Multimodal Multi-Pedestrian Path Prediction for Autonomous Cars

Atanas Poibrenski, Matthias Klusch, Igor Vozniak, Christian Müller
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
{atanas.poibrenski, matthias.klusch, igor.vozniak, christian.mueller}@dfki.de

ABSTRACT

Accurate prediction of the future position of pedestrians in traffic scenarios is required for safe navigation of an autonomous vehicle but remains a challenge. This concerns, in particular, the effective and efficient multimodal prediction of most likely trajectories of tracked pedestrians from egocentric view of self-driving car. In this paper, we present a novel solution, named M2P3, which combines a conditional variational autoencoder with recurrent neural network encoder-decoder architecture in order to predict a set of possible future locations of each pedestrian in a traffic scene. The M2P3 system uses a sequence of RGB images delivered through an internal vehicle-mounted camera for egocentric vision. It takes as an input only two modes, that are past trajectories and scales of pedestrians, and delivers as an output the three most likely paths for each tracked pedestrian. Experimental evaluation of the proposed architecture on the JAAD, ETH/UCY and Stanford Drone datasets reveal that the M2P3 system is significantly superior to selected state-of-the-art solutions.

CCS Concepts

•**Computing methodologies** → **Neural networks;**

Keywords

Autonomous driving, multi-pedestrian path prediction

1. INTRODUCTION

Despite recent advances in autonomous driving, the achievement of pedestrian-safe navigation of autonomous vehicles (AVs) remains a challenge [53]. One prerequisite of collision-free navigation is an effective and efficient multi-pedestrian path prediction in traffic scenes by AVs. In fact, there is a plethora of solution approaches for this problem [75] to be employed in advanced driver assistance systems of AVs. Currently, these systems enable an AV to detect if a pedestrian is actually in the direction of travel, warn the control driver and even stop automatically. Other approaches would allow ADAS to predict whether the pedestrian is going to step on the street, or not [52].

The multimodality of multi-pedestrian path prediction in ego-view is a challenge and hard to handle by many deep learning (DL) models for many-to-one mappings. Given past trajectories of tracked pedestrians in a traffic scene, the distribution of future trajectories as outcomes has not a single but multiple modes. Each pedestrian has unique dynamics and individual goals to reach, and many different trajectory predictions are equally possible for the same traffic scene context with pedestrians. Conditional variational auto-encoders (CVAE) for output representation learning and structured prediction may be applied to cope with this problem in principle [17]. A CVAE models the distribution of a high-dimensional output space as a generative model conditioned on input modes, which modulate the prior on lower dimensional, randomly sampled Gaussian latent variables that are then decoded into a set of probabilistic input reconstructions as outputs [63, 38, 43]. Though, the benefit of using a CVAE-based system for multimodal prediction of most likely pedestrian paths from egocentric vision of a self-driving car remains to be shown. It is not known for which set of input modes or factors of pedestrian dynamics, scene context and social context what kind of CVAE-based system architecture performs best for this purpose [57, 53].

To this end, we developed a novel CVAE-based system, named M2P3, for multimodal multi-pedestrian path prediction by self-driving cars in ego-view [50]. It combines a conditional variational autoencoder as a generative model with a recurrent neural network (RNN) encoder-decoder architecture in order to output a set of possible future paths of each pedestrian in a traffic scene tracked by a self-driving car with egocentric vision. The M2P3 system uses a RGB vehicle-mounted camera for egocentric vision and takes as input only two basic modes, that are past trajectories and scales of tracked pedestrians in traffic scene video. It k-means clusters the set of their trajectories predicted for 1 second into the future and outputs these k future pedestrian paths together with their probability of occurrence. Results of our comparative evaluation on the publicly available datasets - JAAD (joint attention for autonomous driving in ego-view), ETH/UCY and Stanford Drone Dataset (SSD) reveal that the M2P3 system performance is significantly superior to selected state-of-the-art solutions.

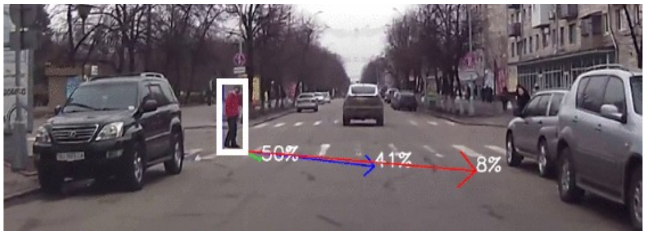

Figure 1: Example of M2P3 prediction of three most likely trajectories of tracked pedestrian in car ego-view video taken from the JAAD dataset.

The remainder of the paper is structured as follows. In Section 2, we briefly summarize related work and describe our novel solution M2P3 to the multi-pedestrian path prediction problem in Section 3. Results of our comparative experimental evaluation of M2P3 are discussed in Section 4 before we conclude in Section 5.

2. RELATED WORK

First-person video or egocentric vision is a sub-field of computer vision which tries to analyze images or videos from a wearable camera (typically on a person's head) or from a mounted camera in the car looking forward. This is a challenging task due to the perspective view of the camera, the narrow field of view, as well as the introduced ego-motion. Most of the works in literature have focused on object detection [32, 5], activity recognition [36, 15, 39, 49],person identification [2, 13, 68, 21], activity forecasting [12, 18, 54], video summarization [31], gaze anticipation [35, 73], and grasp recognition [3, 7, 34, 60].

Recent work [46] also focuses on egocentric future localization but predicts the future location of the camera wearer and not the people around. Another example is the approach presented in [65], which uses a Siamese network to estimate future behaviors of basketball players in first-person videos. However, unlike our M2P3 approach, this method requires multiple cameras to reconstruct the scene.

Recent work that is more related to our M2P3 solution is presented in, for example, Bhattacharyya et al. [6]. The authors propose a Bayesian LSTM (Long Short Term Memory) to predict the future locations of people by taking into account the car's ego-motion as well.

Yagi et al. [69] predict future locations of people observed in first-person videos by using the person's pose, past movement and ego-motion in a multi-stream convolution - deconvolution neural network. However, the method only predicts one possible future location, thus fails to capture multi-modality of the pedestrian motion.

Yao et al. [71] proposes a RNN (recurrent neural network) encoder-decoder model that can predict future vehicle locations from ego-view in traffic scenarios. The approach employs scene optical flow as well as future ego-motion prediction but fails to model the probabilistic nature of the problem and takes no pedestrians into account.

Ma et al. [40] predicts the motion of heterogeneous traffic-agents from ego-view perspective using an LSTM-based real-time traffic prediction algorithm. They model the problem as a 4D-graph and treat traffic agents as points and only take into account their past motions.

The problem of human trajectory prediction has been researched extensively. Most of the works focus of static scenes and crowds. There are many classical approaches to the problem such as a Bayesian formulation [61, 33], Monte Carlo Simulation [10, 45, 56], Hidden Markov Models [16, 41], Kalman Filters [24], linear and non-linear Gaussian models [51, 11], Markov jump process [25]. These methods try to model objects based on their past movements but cannot work reliably in real-world traffic scenarios where uncertainty and multi-modality should be taken into account as well.

Other works explicitly model the interaction between pedestrians for collision avoidance. For example, in [48] the authors propose a linear trajectory avoidance model, and in [66] the social force model is utilized. These approaches are designed for homogeneous interactions in crowds and rely on predetermined models of interaction.

In [1], a "Social LSTM" network is introduced, which predicts the future path of multiple people in a crowd by means of connected neighboring LSTM networks in a social pooling layer. Recently, the authors of [19] propose to generate socially compliant set of trajectories by utilizing a generative adversarial network (GAN) and training against a recurrent discriminator. However, their method is applied to a static crowd scene only.

Some recent works on pedestrian path prediction employ some variant of a RNN and/or combine it with other deep learning models such as convolutional neural networks (CNN), GANs, and variational autoencoders (VAE). For example, the DESIRE framework [30] consists of a CVAE-based RNN encoder-decoder architecture, which can output multiple path predictions to be refined further. However, the likelihood of each future path prediction per pedestrian is not estimated. The latter is achieved in M2P3 by means of k-means clustering to approximate the likelihood of future trajectories. Furthermore, according to our experiments the prior of DESIRE's CVAE appears too restrictive for the modelling of multimodal trajectory distributions (cf. Table 2).

The NEXT model [37] proposes a combined LSTM and focal attention-based approach to the prediction of trajectory and future activity of pedestrians. In particular, it combines visual features of person (appearance, pose), person-scene (segmentation of scene around person) and geometric person-object relations in a visual feature for separate trajectory generation with focal attention and action label prediction per pedestrian. However, in contrast to M2P3, NEXT does not address the above mentioned stochastic nature of human trajectory prediction.

In [17], the use of conditional stochastic networks for multimodal prediction of object future motion trajectory in top-view with single frame as input from the drone Stanford dataset is investigated. However, our CVAE-based M2P3 system architecture and loss function are different, and implements a complete processing pipeline for self-driving car in ego-view.

Moreover, in [8], the authors propose an end-to-end algorithm for predicting future trajectories of road-agents using a LSTM-CNN neural network architecture and RGB camera. The authors of [22] use RNNs to capture the scene information (Scene-LSTM) as well as individual pedestrian movement (Pedestrian-LSTM) which are trained simultaneously on static crowded scenes. The work of [42] combines a LSTM and a CNN and considers the type of the agent (e.g pedestrian, cyclists) as well as the physical environment information as inputs. This allows their model to predict a trajectory that is proper for each agent type. Additionally, a state refinement for LSTM (SR-LSTM) is proposed in [74] which utilizes the intentions of neighbours and jointly refines the current state of all pedestrians in a crowd scenario through a message passing mechanism. The authors also introduce socially-aware information consisting of element-wise motion gate and pedestrian-wise attention to select promising messages from nearby pedestrians. Despite showing promising results, all these methods output only a single deterministic trajectory prediction which is not optimal in real world scenarios.

3. M2P3 SOLUTION

As mentioned above, future prediction of pedestrian movement can be very ambiguous because given the same input state, multiple future states are possible. For example, a pedestrian heading towards a t-intersection, has an equal probability of going either left or right. Moreover, a model which simply learns a deterministic input/output mapping $f : \mathbb{X} \rightarrow \mathbb{Y}$ will under-represent the prediction space and possibly average out all possible outcomes, if a naive loss function is used. In order to tackle this problem, we adopt a generative model, especially a conditional variational auto-encoder (CVAE) based on gated recurrent neural networks for encoding and decoding, which generates a set of future pedestrian trajectories, hence allowing one-to-many input/output mappings.

3.1 Architecture

The architecture of the M2P3 approach is summarized in Figure 2. The pedestrian trajectory prediction problem is modeled with a generative model, a CVAE, where the posterior distribution $P(Y \mid X)$ is learned with the help of a latent variable Z [63]. Our model allows for conditional generation of pedestrian trajectories while taking into account the uncertainty of the future prediction.

The M2P3 gets as an input the trajectory and scale (represented as X) of each detected pedestrian in ego-view and predicts for each input for about two third of one second (n = 10 frames) the three most likely future trajectories $\hat{Y}$ for one second into the future (m = 15 frames). During training M2P3 is also provided with the ground truth of future pedestrian trajectories (given as Y). Its CVAE network learns to map the joint input H of RNN-based encodings of Y and X to latent Z with prior normal distribution in order to generate a model of $P(Y|X)$ which maximizes the probability of Y conditioned on input X.

During testing, the processing of ground truth Y is removed such that only random samples Z from prior normal $\mathcal{N}(0, I)$ extended with encoded input X are used for the prediction of Y with approximated posterior $P(Y|Z, X)$. For a given number of N for such predictions, the system eventually returns k most likely trajectories based on k-means clustering. The whole M2P3 processing pipeline allows to realize many-to-many mappings for multimodal multi-pedestrian path prediction in ego-view.

3.1.1 Training

The training architecture of M2P3 is shown in Figure 2. During training the time-dependent features of two basic input modes, that are the pedestrian location l and scale s (see Sect. 3.2) of X, and the ground truth Y of the future trajectory of the pedestrian are encoded through gated recurrent neural networks in H_X, respectively, H_Y. These encodings are concatenated in the joint input vector H for the variational module, which, in turn, learns to estimate the mean μ_H and co-variance Σ_H of normal distribution $\mathcal{N}(\mu_H, \Sigma_H)$, thus mapping the joint input H to latent Z with conditional normal prior $P(Z|X) \sim \mathcal{N}(0, I)$ as reference for sampling. A random sample of Z from normal distribution together with condition H_X is then fed into the following RNN decoder. The latter decodes this extended sample into a predicted future trajectory with approximated conditional normal posterior distribution $P(Y|Z, X)$.

Each of the encoding of the inputs X into H_X and Y into H_Y is done by a RNN encoder using the following recurrent (GRU) computation:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_z \cdot \mathbf{x}_t + \mathbf{U}_z \cdot \mathbf{h}_{t-1} + \mathbf{b}_z), \\
\mathbf{r}_t &= \sigma(\mathbf{W}_r \cdot \mathbf{x}_t + \mathbf{U}_r \cdot \mathbf{h}_{t-1} + \mathbf{b}_r), \\
\mathbf{h}_t &= (1 - \mathbf{z}_t)\mathbf{h}_{t-1} + \mathbf{z}_t \sigma(\mathbf{W}_h \cdot \mathbf{x}_t + \mathbf{U}_h(\mathbf{r}_t \mathbf{h}_{t-1}) + \mathbf{b}_h)
\end{aligned} \tag{1}$$

where $\mathbf{W}$, $\mathbf{U}$ and $\mathbf{b}$ are learnable weights, $\mathbf{z}$ and $\mathbf{r}$ are update and reset gates; $\mathbf{x}$ and $\mathbf{h}$ are input and output vectors accordingly. σ is a nonlinear function such as tanh. Initially, for $\mathbf{t} = 0$, the output vector is $\mathbf{h}_0 = 0$.

The outputs H_Y, H_X of both encoders are then concatenated into a joint input vector H. This vector is fed into two fully-connected layers for mean μ_H and co-variance Σ_H, which are learned to model the latent Z distribution $Q(Z|H)$ as normal distribution $\mathcal{N}(\mu_H, \Sigma_H)$ with $Z = \mu_H + \Sigma_H \odot \epsilon$ and $\epsilon \sim \mathcal{N}(0, I)$. In other words, it learns to map the joint input H to latent Z with normal distribution $\mathcal{N}(0, I)$ as reference for sampling; $P(Z|X)$ is $\mathcal{N}(0, I)$, because we assume Z is sampled independently of X at test time. This processing part of the M2P3-CVAE network during learning requires the minimization of the Kullback-Leibler divergence (D_{KL}) between the estimated distribution $Q(Z|H)$ and the reference distribution $\mathcal{N}(0, I)$, i.e. $D_{KL}(\mathcal{N}(\mu_H, \Sigma_H)||\mathcal{N}(0, I))$.

In order to allow for backpropagation of errors through a layer that samples Z from $Q(Z|H)$, which is a non-continuous operation without gradient, the standard reparameterization trick to move the sampling to an input layer as introduced in [27] is applied. That is, sampling from $\mathcal{N}(\mu_H, \Sigma_H)$ is done by first randomly sampling $\epsilon \sim \mathcal{N}(0, I)$ and then computing Z with these parameters $(\epsilon, \mu_H, \Sigma_H)$ as mentioned above.

Eventually, the RNN decoder gets a sample of Z extended

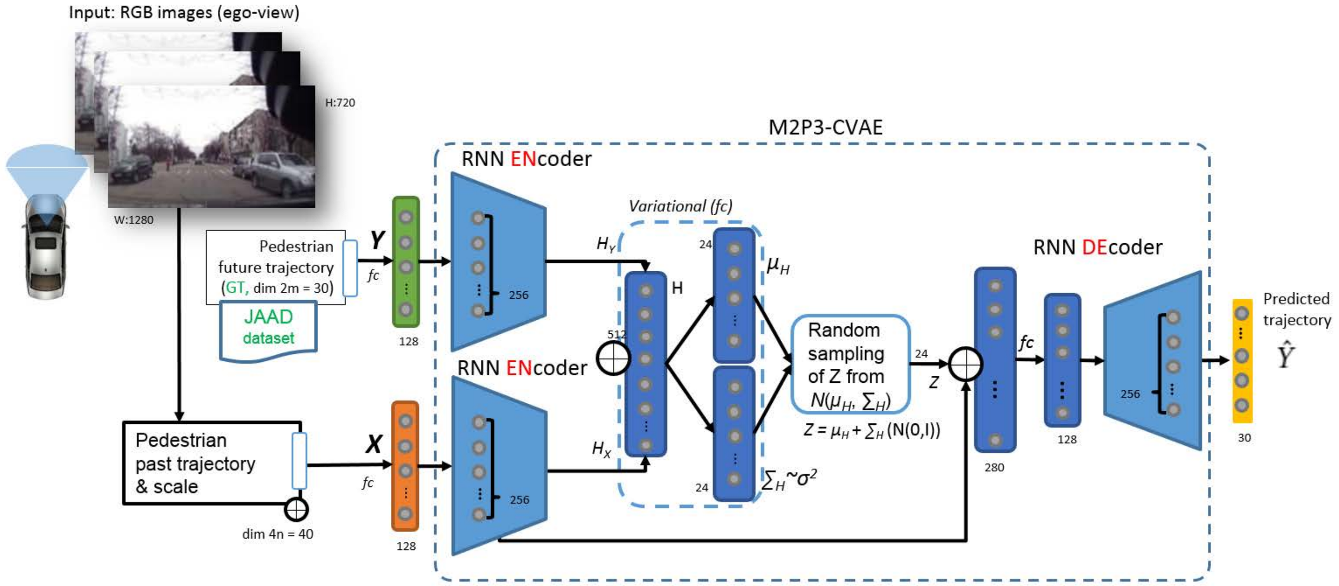

Figure 2: M2P3 system architecture (training) overview. For each tracked pedestrian, the system processes the ground-truth future trajectory (Y) from JAAD dataset, and observed past trajectory (X) by means of encoding, mapping of joint input H to latent Z with normal prior as reference for sampling, and decoding of random sample Z with X into prediction of future pedestrian trajectory $\hat{Y}$ as output.

with condition H_X, performs the recurrent operation (1) on it, and feeds the result into a final dense layer that produces the future trajectory prediction $\hat{Y}$. This processing part of the M2P3-CVAE network during learning requires the minimization of the error between ground truth future trajectory Y and its prediction $\hat{Y}$ according to the L2 loss (Euclidean distance) $\left\|Y - \hat{Y}\right\|^2$.

The whole M2P3-CVAE network architecture is trained with the stochastic gradient descent method to minimize the total loss L defined as

$$L = \left\|Y - \hat{Y}\right\|^2 + \mathrm{D_{KL}}(\mathcal{N}(\mu_H, \Sigma_H)||\mathcal{N}(0, I)) \qquad (2)$$

That is, the latent distribution Q is learned by the M2P3-CVAE network such that it gives a higher probability to Z with which it is more likely to produce predictions $\hat{Y}$ that are close to ground truth Y in the context of X.

3.1.2 Testing

The M2P3 test architecture is shown in Figure 3. At test time, the ground truth of future trajectories Y is not available such that the respective part of the encoding pathway in the M2P3 system is not used (see Figure 3). Besides, we can now sample from distribution $P(Y|X)$ by sampling $Z \sim \mathcal{N}(0, I)$ In fact, the RNN decoder of M2P3 only receives the RNN-encoded condition H_X together with a random sample Z drawn from the prior distribution $\mathcal{N}(0, I)$. This enables probabilistic inference allowing to handle multimodality in the prediction space.

For each input X, the test network is run with N = 1000 random samples Z, thereby generating N possible trajectories of the considered pedestrian, which are then clustered into k = 3 clusters using k-means. Since we do not have explicit access to the posterior trajectory distribution, we choose a large number for N, which allows the future trajectory distribution to be closely approximated. The value of k is chosen arbitrarily, such that the output trajectories are not under- or over-clustered. In particular, each of the generated trajectories is assigned to the closest cluster based on 2D-Euclidean distance. The number of assigned trajectories in each cluster is divided by the total number N of generated trajectories to obtain a probability distribution over the clusters (see Figure 1). In concrete terms, given the set of output trajectories from the model $Y = \{\mathrm{Y}_1, \mathrm{Y}_2, ..., \mathrm{Y}_N\}$ and the set $\mathrm{S} = \{\mathrm{S}_1, \mathrm{S}_2, \mathrm{S}_3\}$ of clusters, M2P3 assigns each output trajectory Y_p, $1 \leq p \leq$ N, to exactly one cluster S_i, $1 \leq i \leq 3$, whose mean m_i has the least squared Euclidean distance:

$$\mathrm{S}_i = \{\mathrm{Y}_p : \left\|\mathrm{Y}_p - \mathrm{m}_i\right\|^2 \leq \left\|\mathrm{Y}_p - \mathrm{m}_j\right\|^2, 1 \leq j \leq 3, i \neq j\} \qquad (3)$$

This is followed by the calculation of (k=3) cluster probabilities as $P(\mathrm{S}_i) = |\mathrm{S}_i|/|\mathrm{Y}|$, where $|\mathrm{S}_i|$ and $|\mathrm{Y}|$ denote the cardinality of respective sets. These cluster probabilities are then displayed by the M2P3 system as probabilities of occurrence of predicted future pedestrian positions.

Alternatively, the most likely future trajectory can be obtained by predicting all N trajectories $\hat{Y}_1$... $\hat{Y}_N$ followed by fitting a bivariate Gaussian distribution to all trajectories and then selecting the one with the highest probability. The predicted 2D locations of each pedestrian $(\hat{x}_n^t, \hat{y}_n^t)$ where n ∈ N at time step t ∈ T are used to fit a bivariate Gaussian distribution $\mathcal{N}(\mu_{xy}, \sigma_{xy}^2, \rho)^t$. Once we have the Gaussian distribution, all the trajectories can be sorted by their joint probability density functions p(.) (Equation 4).

$$\mathrm{P}(\hat{x}_n^t, \hat{y}_n^t) \approx \mathrm{p}[(\hat{x}_n^t, \hat{y}_n^t) | \mathcal{N}(\mu_{xy}, \sigma_{xy}^2, \rho)^t] \tag{4}$$

The most likely trajectory $\hat{Y}^*$ can then be computed by taking the most probable 2D location at each timestep (Equation 5).

$$\hat{Y}^* = \arg\max \sum_{n=1}^{\mathrm{N}} \sum_{t=1}^{\mathrm{T}} \log \mathrm{P}(\hat{x}_n^t, \hat{y}_n^t) \tag{5}$$

3.1.3 *Implementation*

Our M2P3 implementation bases on Keras [9] with Tensorflow as backend. For pedestrian tracking, M2P3 can utilize DeepSORT [67] with underlying mask R-CNN [20] for pedestrian detection. In the implemented M2P3 system, all CVAE input data first passes through a fully-connected embedding layer of size 128 before being fed into an encoder. The hidden size of all encoders and decoder layers is set to 256. The latent dimension of the fully-connected (fc) layer in the CVAE is set to 24. The two latent fc layers are concatenated before deriving latent distribution, that is to match the unknown latent distribution to a known, prior distribution. In order to simplify the training process, in contrast to Long-Short-Term-Memory (LSTM) networks, Gated Recurrent Units (GRU) have been adopted for the RNN-encoders/-decoder.

3.2 Pedestrian Trajectory and Scale

One obvious clue about future pedestrian motions is their motion in the past. Thus, M2P3 also tracks each pedestrian's 2D image location (x,y coordinates) for **n** frames. For each detected pedestrian in the scene, M2P3 collects the following feature vector:

$\mathrm{X_l} = \{\mathrm{x_{T-\mathbf{n}}}, \mathrm{y_{T-\mathbf{n}}}, \mathrm{x_{T-(\mathbf{n}-1)}}, \mathrm{y_{T-(\mathbf{n}-1)}}, ..., \mathrm{x_T}, \mathrm{y_T}\}$,

where T is the current time frame. 2D image distances correspond to different physical distances depending on where the person is situated in the frame. Therefore, M2P3 learns the width and the height (scale) of the pedestrian in order to take the perspective effect of the ego camera into account. In particular, it records the width w and height h in pixels of each pedestrian for the past **n** frames into the following vector:

$\mathrm{X_s} = \{\mathrm{w_{T-\mathbf{n}}}, \mathrm{h_{T-\mathbf{n}}}, \mathrm{w_{T-(\mathbf{n}-1)}}, \mathrm{h_{T-(\mathbf{n}-1)}}, ..., \mathrm{w_T}, \mathrm{h_T}\}$

The final input $\mathrm{X_{l,s}}$ to the underlying M2P3 model (cf. Sect. 3.1) then is: $\mathrm{X_{l,s}} = \mathrm{X_l} \oplus \mathrm{X_s}$, where $\oplus$ denotes the concatenation operator. This input is normalized in the range [0,1] relative to the image resolution. The output Y of the M2P3 model is modeled as the 2D displacement from the last observed frame T:

$\mathrm{Y} = \{\mathrm{x_{T+1}}–\mathrm{x_T}, \mathrm{y_{T+1}}–\mathrm{y_T}, ..., \mathrm{x_{T+\mathbf{m}}}–\mathrm{x_T}, \mathrm{y_{T+\mathbf{m}}}–\mathrm{y_T}\}$,

where **m** is the number of frames in the future. By using a displacement vector rather than absolute coordinates, the M2P3 model is able to learn how a pedestrian moves in the future relative to his starting position. This helps with the generalization of the model to new scenes with different resolution and positions of the pedestrians.

4. EXPERIMENTS

For comparative performance evaluation of our M2P3 system, we conducted experiments based on the publicly available datasets JAAD, ETH/UCY and Stanford Drone Dataset (SSD) against selected state-of-the-art multi-pedestrian path predictors as baselines.

4.1 JAAD

4.1.1 *Dataset*

For our first comparative performance evaluation experiments, we use the publicly available JAAD (Joint Attention for Autonomous Driving) dataset [28]. This dataset contains an annotated collection of short video clips, capturing typical urban traffic scenarios in various weather conditions. The clips are taken from a single RGB camera, mounted behind the windshield of a moving car. All pedestrians are manually annotated with bounding boxes and unique tracking identifier. The resolution of all videos is set to a constant value of 1280 × 720. The frame rate is also re-scaled to a constant value of n = 15 frames per second. All pedestrians which are either too far away from the car (less than 50 pixels in size), or occluded, or tracked for less than 25 frames, are ignored.

4.1.2 *Implementation*

The JAAD dataset is split into training (videos 0-250) and testing (videos 251-346) as done in [14, 64]. The ratio between training and validation videos is 80% to 20% for fine-tuning the hyper-parameters of the implemented M2P3 model. After hyper-parameters are fixed, we train the M2P3 on the full training set of JAAD (videos 0-250).

For all experiments, ground truth bounding boxes provided by the JAAD dataset are used for extracting past trajectory and scale of pedestrian. The numbers of past and future frames are set to $n = 15, m = 10$. The ADAM [26] optimizer is used with learning rate of 1e-4 and trained the M2P3 for 6000 epochs. The model has 948,914 trainable parameters in total. The training takes approximately 2 hours on desktop machine with NVIDIA GTX 1080ti GPU and Intel i7-7800X CPU. The average inference time is 29ms per pedestrian.

4.1.3 *Baselines and Metrics.*

For the comparative performance evaluation, we selected the following five state-of-the-art solution models as baselines.

1. **CV** (Constant Velocity) model. The CV model as in [62] assumes that the pedestrian maintains constant velocity through time. The horizontal and vertical components of the velocity at time t are denoted as $\upsilon_{\mathrm{t}}^{x}$ and $\upsilon_{\mathrm{t}}^{y}$ and defined as $\upsilon_{\mathrm{t}}^{x} = \frac{\mathrm{x_t\text{-}x_{t\text{-}n}}}{\mathrm{n}}$ and $\upsilon_{\mathrm{t}}^{y} = \frac{\mathrm{y_t\text{-}y_{t\text{-}n}}}{\mathrm{n}}$ for the past n observed frames. Therefore, the future position of a pedestrian is defined as $\tilde{\mathrm{x}}_{\mathrm{t+m}} = \mathrm{x_t} + \upsilon_{\mathrm{t}}^{\mathrm{x}} \cdot \mathrm{m}$ and $\tilde{\mathrm{y}}_{\mathrm{t+m}} = \mathrm{y_t} + \upsilon_{\mathrm{t}}^{\mathrm{y}} \cdot \mathrm{m}$ in the next m frames.

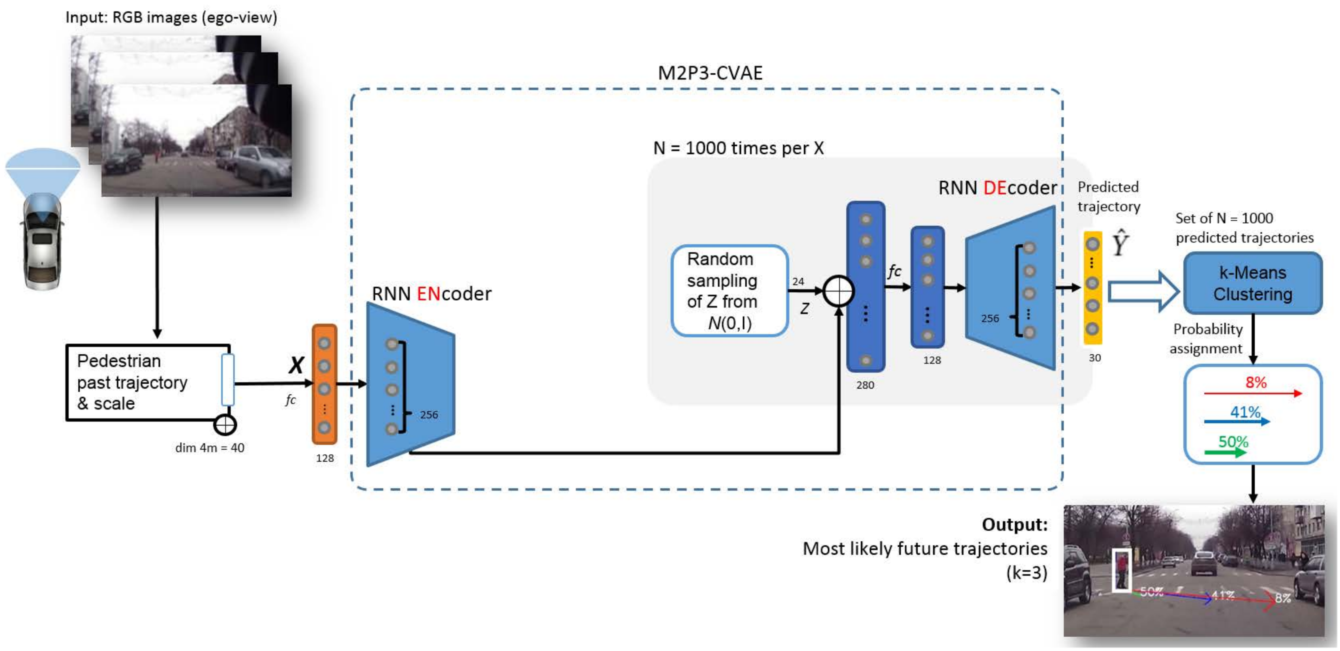

Figure 3: M2P3 architecture (testing) overview. The input only consists of X for observed pedestrian trajectory and scale, which RNN-based encoding combined with random sample of Z from normal distribution is decoded into trajectory prediction $\hat{Y}$ as output.

2. **CA** (Constant Acceleration) model. The implemented CA model is the same as the CV model above but the acceleration of a pedestrian is assumed to be constant.

3. **RNN** Encoder-decoder model in [71] is the same as the M2P3 model but without the CVAE module.

4. **MSCD** model. The MSCD model proposed in [69] uses a multi-stream convolution-deconvolution framework.

5. **DTP** model. The DTP model proposed in [64] utilizes the optical flow of the pedestrians and a residual network.

For reasons of comparability of the results, we applied the same evaluation scheme as in [64]. The M2P3 model observes $\mathbf{n} = 10$ frames (2/3 of a second) and predicts $\mathbf{m} = 15$ frames (1 second) into the future. The following performance evaluation metrics for pedestrian path prediction are computed for all systems on the test set of the JAAD dataset:

- The mean squared error (MSE) in pixels (1280x720 resolution)is defined as $\frac{1}{N}\frac{1}{m}\sum_{i=1}^{N}\sum_{t=1}^{m}(\hat{\mathrm{Y}}^{\mathrm{i}}_{\mathrm{t}}-\mathrm{Y}^{\mathrm{i}}_{\mathrm{t}})^2$, where $\hat{\mathrm{Y}}$ denotes the prediction, Y the ground truth, and N the number of test sequences.

- The displacement error (DE) in pixels (1280x720 resolution) at last time step $\mathbf{m} = 15$ (1 second) is defined as $\frac{1}{N}\sum_{i=1}^{N}\left\|\hat{Y}^i_m - Y^i_m\right\|_2$.

4.1.4 Results and Analysis.

The results of the comparative performance evaluation over the JAAD dataset are given in Table 1.

Table 1: Experimental results for M2P3 and baselines over the JAAD dataset.

Method	MSE	DE
CA[62]	1426	52.8
CV	1148	47.5
RNN [71]	983	49.1
MSCD [69]	881 ± 44	41.3 ± 1.2
DTP [64]	610 ± 21	34.6 ± 0.5
M2P3 (1 sample)	584 ± 5	35.9 ± 0.15
M2P3 (1000 samples, k=3 clusters)	**483 ± 2**	**29.02 ± 0.06**

The results reveal that with just a single random sample (prediction) our CVAE-RNN based model M2P3 is already able to reach a performance comparable to that of the selected state-of-the-art baselines. When the number of samples is increased to 1000, clustered into 3 clusters, where the cluster closest to the ground truth is chosen, our model notably even outperforms all selected baselines. This confirms the multi-modal nature of the prediction problem, which is not captured by the selected alternative methods. Having access to the ground truth of pedestrian path prediction in the real world of autonomous driving is, of course, not possible, hence one cannot simply pick the best prediction. This case is handled by the M2P3 by means of clustering of

Table 2: Experimental results for M2P3 and selected baselines over the ETH/UCY dataset. For each method the best out of 20 predictions (samples) is chosen.

ADE/FDE in meters (20 samples)						
Method	ETH	HOTEL	UNIV	ZARA1	ZARA2	Average
Social-GAN (2018)	0.81/1.52	0.72/1.61	0.60/1.26	0.34/0.69	0.42/0.84	0.58/1.18
Sophie (2018)	0.70/1.43	0.76/1.67	0.54/1.24	**0.30/0.63**	0.38/0.78	0.54/1.15
NEXT (2019)	0.73/1.65	**0.30/0.59**	0.60/1.27	0.38/0.81	0.31/0.60	0.46/1.00
DESIRE (2017)	0.93/1.94	0.52/1.03	0.59/1.27	0.41/0.86	0.33/0.72	0.53/1.11
SGN LSTM (2019)	0.75/1.63	0.63/1.01	**0.48/1.08**	**0.30**/0.65	0.26/0.57	0.48/0.99
FSGAN (2019)	0.68/1.16	0.43/0.89	0.54/1.14	0.35/0.71	0.32/0.67	0.46/0.91
M2P3	1.04/2.16	0.54/1.13	0.64/1.34	0.45/0.95	0.37/0.79	0.60/1.27
M2P3 MoG	**0.57/1.01**	0.40/0.87	0.61/1.31	0.33/0.70	**0.21/0.42**	**0.42/0.86**

and probability assignment to the predictions in the output set. These clusters can then be considered one by one in a decreasing probability fashion by an AV navigation algorithm. Besides, the M2P3 also implicitly learns both pedestrian and ego motion instead of ego motion-free trajectories for learning individual motion patterns of pedestrians.

4.2 ETH/UCY

4.2.1 Dataset.

For our second comparative performance evaluation, we used two prominent, publicly available datasets for trajectory prediction: ETH [65] and UCY [29]. Both datasets are converted to world coordinates (meters) and pedestrian positions are obtained every 0.4 seconds (1 timeframe). The data is split into 5 sets (ETH - 2, UCY - 3) and we follow the standard leave-one-scene-out data split as in [19] for evaluation, such that training is performed on 4 sets and test on the remaining one. Past trajectories are observed for 8 timesteps (3.2 seconds) and predicted for the next 12 timesteps (4.8 seconds).

4.2.2 Implementation.

Here, we experimented with a more powerful prior, that is the Mixture-of-Gaussians (MoG), which can capture more modes of the trajectory distribution compared to just a unit Gaussian. The loss function L for M2P3-MoG training is as follows:

$$L = \left\|Y - \hat{Y}\right\|^2 + \mathrm{D_{KL}}(\mathrm{q(z,c|X,Y)}||p(\mathrm{z,c|X})) \qquad (6)$$

This essentially means that the variational encoder of the M2P3 now learns a posterior distribution q(z,c|X,Y), where the latent embedding **z** is regularized by the prior p(z,c|X) to lie on Mixture-of-Gaussians manifold, where **z** $\sim \mathcal{N}(\mu_c, \sigma_c^2 \mathrm{I})$ and **c** $\sim$ Category(π) such that K is a predefined number of components of the mixture and π=[π_1, π_2,...,π_K] is the prior probability of the Gaussian mixture components. More details for the derivation of the loss function can be seen in [23], where a variational autoencoder with MoG was originally used for the task of clustering.

Since the ETH/UCY dataset is captured from a fixed top-down view, we do not use the person's scale anymore but the normalized past trajectory in meters. For a stable training, the model gets first pre-trained based on just the first term of the loss in (6) for a few epochs. After that a gaussian mixture from the latent space (Z) of the model is initialized for continued training with the full loss for 100 epochs and the ADAM optimizer with a learning rate of 1e-5.

4.2.3 Baselines and Metrics.

For the evaluation, 20 predictions are generated for each observed trajectory and the closest one to the ground truth is chosen. This allows us to test the multi-modality and diversity of the predictions. We compare our model to the following state-of-the-art baselines:

1. **Social-GAN** [19] uses a recurrent sequence-to-sequence model with a novel social pooling mechanism and a generative adversarial network.
2. **Sophie** [58] uses a generative adversarial network to generate realistic trajectory by utilizing social and physical scene constraints.
3. **NEXT** [37] uses a LSTM encoder-decoder architecture to predict persons' movements and utilizes rich visual features about human behavioral information and interaction with their surroundings.
4. **DESIRE** [30] combines a RNN encoder-decoder with a CVAE and uses the person's past trajectory and scene context to predict the future trajectory.
5. **SGN LSTM** [72] is a stochastic trajectory predictor which uses LSTM and directed social graph which is dynamically constructed on timely location and speed direction.
6. **FSGAN** [47] extends Social GAN [19] by incorporating adversarial loss in the trajectory prediction task.

For comparison, we adopt the error metrics from prior work [1, 30]:

1. *Average Displacement Error* (ADE) is the average L2 distance between the prediction and the ground truth over all time steps defined as:

$$ADE = \frac{1}{N \cdot T_{pred}} \sum_{i=1}^{N} \sum_{t=1}^{T_{pred}} ||Y_{i,t} - \widetilde{Y}_{i,t}||_2 \qquad (7)$$

where Y is the ground truth trajectory, $\widetilde{Y}$ is the predicted trajectory, T_{pred} is the total number of time steps and N is the total amount of trajectories (or datapoints).

2. *Final Displacement Error* (FDE) is the L2 distance between the prediction and the ground truth at the last time step (in our experiments: 4.8 seconds) defined as:

$$FDE = \frac{1}{N} \sum_{i=1}^{N} \|Y^i_{T_{pred}} - \widetilde{Y}^i_{T_{pred}}\|_2 \tag{8}$$

4.2.4 *Results and Analysis.*

The results for the ETH/UCY dataset are summarized in Table 2. Our M2P3 model with a unit Gaussian prior performs the worst as it is unable to fully capture all of the modes of trajectory distribution. Its predictions are simply forced around the mean of this single Gaussian. However, by exchanging the prior with a Mixture-of-Gaussians (MoG) one, the M2P3-MoG was able to successfully capture multiple modes of the data. Even though the M2P3-MoG uses only the past trajectory as an input, it achieved the lowest prediction error in our experiments. This suggests that a generative model with a diverse prior is crucial for achieving state-of-the-art results on this particular dataset. A disadvantage of the MoG prior is that one needs to manually choose the amount of mixture components (in this case 5) but that can be addressed by hyper-parameter tuning on a validation set.

4.3 Stanford Drone Dataset

4.3.1 *Dataset.*

For our third comparative performance evaluation, we used the Stanford drone dataset (SSD) [55]. The SSD consists of trajectories of pedestrians, cyclists, vehicles and skateboarders captured from a bird's eye view with a drone in 60 different scenes of the Stanford university campus. We have used the dataset split (train/test) as defined in the TrajNet benchmark [4]. Table 4 shows the scenes used for training and testing. Each unique scene is defined as a name-number pair. The dataset provides the locations of the agents in pixel coordinates and serves as a good benchmark for the multimodality of the output of a path-predictor since it contains a diverse set of agents and scene elements such as walkways, roads, roundabouts and buildings. The SSD is captured at 30 frames per second, therefore we observe agents' trajectories for 96 frames (3.2 seconds) and predict their trajectories in the next 144 frames (4.8 seconds) in order to match the setting of previous works [58, 30].

4.3.2 *Implementation.*

Similar to the setting of Section 4.2 we replace the unit gaussian prior with Mixture-of-Gaussians (MoG) in order to better capture the multimodal distribution of trajectories in the dataset. Since the SSD is captured from a bird-eye-view, we don't utilize the object scale but just its past trajectory in the pixel space. Instead of predicting absolute coordinates, we predict pixel displacements relative to the last observed frame. This solves the problem of varying frame resolutions between the train and the test set. The training is carried out with the ADAM optimizer with a learning rate of 1e-5 for 100 epochs.

Table 3: Stanford drone dataset (SSD) train/test split as defined in the TrajNet benchmark.

SSD Dataset	
Split	Scenes
Train	bookstore: 0-3, coupa: 3, deathCircle: 0-4, gates: 0, 1, 3-8,hyang: 4-7, 9, nexus: 0, 1, 3, 4, 7-9
Test	coupa: 0,1, gates: 2, hyang: 0, 1, 3, 8, little: 0-4, nexus: 5, 6,quad: 0-4

4.3.3 *Baselines and Metrics.*

For the evaluation, we follow the protocol from [19] and [58] where each method predicts k trajectories and the one closest to the ground truth is selected. The error metrics are the average (ADE) and final (FDE) displacement errors as defined in Section 4.2.

We compare our model to the following baselines:

1. **Linear** estimates a single (k=1) linear trajectory by finding the parameters of linear regression model by minimizing the least square error.

2. **Social Force** [70] uses a behavioral model as an energy function to express how desirable a certain direction is for the pedestrian.

3. **Social-LSTM** [1] combines LSTM with a social pooling mechanism to the future trajectory of each pedestrian.

4. **CAR-Net** [59] combines an attentive recurrent network with a convolutional neural network (CNN) to predict the trajectory of an agent.

5. **DESIRE** [30], **Social-GAN** [19] and **Sophie** [58] as described in Section 4.2

Table 4: ADE/FDE Results on the Stanford Drone Dataset (SSD) in pixel coordinates for a prediction horizon of 4.8 seconds.

SSD Dataset Results	
Method	ADE/FDE in pixels
Linear (k=1)	37.11 / 63.51
Social Force (k=1)	36.48 / 58.14
Social-LSTM (k=20)	31.19 / 56.97
Social-GAN (k=20)	27.25 / 41.44
CAR-Net (k=1)	25.72 / 51.8
DESIRE (k=5)	19.25 / 34.05
SoPhie (k=20)	**16.27 / 29.38**
Ours (k=20)	27.14 / 56.85

4.3.4 Results and Analysis.

The results for the Stanford Drone Dataset (SSD) are summarized in Table 4. Our model is able to outperform about half of the baselines by just using the past trajectory as an input. The agent trajectories in the SSD dataset are more difficult to predict than the ones in the ETH/UCY dataset because there are 60 different scenes, each having its own layout and agents interacting between each other. Relying just on past trajectory in such scenarios is a strong limitation for a path-prediction model. We experiment with different number of encoder/decoder units (from 128 to 256) as well as the number of gaussians but the results of our model didn't change significantly. This suggests that for this particular dataset, additional input modalities such as scene context, are required for more accurate results. However, we strongly believe that our model can be used as a solid foundation to expand upon.

5. CONCLUSIONS

In this paper, we presented a novel solution, named M2P3, for the egocentric multi-modal multi-pedestrian path prediction problem. M2P3 combines a conditional variational autoencoder with a recurrent neural network encoder-decoder architecture. It uses a vehicle-mounted RGB camera for egocentric vision, takes two inputs by computing past trajectories and scales of tracked pedestrians in the field of car perception with egocentric vision, and then outputs diverse trajectories together with their probability of occurrence. Results of comparative experimental evaluation on the JAAD dataset showed that the M2P3 model can outperform selected state-of-the-art solutions. Furthermore, the M2P3 with a simple change of the prior to a Mixture-of-Gaussians already showed comparable performance to that of more complex state-of-the-art path predictors over the prominent ETH/UCY and SSD datasets.

Ongoing work is concerned with the separation of car ego motion from pedestrian motion, and the ablative investigation of integrating additional inputs to the model such as scene context and social interaction-based intention of pedestrians across street [44]. Additionally, increasing the diversity of the output as well as incorporating even more sophisticated prior, will be further investigated.

6. ACKNOWLEDGEMENT

This research was supported by iMotion Germany GmbH and the German Federal Ministry for Education and Research (BMB+F) in the project REACT.

7. REFERENCES

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, June 2016.

[2] S. Ardeshir and A. Borji. Ego2top: Matching viewers in egocentric and top-view videos. In *European Conference on Computer Vision (ECCV)*, 2016.

[3] S. Bambach, S. Lee, D. J. Crandall, and C. Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1949–1957, 2015.

[4] S. Becker, R. Hug, W. Hübner, and M. Arens. An evaluation of trajectory prediction approaches and notes on the trajnet benchmark. *ArXiv*, abs/1805.07663, 2018.

[5] G. Bertasius, H. S. Park, S. X. Yu, and J. Shi. First-person action-object detection with egonet. *ArXiv*, abs/1603.04908, 2017.

[6] A. Bhattacharyya, M. Fritz, and B. Schiele. Long-term on-board prediction of people in traffic scenes under uncertainty. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4194–4202, 2018.

[7] M. Cai, K. M. Kitani, and Y. Sato. A scalable approach for understanding the visual structures of hand grasps. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1360–1366, 2015.

[8] R. Chandra, U. Bhattacharya, C. Roncal, A. Bera, and D. Manocha. Robusttp: End-to-end trajectory prediction for heterogeneous road-agents in dense traffic with noisy sensor inputs. *ArXiv*, abs/1907.08752, 2019.

[9] F. Chollet et al. Keras. https://keras.io, 2015.

[10] S. Danielsson, L. Petersson, and A. Eidehall. Monte carlo based threat assessment: Analysis and improvements. In *2007 IEEE Intelligent Vehicles Symposium*, pages 233–238, June 2007.

[11] D. Ellis, E. Sommerlade, and I. Reid. Modelling pedestrian trajectory patterns with gaussian processes. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 1229–1234, Sep. 2009.

[12] C. Fan, J. Lee, and M. S. Ryoo. Forecasting hand and object locations in future frames. *CoRR*, abs/1705.07328, 2017.

[13] C. Fan, J. Lee, M. Xu, K. K. Singh, Y. J. Lee, D. J. Crandall, and M. S. Ryoo. Identifying first-person camera wearers in third-person videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4734–4742, July 2017.

[14] Z. Fang and A. M. López. Is the pedestrian going to cross? answering by 2d pose estimation. *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1271–1276, 2018.

[15] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. *2011 International Conference on Computer Vision*, pages 407–414, 2011.

[16] J. Firl, H. Stübing, S. A. Huss, and C. Stiller. Predictive maneuver evaluation for enhancement of car-to-x mobility data. In *2012 IEEE Intelligent Vehicles Symposium*, pages 558–564, June 2012.

[17] K. Fragkiadaki, J. Huang, A. Alemi, S. Vijayanarasimhan, S. Ricco, and R. Sukthankar. Motion prediction under multimodality with conditional stochastic networks. *arXiv preprint arXiv:1705.02082*, 2017.

[18] A. Furnari, S. Battiato, K. Grauman, and G. M. Farinella. Next-active-object prediction from egocentric videos. *ArXiv*, abs/1904.05250, 2017.

[19] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[20] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[21] Y. Hoshen and S. Peleg. Egocentric video biometrics. *CoRR*, abs/1411.7591, 2014.

[22] M. Huynh Trung and G. Alaghband. *Trajectory Prediction by Coupling Scene-LSTM with Human Movement LSTM*, pages 244–259. 10 2019.

[23] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. In *IJCAI*, 2016.

[24] R. E. Kálmán. A new approach to linear filtering and prediction. In *Journal of Basic Engineering, 82(1):35-45*, 1960.

[25] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatto. Intent-aware long-term prediction of pedestrian motion. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2543–2549, 2016.

[26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[27] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

[28] I. Kotseruba, A. Rasouli, and J. K. Tsotsos. Joint Attention in Autonomous Driving (JAAD). *arXiv e-prints*, page arXiv:1609.04741, Sep 2016.

[29] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. 06 2014.

[30] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. K. Chandraker. DESIRE: distant future prediction in dynamic scenes with interacting agents. *CoRR*, abs/1704.04394, 2017.

[31] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1346–1353, 2012.

[32] Y. J. Lee and K. Grauman. Predicting important objects for egocentric video summarization. *International Journal of Computer Vision*, 114:38–55, 2014.

[33] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán. Exploiting map information for driver intention estimation at road intersections. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 583–588, June 2011.

[34] C. Y. Li and K. M. Kitani. Pixel-level hand detection in ego-centric videos. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3570–3577, 2013.

[35] Y. Li, A. Fathi, and J. M. Rehg. Learning to predict gaze in egocentric video. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 3216–3223, Washington, DC, USA, 2013. IEEE Computer Society.

[36] Y. Li, Z. Ye, and J. M. Rehg. Delving into egocentric actions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 287–295, 2015.

[37] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. *CoRR*, abs/1902.03748, 2019.

[38] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret. Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot. *Sensors*, 17(9):1967, 2017.

[39] M. Ma, H. Fan, and K. M. Kitani. Going deeper into first-person activity recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1894–1903, 2016.

[40] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. *ArXiv*, abs/1811.02146, 2019.

[41] D. Makris and T. J. Ellis. Spatial and probabilistic modelling of pedestrian behaviour. In *BMVC*, 2002.

[42] H. Minoura, T. Hirakawa, T. Yamashita, and H. Fujiyoshi. Path predictions using object attributes and semantic environment. pages 19–26, 01 2019.

[43] A. Mishra, S. Krishna Reddy, A. Mittal, and H. A. Murthy. A generative model for zero shot learning using conditional variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2188–2196, 2018.

[44] N. Muscholl, A. Poibrenski, M. Klusch, and P. Gebhard. Simp3: Social interaction-based multi-pedestrian path prediction by self-driving cars. In *IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS)*, 2020.

[45] S. M. Oh, J. M. Rehg, T. R. Balch, and F. Dellaert. Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *International Journal of Computer Vision*, 77:103–124, 2007.

[46] H. S. Park, J.-J. Hwang, Y. Niu, and J. Shi. Egocentric future localization. June 2016.

[47] A. A. Parth Kothari. Human trajectory prediction using adversarial loss. In *Proceedings of 19th Swiss Transportation Research Conference (STRC)*, April 2019.

[48] S. Pellegrini, A. Ess, K. Schindler, and L. V. Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268, 2009.

[49] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2847–2854, 2012.

[50] A. Poibrenski, M. Klusch, I. Vozniak, and C. Mueller. M2p3: Multimodal multi-pedestrian path prediction by self-driving cars with egocentric vision. In *Proceedings of 35th ACM International Symposium on Applied Computing (SAC)*, 2020.

[51] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[52] A. Rasouli, I. Kotseruba, and J. K. Tsotsos. Agreeing to cross: How drivers and pedestrians communicate. *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 264–269, 2017.

[53] A. Rasouli and J. K. Tsotsos. Autonomous vehicles that interact with pedestrians: A survey of theory and practice. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–19, 2019.

[54] N. Rhinehart and K. M. Kitani. First-person activity forecasting with online inverse reinforcement learning. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3716–3725, 2017.

[55] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European Conference on Computer Vision (ECCV)*, 2016.

[56] A. V. I. Rosti and M. J. F. Gales. Rao-blackwellised gibbs sampling for switching linear dynamical systems. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–809, May 2004.

[57] A. Rudenko and et al. Human motion trajectory prediction: A survey. In *arXiv preprint arXiv:1905.06113*, 2019.

[58] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, and S. Savarese. Sophie: An attentive GAN for predicting paths compliant to social and physical constraints. *CoRR*, abs/1806.01482, 2018.

[59] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi, and S. Savarese. Car-net: Clairvoyant attentive recurrent network. In *European Conference on Computer Vision (ECCV)*, 2018.

[60] A. Saran, D. Teney, and K. M. Kitani. Hand parsing for fine-grained recognition of human grasps in monocular images. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5052–5058, Sep. 2015.

[61] N. Schneider and D. M. Gavrila. Pedestrian path prediction with recursive bayesian filters: A comparative study. In J. Weickert, M. Hein, and B. Schiele, editors, *Pattern Recognition*, pages 174–183, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[62] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll. The simpler the better: Constant velocity for pedestrian motion prediction. *ArXiv*, abs/1903.07933, 2019.

[63] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

[64] O. Styles, A. Ross, and V. R. Sánchez. Forecasting pedestrian trajectory with machine-annotated training data. *ArXiv*, abs/1905.03681, 2019.

[65] S. Su, J. Pyo Hong, J. Shi, and H. Soo Park. Predicting behaviors of basketball players from first person videos. pages 1206–1215, 07 2017.

[66] J. P. van den Berg, S. J. Guy, M. C. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *ISRR*, 2009.

[67] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. *CoRR*, abs/1703.07402, 2017.

[68] M. Xu, C. Fan, Y. Wang, M. S. Ryoo, and D. J. Crandall. Joint person segmentation and identification in synchronized first- and third-person videos. In *European Conference on Computer Vision (ECCV)*, 2018.

[69] T. Yagi, K. Mangalam, R. Yonetani, and Y. Sato. Future person localization in first-person videos. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7593–7602, 2018.

[70] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, page 1345–1352, USA, 2011. IEEE Computer Society.

[71] Y. Yao, M. Xu, C. Choi, D. J. Crandall, E. M. Atkins, and B. Dariush. Egocentric vision-based future vehicle localization for intelligent driving assistance systems. *CoRR*, abs/1809.07408, 2018.

[72] L. Zhang, Q. She, and P. Guo. Stochastic trajectory prediction with social graph network. *CoRR*, abs/1907.10233, 2019.

[73] M. Zhang, K. T. Ma, J. H. Lim, Q. Zhao, and J. Feng. Deep future gaze: Gaze anticipation on egocentric videos using adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3539–3548, July 2017.

[74] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. pages 12077–12086, 06 2019.

[75] Y. Zhang, Y. Qi, J. Liu, and Y. Wang. Decade of vision-based pedestrian detection for self-driving: An experimental survey and evaluation. In *SAE Technical Paper*. SAE International, 08 2018.

ABOUT THE AUTHORS:

Atanas Poibrenski received a B.Sc. in Computer Science from the University of York in 2014 and a M.Sc. in Visual Computing from the Saarland University in 2018. He is currently a researcher at the German Research Center for Artificial Intelligence (DFKI). His research interests include the use of synthetic data in deep learning as well as multi-pedestrian path prediction for autonomous cars.

Matthias Klusch is a principal researcher and research fellow of the German Research Center for Artificial Intelligence (DFKI) where he heads the Intelligent Information Systems research team of the DFKI department for Agents and Simulated Reality. He is also private docent of computer science at the Saarland University, and co-chair of the AI section of the German Society for Informatics. His research areas in applied AI are hybrid learning and reasoning, agent-based semantic service coordination, and Quantum AI. He obtained his Ph.D. and habilitation (PD) degree in computer science from the University of Kiel, respectively, the Saarland University. Among other, he contributed to many funded AI research projects, served the scientific community on numerous program committees, is on the editorial board of several international scientific journals, and published more than 200 papers in his research areas.

Igor Vozniak is a researcher at the German Research Center for Artificial Intelligence (DFKI), where he is part of the Autonomous Driving team led by Dr.-Ing. Christian Müller under the supervision of Prof. Dr.-Ing. Philipp Slusallek. Mr. Vozniak received his B.Sc. and M.Sc. degrees in Informatics in 2008 and 2010 respectively from the National Technical University of Ukraine “Kyiv Polytechnic Institute” (NTUU “KPI”), renamed to the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". In 2016, Mr. Vozniak received his M.Sc. of Science degree in Informatics from Saarland University. His research area and interests include critical scenario generation in the scope of the autonomous driving domain and related to his field challenges such as human-like pedestrian-agents simulation and hardware in the loop simulations. He is co-author of the renown open-source driving simulation software openDS.

Christian Müller is deputy head of the research area Agents and Simulated Reality at the German Research Center for Artificial Intelligence (DFKI) as well as head of the Competence Center for Autonomous Driving. At the DFKI, he is a Principal Researcher and one of a small number of named DFKI Research Fellow. He regularly leads nationally- and EU-funded large consortium projects and coordinates large research clusters. Christian Müller is a computer scientist and has been working in the field of artificial intelligence for over 20 years. His earliest influences go back to the Collaborative Research Center 378 "Resource-Adaptive Cognitive Processes" consisting of computer scientists, psychologists, and philosophers. Besides his experience in interdisciplinary cooperation, he possesses long-standing philosophical interests and profound knowledge of philosophy in general, and philosophical questions of AI in particular.