Semantic Synthesis of Pedestrian Locomotion

Maria Priisalu[1], Ciprian Paduraru[2,3], Aleksis Pirinen[1], and Cristian Sminchisescu[1,3,4]

[1] Department of Mathematics, Faculty of Engineering, Lund University
[2] The Research Institute of the University of Bucharest (ICUB), Romania
[3] Institute of Mathematics of the Romanian Academy
[4] Google Research
{maria.priisalu,aleksis.pirinen, cristian.sminchisescu}@math.lth.se
ciprian.paduraru@fmi.unibuc.ro

Abstract. We present a model for generating 3d articulated pedestrian locomotion in urban scenarios, with synthesis capabilities informed by the 3d scene semantics and geometry. We reformulate pedestrian trajectory forecasting as a structured reinforcement learning (RL) problem. This allows us to naturally combine prior knowledge on collision avoidance, 3d human motion capture and the motion of pedestrians as observed e.g. in Cityscapes, Waymo or simulation environments like Carla. Our proposed RL-based model allows pedestrians to accelerate and slow down to avoid imminent danger (e.g. cars), while obeying human dynamics learnt from in-lab motion capture datasets. Specifically, we propose a hierarchical model consisting of a semantic trajectory policy network that provides a distribution over possible movements, and a human locomotion network that generates 3d human poses in each step. The RL-formulation allows the model to learn even from states that are seldom exhibited in the dataset, utilizing all of the available prior and scene information. Extensive evaluations using both real and simulated data illustrate that the proposed model is on par with recent models such as S-GAN, ST-GAT and S-STGCNN in pedestrian forecasting, while outperforming these in collision avoidance. We also show that our model can be used to plan goal reaching trajectories in urban scenes with dynamic actors.

1 Introduction

Pedestrian trajectory prediction is an important sub-problem for safe autonomous driving. Recent 3d traffic datasets [1–6] focus on bounding box detection and prediction of cars and pedestrians. Bounding boxes are popular since they provide information on the location and velocity of the travelling object, and are relatively well suited to model cars, but neglect the detailed motion cues present in pedestrian posture. Pedestrian poses compactly model posture and motion cues and have been shown effective in pedestrian intent prediction [7–9]. However, to our knowledge there exists no large-scale datasets with ground truth annotations of pedestrian poses in traffic. Moreover, most previous work in pedestrian pose modelling has been performed without spatial reasoning [7, 9] or using action-conditioned human models [8]. In contrast, we formulate pedestrian synthesis

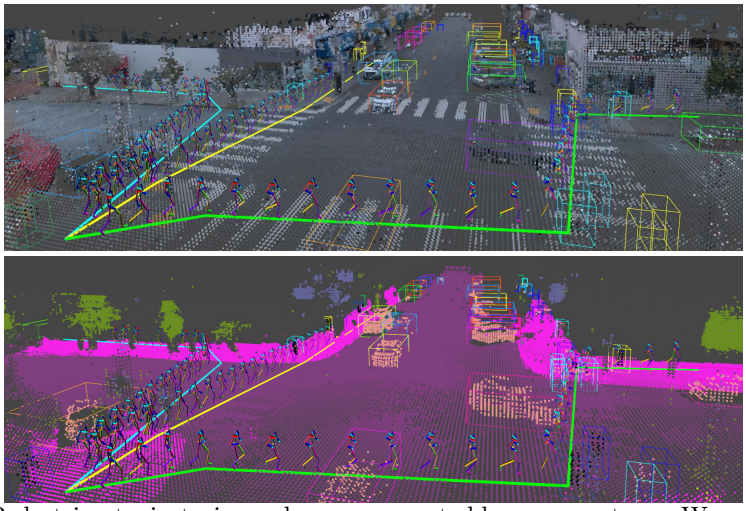

Fig. 1: Pedestrian trajectories and poses generated by our agent on a Waymo scene. RGB and semantic pointclouds of the scene are shown in the top and bottom images, respectively. A local neighborhood of these pointclouds are observed by the agent. Coloured lines on the ground show different trajectories taken by the agent when initialized with varying agent histories, cf. §2.2. The agent crosses the roads without collisions. Cars and other pedestrians in the scene are shown as positioned in the first frame and are surrounded by bounding boxes for clarity.

as a 3d scene reasoning problem that is constrained by human dynamics and where the generated motion must follow the scene's 3d geometric and semantic properties as seen in Fig. 1. To impose human dynamics, the articulated pose trajectories are conditioned on the current and past poses and velocities.

Specifically, we propose a *semantic pedestrian locomotion* (SPL) agent, a hierarchical articulated 3d pedestrian motion generator that conditions its predictions on both the scene semantics and human locomotion dynamics. Our agent first predicts the next trajectory location and then simulates physically plausible human locomotion to that location. The agent explicitly models the interactions with objects, cars and other pedestrians surrounding it, as seen in Fig. 2. We develop two different pedestrian locomotion generators – one without any restrictions that can roll forward from a given starting location, and one which is additionally conditioned on a target location. The former is useful for simulating generic pedestrian motion in traffic situations, while the latter can be used to control the simulation target, for example when generating high-risk scenarios. Moreover, our model can be used to augment existing traffic datasets with articulated poses. For example, the 3d poses generated by the SPL agent can be used to produce dense pedestrian predictions by applying a pose conditioned human mesh such as SMPL [10]. Augmented pedestrians can then be produced in semantic segmentation masks by projecting the dense pedestrian mesh onto the image plane. RGB images can be augmented similarly, but this may additionally require a photorealistic style transfer similar to [11, 12]. Alternatively, LiDAR augmentations can be generated by sampling [13] from the dense bodies.

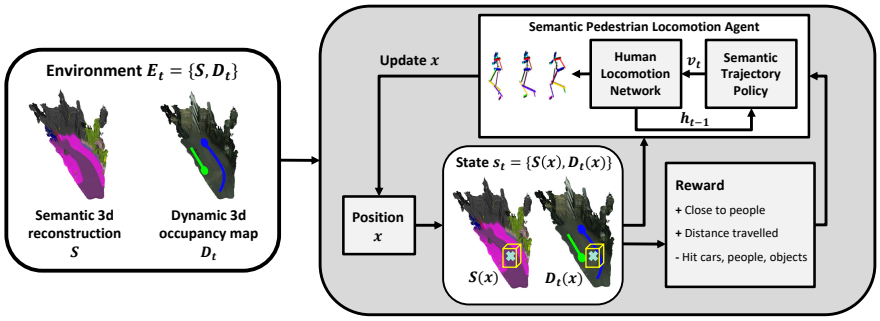

Fig. 2: Semantic pedestrian locomotion (SPL) agent and framework. The 3d environment $E_t = \{S, D_t\}$ consists of a semantic map S of static objects and a dynamic occupancy map D_t of cars and people at time t (shown as blue and green trajectories, ellipsoids indicate the positions at time t). The agent observes a top-view projection of a local crop (yellow box) of E_t. A velocity $\boldsymbol{v}_t$ is sampled from the semantic trajectory policy network (STPN). The human locomotion network (HLN) models the articulated movement of the step $\boldsymbol{v}_t$. Note that the STPN observes pose information via the previous hidden state $\boldsymbol{h}_{t-1}$ from the HLN. In training, a reward evaluating the subsequent state is given to the agent.

Learning to synthesize pedestrian motion is difficult, since the diversity among expert pedestrian trajectories is often limited in the training data, especially for high-risk scenarios. A trajectory generation model trained via imitation learning is unlikely to act reliably in situations that are not present in the training data. This implies e.g. that such an agent will likely behave poorly in near collision scenarios, as these are not present in existing datasets. Recent work on generative adversarial imitation learning (GAIL) [14] has recently gained popularity within trajectory forecasting [15–18] since it models the data distribution rather than cloning expert behaviour. GAIL is an inverse RL method where a policy tries to mimic the experts and the reward function aims to discriminate policy trajectories from expert trajectories. However, as for behaviour cloning, GAIL cannot learn reliable behaviour in situations that are highly different from those available in the training data, since the discriminator will in such cases be able to trivially distinguish between generated and expert trajectories.

To allow our SPL agent to learn also from states outside the training set, in §2 we pose the trajectory forecasting problem in the framework of reinforcement learning (RL). We extrapolate the learning signal with an optima-preserving reward signal that additionally involves prior knowledge to promote e.g. collision avoidance. We adapt the RL policy sampling process to simultaneously optimize the trajectory forecasting loss and maximize the reward. Moreover, our analysis in §2 can be used to adapt any trajectory forecasting model into a robust articulated pedestrian synthesis model. By sampling initial positions of the agent in different locations, all of the spatio-temporal data in the driving dataset can be utilized. Because we train on a large number of different spatial locations and in near-collision scenarios, our motion synthesis model learns to generate

plausible trajectories even in states that are far from expert trajectories such as near-collision scenarios. In summary, our contributions are as follows:

- We propose an articulated 3d pedestrian motion generator that conditions its predictions on both the scene semantics and human locomotion dynamics. The model produces articulated pose skeletons for each step along the trajectory.
- We propose and execute a novel training paradigm which combines the sample-efficiency of behaviour cloning with the open-ended exploration of the full state space of reinforcement learning.
- We perform extensive evaluations on Cityscapes, Waymo and CARLA and show that our model matches or outperforms existing approaches in three different settings: i) for pedestrian forecasting; ii) for pedestrian motion generation; and iii) for goal-directed pedestrian motion generation.

1.1 Related Work

In pedestrian trajectory forecasting, social interactions of pedestrians have been modelled with different GAN-based approaches [19, 20], by social graphs [21–23], by recurrent networks [24, 25] and by temporal convolutions [26, 27]. Differently from us, these approaches only model the social interactions of pedestrians and ignore cars and obstacles. An attention model is used by [28] to forecast pedestrian trajectory given environmental features and GAN-based social modelling that neglects cars. Differently from [19–28] we utilize a locomotion model and therefore do not need to learn human dynamics from scratch. All of the mentioned supervised models in pedestrian forecasting can in principle be trained with our proposed methodology (cf. §2) to extend to unobserved states.

Our model does not rely on action detection (e.g. "walking" or "standing") of the expert dataset for trajectory forecasting, as opposed to action conditioned intention detection networks [8, 29] and motion forecasting models [30, 31]. Instead the pedestrian's future trajectory is conditioned on its past trajectory. A benefit of our approach is thus that it avoids dealing with temporal ambiguities associated with action detection. Recently it has been shown that pedestrian future augmentation can improve pedestrian forecasting features [32]. Our generator produces articulated 3d trajectories on real data, and in comparison to [33] we do not require the recreation of the full dataset in a simulation environment. We note that the goal-reaching version of our model could be utilized with a goal proposal method [34] to provide multiple future augmentations to data.

Human synthesis models for still images [18, 11, 35–37] aim to synthesize poses in semantically and geometrically plausible ways in images, and have no temporal modelling, but could be used to initialize the SPL's pose trajectories. The works [36, 37] model likely locations for humans in images. The models in [18, 11, 35] synthesize pedestrians with 3d models, but do this only in static scenes. Similar to us, the affordance model of [38] explicitly incorporates 3d scene semantics to propose plausible human poses, but only for static scenes. Synthetic videos are generated in [39] by cropping humans from sample videos and pasting them into target videos followed by visual smoothing with a GAN, but this approach does not guarantee semantic plausibility.

The majority of 3d human pose forecasting models concentrate on predicting future poses given only the past pose history [40–42]. In [43] human pose futures are predicted on a static dataset by forecasting a trajectory, to which poses are fitted by a transformer network. Differently from our work the reasoning is performed in 2d which leads to geometrically implausible failure cases. [44] forecast pedestrian motion by combining a pose predicting GRU [45] with social pooling and a 2d background context layer. Both [43, 44] are not readily applicable to driving datasets as they lack modelling of cars and require access to high quality 2d human poses which are in general hard to obtain in driving datasets.

2 Methodology

The pedestrian trajectory forecasting problem on a dataset $\mathcal{D}$ of pedestrian trajectories can be formulated as follows. Let $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \ldots \boldsymbol{x}_T$ be a pedestrian trajectory[5] of length T in $\mathcal{D}$. Given the trajectory $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_t$ up to timestep t we would like to predict the pedestrian's position in the next timestep $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{v}_t$, where $\boldsymbol{v}_t$ is the step taken by the pedestrian from $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t+1}$. Each position $\boldsymbol{x}_t$ is associated with a state $\boldsymbol{s}_t$, described in detail in §2.1, that includes the pedestrian's past trajectory and other relevant scene information at position $\boldsymbol{x}_t$. We denote the density function of the random variable $\boldsymbol{v}_t$ conditioned on $\boldsymbol{s}_t$ as $p(\boldsymbol{v}_t|\boldsymbol{s}_t)$. The prediction task is to estimate $p(\boldsymbol{v}_t|\boldsymbol{s}_t)$ by a parametric function $p_\Theta(\boldsymbol{v}_t|\boldsymbol{s}_t)$ where the step forecast is $\hat{\boldsymbol{v}}_t = \max_{\boldsymbol{v}_t} p_\Theta(\boldsymbol{v}_t|\boldsymbol{s}_t)$. The maximum likelihood estimate of the model parameters Θ is then given by

$$\Theta^* = \arg\max_\Theta \log \mathcal{L}(\Theta|\mathcal{D}) = \arg\max_\Theta \sum_{\mathcal{D}} \sum_{t=0}^{T-1} \log p_\Theta(\boldsymbol{v}_t|\boldsymbol{s}_t) \tag{1}$$

From the RL perspective on the other hand, an agent has an initial position $\boldsymbol{x}_0$ and takes steps by sampling from a parametric policy: $\boldsymbol{v}_t \sim \pi_\Theta(\boldsymbol{v}_t|\boldsymbol{s}_t)$. After taking a step $\boldsymbol{v}_t$ the agent finds itself in a new location $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{v}_t$ and in training receives a reward $R(\boldsymbol{s}_t, \boldsymbol{v}_t)$. The objective is to find a policy π_Θ that maximizes the expected cumulative reward,

$$J(\Theta) = \mathbb{E}_{\pi_\Theta}\left[\sum_{t=0}^{T-1} R(\boldsymbol{s}_t, \boldsymbol{v}_t)\right] \tag{2}$$

Comparing the RL perspective with the standard forecasting formulation, we first note that $\pi_\Theta(\boldsymbol{v}_t|\boldsymbol{s}_t) = p_\Theta(\boldsymbol{v}_t|\boldsymbol{s}_t)$. Furthermore, the optima of (1) is unchanged if it is multiplied by a function $R(\boldsymbol{s}_t, \boldsymbol{v}_t)$ that obtains its maximum at all $(\boldsymbol{s}_t, \boldsymbol{v}_t) \in \mathcal{D}$, i.e. on the expert trajectories. Thus, assuming that the actions taken by the pedestrians in $\mathcal{D}$ are optimal in the reward function R, we can rewrite the maximum likelihood objective (1) as a Monte Carlo estimate of the

[5] The $\boldsymbol{x}_t$ are 2d locations in the movement plane.

policy gradient objective [46], sampled from the expert trajectories $(\boldsymbol{s}_t, \boldsymbol{v}_t) \in \mathcal{D}$:

$$\Theta^* = \arg\max_{\Theta} \sum_{\mathcal{D}} \sum_{t=0}^{T-1} \log \pi_\Theta(\boldsymbol{v}_t|\boldsymbol{s}_t) R(\boldsymbol{s}_t, \boldsymbol{v}_t) \tag{3}$$

We can now unify the policy gradient objective (3) and the supervised objective (1) by sampling respectively from $(\tilde{\boldsymbol{s}}_t, \tilde{\boldsymbol{v}}_t) \sim \pi_\Theta$ and $(\boldsymbol{s}_t, \boldsymbol{v}_t) \in \mathcal{D}$. Optimizing (3) while sampling from both the expert trajectories and the current parametric policy equates to iteratively optimizing the policy gradient objective and the maximum likelihood objective. Thus we have shown that (1) can be rewritten as a policy gradient objective assuming a reward function that obtains its optima on $\mathcal{D}$. In §2.4 we construct a reward function that fulfills this criteria.

By posing the supervised learning problem of pedestrian trajectory forecasting as an RL problem, the detailed human dynamics model HLN becomes part of the observable environment dynamics and does not need to be modelled explicitly in the trajectory prediction model π_Θ. This is a natural way of combining accurate human motion models trained on in-laboratory motion capture data [47, 48] with trajectories available in autonomous driving datasets [1–6].

In the following subsections we present our SPL agent, which performs human 3d motion synthesis within two modules. First a semantic pedestrian locomotion network (STPN) samples a step $\boldsymbol{v}_t$ based on $\boldsymbol{s}_t$, and then a human locomotion network (HLN) generates realistic body joint movements to the next position $\boldsymbol{x}_{t+1}$. The HLN is first trained in a supervised fashion (see §2.3). Then the STPN and HLN modules are combined, and the STPN is trained by alternating[6] between sampling from expert trajectories and from arbitrary states, following the objective (3). Fig. 2 provides an overview of the SPL model.

2.1 States and Actions

The agent acts in the voxelized 3d environment $E_t = \{S, D_t\}$ over the time horizon $\{0, \ldots, T\}$, where E_t is a 3d pointcloud reconstruction of a scene with resolution 20 cm × 20 cm × 20 cm. The reconstruction E_t consists of stationary objects S and a dynamic occupancy map D_t of moving objects. Specifically, the dynamic occupancy map marks the timestamps of voxel occupancies by other pedestrians and cars (in separate channels) in the time horizon $\{0, \ldots, T\}$. For past timesteps $0 - t$ the dynamic occupancy map contains the past trajectories of cars and pedestrians, while a constant velocity model is used to predict the future $t + 1, \ldots, T$. Further details of D_t are in the supplement. Each 3d point in E_t is described by a semantic label l and an RGB-color label $\boldsymbol{c}$. We let $E_t(\boldsymbol{x}_t) = \{S(\boldsymbol{x}_t), D_t(\boldsymbol{x}_t)\}$ denote a 5 m × 5 m × 1.8 m rectangular 3d crop of E_t centered at the agent's current position $\boldsymbol{x}_t$ and touching the ground.

The agent's state at time t consists of its external semantic state $\boldsymbol{s}_t$ and the internal locomotion state $\boldsymbol{l}_t$. The external semantic state is defined as

$$\boldsymbol{s}_t = \{E_t^{2d}(\boldsymbol{x}_t), \boldsymbol{v}_{t-N}, \ldots, \boldsymbol{v}_{t-1}, \boldsymbol{d}_v, \boldsymbol{h}_{t-1}\} \tag{4}$$

[6] See details of the alternating training in the supplement.

where $E_t^{2d}(\boldsymbol{x}_t)$ is a top-view projection of $E_t(\boldsymbol{x}_t)$, $\boldsymbol{v}_{t-N}, \ldots, \boldsymbol{v}_{t-1}$ constitute the agent's movement history for the past $N = 12$ timesteps, $\boldsymbol{d}_v$ is the displacement[7] to the closest vehicle, and $\boldsymbol{h}_{t-1}$ is the hidden layer of the HLN (cf. §2.3) which informs about the agent's posture, pose and acceleration. The locomotion state

$$\boldsymbol{l}_t = \{\boldsymbol{x}_{t-M}, \ldots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t, \boldsymbol{g}_{t-M}, \ldots \boldsymbol{g}_{t-1}, \boldsymbol{g}_t, \boldsymbol{j}_t, \boldsymbol{i}_t, \boldsymbol{x}_{t+1}, |\boldsymbol{v}_t|\} \tag{5}$$

consists of the past positions $\boldsymbol{x}_{t-M}, \ldots, \boldsymbol{x}_{t-1}$ of the agent ($M = 11$), the current position $\boldsymbol{x}_t$, the past gait characteristics $\boldsymbol{g}_{t-M}, \ldots, \boldsymbol{g}_{t-1}$, the next step's gait $\boldsymbol{g}_t$, the joint positions and velocities $\boldsymbol{j}_t$ and $\boldsymbol{i}_t$, the next trajectory position $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{v}_t$, and the speed $|\boldsymbol{v}_t|$. The gait characteristic $\boldsymbol{g}_t$ is a binary vector indicating if the agent is standing, walking or jogging and is regressed from $|\boldsymbol{v}_t|$. The joint positions $\boldsymbol{j}_t$ are the 3d positions of the root-joint centered 30 BVH joints of the CMU motion capture data [49].

2.2 Semantic Trajectory Policy Network (STPN)

The STPN is a neural network that parametrizes $\pi_\Theta(\boldsymbol{v}_t|\boldsymbol{s}_t)$, the velocity distribution of the agent in position $\boldsymbol{x}_t$ with state $\boldsymbol{s}_t$. We factorize $\pi_\Theta(\boldsymbol{v}_t|\boldsymbol{s}_t)$ into a Gaussian distribution over speed $|\boldsymbol{v}_t|$, and a multinomial distribution over discretized unit directions $\boldsymbol{u}_t$. Since the agent is acting and observing the world in a regular voxel grid, the movement directions are discretized into the eight directions North (N), North-East (NE) and so on: N, NE, E, SE, S, SW, W, NW, as well as a no-move action. After the velocity $\boldsymbol{v}_t$ is sampled, the agent's next position $\boldsymbol{x}_{t+1}$ is given by the HLN in §2.3. The new position is often close to $\boldsymbol{x}_t + \boldsymbol{v}_t$ but could be adjusted by the HLN to ensure physical plausibility.

The policy $\pi_\Theta(\boldsymbol{v}_t|\boldsymbol{s}_t)$ is parameterized by a neural network, consisting of a convolutional features extractor, an agent history encoder and two parallel fully connected (FC) layers. The convolutional features extractor consists of two convolutional layers of size $(2, 2, 1)$ with ReLU activations and max pooling. The agent history encoder is a a 32-unit LSTM [50] that extracts a temporal feature vector $\boldsymbol{f}_t$ from the agent's past trajectory $\boldsymbol{v}_{t-N}, \ldots, \boldsymbol{v}_{t-1}$. The parallel FC layers both receive[8] as input the convolutional features, the temporal features $\boldsymbol{f}_t$, the displacement vector $\boldsymbol{d}_t$ and the hidden state[9] $\boldsymbol{h}_{t-1}$ of the HLN. The previous unit direction $\boldsymbol{u}_{t-1}$ is added as a prior to the output of the first FC layer, and the result is then fed through a softmax activation to output a probability distribution over the unit directions $\boldsymbol{u}_t$. The second FC layer is activated by the sigmoid function which is scaled with the maximal speed 3 m/s to produce μ_t, the mean of the normal distribution that models the speed taken at time t. Hence $|\boldsymbol{v}_t| \sim \mathcal{N}(\mu_t, \sigma)$, where σ is exponentially decreased from 2 to 0.1 in training. Finally, the sampled velocity $\boldsymbol{v}_t$ is given by $\boldsymbol{v}_t = |\boldsymbol{v}_t|\boldsymbol{u}_t$.

[7] This is comparable to a pedestrian being aware of cars in its vicinity.
[8] The goal-directed agent additionally includes the direction to the goal at this stage.
[9] The previous hidden state is used, as the HLN is executed after the STPN.

2.3 Human Locomotion Network (HLN)

The HLN produces 3d body joint positions to take a step $\boldsymbol{v}_t$ from $\boldsymbol{x}_t$. The HLN is adapted from [51] with the addition of a velocity regression layer that estimates $\boldsymbol{g}_t$ in (5) from $\boldsymbol{v}_t$. Network weights are learnt following the data and procedure in [51]. The HLN is a phase function network that is conditioned on the walking phase of the body at time t, where the phase varies from 0 to 2π for a full cycle from the right foot touching the ground until the next occurrence of the right foot touching the ground. The HLN regresses $\boldsymbol{j}_{t+1}, \boldsymbol{i}_{t+1} = h(\boldsymbol{l}_t)$, i.e. the joint positions $\boldsymbol{j}_{t+1}$ and velocities $\boldsymbol{i}_{t+1}$, conditioned on the current state $\boldsymbol{l}_t$ (see §2.1).

The next position $\boldsymbol{x}_{t+1}$ of the agent is set to the plane coordinates of the pelvis joint in $\boldsymbol{j}_{t+1}$ at timestep $t+1$ (the agent is not allowed to move through objects). The HLN architecture consist of three fully connected layers with 512 hidden units per layer and an exponential rectified linear function [52] as the activation function. The last hidden layer $\boldsymbol{h}_t$ is observed by the STPN in the next timestep, informing it of the agent's current posture. Network weights are trained for different walking phases by augmenting surface curvature for constant feet to ground distances from motion capture data as reported in [51].

2.4 Reward Signal

In training the agent's state is evaluated by the reward function $R_t = R(\boldsymbol{x}_t, \boldsymbol{v}_t)$ at each step. We wish to estimate the optimal policy $\pi_{\boldsymbol{\Theta}^\star}(\boldsymbol{v}_t|\boldsymbol{s}_t)$ that maximizes the total expected reward. The reward function is designed so that its maximal value occurs on the expert trajectories, as discussed in §2. A reward $R_d = 1$ is given for visiting a pedestrian trajectory in the dataset $\mathcal{D}$, otherwise $R_d = 0$. The reward is given only for newly visited locations to promote the agent to move. We also encourage the agent to move close to positions where pedestrians tend to appear. To approximate a pedestrian density map from $\mathcal{D}$ we apply an exponential kernel on the trajectory locations in $\mathcal{D}$, i.e.

$$R_k(\boldsymbol{x}_t, \boldsymbol{v}_t) = \log\left\{\frac{1}{b}\sum_{\boldsymbol{x}^i \in D}\sum_{t'=0}^{T}\exp\{-\|\boldsymbol{x}^i_{t'} - \boldsymbol{x}_{t+1}\|\}\right\} \tag{6}$$

where b is the bandwidth (we set $b = 0.0001$) and the sum is over all pedestrian trajectory positions $\boldsymbol{x}^i$ in the dataset $\mathcal{D}$. We gather the terms that encourage the agent to stay near trajectories in $\mathcal{D}$ as $R_{ped}(\boldsymbol{x}_t, \boldsymbol{v}_t) = R_k(\boldsymbol{x}_t, \boldsymbol{v}_t) + R_d(\boldsymbol{x}_t, \boldsymbol{v}_t)$.

To penalize collisions, let R_v, R_p and R_s be negative indicator functions that are active if the agent collides with vehicles, pedestrians and static objects, respectively. The terms are gathered as $R_{coll}(\boldsymbol{x}_t, \boldsymbol{v}_t) = R_v(\boldsymbol{x}_t, \boldsymbol{v}_t) + R_p(\boldsymbol{x}_t, \boldsymbol{v}_t) + R_s(\boldsymbol{x}_t, \boldsymbol{v}_t)$. Note that $R_{ped}(\boldsymbol{x}_t, \boldsymbol{v}_t)$ is only given when $R_p(\boldsymbol{x}_t, \boldsymbol{v}_t) = 0$.

To encourage smooth transitions between the exhibited poses and to penalize heavy effort motions, we penalize the average yaw ϕ (in degrees) of the joints in the agent's lower body as $R_\phi(\boldsymbol{x}_t, \boldsymbol{v}_t) = \max(\min(\phi - 1.2, 0), 2.0)$. Thus the full reward[10] is $R(\boldsymbol{x}_t, \boldsymbol{v}_t) = R_{coll}(\boldsymbol{x}_t, \boldsymbol{v}_t) + R_{ped}(\boldsymbol{x}_t, \boldsymbol{v}_t) + R_\phi(\boldsymbol{x}_t, \boldsymbol{v}_t)$.

[10] Each term weighted with the respective weights, $\lambda_v = 1$, $\lambda_p = 0.1$, $\lambda_s = 0.02$, $\lambda_k = 0.01$, $\lambda_d = 0.01$, $\lambda_\phi = 0.001$.

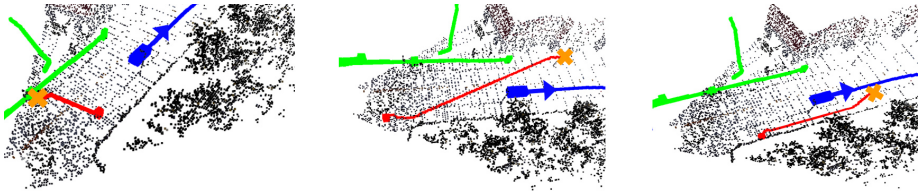

Fig. 3: Several 1-minute trajectories of our SPL-goal agent reaching its goal location in orange (maximum distance to goal: 120 m) on the CARLA test set. Car, person and agent trajectories are shown in blue, green and red respectively. *Left:* Agent sharply but safely crossing the street to reach a goal. *Middle:* Agent safely crossing the street as no cars are approaching. *Right:* Agent safely moving along the pavement when given a goal on the road. The shortest path to the goal would involve walking on the road for a longer amount of time, so the agent balances its desire to reach the goal with the risk of being on the road.

When the agent is given a goal location, every step taken towards the goal should provide a reward for the improvement made relative to the initial goal distance. Thus, given a goal location $\boldsymbol{x}_g$ we define

$$R_g(\boldsymbol{x}_t, \boldsymbol{v}_t) = \begin{cases} 1 & \text{if } \|\boldsymbol{x}_{t+1} - \boldsymbol{x}_g\| < \epsilon \\ 1 - \frac{\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_g\|}{\|\boldsymbol{x}_t - \boldsymbol{x}_g\|} & \text{otherwise} \end{cases} \tag{7}$$

where ϵ defines the distance from the goal location to the agent center.[11] The full reward[12] of the goal reaching agent is $R(\boldsymbol{x}_t, \boldsymbol{v}_t) = R_{coll}(\boldsymbol{x}_t, \boldsymbol{v}_t) + R_{ped}(\boldsymbol{x}_t, \boldsymbol{v}_t) + R_{\phi}(\boldsymbol{x}_t, \boldsymbol{v}_t) + R_g(\boldsymbol{x}_t, \boldsymbol{v}_t)$. Note that the goal-driven reward does not necessarily reach its optima on expert trajectories, as the it is not assumed that $\boldsymbol{x}_g \in \mathcal{D}$.

2.5 Policy Training

With a finite sequence length T, a large number of states are in practice unreachable for the agent with an initial location $\boldsymbol{x}_0$. However, thanks to the RL reformulation the agent can be initialized in any location. By regularly choosing information dense $\boldsymbol{x}_0$, the number of samples needed to learn critical behaviours such as collision avoidance can be reduced, and thus the agent is initialized in front of cars, near pedestrians, randomly, on pavement and on pedestrians. Agents are trained in Tensorflow [53] using Adam [54] with a batch size of 30 trajectories, learning rate of 10^{-3}, and a discount rate of 0.99.

3 Experiments

The proposed pedestrian motion generation agent is evaluated on both simulated and real data, with and without target goals. The goal-free and goal-directed

[11] We set $\epsilon = 20\sqrt{2}$ cm, i.e. the agent must overlap the goal area.

[12] The weights except for $\lambda_v = 2$, $\lambda_g = 1$ are the same. The fraction term of R_g is weighted by 0.001.

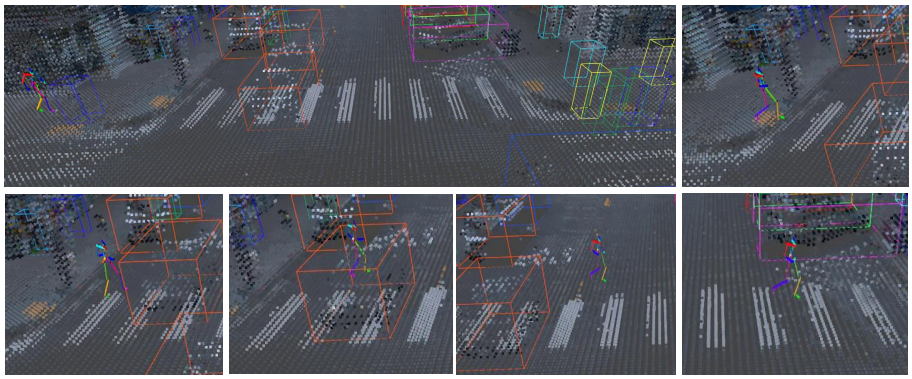

Fig. 4: Subsampled pose sequence in a Waymo test scene, showing the SPL agent walking behind a car (indicated with an orange 3d bounding box) to avoid a collision, and then returning to the crosswalk. A zoomed out view of the scene at the beginning of the agent's trajectory is shown in the top left.

agents are denoted SPL and SPL-goal, respectively. Since the human locomotion network (HLN) described in §2.3 imposes realistic human dynamic constraints, we present all results with the HLN performing joint transformations along the trajectories. We compare SPL with the following methods:

- *Behaviour cloning (BC)* is an imitation learning baseline. BC is trained with the same network structure as SPL, but by only sampling from $\mathcal{D}$, i.e. max-likelihood forecasting. The same hyperparameters as for SPL are used.
- *Constant velocity (CV)* models pedestrian motion with a constant velocity, which as shown in [55] is surprisingly effective in many cases. When initialized on a pedestrian it continues with the last step velocity of that pedestrian. When initialized elsewhere, a Gaussian with $\mu = 1.23$ and $\sigma = 0.3$ (same as [56]) is used to estimate speed and the direction is drawn at random.
- *S-GAN* is the Social-GAN [19] used for pedestrian forecasting.
- *S-STGCNN* (*S-STG* in tables) the Social Spatio-Temporal Graph Convolutional Neural Network [23], a pedestrian trajectory forecasting network.
- *ST-GAT* is the Spatial-Temporal Graph Attention Network [22], another recent pedestrian trajectory forecasting network.
- *CARLA-simulated (GT)* are the pedestrians simulated in CARLA, here considered ground truth. These pedestrians follow hand-designed trajectories.

S-GAN, S-STGCNN and ST-GAT are trained with default hyperparameters from the the official implementations. We compare SPL-goal with the following:

- *Goal direction (GD)* takes the shortest Euclidean path to the goal.
- *Collision avoidance with deep RL (CADRL)* [57] walks towards the goal location while avoiding moving objects around itself. CADRL is a learning based model for collision avoidance with dynamic obstacles.

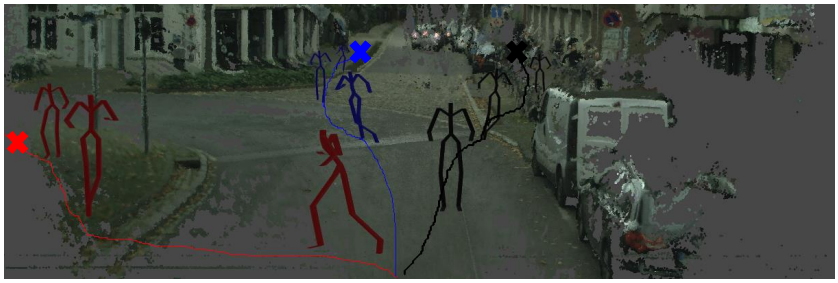

Fig. 5: Multiple SPL-goal agent trajectories generated from the same initial position in Cityscapes. The agent can be seen reaching different goals (marked by crosses). The agent chooses to walk on pavement when nearby.

3.1 Datasets

Simulated data from CARLA. The CARLA package [58] is a simulator for autonomous driving. RGB images, ground truth depth, 2d semantic segmentations and bounding boxes of pedestrians and cars are collected from the simulator at 17 fps. Town 1 is used to collect training and validation sets, with 37 and 13 scenes, respectively. The test set consists of 37 scenes from Town 2.

3d reconstructions from Cityscapes. This dataset [59] consists of on-board stereo videos captured in German cities. The videos are 30 frames long with a frame rate of 17 fps (video length: 1.76 seconds). We use GRFP [60] to estimate the semantic segmentation of all frames. The global reconstructions are computed by COLMAP [61] assuming a stereo rig with known camera parameters. The density of the dense reconstructions from COLMAP varies; an example reconstruction can be seen in Fig. 5. Cars and people are reconstructed frame-by-frame from PANnet [62] 2d bounding boxes and instance level segmentation masks. Triangulation is used to infer 3d positions from 2d bounding boxes. The dataset consists of 200 scenes; 100 for training, 50 for validation and 50 for testing.

LiDAR Waymo. The Waymo dataset [2] consist of 200 frame 10Hz LiDAR 3d scans, traffic agent trajectories and RGB images in 5 directions from the top of a data gathering car. We subsample a dataset of the 100 most pedestrian dense scenes in a 50 m radius to the collecting car. We use 70, 10 and 20 scenes for training, validation and testing, respectively. The images are segmented by [63] and the semantic labels are mapped to the 3d scans by the mapping between LiDAR and cameras provided by the Waymo dataset.

3.2 Training and Evaluation Details

In CARLA and Waymo the training sequence length is 30 timesteps, and in testing 300 timesteps (≈17s). The agents are trained for 20 epochs, 10 of which are STPN-pretraining without the HLN, and 10 of which are further refinements with the HLN attached (cf. §2.2 and §2.3). Agents tested on Cityscapes are first

Table 1: *Left:* Evaluation of pedestrian motion generation with 17s rollouts on the CARLA test set. The SPL (goal-free) agent is compared to the behaviour cloning (BC), constant velocity (CV) heuristics, as well as to to S-GAN [19], ST-GAT [22] and S-STG(CNN) [23]. The average of five different starting scenarios is shown (on pedestrian, random, close to a car, near a pedestrian, or on pavement). Our SPL agent collides with objects and people (f_o) and cars (f_v) less frequently than any other method, while travelling (d) only slightly shorter than ST-GAT. *Right:* Our SPL-goal agent outperforms or matches the goal direction (GD) heuristic and CADRL in success rate (f_s), while colliding much less (f_v, f_o).

	SPL	**BC**	**CV**	**S-GAN**	**ST-GAT**	**S-STG**
f_o	**0.02**	0.03	0.13	0.14	0.14	0.02
f_v	**0.07**	0.13	0.16	0.16	0.15	0.12
d	7.0	1.6	3.7	5.1	**7.9**	0.47

	f_o	f_v	f_s
SPL-goal	**0.09**	**0.01**	**0.78**
CADRL	0.24	0.08	0.75
GD	0.14	0.07	**0.78**

Table 2: *Left:* Average displacement error (m) for pedestrian forecasting on CARLA and Waymo. Our SPL agent receives the second lowest forecasting error on both datasets. The ST-GAT outperforms SPL on the CARLA dataset but yields the worst results on the Waymo dataset. On the Waymo dataset our SPL and BC models outperform the others with a large margin. *Right:* Our SPL agent avoids more collisions ($f_o + f_v$), walks further (d) and stays more on pavements (f_p) than ground truth simulated pedestrians (GT) on CARLA. The SPL agent is initialized on the same positions as the simulated pedestrians.

	SPL	**BC**	**S-GAN**	**ST-GAT**	**S-STG**
CARLA	0.11	0.22	0.16	**0.09**	0.12
WAYMO	0.06	**0.03**	0.11	0.13	0.11

	f_o	f_v	d	f_p
SPL	**0.00**	**0.0**	**17.0**	**0.46**
GT	0.08	**0.0**	16.0	0.35

trained on CARLA for 10 epochs and refined on Cityscapes for 22 epochs. Agents that are given a goal are trained with a sequence length of 10 timesteps for the first 5 epochs, after which the sequence length is increased to 30. The SPL-goal agents are refined from the weights of goal-free SPL agent that was trained on CARLA, with the addition of a feature indicating the direction to the goal. Each test scene is evaluated for 10 episodes with different spatial and agent history initializations to compute mean metrics.

3.3 Results

Evaluation metrics are adapted from the benchmark suite of CARLA and are:

- f_o, average frequency of collisions with static objects and pedestrians;
- f_v, average frequency of collisions with vehicles;
- d, average Euclidean distance (in meters) between agent's start and end location in episodes;
- f_p, average frequency of the agent being on pavements;

Table 3: *Left:* The SPL agent has learnt to avoid collisions with cars and pedestrians significantly better than BC, CV, S-GAN, ST-GAT and S-STG(CNN) on the Waymo data. *Right:* SPL-goal outperforms CADRL and GD on all metrics on Cityscapes. SPL-goal can reach goals while avoiding cars even in noisy scenes.

	SPL	**BC**	**CV**	**S-GAN**	**ST-GAT**	**S-STG**
f_o	**0.07**	0.16	0.22	0.60	0.71	0.15
f_v	**0.03**	0.06	0.10	0.26	0.12	0.07
d	1.4	**4.0**	1.2	2.9	2.5	0.34

	f_o	f_v	f_s
SPL-goal	**0.23**	**0.03**	**0.71**
CADRL	0.28	0.10	0.70
GD	0.28	0.09	0.70

– f_s, success rate in reaching a goal (only applicable for goal reaching agents).

CARLA. Table 1 (left) shows that our SPL agent generates long trajectories and yields significantly fewer collisions than the compared methods. The SPL average trajectory length of 7.0m is 11% less than the furthest travelling ST-GAT of 7.9m, but the SPL agent collides 53% less with vehicles and 86% less with objects and pedestrians. As shown in Table 2 (right), SPL even outperforms the CARLA-simulated (GT) trajectories in collision avoidance, and learns to stay on the sidewalk more, despite GT being the experts. To show the effect on the loss (1) of training on states outside of the expert trajectories, we compute the average negative log-likelihood loss (NLL) with respect to expert trajectories on the test set for the STPN module of SPL and of the BC baseline, obtaining losses of 0.009 and 0.013, respectively. The lower NLL of STPN indicates that training on states outside the expert trajectories provides more informative features and a model that acts more similar to ground truth data (i.e. expert trajectories). Finally, the SPL agent obtains the second lowest one-step trajectory forecasting error, or average displacement error (ADE), as seen in Table 2 (left).

To the right in Table 1 the SPL-goal agent is compared to CADRL and to the goal direction (GD) heuristic when given a goal at a distance of 6m. Our SPL-goal agent achieves a slightly higher success rate (f_s) than CADRL while being on par with GD. Moreover, SPL-goal is significantly better at avoiding collisions with cars, people and obstacles than the compared methods. In Fig. 3, the SPL-goal agent can be seen safely crossing streets to reach its goals.

Cityscapes. The 3d reconstructions of moving objects in the Cityscapes data can be noisy due to errors in depth estimation in frame-by-frame reconstruction, as well as noise in bounding boxes and semantic segmentation. Therefore the goal reaching task is harder in Cityscapes than in CARLA. Agents are initialized on pavement, near cars or randomly. The SPL-goal agent outperforms the GD heuristic and CADRL in collision avoidance as seen in table Table 3 (right). Sample trajectories of our agent can be seen in Fig. 5.

Waymo. In Table 3 (left), our SPL agent, BC, CV, S-GAN, ST-GAT and ST-GCNN are evaluated on 4 second trajectories. The SPL agent is significantly better at collision avoidance than any other model that is only trained on expert

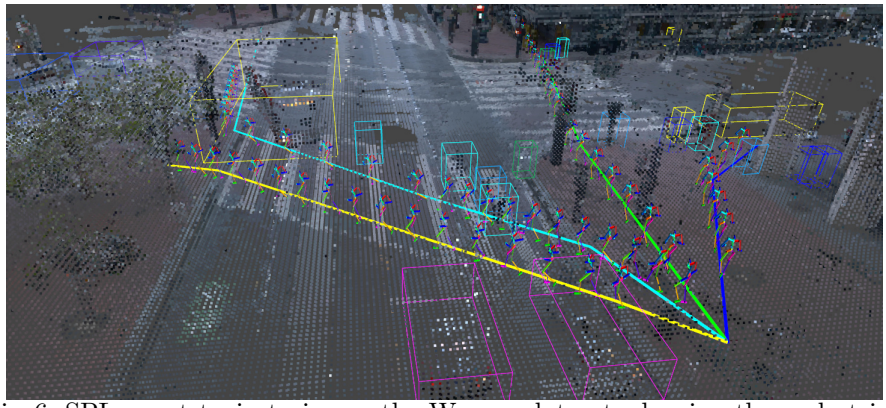

Fig. 6: SPL agent trajectories on the Waymo dataset, showing the pedestrian taking a number of different paths depending on how the agent history is initialized (cf. §2.2). Cars and other pedestrians are indicated with 3d bounding boxes.

pedestrian trajectories. It should be noted that the collision-aware SPL agent travels slower than BC to avoid collisions, which results in shorter trajectories on average. However SPL's trajectories are three times longer than S-STG(CNN) with half of the collisions. The SPL model has the second lowest ADE after BC (which shares SPL's architecture) on the Waymo dataset as seen in Table 2 (left). The SPL model is the only model to perform well on trajectory forecasting on both simulated and real data, while outperforming all models in collision avoidance. Qualitative examples of the SPL agent (without goals) are shown in Fig. 1, Fig. 6 and frame-by frame car avoidance in Fig. 4.

4 Conclusions

We have introduced a novel hierarchical 3d pedestrian locomotion generation model, based on explicit 3d semantic representations of the scene and 3d pedestrian locomotion model. By training the generator with a unified reward and likelihood maximization objective, the model learns to forecast well on both real and simulated data, while outperforming even expert trajectories in collision avoidance. More generally, our formulation can be used to adapt or refine any maximum likelihood-based trajectory forecasting method to simultaneously handle collision avoidance and forecasting. Our formulation also enables the use of articulated human models to enforce human dynamics on the trajectory forecasting model. Finally, the proposed pedestrian motion generator can also be refined to plausibly navigate among other pedestrians and traffic to specific goals. Future work includes studying finer grained agent-scene interactions, for example modelling traffic signs, crossroads, and other relevant objects in urban scenes.

Acknowledgments: This work was supported by the European Research Council Consolidator grant SEED, CNCS-UEFISCDI PN-III-P4-ID-PCE-2016-0535 and PCCF-2016-0180, the EU Horizon 2020 Grant DE-ENIGMA, and the Swedish Foundation for Strategic Research (SSF) Smart Systems Program.

References

1. Chang, M.F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D., et al.: Argoverse: 3d tracking and forecasting with rich maps. In: CVPR. (2019)
2. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: Scalability in perception for autonomous driving: Waymo open dataset (2019)
3. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR. (2020)
4. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: ICCV. (2019)
5. Huang, X., Cheng, X., Geng, Q., Cao, B., Zhou, D., Wang, P., Lin, Y., Yang, R.: The apolloscape dataset for autonomous driving. In: CVPR Workshops. (2018)
6. Kesten, R., Usman, M., Houston, J., Pandya, T., Nadhamuni, K., Ferreira, A., Yuan, M., Low, B., Jain, A., Ondruska, P., et al.: Lyft level 5 av dataset 2019. urlhttps. level5. lyft. com/dataset **2** (2019) 5
7. Mangalam, K., Adeli, E., Lee, K.H., Gaidon, A., Niebles, J.C.: Disentangling human dynamics for pedestrian locomotion forecasting with noisy supervision. In: The IEEE Winter Conference on Applications of Computer Vision. (2020) 2784–2793
8. Mínguez, R.Q., Alonso, I.P., Fernández-Llorca, D., Sotelo, M.Á.: Pedestrian path, pose, and intention prediction through gaussian process dynamical models and pedestrian activity recognition. IEEE Transactions on Intelligent Transportation Systems **20** (2018) 1803–1814
9. Rasouli, A., Kotseruba, I., Tsotsos, J.K.: Pedestrian action anticipation using contextual feature fusion in stacked rnns. arXiv preprint arXiv:2005.06582 (2020)
10. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In: ECCV. (2016)
11. Zanfir, M., Oneata, E., Popa, A.I., Zanfir, A., Sminchisescu, C.: Human synthesis and scene compositing. In: AAAI. (2020) 12749–12756
12. Wang, M., Yang, G.Y., Li, R., Liang, R.Z., Zhang, S.H., Hall, P.M., Hu, S.M.: Example-guided style-consistent image synthesis from semantic labeling. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
13. Cheng, S., Leng, Z., Cubuk, E.D., Zoph, B., Bai, C., Ngiam, J., Song, Y., Caine, B., Vasudevan, V., Li, C., et al.: Improving 3d object detection through progressive population based augmentation. arXiv preprint arXiv:2004.00831 (2020)
14. Ho, J., Ermon, S.: Generative adversarial imitation learning. In: NIPS. (2016)
15. Rhinehart, N., Kitani, K.M., Vernaza, P.: R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In: ECCV. (2018)
16. Li, Y.: Which way are you going? imitative decision learning for path forecasting in dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
17. van der Heiden, T., Nagaraja, N.S., Weiss, C., Gavves, E.: Safecritic: Collision-aware trajectory prediction. In: British Machine Vision Conference Workshop. (2019)

18. Zou, H., Su, H., Song, S., Zhu, J.: Understanding human behaviors in crowds by imitating the decision-making process. ArXiv **abs/1801.08391** (2018)
19. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: CVPR. (2018)
20. Kosaraju, V., Sadeghian, A., Martín-Martín, R., Reid, I., Rezatofighi, H., Savarese, S.: Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In: NeurIPS. (2019)
21. Zhang, L., She, Q., Guo, P.: Stochastic trajectory prediction with social graph network. CoRR **abs/1907.10233** (2019)
22. Huang, Y., Bi, H., Li, Z., Mao, T., Wang, Z.: Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In: The IEEE International Conference on Computer Vision (ICCV). (2019)
23. Mohamed, A., Qian, K., Elhoseiny, M., Claudel, C.: Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2020)
24. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Li, F., Savarese, S.: Social LSTM: human trajectory prediction in crowded spaces. In: CVPR. (2016)
25. Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H., Chandraker, M.: Desire: Distant future prediction in dynamic scenes with interacting agents. In: CVPR. (2017)
26. Luo, W., Yang, B., Urtasun, R.: Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In: CVPR. (2018)
27. Zhao, T., Xu, Y., Monfort, M., Choi, W., Baker, C., Zhao, Y., Wang, Y., Wu, Y.N.: Multi-agent tensor fusion for contextual trajectory prediction. In: CVPR. (2019)
28. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In: CVPR. (2019)
29. Malla, S., Dariush, B., Choi, C.: Titan: Future forecast using action priors. In: CVPR. (2020)
30. Tanke, J., Weber, A., Gall, J.: Human motion anticipation with symbolic label. CoRR **abs/1912.06079** (2019)
31. Liang, J., Jiang, L., Niebles, J.C., Hauptmann, A.G., Fei-Fei, L.: Peeking into the future: Predicting future person activities and locations in videos. In: CVPR. (2019)
32. Liang, J., Jiang, L., Murphy, K., Yu, T., Hauptmann, A.: The garden of forking paths: Towards multi-future trajectory prediction. In: CVPR. (2020)
33. Liang, J., Jiang, L., Hauptmann, A.: Simaug: Learning robust representations from 3d simulation for pedestrian trajectory prediction in unseen cameras. arXiv preprint arXiv:2004.02022 (2020)
34. Makansi, O., Cicek, O., Buchicchio, K., Brox, T.: Multimodal future localization and emergence prediction for objects in egocentric view with a reachability prior. In: CVPR. (2020)
35. Zhang, Y., Hassan, M., Neumann, H., Black, M.J., Tang, S.: Generating 3d people in scenes without people. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 6194–6204
36. Hong, S., Yan, X., Huang, T.S., Lee, H.: Learning hierarchical semantic image manipulation through structured representations. In: Advances in Neural Information Processing Systems. (2018) 2708–2718
37. Chien, J.T., Chou, C.J., Chen, D.J., Chen, H.T.: Detecting nonexistent pedestrians. In: CVPR. (2017)

38. Li, X., Liu, S., Kim, K., Wang, X., Yang, M.H., Kautz, J.: Putting humans in a scene: Learning affordance in 3d indoor environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 12368–12376
39. Lee, D., Pfister, T., Yang, M.H.: Inserting videos into videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 10061–10070
40. Wang, B., Adeli, E., Chiu, H.K., Huang, D.A., Niebles, J.C.: Imitation learning for human pose prediction. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019) 7123–7132
41. Wei, M., Miaomiao, L., Mathieu, S., Hongdong, L.: Learning trajectory dependencies for human motion prediction. In: ICCV. (2019)
42. Du, X., Vasudevan, R., Johnson-Roberson, M.: Bio-lstm: A biomechanically inspired recurrent neural network for 3-d pedestrian pose and gait prediction. IEEE Robotics and Automation Letters **4** (2019) 1501–1508
43. Cao, Z., Gao, H., Mangalam, K., Cai, Q., Vo, M., Malik, J.: Long-term human motion prediction with scene context. In: ECCV. (2020)
44. Adeli, V., Adeli, E., Reid, I., Niebles, J.C., Rezatofighi, H.: Socially and contextually aware human motion and pose forecasting. IEEE Robotics and Automation Letters **5** (2020) 6033–6040
45. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
46. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning (1992)
47. Hodgins, J.: Cmu graphics lab motion capture database (2015)
48. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE transactions on pattern analysis and machine intelligence **36** (2013) 1325–1339
49. Joo, H., Liu, H., Tan, L., Gui, L., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., Sheikh, Y.: Panoptic studio: A massively multiview system for social motion capture. In: ICCV. (2015)
50. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9** (1997) 1735–1780
51. Holden, D., Komura, T., Saito, J.: Phase-functioned neural networks for character control. ACM Trans. Graph. **36** (2017) 42:1–42:13
52. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289 (2015)
53. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: a system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2-4, 2016. (2016)
54. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR. (2015)
55. Schöller, C., Aravantinos, V., Lay, F., Knoll, A.: What the constant velocity model can teach us about pedestrian motion prediction. IEEE Robotics and Automation Letters **5** (2020) 1696–1703
56. Chandra, S., Bharti, A.K.: Speed distribution curves for pedestrians during walking and crossing. Procedia-Social and Behavioral Sciences **104** (2013) 660–667
57. Everett, M., Chen, Y.F., How, J.P.: Motion planning among dynamic, decision-making agents with deep reinforcement learning. In: IROS. (2018)

58. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: CoRL. (2017)
59. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR. (2016)
60. Nilsson, D., Sminchisescu, C.: Semantic video segmentation by gated recurrent flow propagation. In: CVPR. (2018)
61. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR. (2016)
62. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: CVPR. (2018)
63. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. IJCV (2018)