# A Survey on Trajectory Encoding Methods for Social Robots

LEANDRO DE SOUZA ROSA and LUCA IOCCHI*, La Sapienza University of Roma, Italy

JULIAN F. SCHUMANN, Delft University of Technology, Netherlands

ANNA MÉSZÁROS, Delft University of Technology, Netherlands

## 1 INTRODUCTION

Social robots ...

Navigation in human-cluttered environments is a complex task due to the unpredictable behavior of humans.

One way to tackle this problem is create models for the human behaviors, which are used to predict their possible trajectories. However these methods capture a set of pre-defined behaviors and may fail in mode generic cases.

An alternative is learning embeddings for trajectories, which map possible trajectories in lower-dimensional spaces which are used to map possible different behaviors, or used to compute trajectory distances.

We are focusing on trajectories embeddings, and how they are used on the context of social navigation.

The structure of this survey follows: Section XX presents ... Section XX presents ...

## 2 TRAJECTORY SIMILARITY

This section focuses on trajectory similarity methods.

Such methods have been used in different scopes.

Early similarity metrics are based on numeric methods. These methods focus on computing a distance between trajectories based on heuristics or spatio-temporal characteristics observed in the trajectories. They commonly are agnostic to the type of trajectory and thus have found use in many applications [give a sample with citations]. Furthermore, they are commonly grouped as non-learning methods, since the provided distance metric between trajectories depends only on the input trajectories, while learning-based methods depend on an available database of trajectories used implicitly learn a distance metric.

For robot manipulation these methods are used for ....

For navigation in urban environments these methods are used for ....

For Social navigation these methods are used for ....

### 2.1 Numerical Methods for Trajectory Distance

Numerical methods have been widely used in the literature for computing the distance between trajectories.

**Hausdorff Distance**.

Hausdorff Distance (HD) has been widely used in image matching [36, 38] and trajectories comparison [5]. It is based between the maximum distance between each point from a set to all points in the other set, being the sets curves or images.

**Fréchet Distance**.

Fréchet Distance (FD) [21] is similar to HD, but considers the directions of curves to compute the minimal trajectory.

**Longest Common Subsequence**.

Longest Common Subsequence (LCSS) [51] considers as metric only the part of trajectories which match the most, given a tolerance parameter. This also makes it sensitive to the parameter selection and noise.

**Dynamic Time Warp**.

Dynamic Time Warp (DTW) [68, 91] is based on aligning one or more adjacents points for computing a distance between trajectories. It handles trajectories of different lengths, however it causes distortions which might not have been in the trajectories.

**Edit distance with Real Penalty**.

Edit distance with Real Penalty (ERP) [9] it is an "edit-based distance" meaning that this metric takes in account a cost which is based in how many edits one trajectory needs to match the other.

In this distance the points of a trajectory are taken as reference and the points and edits are performed on the other trajectory for reducing the distance.

**Edit Distance on Real Sequence**.

Edit Distance on Real Sequence (EDR) [10] considers a tolerance parameter like LCSS and adds a cost value on its evaluation. This cost increases with the number of editions that are necessary to match both trajectories within the tolerance error. This makes EDR more robust to gaps in the trajectory and outliers.

**Edit Distance with Projections**.

Edit Distance with Projections (EDwP) [65] performs interpolations between points and tries to match segments of trajectories.

**Locality In-between Polylines**.

Locality In-between Polylines (LIP) [62] computes distance based on the area of polygons made between the two curves. It does not consider warping, but instead it is sort of an integral of the area between curves.

Variations are introduced in [62] to support time and direction.

**One-Way Distance**.

One-Way Distance (OWD) [46] is a measure based on the spatial shapes of moving objects. It is based on the average distance of each point of one trajectory and the other trajectory.

## 2.2 Trajectory Clustering

Gotta check some papers on trajectory clustering

[check fernando2018soft+ for some more clustering methods to add to this section]

**2009 - Learning trajectory patterns by clustering: Experimental studies and comparative evaluation**.

[57]

## 2.3 Trajectory Learning

[check this out https://tslearn.readthedocs.io/en/latest/user_guide/shapelets.html]

**2014 - Sequence to Sequence Learning with Neural Networks**.

**Name: seq2seq**

[76]

**2017 - Identifying Human Mobility via Trajectory Embeddings**.

**Name: TULER**

**Code on:** https://github.com/gcooq/TUL

[28] classifies users based on trajectory data. The problem is hard because there are many more trajectories than users.

Recurrent Neural Network (RNN) is used, and said to be good for classification when the number of labels is small. In particular uses a Long Short-Term Memory (LSTM) for processing sub-trajectories. In particular, using a bi-directional LSTM yields best results.

There is a location embedding, not sure how that is computed. But the trajectories are points on google maps, so there maybe be semantic information in there. The sequence of location embedding is passed onto the LSTM, not sure how they handle different trajectory lengths.

**2018 - Self-Consistent Trajectory Autoencoder: Hierarchical Reinforcement Learning with Trajectory Embeddings**.

**Name: SeCTAR**

**Code on:** https://github.com/wyndwarrior/Sectar

[16] learns an embedding for trajectories with one encoder and two decoders: a state decoder to decode from the latent space back into trajectories, and a policy decoder, which generates the trajectory in the environment. As such the state decoder predict the trajectory of the policy. The encoder is used in a hierarchical Reinforcement Learning (RL) setup.

The state encoder and decoder are RNNs and the policy decoder is a feed-forward Neural Network (NN).

**note:** if the encoder is trained with trajectories from different tasks, the policy will be conditioned to each task, what is sort of parameterzing the policy to tasks.

In the paper, the policy has unknown dynamics, and hence the RL setup. Trajectories are continuos poses of joints over time. Tested in simulation.

**2018 - Anomalous Trajectory Detection Using Recurrent Neural Network**.

**Name: ATD-RNN**

**Code on:** https://github.com/LeeSongt/ATD-RNN

[73] proposes anomalous trajectory detection using RNN.

The trajectories are discretized, using a grid, and feed to a stacked RNN for learning the embedding, then a multi-layered perceptron and a soft-max layer detects if the trajectory is anomalous. The stacked RNN is made by feeding the hidden states od the previous to the next RNN.

The trajectories are padded in order to get trajectories of the same length.

LSTM and Gated Recurrent Unit (GRU) are two special types of RNNs are tested. GRUs seems to work better.

**2018 - Deep Representation Learning for Trajectory Similarity Computation**.

**Name: t2vec Code on:** https://github.com/boathit/t2vec

[44] presents t2vec. A Deep Learning (DL) approach for trajectory similarity. States that using RNN is not a very good idea because you cannot reconstruct the trajectory and it fails to consider spatial proximity, which is inherited in trajectory data.

called in the paper as t2vec or seq2seq?

The approach is based on the encoder-decoder framework. Handling varying sampling rates is done by augmenting the training data creating sub-trajectories by sub-sampling and noise addition. They also propose a spatial-aware loss, and pre-train the ??cells?? and let them to be optimized during training.

**Notes:** The paper is very confusing. I do not really know that are the inputs and outputs or how the sequences are fed in the RNN inside the encoder.

### 2020 - Trembr: Exploring Road Networks for Trajectory Representation Learning.

### Name: Trembr

[26] uses two embedding one for trajectories traj2vec and another for the road network road2vec. It preprocesses the trajectories by projecting them in a road network and the trajectory is a sequence of road segments and travel time.

The RNN decoder is conditioned to the road network, and the training is made by optimising a loss for the trajectory and another for time.

**Notes:** Maybe the secret for velocities profiles is in the addition of time to the loss.

### 2019 - Computing Trajectory Similarity in Linear Time: A Generic Seed-Guided Neural Metric Learning Approach.

### Name: NeuTraj

### Code on: https://github.com/yaodi833/NeuTraj

[90] proposes a method for accelerating trajectory similarity computation by sampling seeds of trajectories, computing their similarity, and approximating them with a neural metric.

States that RNNs, LSTMs, and GRUs can only model one sequence without considering the between-sequence correlation.

Does not consider time in the trajectory. Starts sampling from the trajectories and computes a distance matrix between the samples using a given trajectory distance metric which is then normalized.

The RNN is augmented with a memory, which is created by dividing the space into a grid, and for each grid slot, the memory stores the hidden vector of the RNN. This memory is used to extend the RNN cell, sort of like an LSTM.

The loss for training is $\mathcal{L}\tau_i, \tau_j = \sum_k w_k(f(\tau_i, \tau_j) - \exp(-||e_i - e_j||)$, a weighted difference between the similarity metric $f$ and the distance in the embedding space $(e_i - e_j)$. The weight $w_k$ is obtained using the normalized distance matrix, computing pairs of similar and dissimilar trajectories and more fancy stuff.

**Notes:** map is like google map.

### 2020 - Trajectory similarity learning with auxiliary supervision and optimal matching.

### Name:Traj2SimVec

[94] follows the same idea as in [90] which selects some trajectories for pre-training [something], the training samples are divided in three sub-trajectories [because it seems to help learning].

A distance matrix is computed which is used as supervision signal, similar to [90].

### 2020 - MARC: a robust method for multiple-aspect trajectory classification via space, time, and semantic embeddings.

### Name: MARC

[64] Embeds semantics on the trajectories. Each semantic information (weather, time, type of place) has an encoding, and a weight matrix which transform them into a fixed size vector. The semantic trajectory is fed to an LSTM, which encodes the trajectories, having the hidden states used for classification.

### 2021 - Embedding-Based Similarity Computation for Massive Vehicle Trajectory Data.

**Name: L2R***

[11] seems to propose the exact same thing as [90], but with interpolation for de-noising.

Adds a ranking learning loss on [90]

**2021 - STENet: A hybrid spatio-temporal embedding network for human trajectory forecasting**.

**Name: STENet**

[93] Focuses on predicting pedestrian trajectories. Uses a LSTM with Convolutional Neural Networks (CNNs) to embed position features in multiple temporal time-scales. The encoder-decoder structure stack a CNN and a graph attention model. The decodes stacks many LSTMs.

They give related works on social trajectory learning.

**Notes:** They point to Variational Auto-Encoders (VAEs) for modelling multi-modality and for the generative capabilities.

**2021 - A Graph-Based Approach for Trajectory Similarity Computation in Spatial Networks**.

**Name: GTS**

[32] Propose a Graph Neural Network (GNN)-based trajectory embedding. The framework measures trajectory similarities, learns Points of Interest (PoIs), and leans a trajectory embedding.

A trajectory is encoded as the points in a graph map. Then they define a trajectory similarity metric on the PoI graph, based on the graph distance between the points and trajectories. An embedding capturing the neighbours and graph trajectory is learned. The PoI embeddings and their neighbours are used to learn another embedding using its neighbours information. Finally, LSTMs are used to learn the trajectory over the graph embeddings. The loss function minimizes the above defined distance between trajectories and the distance between the two closest trajectories.

**2021 - T3S: Effective Representation Learning for Trajectory Similarity Computation**.

**Name: T3S**

[89] combines LSTMs and attention NNs over the grid graph for learning the embedding. Close to [11, 90, 94].

**2021 - How meaningful are similarities in deep trajectory representations?**.

**Code on:** https://dbis.ipd.kit.edu/2652.php

[78] presents a survey and evaluation of t2vec [44] and other methods. Seems like t2vec with some variations outperform the rest. t2vec seems to be stacked LSTMs.

Evaluate how changing t2vec parameters affect similarity values. t2vec seems robust to parameters.

Evaluate t2vec against non learning metrics. Seems like associating them lead to better results.

[They DO ignore the whole literature on learning methods?]

Concludes that using LCSS and t2vec leads to a better trajectory similarity, covering overlap, shape, direction and distance.

**Notes:** Maybe that should be 4 characteristics to consider for explainability.

**2022 - Spatio-Temporal Trajectory Similarity Learning in Road Networks**.

**Name: ST2vec**

[23] learns a spatio-temporal representation. Two steps, which is based on learning a spatial model, a temporal model and a co-attention fusion module. It is based on a road network, trajectories are sequences of vertex on the road network.

Define the distance of a spatio-temporal trajectory as a weighted sum for a spatio-distance ($d_s$) and a temporal distance ($d_t$):

$$d(\tau_i, \tau_j) = \alpha d_s(\tau_i, \tau_j) + (1 - \alpha)d_t(\tau_i, \tau_j)|\alpha \in [0, 1]$$

Later uses LSTMs to learn using two strategies, using one LSTM for space and another for time, or using one for both.

**2022 - Deep Fuzzy Contrast-Set Deviation Point Representation and Trajectory Detection**.

[1] Grid-map based, contrastive learning.

**notes:** hard to understand what they are doing here.

**2022 - Contrastive Pre-training of Spatial-Temporal Trajectory Embeddings**.

**Name: CSTTE**

[47] employs contrastive learning for learning an embedding which retains high-level travel semantics.

Recovering the original trajectory is not a good approach when learning representations with RNNs since it fails to capture the high-level information of trajectories. Contrastive learning with noisy augmentation can handle the high-level information while being robust to noise. However data augmentation needs to be well designed.

The positive samples are created with subsampling the query trajectory, while the negative samples come from different trajectories.

**Notes:** Not sure this is correct, I think the "different trajectories" should be far enough from the query trajectory to be a negative sample.

The encoder stacks a spatio-temporal encoding layer and attention layers. For the first, a learnable encoding of locations is learned (each location leads to a vector) and location and time are passed to a trigonometric vector transformation to compute features which can capture periodic information; those vectors are then summed up. The attention layer is actually 2 stacked attention layers.

**2022 - TMN: Trajectory Matching Networks for Predicting Similarity**.

**Name: TMN**

[88] uses attention to compute intra-trajectory similarities, and then uses a LSTM.

**Notes:** Comparison ignores many methods.

**2022 - TSNE: Trajectory Similarity Network Embedding**.

**Name: TSNE**

[20] uses a pre-defined trajectory measure function to construct a k-NNG (K nearest neighbours graph) and computes the embedding based on the graph.

**Notes:** Not sure how they compute the embedding from the graph. Seems like the graph representation allows to handle partial similarity and unordered similarity.

**2022 - Towards robust trajectory similarity computation: Representation-based spatio-temporal similarity quantification**.

**Name: RSTS**

[14] splits the spatio-temporal trajectories into cells, and uses a triplet loss for the learning. It enforces that if the time and space similarities are higher, then the distance in the encoded space must be smaller, and that, in the encoded space, the distance between two trajectories variations (noise and downsampling) must obey the distance of the trajectories.

An embedding is used for the tokens, which are then passed to a RNN encoder-decoder. The tokens for the embedded are an ID computed by splitting the space-time into cells. The input is grid-cells (gps + time).

**Notes:** Analysis is poor. Ignores all other works on learning. Seems like there is little innovation besides the loss.

**2023 - Spatial-temporal fusion graph framework for trajectory similarity computation**.

**Name: GTS**

Sort of the same thing as [32]. Name is also the same. Even results are the same.

[98] first learns a point of interest representation on the road network, which is passed to a GNN for learning neighbours information as embeddings, and then a LSTM for learning the sequencing.

A symmetric distance between trajectories is defined based on the distance between each point of the trajectories and the other trajectory:

$$d(\tau_1, \tau_2) = \sum_{v \in \tau_1} e^{-d(v, \tau_2)} + \sum_{v \in \tau_2} e^{-d(v, \tau_1)}$$

The time is considered in an extension called ST-LSTM, which adds a time one-hot encoding into the gating functions of the LSTM.

**Notes:** Comparisons goes as far as traj2SimVec [94].

**2023 - GRLSTM: Trajectory Similarity Computation with Graph-Based Residual LSTM**.

**Name: GRLSTM**

[99] combines Knowledge Graph Embedding (KGE), GNN and a multi-layer residual-LSTM. KGE is used to learn point and relation embeddings for constructing a graph, which is passed to the GNN for learning the topology in the point-structure graph. Then the LSTM is used to learn the embeddings trajectories. Uses two losses: a graph-based loss and a trajectory-base loss.

The input is trajectories in a graph road network. The interesting thing here is that adjacent points in the trajectory may not be adjacent in the graph (due to data loss or lower sample rate).

The stacked LSTM is augmented with a residual layer for handling the gradient forgetting of traditional LSTM. It is stated that it does not add parameters so it does not affect training time considerably.

**Notes:** does not really say how the residual function is computed. Similarly to [32] they implement point and trajectory distances.

**2023 - Contrastive Trajectory Similarity Learning with Dual-Feature Attention**.

**Name: TrajCL**

[8] introduces four trajectory augmentation and a dual feature self-attention encoder, for learning structural and spatial patterns of trajectories. It does not involve any recurrent structure. Instead, it uses a dual self-attention-based trajectory encoder.

Augmentations:

**point shifting:** adds an offset to the points

**point masking** randomly removes points from the trajectory

**tuncation** cuts a prefix, suffix, or both from the trajectory

**simplification** uses the Douglas–Peucker algorithm which removes non critical points from the trajectories (like points in a straight line).

The augmented trajectories are used to create two trajectory views to learn structural and spatial features. The augmented trajectories are used to compute two trajectory views. The structural features, the map is converted into a grid, and used to create a graph in which the grid locations are the vertices and the trajectory transitions the edges. Then a graph embedding (node2vec) is used to learn an embedding. For the spatial features, the angle and length of trajectory segments is computed. Both views are augmented by adding a [sketchy] sine and cosine value to the points to capture position information.

7

Finally the two views are passed to a two-head self attention module to learn the embeddings.

**2023 - Spatio-Temporal Trajectory Similarity Measures: A Comprehensive Survey and Quantitative Study**.

**Code on:** https://github.com/ZJU-DAILY/TSM

[34] presents a survey with several methods, and benchmark for evaluating them. Apparently `Traj2SimVec` [94] is the learning method, which is not grid-based that handles our problem.

### 2.4 Trajectory Learning on Robotics

**2020 - Controlling Assistive Robots with Learned Latent Actions**.

[50] Use encoders to learn latent task representations for assistive robot remote controlling. In this setup, VAEs are used, encoding states into a task representation, the user gives input from a joystick which are decoded together with the latent space representation.

The latent representation is encoded ($\phi$) from state ($s$) and action ($a$), and decode from latent state ($z$) and state into action.

There is an emphasis on desired characteristics of the latent representation:

**controlability:** $s_i = T(s_{i-1}, \phi(s_{i-1}, z_{i-1}))$, where $T$ is the transition function. Meaning that using the latent states should lead to the same actions as the ones in the training set.

**consistency:** $|T(s_i, \phi(s_i, z_i)) - T(s_j, \phi(s_j, z_j))| \leq \epsilon$ if $|s_i - s_j| \leq \delta$.

**scalability:** $|T(s_i, \phi(s_i, z_i)) - T(s_j, \phi(s_j, z_j))| \rightarrow \inf$ if $|s_i - s_j| \rightarrow \inf$

**Notes:** Seems like VAE is used straight up with the trajectories. But it is a bit blurry how the actions are being defined or learned (seem pre defined). conditional Variational Auto-Encoder (cVAE) seems to outperform other encoders.

**2020 - DiversityGAN: Diversity-Aware Vehicle Motion Prediction via Latent Semantic Sampling**.

**Name: DiversityGAN**

[35] extends Generative Adversarial Network (GAN) using a low-dimensional approximate semantic (encoding) which is shaped to capture semantics. Sampling from this space allows to cover semantically distinguish outcomes. The work focuses on predicting vehicle trajectories.

An intermediate layer avoids the need of taxonomy [?] by using metric learning, in which a latent representation is trained to match annotations of high-level labels, and forcing the distance to be large if they represent two distinguish semantic labels. The latent space is trained to match human similarity measures.

Past trajectories and map information are embedded, and their embeddings are passed to an LSTM whose latent space is divided intro a high- and low-level parts. The decoder takes both parts to produce trajectory samples. The trajectory network is a series of fully connected layers that embed a trajectory into a vector [2] [seems this work uses LSTMs for the embeddings]. The map embedding is a fully connected network that maps polynomial coefficients (quadratic) into an embedding. The encoder is a LSTM, whose hidden states are added a Gaussian noise and passed to a non-linear fully-connected network to compute the high and low-level embedding representation. The high-level embedding part is not correlated with the low-level one, and is trained for learning semantic similarities from the human teacher (they use a hand coded oracle though). The decoder is a LSTM. There is also a discriminator trained for identifying if samples are generated by the architecture or if they are real data.

The loss design incorporates minimal and final displacement losses, a term to enforce the non-correlation between the high and low-level embeddings, and another to enforce that semantically related pairs should also be close in the encoding space.

Sampling is performed using Farthest Point Sampling.

**Notes:** It is interesting that they added semantics to the network.

### 2022 - Controlling Assistive Robots with Learned Latent Actions.

[49] expands on [50] for assistive robotics. Here the latent space takes in consideration human input from a low-dimensional joystick, allowing to encode actions (e.g., cut, pour, dance) using buttons.

A Multi-Layer Perceptron (MLP) is used to learn an alignment between the user's intention/preferences and the context.

### 2022 - Promoting Quality and Diversity in Population-based Reinforcement Learning via Hierarchical Trajectory Space Exploration.

**Name: HTSE**

[56] propose a trajectory embedding using VAE and LSTM with similarity constrains, which is used with a hierarchical trajectory space exploration to generate diverse samples in a reinforcement learning framework.

The encoder is a double layer bi-directional LSTM, and the hidden state is formed by the last state of both encoding-LSTMs. The decoder is an one layer LSTM which take as input the first trajectory state and the hidden variable. The constraint is computed by sampling a batch of trajectories and ordering them according to [point location distance?], the closest one in the batch is the positive sample and the bottom half are negative samples, and a loss function is computed using the encodings of the anchor, positive and negative samples. A hidden-state conditioned policy is added, learning $\pi : z, s \rightarrow a$, which is trained together with the encoder-decoder.

### 2023 - SIRL: Similarity-Based Implicit Representation Learning.

**Name: SIRL**

[7] propose to ask humans what are similar trajectories (robotics manipulation), allowing to distinguish high- and low-level features for learning tasks. [sort of evolution of "learning one feature at a time".]. A trajectory query is a triplet of trajectories which are presented to the user, who is asked which are the two most similar, forming a tripled (anchor, positive and negative) [vae?].

The triples are used to learn am embedding space such similar trajectories are close in the representation space, and dissimilar ones are far apart. The features are learned using a fully connected NNs, which are trained based on the distance in the embedding space using a contrastive loss based on the human triplet selection.

## 2.5 Trajectory Prediction

### 2018 - Convolutional Social Pooling for Vehicle Trajectory Prediction.

**Name: CS-LSTM**

[19] Convolution Social LSTM (CS-LSTM) uses an LSTM encoder-decoder together with a convolutional social pooling for learning interdependencies between vehicles in the street, outputting multi-modal predictions of trajectories based on maneuver classes. The interdependencies are important because the decision possible trajectory of one car depends on the predicted trajectory of the other ones.

Tests are a simple simulator, maneuvers are `keep lane`, `change to right`, `change to left`, so it is a very simple test case.

The encoder is one LSTM for each vehicle, but all LSTMs have the same weights. The encoded state for the target vehicle is concatenated with an encoded state for all other vehicles, which is obtained by passing their LSTM states into the convolutional maxpooling. There is one decode for each maneuver for enabling multi-modality.

**Notes:** This interdependency is similar to the idea of the robot-human case.

**2018 - Social GAN: Socially Acceptable Trajectories With Generative Adversarial Networks**.

**SGAN**

[31] proposed an LSTM encoder-decoder with max-pooling layer for handling interdependencies (the trajectory of a person depends on the trajectory of others). The encoder decoder is trained in a GAN fashion, in which uses an LSTM-based discriminator.

There is one LSTM encoder, decoder and discriminator for each tracked individual. On the encoder, first the individual's position is passed through an 1 layer MLP to be transformed into a fixed-sized vector, which is then passed to the LSTM. The weights of the encoders are shared among all individuals. The pooling module converts the hidden-state of every encoding LSTM into tensor for each individual. The decoder is a straight-forward LSTM and the discriminator takes the predicted trajectories and classifies encodes them using LSTMs into "good" or "bad".

The pooling mechanism is what handles multiple people.

**2018 - 3DOF Pedestrian Trajectory Prediction Learned from Long-Term Autonomous Mobile Robot Deployment Data**.

[add to table]

**Name: T-Pose-LSTM**

[75] propose learning 3Degrees of Freedom (DoFs) (planar position and orientation) trajectories from lidar data. The approach computes an embedding for the observations which are passed to a 3 layer stacked LSTM, the stacks prevent short-memory from vanishing quickly.

[not sure if this goes to social robots or human trajectory prediction]

**2020 - CNN, Segmentation or Semantic Embeddings: Evaluating Scene Context for Trajectory Prediction**.

[add to table?]

**Name: RAE-PSPNet-emb**

[77] evaluates the use of scene embeddings, social embeddings, and scene segmentation when learning to predict pedestrian trajectories using LSTMs. Results indicate that using scene segmentation may be useful. [however it is not clear what they call their approach and the other ones used in comparison. Hard to really conclude anything from the paper.]

## 3  SOCIAL NAVIGATION

We find two big clusters of works on social robots.

The first, composed by early approaches which try to make a model humans behaviors in crowded spaces. The second is based on learning methods.

### 3.1  Human Models for Social Navigation

[check fernando2018soft+ for more methods to add here]

[check [20-26] in [80] also]

[check [58] for works that joint multi-modal information like facial expression and voice for predicting the intention, emotions and etc... probably fall out-of-scope here. I think this could be an interesting line to show in the survey]

[check [39] for modeling crowds with fluid dynamics and social invasiveness]

**1995 - Social force model for pedestrian dynamics**.

[33] is probably the earliest work on modeling human interactions for social navigation.

**2011 - Who are you with and where are you going?**.

[87] presents Social Force (SF) model

**2016 - Anticipating Human Activities Using Object Affordances for Reactive Robotic Response**.

[40]

## 3.2 Pedestrian Prediction

**2018 - Soft + Hardwired attention: An LSTM framework for human trajectory prediction and abnormal event detection**.

[24] proposes a framework which uses soft and hardwired attention mechanisms to predict human trajectories based on a brief history of the target's and its neighbors trajectories.

Trajectories for each pedestrian are encoded and decoded using LSTMs encoders and passes to the soft and hard attention layers to compute the final encoding, which is then decoded. The key idea is to use the distance to compute weights for the hard attention layer, since these are key features that influence in the trajectory.

The approach is also evaluated in computing abnormal trajectories based on the proposed encoding.

**2019 - Pedestrian Trajectory Prediction Using RNN Encoder-Decoder with Spatio-Temporal Attentions**.

[6] proposes using not only the humans trajectories, but also information from the scene for predicting trajectories, using an RNN for learning human-human and human-scene interactions, using attention mechanisms to find semantic alignment between the encoder and decoder.

Images as processed with a pre-trained CNN for extracting environment features and attention mechanisms are computed before the decoding part.

[this paper is very bad and should probably be kicked out]

**2019 - GD-GAN: Generative Adversarial Networks for Trajectory Prediction and Group Detection in Crowds**.

[25] proposes a framework which predicts trajectories and group memberships through clustering. It builds on top of [24], which uses LSTMs for learning an embedding for trajectories with attention mechanisms, here, the decoder is a GAN architecture with generators and discriminators being LSTMs. The embedding is passed through a t-Distributed Stochastic Neighbor Embedding (t-SNE) module for dimensionality reduction and a further clustering predicts group membership.

**2019 - The Trajectron: Probabilistic Multi-Agent Trajectory Modeling With Dynamic Spatiotemporal Graphs**.

[add to table]

[37] presents the trajectron, a graph-structured model for predicting many (multi-modal) trajectories of multiple agents simultaneously (NxN).

They say their model is the first one to use graph representation, which has advantages over RNNs since graphs naturally accept an arbitrary number of inputs [as many agents as you want], provide an intermediate representation and encourage model reuse.

First, past trajectories are encoded with LSTMs, which is then passed together with the LSTM-based "edge-encoders" for computing the influence of neighboring nodes and an additive attention module. The encoded values are passed to a cVAE and the decoder is also LSTM-based with a Gaussian Mixture Model (GMM) output which is sampled for generating multiple possible trajectories (achieving multi-modality).

They also propose a Kernel Density Estimate (KDE)-based evaluation metric, which is a bit more fair since multi-modal methods in the literature were generating thousands of trajectories and counting as a success if one of those hit the target.

Tested on datasets of ppl walking around.

**2020 - Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data**.

[67] presents a graph structured recurrent model for learning trajectories considering dynamics and environment constraints (maps) by extending [37] by adding support to multi-agents and heterogeneous data. The robot in question is an autonomous car traveling in a city.

The scene is represented as a graph, with nodes representing agents (cars and ppl), and edges representing interactions. The scene evolution is encoded by a LSTM and attention is used to balance the weights in the interactions. Finally a CNN is used to aggregate heterogeneous data from a map, and semantic semantic information (pedestrian crossing", "drivable area", "walkway"). Multi-modality is achieved through the use of cVAE [but it is not very clear where it is used].

**2020 - Transformer networks for trajectory forecasting**.

[29] Uses transformers to predict trajectories. Without considering social interactions, the model was able to compete with state of the art social modals like Trajectron++, showing the promise of transformers

**2020 - It Is Not the Journey But the Destination: Endpoint Conditioned Trajectory Prediction**.

**Name: PECNet**

[53] focus on learning long-range multi-modal trajectory prediction. They propose a social-pooling layer which allows for improving the diversity of the predicted social-compliant trajectories.

Sets of trajectories and destination points are embedded, and the embeddings are used by a VAE for computing predicted destination encoding, which is then passed to the social pooling for estimating which is the probable future destination. Finally, the future destination encoding is used to estimate the trajectory. [the paper is quite confusing, and at least to Anna's and Julian's experiments, the results on other datasets are much worse]

Tests are made with datasets of humans walking around.

[ —————————

The more or less line separating behaviour prediction instead of trajectory prediction.

————————— ]

**2020 - Vectornet: Encoding hd maps and agent dynamics from vectorized representation**.

[27] In this representation, agent trajectories and scene elements (walls, lane markers, etc.) are all represented as vectors in polylines (with class anotations). The use a lower level gnn to encode the polylines, and then a higher level gnn to encode the interaction between those polylines.

**2020 - Learning lane graph representations for motion forecasting**.

[45] In LaneCGN, in a first step, one uses 1D-CNN to encode the the agent trajectories along the time domain. Meanwhile, the environment is encoded using polylines respresenting lane segment (who are related to other lane segment by their positions, e.g., predecessor and successor along the same lane), who are represented as the average position of their start and end node as well as the distance between them that feed through separate MLPs and then added together as the representation of the lane segment. They then use a adjusted GNN, whose result are based on the addition of four different adjacency matrices, (based on the different relationships), which can be set to certain powers to look into the future.

Having encoded the agents and the lane node, attention (an MLP based on node position (average position for lanes, last position for agents) is used to go through the different connection of L-L, A-L, L-A, and A-A,

### 2021 - Human Trajectory Forecasting in Crowds: A Deep Learning Perspective.

### Name: DirectContact, TrajNet++

[41] presents a review on deep learning for human trajectory prediction, presents two methods for capturing social interaction and present TrajNet++, a benchmark for evaluating trajectory predictions. [quite a lot of things in one paper]

They focus on short-term trajectory prediction (5 sec), since the long term objective cannot be observed. They focus on the interactions of trajectory predictors, not on the predictor (LSTM) itself. Specifically, to handle a variable number of neighbors and how they collective influence one's trajectory.

A pipeline is composed of an encoder, followed by a social model, and then a decoder. Social model are classified in grid-based and non-grid-based, they propose a grid based method, using the velocities as observations, as it is natural for learning collision avoidance and leader-follower relations. For grid-less method, they propose the Layer-wise Relevance Propagation (LRP) which traces back which trajectories generate the prediction, and hence improving explainability.

[This paper should also appear on the survey sections since it has a really good and complete literature review and comparison.]

### 2021 - Tra2Tra: Trajectory-to-Trajectory Prediction With a Global Social Spatial-Temporal Attentive Neural Network.

[add to the table]

[86] proposes a global social spatial-temporal attentive NN. It separates spatial interaction features with a ?decentralization operation? and attention mechanism, which is then passed to a LSTM, which is then passed together with velocities an auto-encoder-decoder. Multi-modality is achieved by adding a random noise before decoding.

The decentralization subtracts the average positions from the trajectory. The attention is a standard attention which gives weights for the features. The LSTM are also standard, and their output is concatenated with velocity information.

tested on datasets with ppl walking around.

### 2021 - FloMo: Tractable Motion Prediction with Normalizing Flows.

[69] This work uses normalizing flows to make predictions, as well as gnn based encoding of surrounding agents.

### 2021 - From goals, waypoints & paths to long term human trajectory forecasting.

[52] This approach builds on the following steps, under the assumption that an image is availble with pixel positions $x_{i,j}$ for pixel i,j.

- Create heatmap $H$ for an agents position $x_t$ at timestep $t$, with

$$h_{t,i,j} = 2 \frac{\left\| x_t - x_{i,j} \right\|}{\max\limits_{k,l} \left\| x_t - x_{k,l} \right\|} \tag{1}$$

- Create segmentation map using U-Net, which is pretrained on hand-labeled maps from the training set
- The semantic map and trajectory heatmap are concatenated and encoded in a U-Net encoder.

### 2021 - AgentFormer: Agent-Aware Transformers for Socio-Temporal Multi-Agent Forecasting.

[92] First model proposing the use of transformers to learn the joint future trajectories of all agents involved in a scene. Uses only the agents positions and velocities as input (plus a timestep encoding based on sin/cos function with varying wavelengths).

**2022 - Social-PatteRNN: Socially-Aware Trajectory Prediction Guided by Motion Patterns**.

[59]

**2022 - Stochastic trajectory prediction via motion indeterminacy diffusion**.

[30] This paper uses Denoising Diffusion process for predictiong trajectories, using a transformer based decoder to predict the respective added noise at each step, while using Trajectron++ to encode the past observations. Consequently, this work provides no new encoding method.

**2022 - HiVT: Hierarchical Vector Transformer for Multi-Agent Motion Prediction**.

[100] In this paper, the encoding will be done in multiple steps.

- Extract for each agent its surrounding region (other agents and lanes) and encoded the corresponding past observations, using a coordinate system that is normalized to this agent last observed position and velocity, where we embed the agents own trajectory as well as the surrounding agents trajectory.
- We apply multi head cross attention with the agent as the query and its neighbors as key and values, and a recursive layer afterwards.
- The spatial encodings are then encoded temporally using a time axis transformer block
- Similar to the encoding of the neighbors, the encoded agent position is enhanced by cross attention with the neighboring lane segments (here, each vector in a polyline is treated individually)
- An transformer is used to encode the interactions with each agent, where their states are expanded by the translations and rotations between the coordinate systems.
- The decoder used is simple, however, the joint prediction are not really joint, just parallel

Here, using for the local encoding two different transformers, first along the agent axis and then along the time axis, significantly reduces the computational cost of the model.

**2022 - Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction**.

[81] In this paper, the authors propose replacing the attention mechanisms with something called *multi-context gating*, which basically updates multiple inputs with a shared context vector.

**2022 - Path-aware graph attention for hd maps in motion prediction**.

[18] In this paper, they suggest PAGA, which is an improvement on LaneCGNB [45]. Namely, the update the method for calculating the attention weights in the GNN used to encode the map. Instead of simply using an the four different adjacency matrices, the replaces that with a more complex matrix able to include connection via intermediate steps.

**2022 - Socialvae: Human trajectory prediction using timewise latents**.

[85] The paper uses relatively normal combination of rnn and attention to encode the past, its most important contribution comes in the recursive reasoning of the decoder.

**2022 - Motion transformer with global intention localization and local movement refinement**.

[71] This is a transformer based model. Here, this model uses a polyline based map representation [27], but instead of using an inner gnn, they use a combination of mlp and max pooling, and instead of the outer gnn, a transformer is used. While the transformer does return encodings for each agent, the input positions are normalized around a single agent, which therefore becomes the target of the prediction

**2023 - CSR: Cascade Conditional Variational Auto Encoder with Socially-aware Regression for Pedestrian Trajectory Prediction**.

[97]

**2023 - MRGTraj: A Novel Non-Autoregressive Approach for Human Trajectory Prediction**.

[63]

**2023 - Gatraj: A graph-and attention-based multi-agent trajectory prediction model**.

[15] This model applies first an 1D CNN over the time-axis of the recorded data. Then, it applies a 2-layer MLP over the position axis, and then applies 3 layers of self attention accross all agents along the combined agent-time axes, and then lstm along the time axes. Afterwards, a recurrent graph neural network is applied on those encoded agents trajectories.

During training, the prediction loss is only applied to the best predicted trajectory ('Winner-takes-all', which is said to be more resilient against mode collaps)

**2023 - EWareNet: Emotion-Aware Pedestrian Intent Prediction and Adaptive Spatial Profile Fusion for Social Robot Navigation**.

[58]

**2023 - Fend: A future enhanced distribution-aware contrastive learning framework for long-tail trajectory prediction**.

[84]

- Combined past and future trajectories in a dataset are encoded using 1D-CNN and LSTM. These encoded trajectories are then clustered using k-means.
- Given those labels, one can use them to adjust any arbitrary model. Namely, during training, one uses Prototypical Contrastive Learning losses on the encoded trajectories (after feeding them through a MLP to get to the cluster space), which encourages samples belonging to the same cluster to stick closer together. It is unclear if the prototypes (i.e., the means) are here taken as the ones from the initial clustering, or if they are recalculated as the means of all datapoints belonging to the cluster after every batch, although the latter seems to be applied, although only calculated for the current samples in the batch.
- After training, one can calculate the prototypes based on the whole datasets
- During inference, one can assign a label based on the known prototypes, and then use slight varied weights (either completely new, or slightly multiplied in certain structures such as LSTMs) to make more targeted predictions

**2023 - ForceFormer: exploring social force and transformer for pedestrian trajectory prediction**.

[95] This is model based on AgentFormer [92], expanding it by not only providing positions and velocities, but expands it with an estimated goal of the agent, as well as a repulsive force part. This combination has comparable accuracy, but is able to lower collision rates.

**2023 - Wayformer: Motion forecasting via simple & efficient attention networks**.

[60] In this work, they use transformers on the concatenated data of the agent, its surrounding agents as well as further context information to get improved results. Additionally, they use a reduced dimension latent key query in the first attention layer to allow for faster training/inference

**2023 - Adapt: Efficient multi-agent trajectory prediction with adaptation**.

[3] Uses polyline representation based on [27] for environment. They use the same inner gnn, but instead of the outer gnn, they use the appraoch proposed by LaneCGN [45]. Given the for types of connections (lane to lane, lane to agent, agent to lane, agent to agent), instead of putting each type through a number of attention layeres, here, each type is put through one attention layer, before the process is repeated n times. Compared to the previous LaneCGN, they also uses multi-head attention blocks (i.e., standard transformers) instead of simpler multi-head attention.

**2023 - Trajectory unified transformer for pedestrian trajectory prediction**.

[70] The have the approach of clustering the future trajecotries in the dataset, and then emebedding each cluster and adding that to the embiddings of the past trajectories to create a multimodal predictions. Thos are then feed through a transformer.

**2023 - Eigentrajectory: Low-rank descriptors for multi-modal trajectory forecasting**.

[4] They extract prinicial components from the dataset for past and future trajectories, and project the recorded trajectories onto them. This is can then be used as a pre- and post-processing (inverse PCA for the latter) for any arbitrary prediction model, increasing the performance in two test cases.

**2023 - Prophnet: Efficient agent-centric motion forecasting with anchor-informed proposals**.

[83]

- In a first step, the target agent, neighbors and polylines are encoded using 3 different gated MLPs (some RNN)
- One learns queris, which are combined in cross attention to the encoding of the target agent (i.e., the hidden states of all timesteps are used there).
- the concatenated encodings of all agents (only last timesteps) and lanes are encoded using self attention between the objects, and the so encoded target agent encoding is also used.

**2024 - TrajFlow: Learning Distributions over Trajectories for Human Behavior Prediction**.

[55] Based on [69], it adds an RNN autoencoder to give the model better accuracy and greater flexibility.

**2024 - MapFlow: Multi-Agent Pedestrian Trajectory Prediction Using Normalizing Flow**.

[74] This is seemingly more or less the same as TrajFlow [55], except written much worse and a method for context encoding which is highly unclear.

**2024 - Hpnet: Dynamic trajectory forecasting with historical prediction attention**.

[79]

**2024 - Real-Time Motion Prediction via Heterogeneous Polyline Transformer with Relative Pose Encoding**.

[96] In this work, the authors use a polyline representation similar to [27]. However, compared to the approach they split each polyline into two parts: A beginning state (position and orientation), and the a self normalized vector representation of the polyline (i.e., those polyines start at the origin with angle 0). One then uses the lower level GNN to encode those polylines. One then extracts the $k$-closests polylines (i.e, their concatenated state and ecnoded polyline) around the target agent to predict, which are then filtered pushed through four levels of transformers. The beginiing states of those polylines are represented in a coordinate system that is normalized around the target agent

**2024 - MTR++: Multi-Agent Motion Prediction With Symmetric Scene Modeling and Guided Intention Querying**.

[72] This is an improvement upon a previous work [71], which removes the agent centric approach and therefore allows for the prediction of all agents simultaneously. However, the encoder part stays unchanged.

## 3.3 Social Robots

**2014 - Inverse Reinforcement Learning algorithms and features for robot navigation in crowds: An experimental comparison**.

[82] Learning from demonstrations, they define a good number of hard-coded features, like social force ones.

[no encoding, out-of-scope]

16

**2016 - Learning socially normative robot navigation behaviors with bayesian inverse reinforcement learning**.

[61] Learns behaviors from demonstrations using a control graph-structured representation.

[no encoding, out-of-scope]

**2016 - Socially compliant mobile robot navigation via inverse reinforcement learning**.

[42] use a Hamiltonian Markov chain Monte Carlo sampling for learning trajectories from demonstrations or pedestrians. The map is a parsed into a Voronoi-based graph for improving efficiency.

[no encoding, out-of-scope]

**2017 - DESIRE: Distant Future Prediction in Dynamic Scenes With Interacting Agents**.

[add to table]

**Name: DESIRE**

[43] proposes a deep stochastic inverse optimal controller RNN encoder-decoder for multiple agents considering multi-modal predictions, which uses the past trajectories and the scene for making prediction.

The model first encodes trajectories with RNNs, and then creates predictions with a cVAE before being decoded by another RNN. Then a RNN-based refinement module, which also uses a CNN feature scene extractor is used for refining the score of the generated trajectories.

**2017 - Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning**.

[13] A little bit similar with [12], the reward function is a bit simpler, and here they do not handle multi-agents.

The idea is to learn one policy, which is used by multiple agents, and having the social behaviors emerge from it.

[no encoding, out-of-scope]

**2017 - Socially aware motion planning with deep reinforcement learning**.

**Name: SA-CADRL**

[12] (a bit similar to [13]) proposes a motion planning using Deep reinforcement Learning (DRL) which focus on what **not to do** instead of trying to learn all possible socially acceptable rules. The approach also handles cases with multiple agents using a ?symetrical NN?. The idea behind this is that each agent has a simple collision avoidance mechanism from which social patterns emerge. This is called reciprocity, and is mathematically captured as both agents having the same policy ($\pi = \tilde{\pi}$).

Some social norms (like passing through the right) is motivated through reward manipulation in the loss [kind of cheating]

To handle multiple agents, they make a fully connected NN with maxpooling layers, so the number of agents is hard-encoded in the network.

The approach is tested on a real robot.

[no encoding, out-of-scope]

**2018 - Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning**.

[add to table]

**Name: GA3C-CADRL**

[22] propose a DRL framework for robots interacting and cooperating with humans walking around. They propose an LSTM based approach which can observe an arbitrary number of agents (1xN)

The policy observes the distance of the agent w.r.t. to the goal and other agents, and learns an Asynchronous Advantage Actor-Critic (A3C) policy which computes actions for collision avoidance. The trajectory of the distance from agents is encoded with an LSTM, which is then passed to the A3C policy.

Tested with simulations and also with real robot.

**2018 - Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning**.

[48] proposes learning a policy for decentralized collision avoidance. The policy is trained in a multi-agent [simulation?] using policy gradient and RL. The policy gets velocities and sensor data, which are passed to a CNN for computing actions. Tested in simulation. [not really using encoding, out-of-scope]

**2020 - Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data**.

[67] presents a graph structured recurrent model for learning trajectories considering dynamics and environment constraints (maps) by extending [37] by adding support to multi-agents and heterogeneous data. The robot in question is an autonomous car traveling in a city. [[37] already handles multiple agents]

The scene is represented as a graph, with nodes representing agents (cars and ppl), and edges representing interactions. The scene evolution is encoded by a LSTM and attention is used to balance the weights in the interactions. Finally a CNN is used to aggregate heterogeneous data from a map, and semantic semantic information (pedestrian crossing", "drivable area", "walkway"). Multi-modality is achieved through the use of cVAE [but it is not very clear where it is used].

**2020 - A Generative Approach for Socially Compliant Navigation**.

**NaviGAN**

[80] focus on learning human-compliant behaviors for robots navigating in human crowded environments, by optimizing trajectories both for comfort (the absence of annoyance and stress for humans) and naturalness (the similarity between robots and humans).

They state "reinforcement learning approaches tend to optimize on the comfort aspect of the socially compliant navigation, whereas the inverse reinforcement learning approaches are designed to achieve natural behavior."

A LSTM encoder-decoder is defined for each of three social-interaction force (intention, social interaction and fluctuation) from [33] to improves interpretability, which are used with a adversarial training to reduce the data-bias tendency of LSTMs.

The trajectories of the robot and other agents (humans) are encoded with LSTMs, for intention just the robot trajectory is encoded, for social interaction and fluctuation all trajectories are encoded and passed through a maxpooling layer before the LSTM. Both encoding are passed to an adversarial training using demonstrations.

**2021 - Learning World Transition Model for Socially Aware Robot Navigation**.

[add to table]

[17] propose learning a world transition model from 2D laser scans. The model is used in simulation for learning a policy using a lot of simulated data and few real data.

The laser data is transformed into an obstacle map which is said to better differentiate static from moving obstacles, and the moving ones are encoded with LSTMs.

In the world simulation, a shared policy is sampled by each agent, meaning that the pedestrian predictors are aware about the robot (NxN).

The approach is tested also with the real robot.

**2021 - Probabilistic Dynamic Crowd Prediction for Social Navigation**.

[add to table]

[39] proposed to predict the crowd flow in a macroscopic way using density and velocities, which are passed to a RNN for making predictions which are used for robot planning.

They use a social invasiveness metric which depends on the total number of interactions in the planning and relative velocity magnitudes.

The RNN is convolutional, between each RNN layer there are convolution and down-sampling layers (upsampling for the decoder), which is used to predict the trajectories of each crowd group, represented by density, velocity and velocity variance.

Then planning is made minimizing the invasiveness given the crowds predictions and robot dynamics.

**2021 - Trajectory Prediction for Autonomous Driving based on Multi-Head Attention with Joint Agent-Map Representation**.

[54] proposes a method which considers the map and multi-agents to predict trajectories, as one influences the other using multi-head attention mechanisms for dealing with multi-modality.

The map is processed by a CNN and the trajectory of each agent in the scene (cars) is encoded with an LSTM, forming the context embedding, which is fed to multiple attention heads. The agent of interest's trajectory is also encoded with an LSTM, which is concatenated with the attention heads output to be decoded by LSTMs (one per attention head) to obtain trajectory predictions.

Tests are made on datasets of cars running on roads.

**2023 - EWareNet: Emotion-Aware Pedestrian Intent Prediction and Adaptive Spatial Profile Fusion for Social Robot Navigation**.

**Name: EWareNet**

[58] presents pedestrian intent using a transformer model from RGB images which is integrated with path navigation schemes. The pedestrian skeleton is passed through a convolutional encoder-decoder, and its image is used with a CNN for detecting emotion. Both parts are decoded fo predicting the trajectory, which is then used in a path-planning for the robot.

[not sure if this is out-of-the-scope]

## 4  DATASETS AND BENCHMARKS

### 4.1  Datasets

From [86]:

    **41 ETH:**  2 pedestrians walking

    **42 UCY:**  2 pedestrian walking

    **43 Collisions:**  Many simulations of balls rolling on a 2D plane without friction

    **44 NGSim:**  Freeway data with vehicles.

    **45 Charges:**  Scenes of electric charges moving on an electric field

From [37, 77]:

    **41 ETH:**  2 pedestrians walking

    **42 UCY:**  2 pedestrian walking

From [43]:

**12 KITTI Raw Data:** scenes of driving scenes with 3D velodyne laser scans

**36 Stanford Drone Dataset** videos captures with a drone flying over an university

From [75]:

**10 STRANDS** long-term robot deployment in a care facility for 19 weeks where the robot was performing tasks and interacting with humans, which recorded a 2D upped body laser scan for tracking people.

**19 L-CAS** Data collected in a university, using a 3D LiDAR for tracking humans

**9 SDD** Similar to ETH and UCY, 2D poses of humans ([need more info])

**2016 - Learning Social Etiquette: Human Trajectory Understanding In Crowded Scenes.**

[66] provide a benchmark for social navigation with interactions between pedestrians, skaters, bikers, and small vehicles. The benchmark contain images (videos) of these interactions in a campus.

A feature called social sensitivity is proposed, which incorporates a distance which the target prefers for avoiding collision and another in which the targets starts to deviate from its trajectory to avoid collision. These parameters are learned with an energy-like minimization. When plotting these parameters, clusters emerge with different navigation styles.

## 4.2 Simulators

From [61]:

**22** open source pedestrian simulator governed by social force model

From [39]:

**2 Pedsim** pedestrian simulator used for generating dense pedestrian maps

## 4.3 Benchmarks

## ACKNOWLEDGMENTS

## REFERENCES

[1] Usman Ahmed, Jerry Chun-Wei Lin, and Gautam Srivastava. 2023. Deep Fuzzy Contrast-Set Deviation Point Representation and Trajectory Detection. *IEEE Transactions on Fuzzy Systems* 31, 2 (2023), 571–581. https://doi.org/10.1109/TFUZZ.2022.3197876

[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 961–971. https://doi.org/10.1109/CVPR.2016.110

[3] Görkay Aydemir, Adil Kaan Akan, and Fatma Güney. 2023. Adapt: Efficient multi-agent trajectory prediction with adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Paris, France). IEEE Computer Society, Los Alamitos, CA, USA, 8295–8305. https://doi.org/10.1109/ICCV51070.2023.00762

[4] Inhwan Bae, Jean Oh, and Hae-Gon Jeon. 2023. Eigentrajectory: Low-rank descriptors for multi-modal trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Paris, France). IEEE Computer Society, Los Alamitos, CA, USA, 10017–10029. https://doi.org/10.1109/ICCV51070.2023.00919

[5] E Belogay, C Cabrelli, U Molter, and R Shonkwiler. 1997. Calculating the Hausdorff distance between curves. *Inform. Process. Lett.* 64, 1 (1997), 17–22.

[6] Niraj Bhujel, Eam Khwang Teoh, and Wei-Yun Yau. 2019. Pedestrian Trajectory Prediction Using RNN Encoder-Decoder with Spatio-Temporal Attentions. In *2019 IEEE 5th International Conference on Mechatronics System and Robots (ICMSR)* (Singapore). IEEE Computer Society, Los Alamitos, CA, USA, 110–114. https://doi.org/10.1109/ICMSR.2019.8835478

[7] Andreea Bobu, Yi Liu, Rohin Shah, Daniel S. Brown, and Anca D. Dragan. 2023. SIRL: Similarity-Based Implicit Representation Learning. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction* (Stockholm, Sweden) *(HRI '23)*. Association for Computing Machinery, New York, NY, USA, 565–574. https://doi.org/10.1145/3568162.3576989

[8] Yanchuan Chang, Jianzhong Qi, Yuxuan Liang, and Egemen Tanin. 2023. Contrastive Trajectory Similarity Learning with Dual-Feature Attention. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)* (Anaheim, CA, USA). IEEE Computer Society, Los Alamitos, CA, USA, 2933–2945. https://doi.org/10.1109/ICDE55515.2023.00224

[9] Lei Chen and Raymond Ng. 2004. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30* (Toronto, Canada) *(VLDB '04)*. VLDB Endowment, 792–803.

[10] Lei Chen, M. Tamer Özsu, and Vincent Oria. 2005. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data* (Baltimore, Maryland) *(SIGMOD '05)*. Association for Computing Machinery, New York, NY, USA, 491–502. https://doi.org/10.1145/1066157.1066213

[11] Yuanyi Chen, Peng Yu, Wenwang Chen, Zengwei Zheng, and Minyi Guo. 2022. Embedding-Based Similarity Computation for Massive Vehicle Trajectory Data. *IEEE Internet of Things Journal* 9, 6 (2022), 4650–4660. https://doi.org/10.1109/JIOT.2021.3107327

[12] Yu Fan Chen, Michael Everett, Miao Liu, and Jonathan P. How. 2017. Socially aware motion planning with deep reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 1343–1350. https://doi.org/10.1109/IROS.2017.8202312

[13] Yu Fan Chen, Miao Liu, Michael Everett, and Jonathan P. How. 2017. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 285–292. https://doi.org/10.1109/ICRA.2017.7989037

[14] Ziwen Chen, Ke Li, Silin Zhou, Lisi Chen, and Shuo Shang. 2022. Towards robust trajectory similarity computation: Representation-based spatio-temporal similarity quantification. *World Wide Web* 26, 4 (aug 2022), 1271–1294. https://doi.org/10.1007/s11280-022-01085-4

[15] Hao Cheng, Mengmeng Liu, Lin Chen, Hellward Broszio, Monika Sester, and Michael Ying Yang. 2023. Gatraj: A graph-and attention-based multi-agent trajectory prediction model. *ISPRS Journal of Photogrammetry and Remote Sensing* 205 (2023), 163–175.

[16] John Co-Reyes, YuXuan Liu, Abhishek Gupta, Benjamin Eysenbach, Pieter Abbeel, and Sergey Levine. 2018. Self-Consistent Trajectory Autoencoder: Hierarchical Reinforcement Learning with Trajectory Embeddings. In *Proceedings of the 35th International Conference on Machine Learning* (Stockholm, Sweden) *(Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). International Machine Learning Society (IMLS), 1009–1018. https://proceedings.mlr.press/v80/co-reyes18a.html

[17] Yuxiang Cui, Haodong Zhang, Yue Wang, and Rong Xiong. 2021. Learning World Transition Model for Socially Aware Robot Navigation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)* (Xi'an, China). IEEE Computer Society, Los Alamitos, CA, USA, 9262–9268. https://doi.org/10.1109/ICRA48506.2021.9561973

[18] Fang Da and Yu Zhang. 2022. Path-aware graph attention for hd maps in motion prediction. In *2022 IEEE International Conference on Robotics and Automation (ICRA)* (Philadelphia, USA). IEEE Computer Society, Los Alamitos, CA, USA, 6430–6436. https://doi.org/10.1109/ICRA46639.2022.9812100

[19] Nachiket Deo and Mohan M. Trivedi. 2018. Convolutional Social Pooling for Vehicle Trajectory Prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (Salt Lake City, UT, USA). IEEE Computer Society, Los Alamitos, CA, USA, 1549–15498. https://doi.org/10.1109/CVPRW.2018.00196

[20] Jiaxin Ding, Bowen Zhang, Xinbing Wang, and Chenghu Zhou. 2022. TSNE: Trajectory Similarity Network Embedding. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems* (Seattle, Washington) *(SIGSPATIAL '22)*. Association for Computing Machinery, New York, NY, USA, Article 86, 4 pages. https://doi.org/10.1145/3557915.3561022

[21] Thomas Eiter and Heikki Mannila. 1994. Computing discrete Fréchet distance. (05 1994).

[22] Michael Everett, Yu Fan Chen, and Jonathan P. How. 2018. Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 3052–3059. https://doi.org/10.1109/IROS.2018.8593871

[23] Ziquan Fang, Yuntao Du, Xinjun Zhu, Danlei Hu, Lu Chen, Yunjun Gao, and Christian S. Jensen. 2022. Spatio-Temporal Trajectory Similarity Learning in Road Networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 347–356. https://doi.org/10.1145/3534678.3539375

[24] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. 2018. Soft + Hardwired attention: An LSTM framework for human trajectory prediction and abnormal event detection. *Neural Networks* 108 (2018), 466–478. https://doi.org/10.1016/j.neunet.2018.09.002

[25] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. 2019. GD-GAN: Generative Adversarial Networks for Trajectory Prediction and Group Detection in Crowds. In *Computer Vision – ACCV 2018*, C. V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler (Eds.). Springer International Publishing, Cham, 314–330. https://doi.org/10.1007/978-3-030-20887-5_20

[26] Tao-Yang Fu and Wang-Chien Lee. 2020. Trembr: Exploring Road Networks for Trajectory Representation Learning. *ACM Trans. Intell. Syst. Technol.* 11, 1, Article 10 (feb 2020), 25 pages. https://doi.org/10.1145/3361741

[27] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. 2020. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *IEEE/CVF conference on computer vision and pattern recognition* (Seattle, USA). IEEE Computer Society, Los Alamitos, CA, USA, 11525–11533. https://doi.org/10.1109/CVPR42600.2020.01154

[28] Qiang Gao, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Xucheng Luo, and Fengli Zhang. 2017. Identifying Human Mobility via Trajectory Embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (Melbourne, Australia) *(IJCAI'17)*. AAAI Press,

1689–1695. https://doi.org/10.24963/ijcai.2017/234

[29] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. 2020. Transformer networks for trajectory forecasting. In *2020 25th International conference on pattern recognition (ICPR)* (Milan, Italy). IEEE Computer Society, Los Alamitos, CA, USA, 10335–10342. https://doi.org/10.1109/ICPR48806.2021.9412190

[30] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. 2022. Stochastic trajectory prediction via motion indeterminacy diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (New Orleans, USA). IEEE Computer Society, Los Alamitos, CA, USA, 17113–17122. https://doi.org/10.1109/CVPR52688.2022.01660

[31] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. 2018. Social GAN: Socially Acceptable Trajectories With Generative Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 2255–2264. https://doi.org/10.1109/CVPR.2018.00240

[32] Peng Han, Jin Wang, Di Yao, Shuo Shang, and Xiangliang Zhang. 2021. A Graph-Based Approach for Trajectory Similarity Computation in Spatial Networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 556–564. https://doi.org/10.1145/3447548.3467337

[33] Dirk Helbing and Péter Molnár. 1995. Social force model for pedestrian dynamics. *Phys. Rev. E* 51 (May 1995), 4282–4286. Issue 5. https://doi.org/10.1103/PhysRevE.51.4282

[34] Danlei Hu, Lu Chen, Hanxi Fang, Ziquan Fang, Tianyi Li, and Yunjun Gao. 2023. Spatio-Temporal Trajectory Similarity Measures: A Comprehensive Survey and Quantitative Study. https://doi.org/10.48550/arXiv.2303.05012 arXiv:2303.05012 [cs.DS]

[35] Xin Huang, Stephen G. McGill, Jonathan A. DeCastro, Luke Fletcher, John J. Leonard, Brian C. Williams, and Guy Rosman. 2020. DiversityGAN: Diversity-Aware Vehicle Motion Prediction via Latent Semantic Sampling. *IEEE Robotics and Automation Letters* 5, 4 (2020), 5089–5096. https://doi.org/10.1109/LRA.2020.3005369

[36] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. 1993. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 9 (1993), 850–863. https://doi.org/10.1109/34.232073

[37] B. Ivanovic and M. Pavone. 2019. The Trajectron: Probabilistic Multi-Agent Trajectory Modeling With Dynamic Spatiotemporal Graphs. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 2375–2384. https://doi.org/10.1109/ICCV.2019.00246

[38] Oliver Jesorsky, Klaus J. Kirchberg, and Robert W. Frischholz. 2001. Robust Face Detection Using the Hausdorff Distance. In *Audio- and Video-Based Biometric Person Authentication*, Josef Bigun and Fabrizio Smeraldi (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 90–95.

[39] Stefan H. Kiss, Kavindie Katuwandeniya, Alen Alempijevic, and Teresa Vidal-Calleja. 2021. Probabilistic Dynamic Crowd Prediction for Social Navigation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)* (Xi'an, China). IEEE Computer Society, Los Alamitos, CA, USA, 9269–9275. https://doi.org/10.1109/ICRA48506.2021.9561053

[40] Hema S. Koppula and Ashutosh Saxena. 2016. Anticipating Human Activities Using Object Affordances for Reactive Robotic Response. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 1 (2016), 14–29. https://doi.org/10.1109/TPAMI.2015.2430335

[41] Parth Kothari, Sven Kreiss, and Alexandre Alahi. 2022. Human Trajectory Forecasting in Crowds: A Deep Learning Perspective. *IEEE Transactions on Intelligent Transportation Systems* 23, 7 (2022), 7386–7400. https://doi.org/10.1109/TITS.2021.3069362

[42] Henrik Kretzschmar, Markus Spies, Christoph Sprunk, and Wolfram Burgard. 2016. Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research* 35, 11 (2016), 1289–1307. https://doi.org/10.1177/0278364915619772 arXiv:https://doi.org/10.1177/0278364915619772

[43] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B. Choy, Philip H. S. Torr, and Manmohan Chandraker. 2017. DESIRE: Distant Future Prediction in Dynamic Scenes With Interacting Agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[44] Xiucheng Li, Kaiqi Zhao, Gao Cong, Christian S. Jensen, and Wei Wei. 2018. Deep Representation Learning for Trajectory Similarity Computation. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)* (Paris, France). IEEE Computer Society, Los Alamitos, CA, USA, 617–628. https://doi.org/10.1109/ICDE.2018.00062

[45] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. 2020. Learning lane graph representations for motion forecasting. In *Computer Vision–ECCV 2020, Proceedings* (Glasgow, UK). Springer, 541–556. https://doi.org/10.1007/978-3-030-58536-5_32

[46] Bin Lin and Jianwen Su. 2005. Shapes based trajectory queries for moving objects. In *Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems* (Bremen, Germany) *(GIS '05)*. Association for Computing Machinery, New York, NY, USA, 21–30. https://doi.org/10.1145/1097064.1097069

[47] Yan Lin, Huaiyu Wan, Shengnan Guo, and Youfang Lin. 2022. Contrastive Pre-training of Spatial-Temporal Trajectory Embeddings. https://doi.org/10.48550/arXiv.2207.14539 arXiv:2207.14539 [cs.CV]

[48] Pinxin Long, Tingxiang Fan, Xinyi Liao, Wenxi Liu, Hao Zhang, and Jia Pan. 2018. Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 6252–6259. https://doi.org/10.1109/ICRA.2018.8461113

[49] Dylan P Losey, Hong Jun Jeon, Mengxi Li, Krishnan Srinivasan, Ajay Mandlekar, Animesh Garg, Jeannette Bohg, and Dorsa Sadigh. 2022. Learning latent actions to control assistive robots. *Autonomous robots* 46, 1 (2022), 115–147. https://doi.org/10.1007/s10514-021-10005-w

[50] Dylan P. Losey, Krishnan Srinivasan, Ajay Mandlekar, Animesh Garg, and Dorsa Sadigh. 2020. Controlling Assistive Robots with Learned Latent Actions. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (Paris, France). IEEE Computer Society, Los Alamitos, CA, USA,

378–384. https://doi.org/10.1109/ICRA40945.2020.9197197

[51] David Maier. 1978. The Complexity of Some Problems on Subsequences and Supersequences. *J. ACM* 25, 2 (apr 1978), 322–336. https://doi.org/10.1145/322063.322075

[52] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. 2021. From goals, waypoints & paths to long term human trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, Canada). IEEE Computer Society, Los Alamitos, CA, USA, 15233–15242. https://doi.org/10.1109/ICCV48922.2021.01495

[53] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. 2020. It Is Not the Journey But the Destination: Endpoint Conditioned Trajectory Prediction. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 759–776. https://doi.org/10.1007/978-3-030-58536-5_45

[54] Kaouther Messaoud, Nachiket Deo, Mohan M. Trivedi, and Fawzi Nashashibi. 2021. Trajectory Prediction for Autonomous Driving based on Multi-Head Attention with Joint Agent-Map Representation. In *2021 IEEE Intelligent Vehicles Symposium (IV)* (Nagoya, Japan). IEEE Computer Society, Los Alamitos, CA, USA, 165–170. https://doi.org/10.1109/IV48863.2021.9576054

[55] Anna Mészáros, Julian F Schumann, Javier Alonso-Mora, Arkady Zgonnikov, and Jens Kober. 2024. TrajFlow: Learning Distributions over Trajectories for Human Behavior Prediction. In *2024 IEEE Intelligent Vehicles Symposium (IV)* (Jeju Island, Republic of Korea). IEEE Computer Society, Los Alamitos, CA, USA, 184–191. https://doi.org/10.1109/IV55156.2024.10588386

[56] Jiayu Miao, Tianze Zhou, Kun Shao, Ming Zhou, Weinan Zhang, Jianye Hao, Yong Yu, and Jun Wang. 2022. Promoting Quality and Diversity in Population-based Reinforcement Learning via Hierarchical Trajectory Space Exploration. In *2022 International Conference on Robotics and Automation (ICRA)* (Philadelphia, PA, USA). IEEE Computer Society, Los Alamitos, CA, USA, 7544–7550. https://doi.org/10.1109/ICRA46639.2022.9811888

[57] Brendan Morris and Mohan Trivedi. 2009. Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. *2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009* (2009), 312 – 319. https://doi.org/10.1109/CVPRW.2009.5206559 Cited by: 200.

[58] Venkatraman Narayanan, Bala Murali Manoghar, Rama Prashanth RV, and Aniket Bera. 2023. EWareNet: Emotion-Aware Pedestrian Intent Prediction and Adaptive Spatial Profile Fusion for Social Robot Navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* (London, United Kingdom). IEEE Computer Society, Los Alamitos, CA, USA, 7569–7575. https://doi.org/10.1109/ICRA48891.2023.10161504

[59] Ingrid Navarro and Jean Oh. 2022. Social-PatteRNN: Socially-Aware Trajectory Prediction Guided by Motion Patterns. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Kyoto, Japan). IEEE Computer Society, Los Alamitos, CA, USA, 9859–9864. https://doi.org/10.1109/IROS47612.2022.9981486

[60] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. 2023. Wayformer: Motion forecasting via simple & efficient attention networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* (London, UK). IEEE Computer Society, Los Alamitos, CA, USA, 2980–2987. https://doi.org/10.1109/TPAMI.2024.3352811

[61] Billy Okal and Kai O. Arras. 2016. Learning socially normative robot navigation behaviors with Bayesian inverse reinforcement learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2889–2895. https://doi.org/10.1109/ICRA.2016.7487452

[62] Nikos Pelekis, Ioannis Kopanakis, Gerasimos Marketos, Irene Ntoutsi, Gennady Andrienko, and Yannis Theodoridis. 2007. Similarity Search in Trajectory Databases. In *14th International Symposium on Temporal Representation and Reasoning (TIME'07)* (Alicante, Spain). IEEE Computer Society, Los Alamitos, CA, USA, 129–140. https://doi.org/10.1109/TIME.2007.59

[63] Yusheng Peng, Gaofeng Zhang, Jun Shi, Xiangyu Li, and Liping Zheng. 2023. MRGTraj: A Novel Non-Autoregressive Approach for Human Trajectory Prediction. *IEEE Transactions on Circuits and Systems for Video Technology* 34, 4 (2023), 2318–2331. https://doi.org/10.1109/TCSVT.2023.3307442

[64] Lucas May Petry, Camila Leite Da Silva, Andrea Esuli, Chiara Renso, and Vania Bogorny. 2020. MARC: a robust method for multiple-aspect trajectory classification via space, time, and semantic embeddings. *International Journal of Geographical Information Science* 34, 7 (2020), 1428–1450. https://doi.org/10.1080/13658816.2019.1707835 arXiv:https://doi.org/10.1080/13658816.2019.1707835

[65] Sayan Ranu, Deepak P, Aditya D. Telang, Prasad Deshpande, and Sriram Raghavan. 2015. Indexing and matching trajectories under inconsistent sampling rates. In *2015 IEEE 31st International Conference on Data Engineering* (Seoul, Korea (South)). IEEE Computer Society, Los Alamitos, CA, USA, 999–1010. https://doi.org/10.1109/ICDE.2015.7113351

[66] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. 2016. Learning Social Etiquette: Human Trajectory Understanding In Crowded Scenes. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 549–565. https://doi.org/10.1007/978-3-319-46484-8_33

[67] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. 2020. Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 683–700. https://doi.org/10.1007/978-3-030-58523-5_40

[68] David Sankoff and Joseph B Kruskal. 1983. Time warps, string edits, and macromolecules: the theory and practice of sequence comparison. *Reading: Addison-Wesley Publication* (1983).

[69] Christoph Schöller and Alois Knoll. 2021. Flomo: Tractable motion prediction with normalizing flows. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Prague, Czech Republic). IEEE Computer Society, Los Alamitos, CA, USA, 7977–7984. https://doi.org/10.1109/IROS51168.2021.9636445

[70] Liushuai Shi, Le Wang, Sanping Zhou, and Gang Hua. 2023. Trajectory unified transformer for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Paris, France). IEEE Computer Society, Los Alamitos, CA, USA, 9675–9684. https://doi.org/10.1109/ICCV51070.2023.00887

[71] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. 2022. Motion transformer with global intention localization and local movement refinement. *Advances in Neural Information Processing Systems* 35 (2022), 6531–6543.

[72] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. 2024. MTR++: Multi-Agent Motion Prediction With Symmetric Scene Modeling and Guided Intention Querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 5 (2024), 3955–3971. https://doi.org/10.1109/TPAMI.2024.3352811

[73] Li Song, Ruijia Wang, Ding Xiao, Xiaotian Han, Yanan Cai, and Chuan Shi. 2018. Anomalous Trajectory Detection Using Recurrent Neural Network. In *Advanced Data Mining and Applications*, Guojun Gan, Bohan Li, Xue Li, and Shuliang Wang (Eds.). Springer International Publishing, Cham, 263–277. https://doi.org/10.1007/978-3-030-05090-0_23

[74] Antonio Luigi Stefani, Niccolò Bisagno, and Nicola Conci. 2024. MapFlow: Multi-Agent Pedestrian Trajectory Prediction Using Normalizing Flow. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Seoul, Republic of Korea). IEEE Computer Society, Los Alamitos, CA, USA, 3295–3299.

[75] Li Sun, Zhi Yan, Sergi Molina Mellado, Marc Hanheide, and Tom Duckett. 2018. 3DOF Pedestrian Trajectory Prediction Learned from Long-Term Autonomous Mobile Robot Deployment Data. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, Australia). IEEE Computer Society, Los Alamitos, CA, USA, 5942–5948. https://doi.org/10.1109/ICRA.2018.8461228

[76] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems* (Montreal, Canada) *(NIPS'14, Vol. 27)*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (Eds.). Curran Associates, Inc., Cambridge, MA, USA, 3104–3112. https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf

[77] Arsal Syed and Brendan Tran Morris. 2020. CNN, Segmentation or Semantic Embeddings: Evaluating Scene Context for Trajectory Prediction. In *Advances in Visual Computing*, George Bebis, Zhaozheng Yin, Edward Kim, Jan Bender, Kartic Subr, Bum Chul Kwon, Jian Zhao, Denis Kalkofen, and George Baciu (Eds.). Springer International Publishing, Cham, 706–717. https://doi.org/10.1007/978-3-030-64559-5_56

[78] Saeed Taghizadeh, Abel Elekes, Martin Schäler, and Klemens Böhm. 2021. How meaningful are similarities in deep trajectory representations? *Information Systems* 98 (2021), 101452. https://doi.org/10.1016/j.is.2019.101452

[79] Xiaolong Tang, Meina Kan, Shiguang Shan, Zhilong Ji, Jinfeng Bai, and Xilin Chen. 2024. Hpnet: Dynamic trajectory forecasting with historical prediction attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, USA). IEEE Computer Society, Los Alamitos, CA, USA, 15261–15270.

[80] Chieh-En Tsai and Jean Oh. 2020. A Generative Approach for Socially Compliant Navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (Paris, France). IEEE Computer Society, Los Alamitos, CA, USA, 2160–2166. https://doi.org/10.1109/ICRA40945.2020.9197497

[81] Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, et al. 2022. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *2022 IEEE International Conference on Robotics and Automation (ICRA)* (Philadelphia, USA). IEEE Computer Society, Los Alamitos, CA, USA, 7814–7821. https://doi.org/10.1109/ICRA46639.2022.9812107

[82] Dizan Vasquez, Billy Okal, and Kai O. Arras. 2014. Inverse Reinforcement Learning algorithms and features for robot navigation in crowds: An experimental comparison. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1341–1346. https://doi.org/10.1109/IROS.2014.6942731

[83] Xishun Wang, Tong Su, Fang Da, and Xiaodong Yang. 2023. Prophnet: Efficient agent-centric motion forecasting with anchor-informed proposals. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Vancouver, Canada). IEEE Computer Society, Los Alamitos, CA, USA, 21995–22003. https://doi.org/10.1109/CVPR52729.2023.02106

[84] Yuning Wang, Pu Zhang, Lei Bai, and Jianru Xue. 2023. Fend: A future enhanced distribution-aware contrastive learning framework for long-tail trajectory prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Vancouver, Canada). IEEE Computer Society, Los Alamitos, CA, USA, 1400–1409. https://doi.org/10.1109/CVPR52729.2023.00141

[85] Pei Xu, Jean-Bernard Hayet, and Ioannis Karamouzas. 2022. Socialvae: Human trajectory prediction using timewise latents. In *Computer Vision–ECCV 2022, Proceedings* (Tel Aviv, Israel). Springer, 511–528. https://doi.org/10.1007/978-3-031-19772-7_30

[86] Yi Xu, Dongchun Ren, Mingxia Li, Yuehai Chen, Mingyu Fan, and Huaxia Xia. 2021. Tra2Tra: Trajectory-to-Trajectory Prediction With a Global Social Spatial-Temporal Attentive Neural Network. *IEEE Robotics and Automation Letters* 6, 2 (2021), 1574–1581. https://doi.org/10.1109/LRA.2021.3057326

[87] Kota Yamaguchi, Alexander C. Berg, Luis E. Ortiz, and Tamara L. Berg. 2011. Who are you with and where are you going?. In *CVPR 2011*. 1345–1352. https://doi.org/10.1109/CVPR.2011.5995468

[88] Peilun Yang, Hanchen Wang, Defu Lian, Ying Zhang, Lu Qin, and Wenjie Zhang. 2022. TMN: Trajectory Matching Networks for Predicting Similarity. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE Computer Society, Los Alamitos, CA, USA, 1700–1713. https://doi.org/10.1109/ICDE53745.2022.00173

[89] Peilun Yang, Hanchen Wang, Ying Zhang, Lu Qin, Wenjie Zhang, and Xuemin Lin. 2021. T3S: Effective Representation Learning for Trajectory Similarity Computation. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE Computer Society, Los Alamitos, CA, USA, 2183–2188. https://doi.org/10.1109/ICDE51399.2021.00221

[90] Di Yao, Gao Cong, Chao Zhang, and Jingping Bi. 2019. Computing Trajectory Similarity in Linear Time: A Generic Seed-Guided Neural Metric Learning Approach. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE Computer Society, Los Alamitos, CA, USA, 1358–1369. https://doi.org/10.1109/ICDE.2019.00123

[91]  Byoung-Kee Yi, H.V. Jagadish, and C. Faloutsos. 1998. Efficient retrieval of similar time sequences under time warping. In *Proceedings 14th International Conference on Data Engineering*. IEEE Computer Society, Los Alamitos, CA, USA, 201–208. https://doi.org/10.1109/ICDE.1998.655778

[92]  Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. 2021. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, Canada). IEEE Computer Society, Los Alamitos, CA, USA, 9813–9823. https://doi.org/10.1109/ICCV48922.2021.00967

[93]  Bo Zhang, Chengzhi Yuan, Tao Wang, and Hongbo Liu. 2021. STENet: A hybrid spatio-temporal embedding network for human trajectory forecasting. *Engineering Applications of Artificial Intelligence* 106 (2021), 104487. https://doi.org/10.1016/j.engappai.2021.104487

[94]  Hanyuan ZHANG, Xingyu ZHANG, Qize JIANG, Baihua ZHENG, Zhenbang SUN, Weiwei SUN, and Changhu WANG. 2020. Trajectory similarity learning with auxiliary supervision and optimal matching. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, (IJCAI'20)*. Research Collection School Of Information Systems, Yokohama, Yokohama, Japan, Article 444, 7 pages. https://ink.library.smu.edu.sg/sis_research/5276

[95]  Weicheng Zhang, Hao Cheng, Fatema T Johora, and Monika Sester. 2023. ForceFormer: exploring social force and transformer for pedestrian trajectory prediction. In *2023 IEEE Intelligent Vehicles Symposium (IV)* (Anchorage, USA). IEEE Computer Society, Los Alamitos, CA, USA, 1–7. https://doi.org/10.1109/IV55152.2023.10186643

[96]  Zhejun Zhang, Alexander Liniger, Christos Sakaridis, Fisher Yu, and Luc V Gool. 2024. Real-time motion prediction via heterogeneous polyline transformer with relative pose encoding. *Advances in Neural Information Processing Systems* 36 (2024).

[97]  Hao Zhou, Dongchun Ren, Xu Yang, Mingyu Fan, and Hai Huang. 2023. CSR: Cascade Conditional Variational Auto Encoder with Socially-aware Regression for Pedestrian Trajectory Prediction. *Pattern Recognition* 133 (2023), 109030. https://doi.org/10.1016/j.patcog.2022.109030

[98]  Silin Zhou, Peng Han, Di Yao, Lisi Chen, and Xiangliang Zhang. 2023. Spatial-temporal fusion graph framework for trajectory similarity computation. *World Wide Web* 26, 4 (2023), 1501–1523. https://doi.org/10.1007/s11280-022-01089-0

[99]  Silin Zhou, Jing Li, Hao Wang, Shuo Shang, and Peng Han. 2023. GRLSTM: Trajectory Similarity Computation with Graph-Based Residual LSTM. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 4 (Jun. 2023), 4972–4980. https://doi.org/10.1609/aaai.v37i4.25624

[100] Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. 2022. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (New Orleans, USA). IEEE Computer Society, Los Alamitos, CA, USA, 8823–8833. https://doi.org/10.1109/CVPR52688.2022.00862