Applications of distribution modeling and MCMC methods to intention forecasting

Francisco Valente Castro

September 30, 2020

Dr. Jean-Bernard Hayet

Index

What is intention forecasting?

What is intention forecasting on pedestrians?

Figure 1: UCY university data-set. Prediction of possible trajectories. White - Real future trajectory, Green - Observed past, Orange - Sample of possible goals, Red - Sample of possible trajectories.

What is intention forecasting on pedestrians?

- We want to predict the intention of an agent (e.g. a pedestrian, a car) using information of a video recording and misc. information of the scene (e.g. semantic information of the image).
- On pedestrians, we want predict a possible goal (place they want reach, the intention) and a possible trajectory they are going to take, knowing all the information up to the present.
- A vehicle navigation system could predict the future possible positions of a pedestrian and adjust its own trajectory to avoid collisions.

What is intention forecasting on pedestrians?

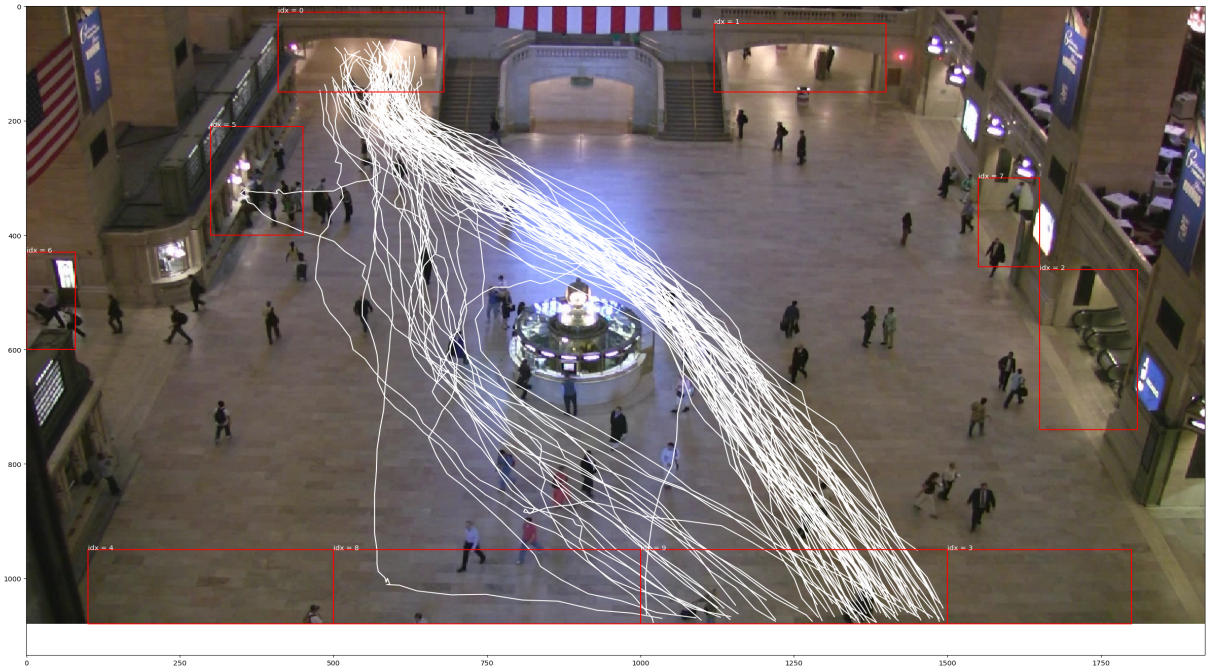


Figure 2: Grand Central Station pedestrian data-set exemplifying the data multi-modality.

Learning the goal distribution

Conditional Variational Auto-Encoder (CVAE)

The following are the random variables describing the principal components of our model

- $\mathbf{m} = (\mathbf{m}_1, ..., \mathbf{m}_t)$: The intermediate points of the path that the pedestrian is taking.
- $\mathbf{s}$: The initial position of the trajectory.
- $\mathbf{g}$: The last position of the trajectory.
- $\mathbf{z}$: The variable in the latent space that describes the trajectory.

We want to train a generative model for the intermediate points $\mathbf{m}$ in the trajectory, *conditioning* the response to the start $(\mathbf{s})$ and goal $(\mathbf{g})$.

What is intention forecasting on pedestrians?

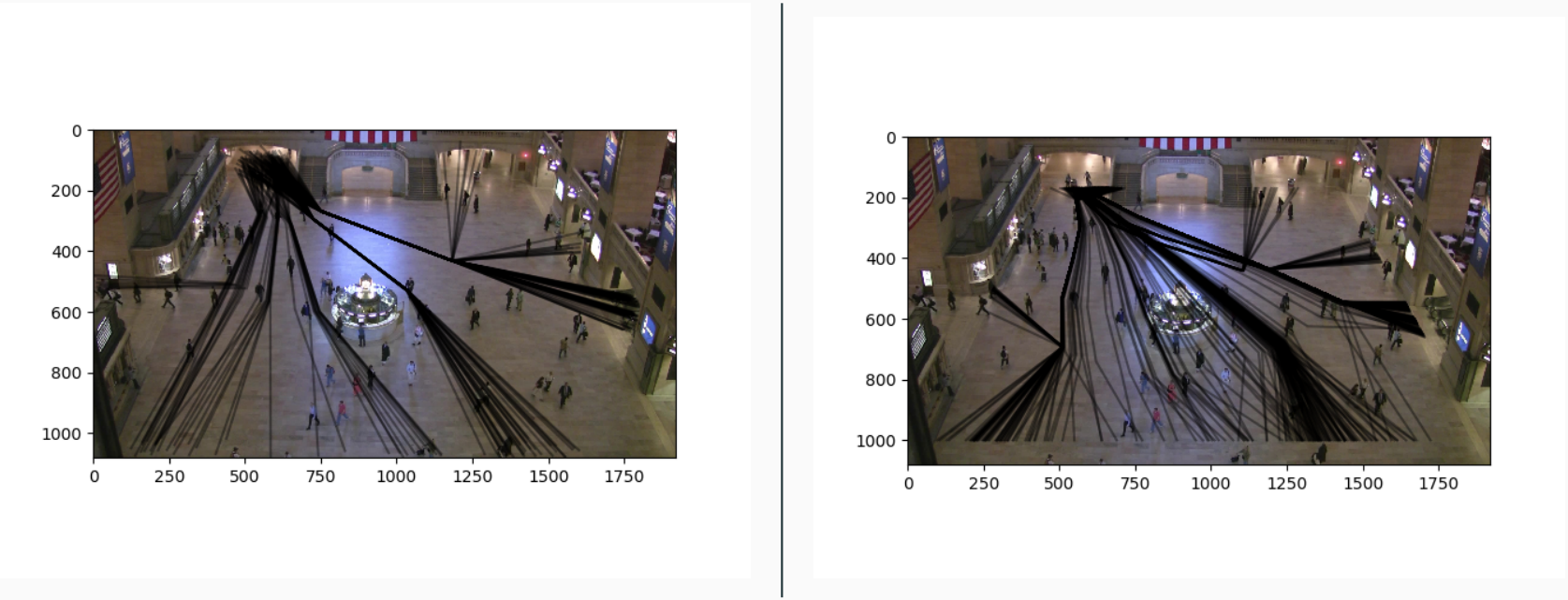


Figure 3: Left: Samples from CVAE trained with a 'start' and 'goal' given as (x, y) coordinates. Right: Samples from CVAE trained with a 'start' and 'goal' given as polar angles to the edge of the scene.

Training the CVAE model

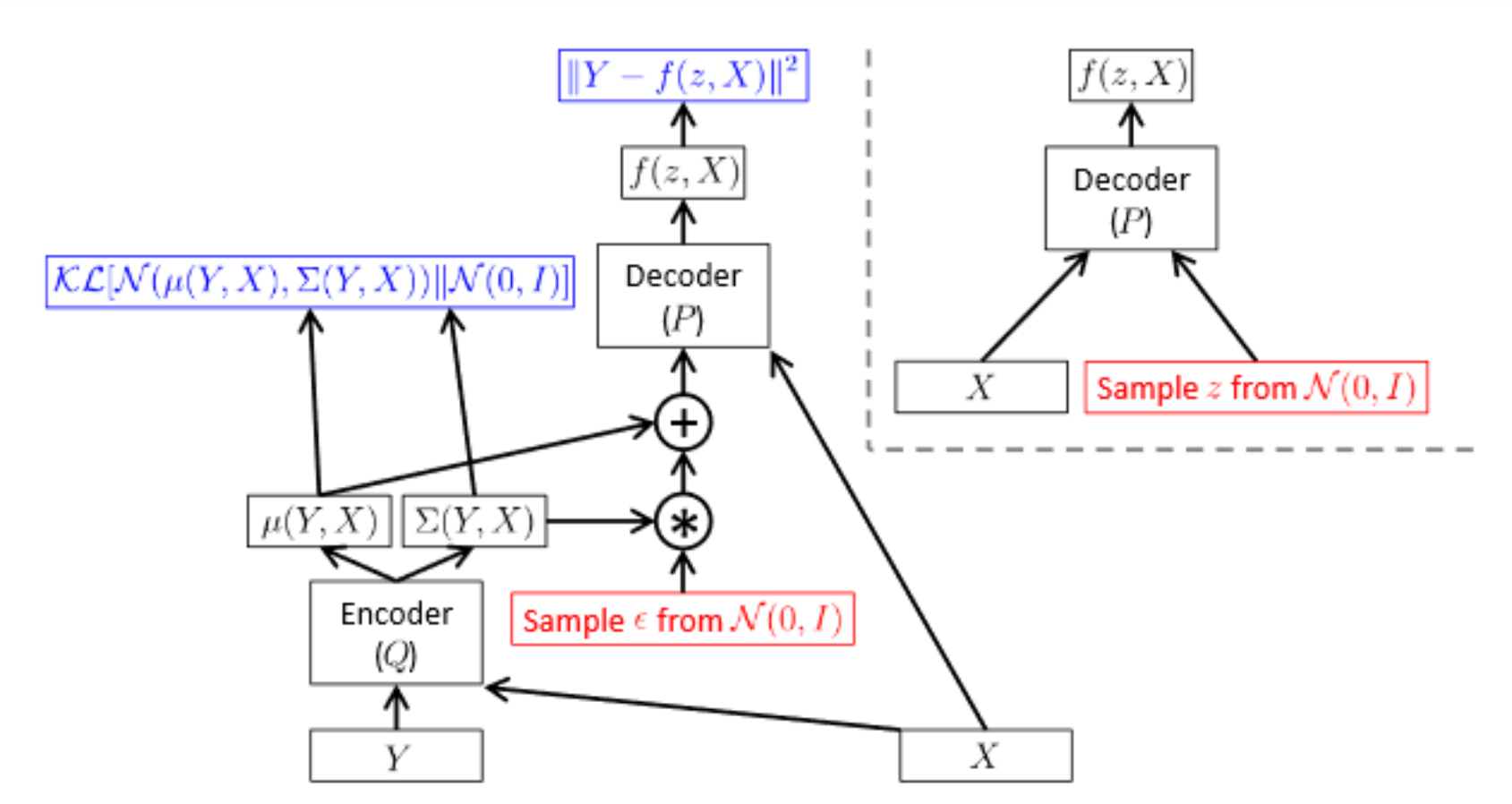


Figure 4: The general structure of a Conditional Variational Auto-encoder.

Training the CVAE model

Enconder

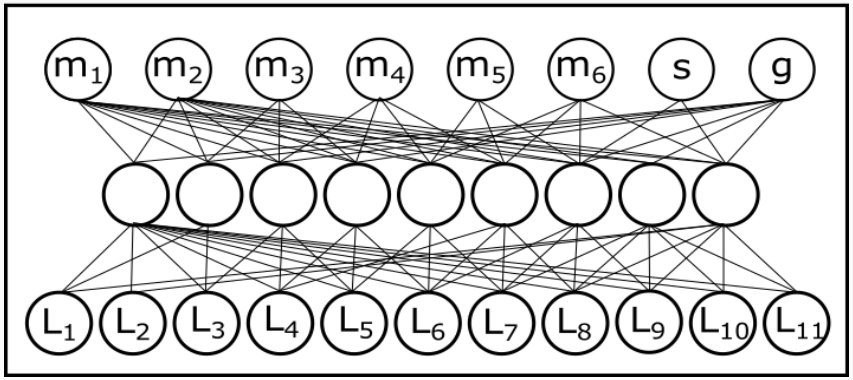


Figure 5: Linear Encoder $Q_\phi(\mathbf{m}, \mathbf{s}, \mathbf{g})$.

Decoder

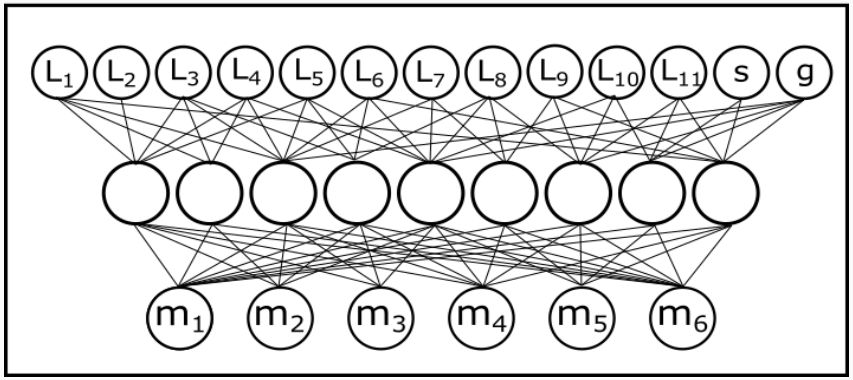


Figure 6: Linear Decoder $P_\theta(\mathbf{z}, \mathbf{s}, \mathbf{g})$.

Training the CVAE model

Minimize the ELBO cost function to train our Variational Auto-encoder.

$$\log P_\theta(\mathbf{m}|\mathbf{s},\mathbf{g}) - D_{KL}[Q_\phi(\mathbf{z}|\mathbf{m},\mathbf{s},\mathbf{g})\|P(\mathbf{z}|\mathbf{m},\mathbf{s},\mathbf{g})]$$
$$= \mathbb{E}_{z\sim Q_\phi}[\log P_\theta(\mathbf{m}|\mathbf{z},\mathbf{s},\mathbf{g})] - D_{KL}[Q_\phi(\mathbf{z}|\mathbf{m},\mathbf{s},\mathbf{g})\|P(\mathbf{z}|\mathbf{s},\mathbf{g})].$$

We are implicitly estimating the distributions $P_\theta(\mathbf{m}|\mathbf{z},\mathbf{s},\mathbf{g})$ (our *decoder*) y $Q_\phi(\mathbf{z}|\mathbf{m},\mathbf{s},\mathbf{g})$ (our *encoder*).

Goal inference using MCMC and CVAE

Goal posterior

We find a way to estimate the posterior distribution of the **goal** given the **intermediate** points $\mathbf{m} = (\mathbf{m}_1, ..., \mathbf{m}_t)$:

$$\begin{aligned}\mathbb{P}[\mathbf{g}|\mathbf{s}, \mathbf{m}] &= \frac{\mathbb{P}[\mathbf{g}, \mathbf{s}, \mathbf{m}]}{\mathbb{P}[\mathbf{s}, \mathbf{m}]}, \\ &= \frac{\mathbb{P}[\mathbf{m}|\mathbf{s}, \mathbf{g}]\mathbb{P}[\mathbf{s}, \mathbf{g}]}{\mathbb{P}[\mathbf{s}, \mathbf{m}]}, \\ &\propto \mathbb{P}[\mathbf{m}|\mathbf{s}, \mathbf{g}]\mathbb{P}[\mathbf{g}|\mathbf{s}]\end{aligned}$$

This **posterior** distribution $\mathbb{P}[\mathbf{g}|\mathbf{s}, \mathbf{m}] \propto \mathbb{P}[\mathbf{m}|\mathbf{s}, \mathbf{g}]\mathbb{P}[\mathbf{g}|\mathbf{s}]$ is the most important ingredient to apply MCMC methods to sample goals.

Goal posterior

To actually calculate the probabilities in our posterior distribution $\mathbb{P}[\mathbf{g}|\mathbf{s},\mathbf{m}] \propto \mathbb{P}[\mathbf{m}|\mathbf{s},\mathbf{g}]\mathbb{P}[\mathbf{g}|\mathbf{s}]$ we have to sample μ means from our decoder.

$$\mathbb{P}[\mathbf{m}|\mathbf{s},\mathbf{g}] = \mathbb{E}_{\mathbf{z}\sim Q}[P(\mathbf{m}|\mathbf{z},\mathbf{s},\mathbf{g})],$$
$$\propto \textbf{exp}\left(-\frac{1}{2N}\sum_{i=1}^{N}(\mathbf{m}-\mu_i)^T(\mathbf{m}-\mu_i)\right).$$

To estimate the required prior distribution $\mathbb{P}[\mathbf{g}|\mathbf{s}]$ we will use simpler methods like using **Gaussian Kernel Density Estimation**. We will show two different ways to do this estimation.

Prior 1D - Gaussian Kernel Density Estimation

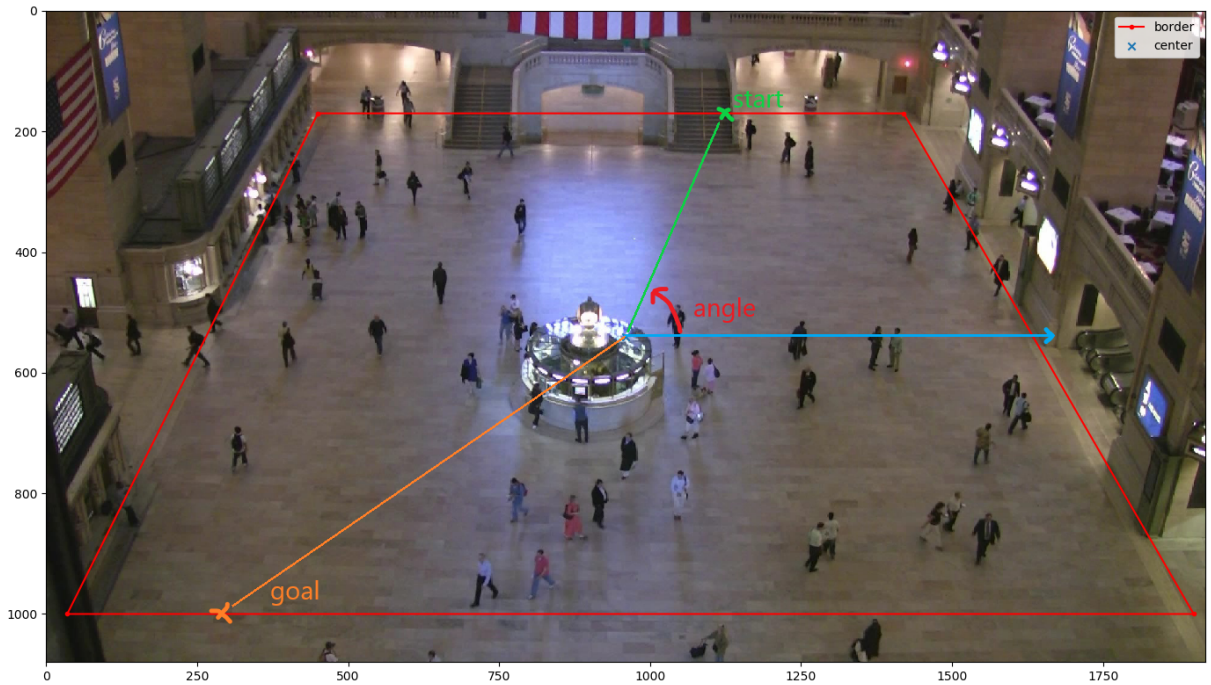


Figure 7: Start, Goal on border.

In this model we determine that **start** and **goal** are uni-dimensional random variables that take values on the range $(0, 2\pi)$ as angles.

Prior 2D - Gaussian Kernel Density Estimation

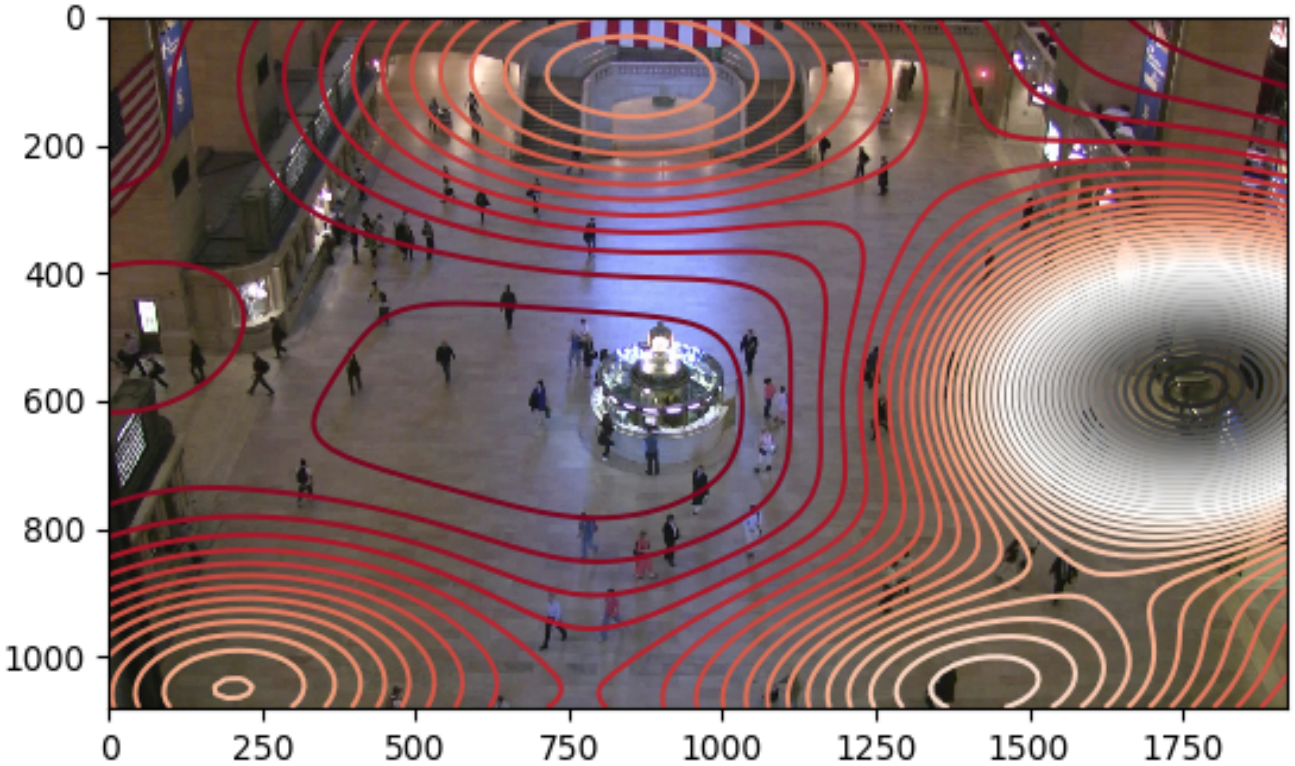


Figure 8: Gaussian Kernel Density Estimation of $\mathbb{P}[g]$.

2D Results on Grand Central dataset

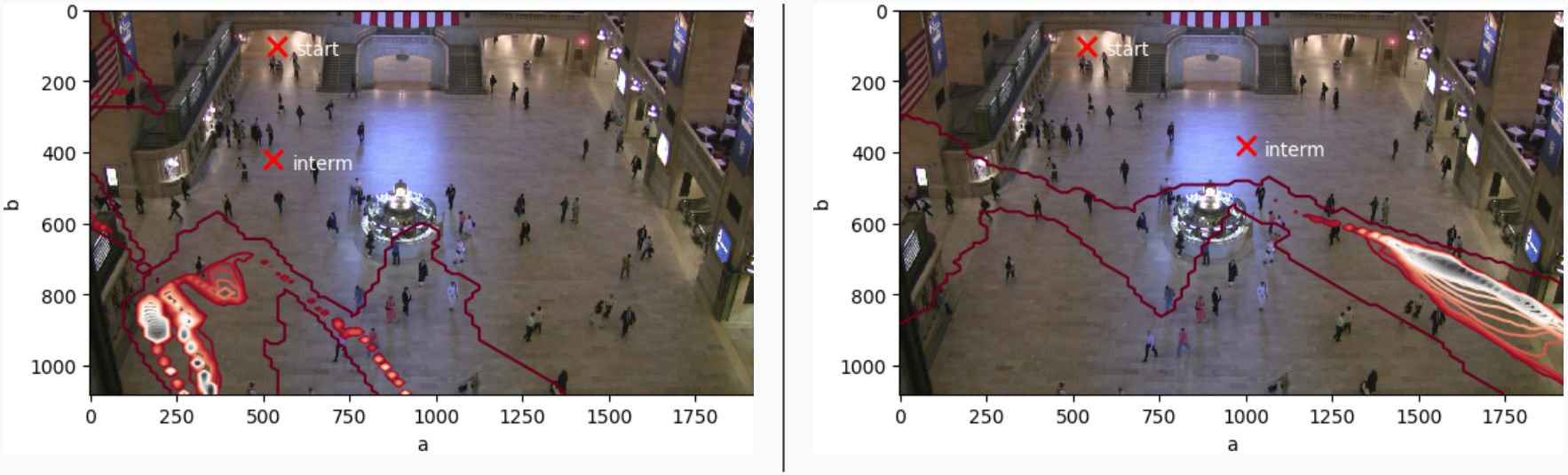
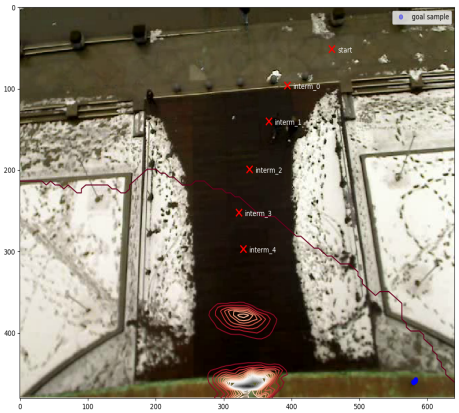
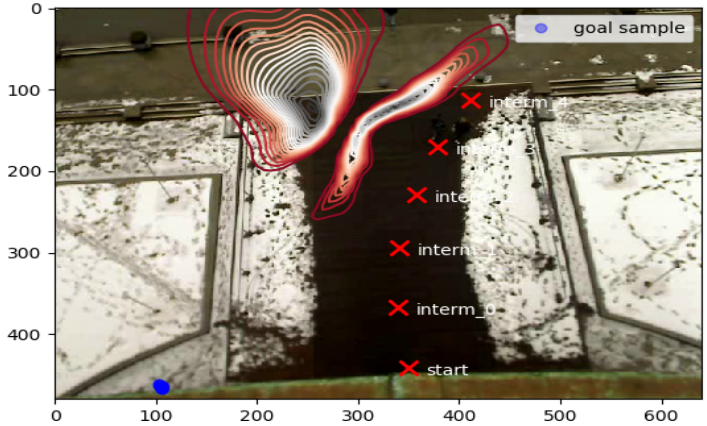
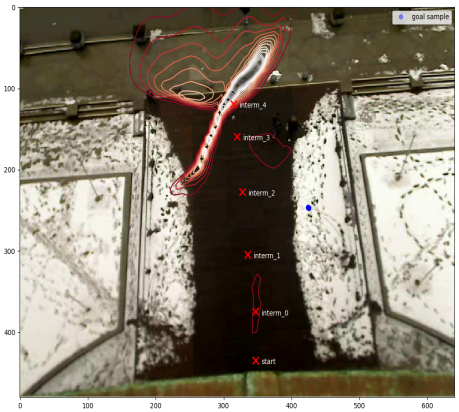


Figure 9: Posterior distribution estimation level sets.

2D Results on ETH dataset







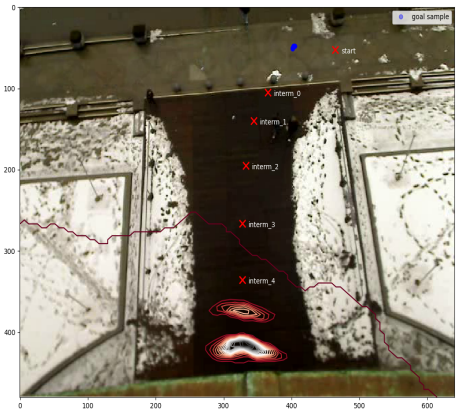


Figure 10: Posterior distribution estimation, ETH Dataset.

Posterior Sampling

Choosing one or more proposals $q(\mathbf{g}^*|\mathbf{g}, \mathbf{s}, \mathbf{m}, \mathbf{M})$ (considering the algorithm using hybrid kernels) we can only consider the **acceptance probability** from the the proposals in each iteration of the algorithm,

$$\alpha(\mathbf{g}^*|\mathbf{g}) = \min\left\{1, \frac{q(\mathbf{g}|\mathbf{g}^*)p(\mathbf{g}^*|\mathbf{s}, \mathbf{M})p(\mathbf{m}|\mathbf{g}^*, \mathbf{s}, \mathbf{M})}{q(\mathbf{g}^*|\mathbf{g})p(\mathbf{g}|\mathbf{s}, \mathbf{M})p(\mathbf{m}|\mathbf{g}, \mathbf{s}, \mathbf{M})}\right\},$$

Posterior Sampling

In the following figure we can observe the **Metropolis-Hastings** algorithm in its simpler form. The variants and other MCMC methods follow a similar procedure.

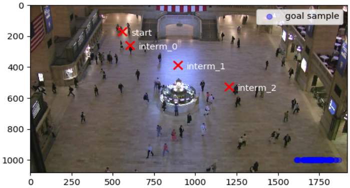
Algorithm 1 Metropolis-Hastings algorithm

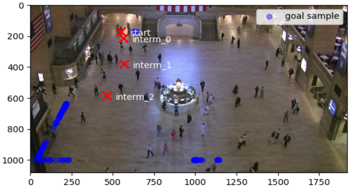
Initialize $x^{(0)} \sim q(x)$
for iteration $i = 1, 2, \ldots$ **do**
  Propose: $x^{cand} \sim q(x^{(i)}|x^{(i-1)})$
  Acceptance Probability:

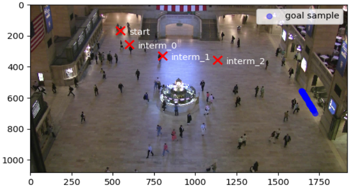
$$\alpha(x^{cand}|x^{(i-1)}) = \min\left\{1, \frac{q(x^{(i-1)}|x^{cand})\pi(x^{cand})}{q(x^{cand}|x^{(i-1)})\pi(x^{(i-1)})}\right\}$$

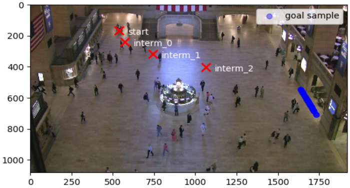
  $u \sim \text{Uniform}\ (u; 0, 1)$
  if $u < \alpha$ **then**
    Accept the proposal: $x^{(i)} \leftarrow x^{cand}$
  else
    Reject the proposal: $x^{(i)} \leftarrow x^{(i-1)}$
  end if
end for

Angle Results on GC using MCMC sampling









Thank you

References

[1] Carl Doersch *Tutorial on Variational Autoencoders*. Carnegie Mellon / UC Berkeley, August 16, 2016

[2] Ilker Yildirim *Bayesian Inference: Metropolis-Hastings Sampling*. Department of Brain and Cognitive SciencesUniversity of Rochester, 2012

[3] Karttikeya Mangalam, Jitendra Malik *It Is Not the Journey but the Destination: Endpoint Conditioned Trajectory Prediction*. University of California, Berkeley