectory
3
Conditional Variational Autoencoder GAN (cVAE-GAN)
4
Conditional Latent Regressor GAN (cLR-GAN)
$\mathcal{L}_1$
5
Bicycle Training

# Social BiGAT

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

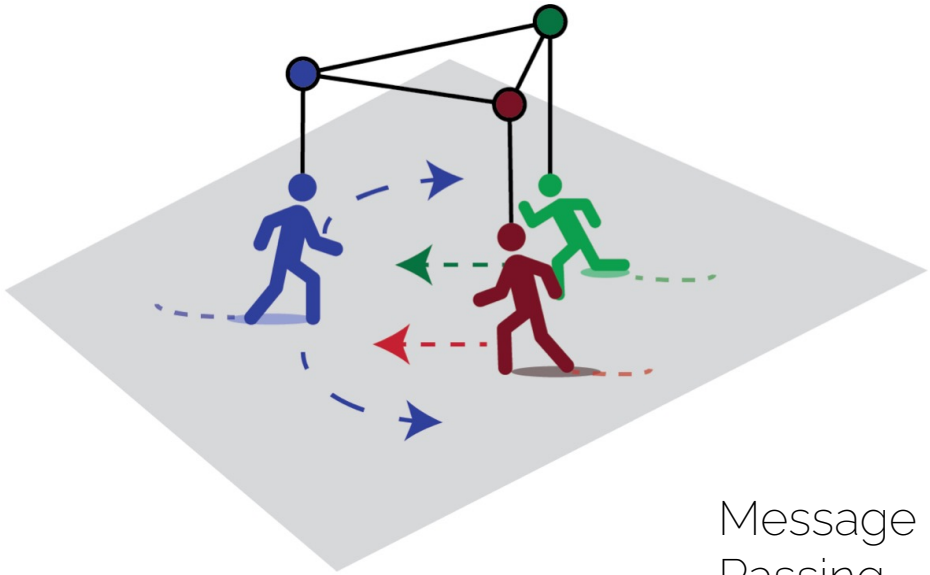





[Kosaraju et al. 19] Social BiGAT

# 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

$$a_{ij} = Softmax_j(e_{ij})$$

Message Passing Steps $\mathbf{l}$

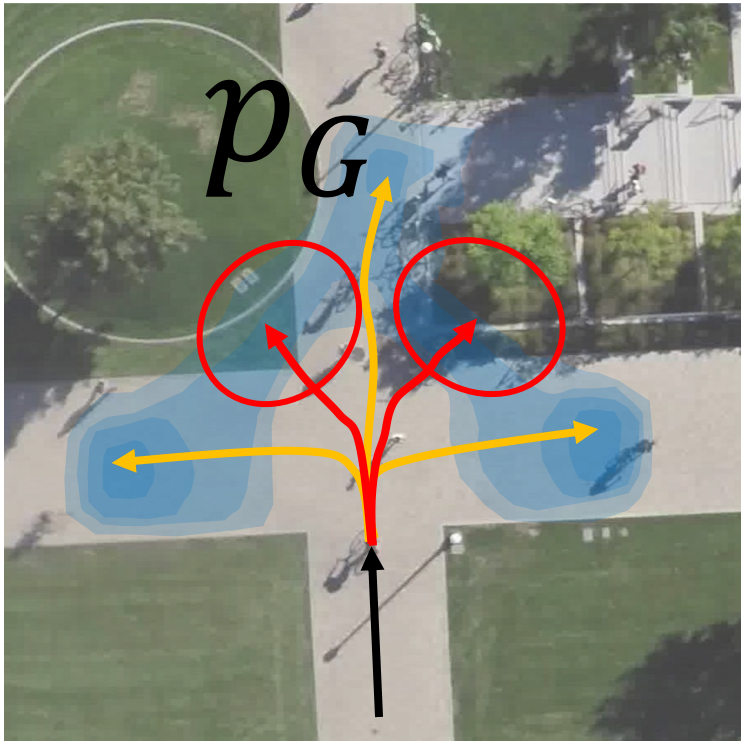
$$C_s^l(i) = \sum_j \alpha_{ij} W_{gat} V_s(j)$$

[Kosaraju et al. 19] Social BiGAT

# Problem with GAN Models

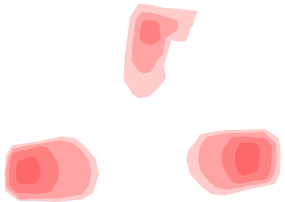
**Single generator GANs and VAEs**

- Social-GAN [Gupta 18]
- SoPhie [Sadeghian 19]
- S-BiGAT [Kosaraju 19]
- PECNet [Mangalam, 20]
- Trajectron++ [Salzmann, 20]

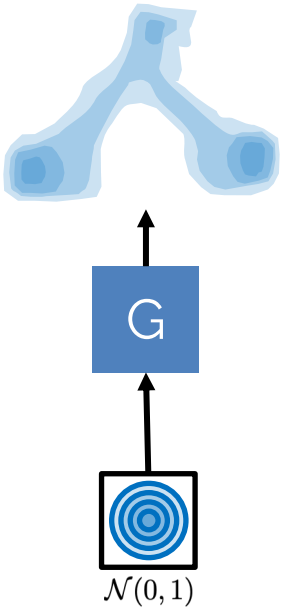




Out-of-distribution samples

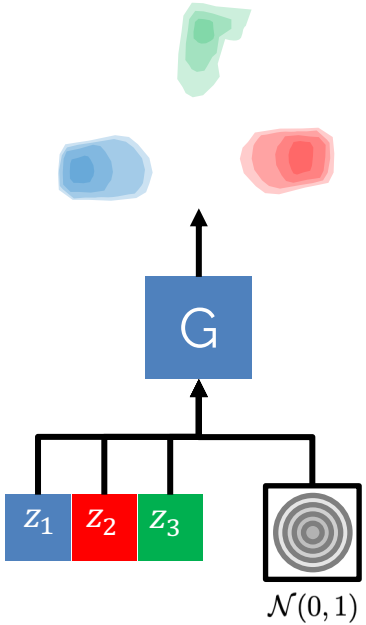
# Modelling Multimodality



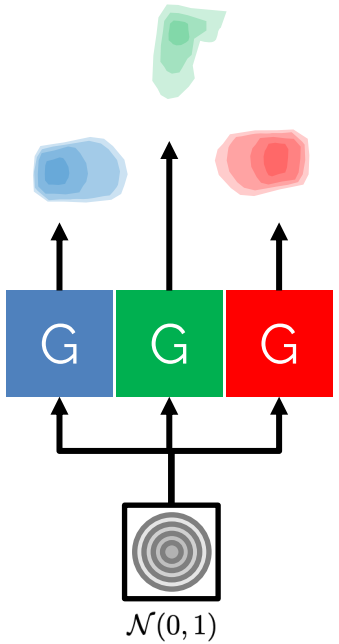
Ground-truth distribution with disconnected support




Single-Generator



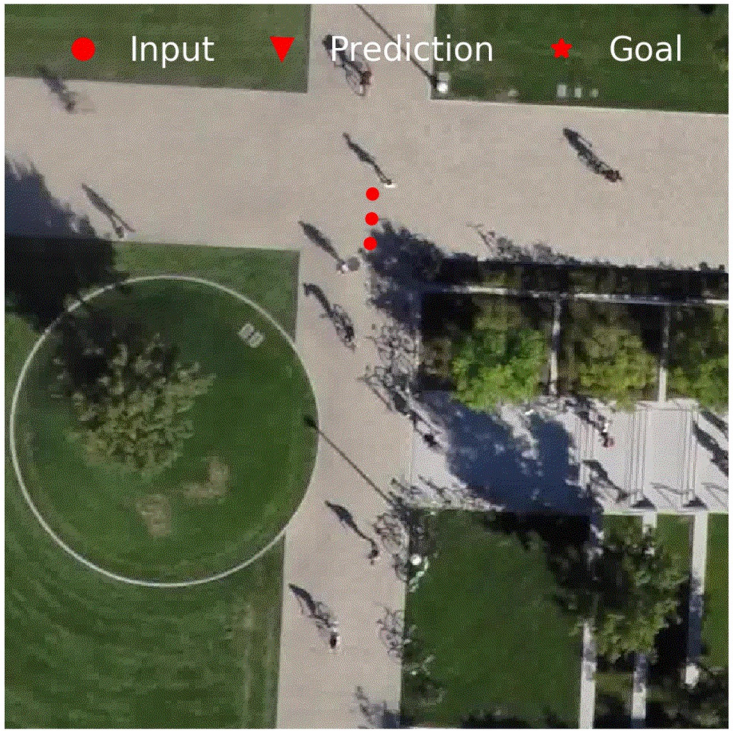

Generator with discrete latent space




Multi-Generator GAN

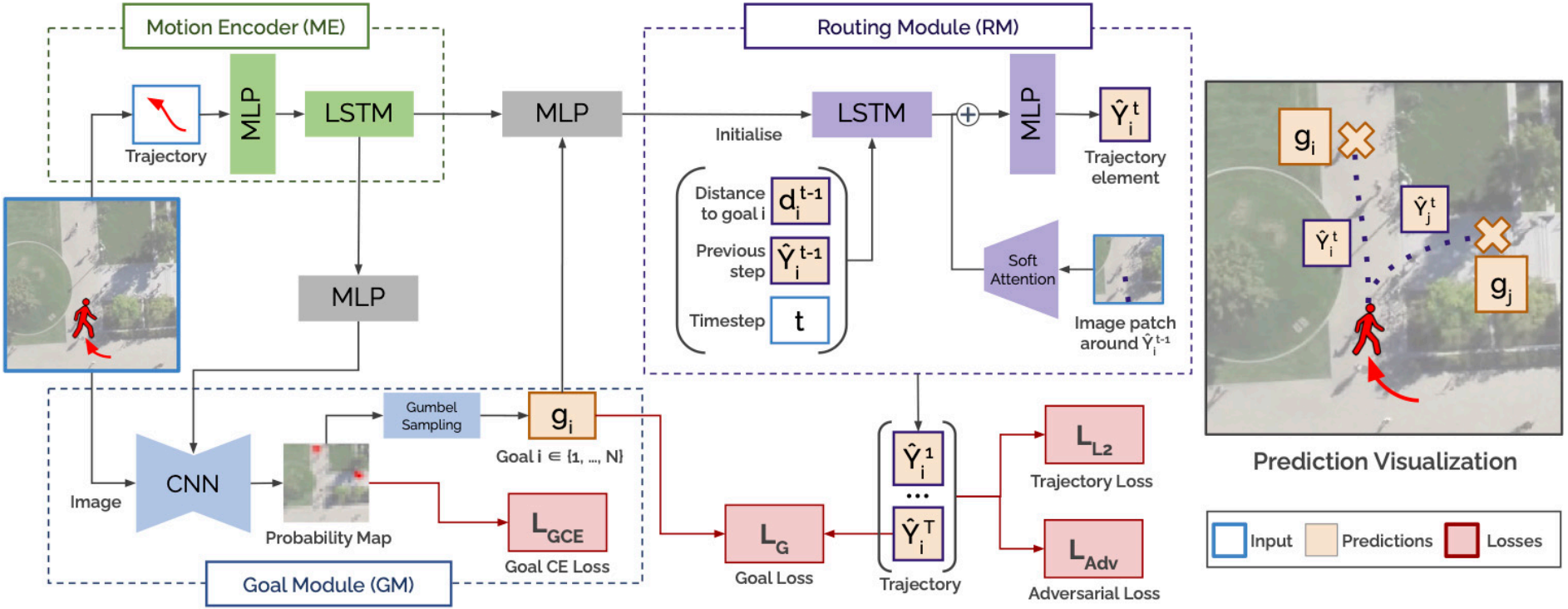
# Goal GAN

## Contribution: Two stage process of predicting goal and routing




[Dendorfer et al. 20] Goal GAN

# Goal GAN

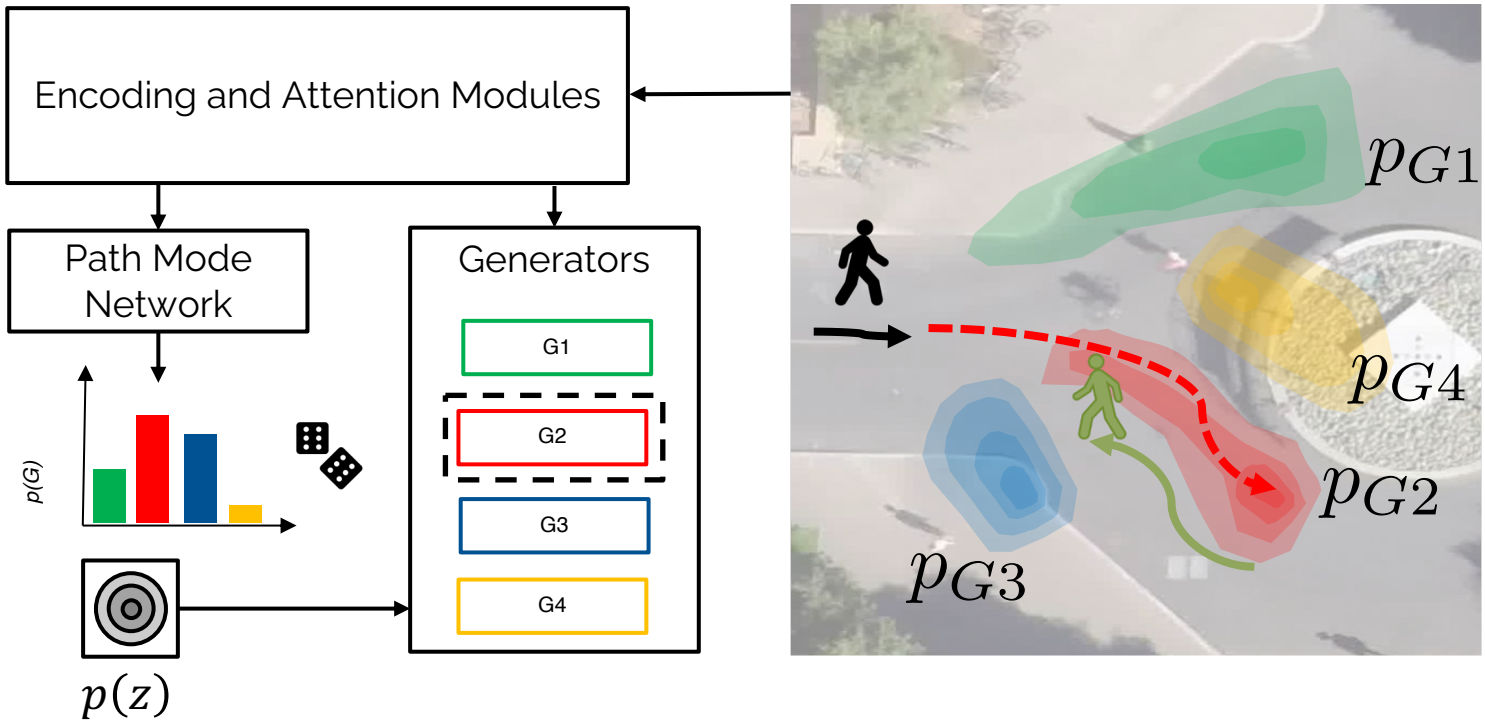





[Dendorfer et al. 20] Goal GAN

# MG-GAN: Multi-Generator GAN

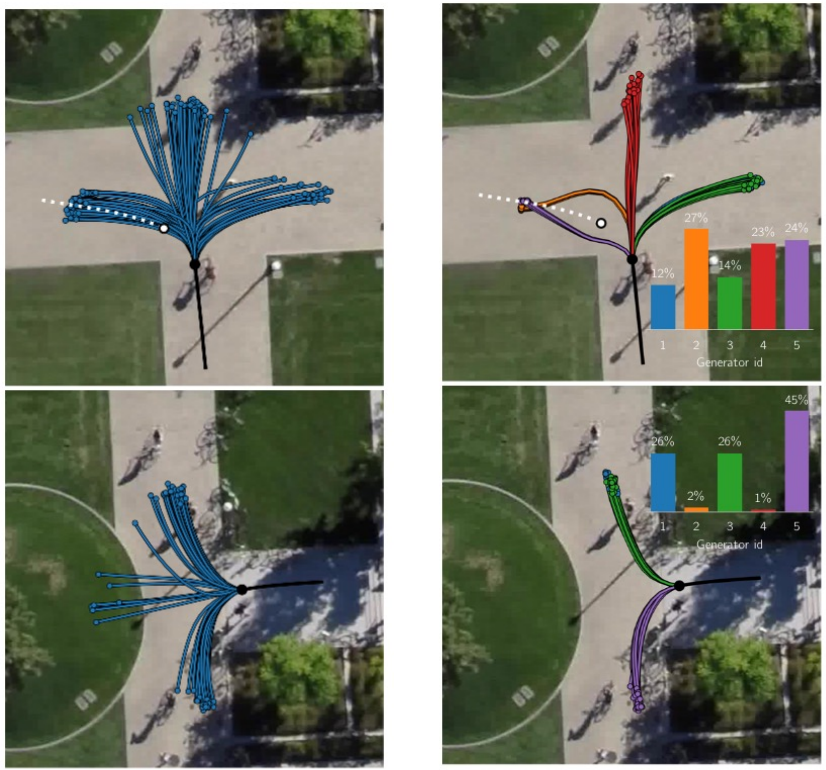
## Contribution: Multi-Generators learn different






[Dendorfer et al. 21] MG-GAN

# MG-GAN: Multi-Generator GAN




(b) GAN L2

(e) MG-GAN (ours)

- MG-GAN produces interpretable distribution over modes
- Controllability of generators during inference

[Dendorfer et al. 21] MG-GAN

# Summary

- Pedestrian Trajectory Prediction is relevant for numerous problem, 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 you attention!