Supplementary Material of Generating Scenarios with Diverse Pedestrian Behaviors for Autonomous Vehicle Testing

Maria Priisalu[1], **Aleksis Pirinen**[*2], **Ciprian Paduraru**[3,4], **and Cristian Sminchisescu**[1,5]

[1]Lund University, [2]RISE Research Institutes of Sweden, [3]University of Bucharest, [4]Institute of Mathematics of the Romanian Academy, [5]Google Research

1 Additional visualizations: supplementary videos

In the attached video `Supplemenatry_Scenarios_with_Diverse_Pedestrian_Behaviors_for_AV_Testing.mp4` a number of sample trajectories are shown. The video contains sample trajectories where the pedestrian initial positions are sampled from the prior P, the model Pμ-D from Table 2 ($P\mu$-D is trained on the dense dataset D) and the model Simultaneous-μ, ρ from Table 3. The trajectories sampled from the prior P and from the model Pμ-D are shown with an untrained AV ρ, as this is what the model Pμ-D is trained on. The pedestrian initial position is sampled from the respective model and the pedestrian behaviour model described in §2.3 of the main paper is used to roll out the pedestrian trajectory. The visualizations show sample trajectories with a length of at most 100 steps (they are edited to stop when the pedestrian and AV collide). The trajectories are performed on a visualization scene that is gathered in the same fashion as the dense dataset, but is not a part of the dense dataset. The first frame is kept still for 6s to ease detecting the pedestrian's and AV's initial positions.

The video shows three sample trajectories where the pedestrian model π is initialized by sampling from the prior $x_0 \sim P$. It can be seen that even when the pedestrian model is initialized near the AV it seeks to reach a sidewalk, or walks along the middle of the road avoiding collisions to reach its goal. Note that goals are placed out as before: by reflecting the pedestrian's position x_0 in the constant velocity prediction of AV's trajectory. These sample trajectories visualize that it is not trivial where to place the pedestrian to ensure a collision. The model μ (see Fig. 1), has initially random weights. Therefore samples drawn from the initial $P\mu$ strongly resemble the samples of prior P.

The next three sample trajectories are from the model Pμ-D trained on the dense dataset. The samples show that the model has learnt to initialize the pedestrian model π such that the pedestrian misjudges the AV's motion and gets hit by the AV. Note that the AV could have 0 speed already at the first timestep/frame and thus avoid collisions by standing. The fourth sample trajectory from the Pμ-D model illustrates a failure case for the initialization model. In the fourth trajectory of the Pμ-D model it can be seen that if the AV and the pedestrian are initialized far away from one another then the initializer has little control over the pedestrian's trajectory and it is harder for the initializer to enforce a collision. Note the collision avoidance behaviour of the pedestrian model. In this trajectory the pedestrian curves around the AV to increase its distance to the AV.

Finally we see a sample failure case for the Simultaneous-μ, ρ model where both μ and ρ are trained simultaneously. The AV and the pedestian model are initialized close by, but both the the pedestrian and AV are good enough at collision avoidance to avoid a collision. This trajectory visualizes that problem of initializing a pedestrian such that it gets hit by a AV is not trivial. Even though the pedestrian is running towards the AV, the AV has learnt to accelerate to avoid collisions.

[*]Work partially done while at Lund University.

5th Conference on Robot Learning (CoRL 2021), London, UK.

2 The proposed μ model

In the following the details of the pedestrian initial distribution model μ are provided. Its objective is to model the distribution of initial positions x_0 of the pedestrian agent locations from where the pedestrian collides with the AV ρ. A sample trajectory of the Monte Carlo estimate of the gradient of J_π is evaluated by sampling the initial position of the pedestrian x_0 from μ, the pedestrian actions $a^\pi \sim \pi$ and the AV actions from $a^\rho \sim \rho$. The roll-outs are evaluated by a reward function r_μ that rewards collisions between the AV with position y_t and the pedestrian x_t, where $t \in [0, T]$ is the timestep. Since the ATS cannot control the actions of the behaviour policy π beyond the first timestep, the model μ does not receive a new state in response to the chosen action, only a reward r_μ. The pedestrian's initial position $x_0 \sim OP\mu(s^\mu)$ is considered to be the action taken by the policy gradient agent μ.

2.1 Model input s^μ

The pedestrian distribution model μ observes the scene as $s^\mu = (S, D, OP)$. Here S contains the top view RGB image and semantic labels of the scene (possibly constructed from a reconstruction). The same semantic labels are used as in the pedestrian model π. The dynamic mapping D contains the constant velocity predictions of the external cars, the AV and the external pedestrians. The dynamic map D is the reciprocal of the dynamic map used in the pedestrian behaviour model π, and contains a separate channel for cars and pedestrians. Finally μ observes the product of the occlusion map and the prior OP.

The proposed model μ takes as input a tensor s^μ of size (128×256×C) where $C = 17$ is the number of channels. The input channels contain the RGB channels of the top view of the scene at timepoint $t = 0$. The s^μ contains 9 channels for the semantic segmentation of the static objects in the scene. The s^μ contains two channels for the inverted dynamic occupancy map D and two channels containing the occupancy of the AV and external pedestrians and cars at timestep 0. Finally the occlusion-map masked prior OP is input as a separate channel to s^μ, to inform μ of which car to challenge.

2.1.1 The prior P

Given that the pedestrian has a maximal speed of $\|v^\pi_{max}\| = 3ms^{-1}$ there exists a cone of points h from which the pedestrian can reach the AV's constant velocity trajectory. The prior for the points in $x \in h$ is $\|x - y_0\|^{-1}$ where y_0 is the initial position of the AV. The prior $P(x)$ is 0 within the braking distance $\|v^\rho_0\|^2/(2g * 0.8)$ of the AV assuming dry road conditions (0.8 as friction coefficient), and v^ρ_0 is the AV's initial velocity. This is to avoid sampling from the trivial initializations within the AV's braking distance, thus leading to an inevitable collision. The points x that are on the constant velocity estimate of the AV's trajectory receive a 0 prior. Finally for all other points the prior is $||x - y_0||^{-2}$. The prior can be summarized as follows,

$$P(x) = \begin{cases} \|x - y_0\|^{-1} & \text{if } x \in h \\ 0 & \text{if } \|x - y_0\| < \frac{(v^\rho_0)^2}{250*0.8} \\ 0 & \text{if } x \text{ is on the line } y_0 + t * v^\rho_0, t > 0 \\ \|x - y_0\|^{-2} & \text{all other x,} \end{cases} \tag{1}$$

where x is a point in the scene, y_0 is the AV's initial location, h is the cone of points from which the pedestrian can reach the AV's constant velocity trajectory, v^ρ_0 is the AV's initial velocity, and t is time. The edges of the cone h are easily found by defining the AV's constant velocity v^ρ_0 (AV's initial velocity at timestep $t = -1$) future motion as a line $\hat{y_t} = y_0 + t * v^\rho_0$. The shortest distance from any point x in front of the AV to the AV's future trajectory $\hat{y_t}$ is the distance from the point x to the orthogonal projection $x_\perp$ of the point in the line $\hat{y_t}$. Let the constant velocity AV reach $x_\perp$ at timepoint $\hat{t}$. Then if $\|x - x_\perp\| < \hat{t}\|v^\pi_{max}\|$, where $\|v^\pi_{max}\| = 3ms^{-1}$ is the maximal speed of the pedestrian, then the point $x \in h$.

2.1.2 The semantic segmentation of static objects in S

The RGB and semantic top view of the scene's static objects is referred to as S. The construction of the semantic map and the top view RGB of S follow the procedure of [1]. The semantic labels used are building, fence, static obstacles, pole, road, sidewalk, vegetation, wall and traffic sign/light. The

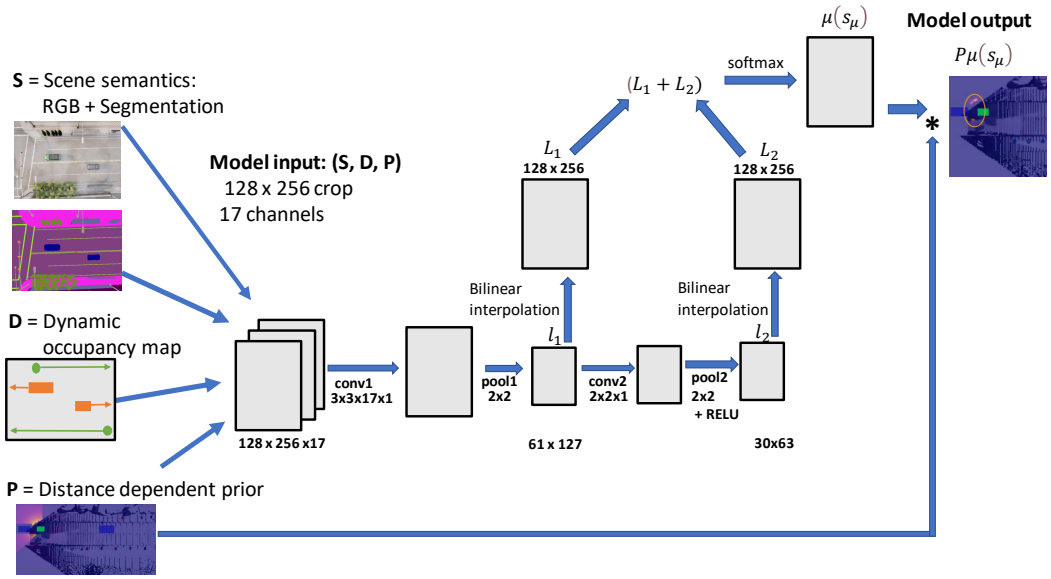

Figure 1: Pedestrian initial spatial distribution $P\mu$ architecture. The model input consists of channels for the top view scene semantic and RGB S, the dynamic occupancy map D, and the prior P that may be replaced by PO to enforce initialization in occluded spaces only. Note that the dynamic occupancy map D of μ and the dynamic occupancy map D_t of π are different. The neural network output is multiplied by the prior P to produce the pedestrian initial spatial distribution.

semantic segmentation is obtained by gathering 2D semantic labeled images from the data gathering stationary AV's perspective together with the depth map of the scene in the same perspective. The 3D points of each pixel can then be reconstructed from the depth map, and a segmentation label and a RGB color can be assigned to each 3D point. This is done for very 50th gathered frame (500 frames in total). Finally mode-voting is applied to obtain the semantic class of a 3D point, and mean to attain the color. The dense dataset is gathered from a moving drone's perspective. Finally the 3D reconstruction is voxellized and projected into the top-view perspective.

2.1.3 The dynamic occupancy map D

The dynamic occupancy map contains the constant velocity estimates of the cars and external pedestrians in separate channels. This provides the model with the most basic car and pedestrian motion estimates. Given that a car's constant velocity estimate at timestep t is the bounding box b_t, the pixels in the bounding box b_t are set to $D(b_t) = 1/t$ if $D(b_t) < 1/t$. That is the earliest occupancy is noted in the inverted dynamic map. The earlier steps in the forecast trajectories are more relevant to μ as they are temporally closer to the pedestrian initialization x_0.

Note that in the pedestrian behaviour model D_t refers to a dynamic occupancy map that is not inverted and that is dependent on the timestep t. D_t contains the AVs, external pedestrians and cars occupancy trajectories up to timestep t, and the constant velocity future trajectory estimates of all pedestrians and cars from timestep t onwards. The map D_t is constructed in the same fashion as D, but D_t also includes occupancy for previous timesteps. Further D_t uses scaled (by constant 0.003 to ensure values lie in range [0,1] for sequence lengths up to 300 timesteps) but not inverted timestamps. The pedestrian observes a local neighborhood of this dynamic map denoted $D_t(x_t)$ which is bigger than the pedestrian.

2.2 Model architecture

The model architecture is visualized in Fig. 1. The proposed μ model has an input of size $(128\times256\times C)$. This is passed through a $(3\times3\times\text{C}\times1)$ convolution followed by a $(2\times2\times1)$ max-pooling layer. Let the output of this max-pooling layer be referred to as l_1. l_1 is convoluted by a $(2\times2\times1\times1)$ filter, $(2\times2\times1)$ max-pooling and passed through ReLu activation. Let the output of the Relu activation be called l_2.

Now l_1 and l_2 are bi-linearly interpolated back to the original image resolution of (128×256), let the upsampled layers be denoted by L_1, L_2. The neural network output is then $\mu(s^\mu) = \text{softmax}(L_1 + L_2)$, where $softmax$ is the softmax all of the pixels of the image to ensure that the model output is a distribution. Finally the $\mu(s^\mu)$ is multiplied by the prior P to get the estimated pedestrian initial spatial location distribution $P\mu(s^\mu)$.

3 Reward functions

Here we will provide the details of the reward functions of μ, π, ρ. A number of the reward components are shared between the model components.

3.1 Reward of μ

A number of the reward components are adapted from π. The reward function r_μ consist of collision terms R_v, R_p, R_s that are indicator functions which are activated when the controlled pedestrian's intended next step position $x_t + a_t^\pi$ collides with external vehicles, external pedestrians and static objects respectively. We introduce a new reward term R_a an indicator function that is positive when $x_t + a_t^\pi$ and the AV collide. The reward terms R_v, R_p, R_s are multiplied with negative factors $\lambda_v = -2, \lambda_p = -0.1, \lambda_s = -0.02$ to discourage collisions with external agents and objects, while R_a is multiplied with a positive factor $\lambda_a = 2$.

Further μ is rewarded for initializations that lead to pedestrian motion in areas often traversed by pedestrians. This is done by the reward components R_d and R_k. To allow for per frame updates of the dataset we adapt R_d to be an indicator function which is 1 when $x_t + a_t^\pi$ is intercepting a past external pedestrian trajectory or a constant-velocity predicted pedestrian trajectory (i.e. interception with a non-zero D_t). In a similar fashion we define R_k to reward the pedestrian for being near external pedestrian trajectories. R_k is evaluated as the ratio of non-zero pixels in D_t in the neighborhood $D_t(x_t)$ of the pedestrian x_t. The terms R_d and R_k are multiplied by $\lambda_d = 0.01$ and $\lambda_k = 0.01$.

Further a positive reward is given to μ for steps taken by π towards the goal g^π. Let the indicator function $I_g(s_t, a_t, g^\pi)$ be positive when $x_t + a_t^\pi$ has reached the goal, i.e. $\|x_t + a_t^\pi - g^\pi\| < \epsilon$, where $\epsilon = \sqrt{2}$ pixels. Then $R_g(s_t, a_t, g^\pi) = 1 - \frac{\|x_t + a_t^\pi - g^\pi\|}{\|x_t - g^\pi\|}$ when the agent has not reached the goal location $I_g(s_t, a_t, g^\pi) < 1$. The goal term R_g is multiplied by $\lambda_g = 0.1$.

The reward components R_p, R_s, R_k, R_d, R_g are multiplied together when non-zero. When the pedestrian collides with a vehicle, the AV or reaches a goal location then r_μ is equal to only the reward term R_v, R_a or R_g, and the model receives a 0 reward in the following time-steps. As shown below and in the main paper the initial distribution model's reward is dependent on the full trajectory of the pedestrian and the AV model, and is thus expressed as

$$R_\mu(\tau) = \sum_{t=0}^{T} \gamma^t r_\mu(s_t, a_t, s_{t+1}). \tag{2}$$

Further we can summarize r_μ, as

$$r_\mu(s_t, a_t, s_{t+1}) = \begin{cases} \lambda_a & \text{if } R_a(s_t, a_t, s_{t+1}) \\ \lambda_p & \text{if } R_v(s_t, a_t, s_{t+1}) \\ 0 & \text{if } R_v(s_k, a_k, s_{k+1}) > 0, \text{ for any } k < t \\ 0 & \text{if } R_a(s_k, a_k, s_{k+1}) > 0, \text{ for any } k < t \\ 0 & \text{if } I_g(s_k, a_k, s_{k+1}) > 0, \text{ for any } k < t \\ \Pi(\lambda_p R_p, \lambda_s R_s, \lambda_k R_k, \lambda_d R_d, \lambda_g R_g) & \text{otherwise} \end{cases}, \tag{3}$$

where $\Pi(.)$ is the product of the non-zero inputs, and where in (3) the general lambda followed by the general reward $\lambda_* R_*$ denotes the following function,

$$\lambda_* R_* = \begin{cases} 1 + \lambda_* R_*(s_t, a_t, s_{t+1}) & \text{if } \lambda_* > 0 \\ 1 - \lambda_* R_*(s_t, a_t, s_{t+1}) & \text{if } \lambda_* < 0 \\ 1 & \text{otherwise .} \end{cases} \tag{4}$$

3.2 Reward of π

The reward function of π consist of pedestrian motion encouraging terms R_{ped}, collision penalizing terms R_{coll}, the terms R_g and I_g encouraging movement towards the goal g^π, and a term discouraging unnatural articulated motion R_ϕ. The collision discouraging terms in $R_{coll}(s_t, a_t, s_{t+1}) = \lambda_p R_p(s_t, a_t, s_{t+1}) + \lambda_s R_s((s_t, a_t, s_{t+1}) + \lambda_a^\pi R_a(s_t, a_t, s_{t+1})$, where R_v, R_p, R_s, R_a are described in §3.1, and $\lambda_a^\pi = -\lambda_a$. The reward term R_ϕ penalizes lathe changes in the average yaw ϕ (in degrees) of the joints in the agent's lower body as $R_\phi(x_t, v_t) = \max(\min(\phi - 1.2, 0), 2.0)$. The term R_g follows the definition given in §3.1. The pedestrian agent also receives a large positive reward $\lambda_G = 2$ for reaching its goal location i.e. $I_g(s_t, a_t, s_{t+1}) > 0$. After collision with a car and after reaching a goal the pedestrian agent is considered dead, and thus receives 0 rewards.

The pedestrian motion encouraging term R_{ped} consists of a reward term that promotes motion on pedestrian trajectories R_d^π. The reward R_d^π is an indicator function which is 1 if any pixel in the pedestrian bounding box coincides with D_T. The pedestrian bounding box is centered at the pedestrian's position x_t and is of size 1.2m×1.2m. The second reward term in R_{ped} is R_k^π that rewards the pedestrian for being near external pedestrian trajectories. The pedestrian trajectories in D_T are blurred by an exponential kernel producing a density map D_k. The pedestrian is rewarded $R_k^\pi(s_t, a_t, s_{t+1}) = D_k(x_{t+1})$ is equal to the kernel smoothed valued of the kernel at the pixel x_{t+1}. The pedestrian like motion promoting terms can be gathered as, $R_{ped}(s_t, a_t, s_{t+1}) = \lambda_k R_k^\pi(s_t, a_t, s_{t+1}) + \lambda_d R_d^\pi(s_t, a_t, s_{t+1})$. A small negative reward $R_\phi(s_t, a_t, s_{t+1}) = \max(\min(\phi - 1.2, 0), 2.0)$ is given for excessively large changes in the average lower body joints yaw ϕ of the pedestrian agent's articulated pose. The term R_ϕ is multiplied by $\lambda_\phi = -0.0001$.

Finally the pedestrian reward can be summarized as

$$r^\pi(s_t, a_t, s_{t+1}) = \begin{cases} 0 & \text{if agent is dead} \\ \lambda_v & \text{if } R_v(s_t, a_t, s_{t+1}) > 0\ , \\ R_{coll} + R_{ped} + \lambda_g R_g + \lambda_G I_g + \lambda_\phi R_\phi & \text{otherwise} \end{cases} \tag{5}$$

where the reward terms $R_{coll}, R_{ped} R_g, I_g, R_\phi$ are evaluated on (s_t, a_t, s_{t+1}) when input parameters are omitted.

3.3 Reward of ρ

The reward function of the AV ρ contains the collision penalizing terms $R_v^\rho, R_p^\rho, R_s^\rho$ that are analogous to the collision penalizing terms R_v, R_p, R_s. The reward terms $R_v^\rho, R_p^\rho, R_s^\rho$ are indicator functions that are 1 if the learnt AV's planned position $y_t + a_t^\rho$ collides with an external car, any pedestrian or static object in the timestep $t + 1$. The terms can again be gathered $R_{coll.}^\rho(s_t, a_t, s_{t+1}) = \lambda_v^\rho R_v^\rho(s_t, a_t, s_{t+1}) + \lambda_p^\rho R_p^\rho(s_t, a_t, s_{t+1}) + \lambda_s^\rho R_s^\rho(s_t, a_t, s_{t+1})$, where $\lambda_v^\rho = -2, \lambda_p^\rho = -2, \lambda_s^\rho = -2$. The reward function r_ρ contains also the term $R_o(y_t, a_t^\rho)$ that is the ratio of pixels in the AV's bounding box that have the semantic label sidewalk, multiplied by $\lambda_o = -0.1$. Finally we introduce a term to encourage the AV to move. The reward term $R_{dist}(s_t, a_t, s_{t+1}) = \|y_0 - y_t\|/t * v_{max}^\rho$, where v_{max}^ρ =70km/h is the maximal speed of the AV. The reward term R_{dist} is multiplied by $\lambda_{dist} = 0.01$. The AV is considered dead after any collision (i.e. when $\max(R_v^\rho, R_p^\rho, R_s^\rho) > 0$), and thus a reward of 0 is given after any collision. The full reward of the alive AV is,

$$r_\rho(s_t, a_t, s_{t+1}) = R_{coll.}^\rho(s_t, a_t, s_{t+1}) + \lambda_o R_o(y_t, a_t^\rho) + \lambda_{dist} R_{dist}(s_t, a_t, s_{t+1}). \tag{6}$$

4 Simultaneous learning of μ, ρ and π

To gradually improve the AV at collision avoidance the ATS can be used to train the AV. Once the AV improves the ATS can be fitted to the new AV model. This gives rise to the possibility of training the AV and the ATS alternatively or even simultaneously. Here we present algorithms for alternative and simultaneous training of the AV and the ATS model. If the pedestrian model is not dependent on an external dataset then even the pedestrian model π can be trained simultaneously with ATS and AV. In the following experiments in §5.3 we utilize the SPL-goal agent from [1] as the pedestrian model π.

4.1 Policy Gradient Framework for the Problem Described in Section 3.1

We would like to find the parametrized Θ pedestrian initial location distribution μ_Θ that maximizes the objective

$$J_\mu(\Theta) = \mathbb{E}_{x_0\sim\mu_\Theta(.|s^\mu),s^\mu\sim q,a_t^\pi\sim\pi,a_t^\rho\sim\rho,s_t\sim p(.|s_t,a_t)}\left[R_\mu(x_0,\tau)\right], \tag{7}$$

where R_μ is the discounted cumulative reward $R_\mu(x_0,\tau) = \sum_{t=0}^{T-1}\gamma^t r_\mu(s_t,a_t,s_{t+1})$, and τ trajectory of an episode be denoted $\tau = (a_0, s_1, ..., a_{T-1}, s_T)$, and x_0 is the initial pedestrian position location, π and ρ are the behaviour model's of the pedestrian and the AV agent respectively, taking actions a_t^π and a_t^ρ. And $p(s_{t+1}|s_t,a_t)$ is the environment dynamics that predicts the successive state s_{t+1}, where $s_t = (s_t^\pi, s_t^\rho)$ and $a_t = (a_t^\pi, a_t^\rho)$ are vectors containing the states and actions of the pedestrian model π and AV ρ respectively. The traffic scene observations s^μ have a distribution $q(s^\mu)$. Further r_μ is μ's reward function, and t is the current timestep and T is the episode length. Finally s_0 is a function of s^μ, x_0 (see Section 3.1 in main paper) and $x_0 \sim \mu_\Theta(x_0|s^\mu)$. Let the discounted cumulative reward $R_\mu = \sum_{t=0}^{T-1}\gamma^t r_\mu(s_t,a_t,s_{t+1})$ we can express (7) as

$$J_\mu(\Theta) = \int_{s^\mu}\int_{x_0}\int_\tau R_\mu(x_0,\tau)p_\tau(\tau|x_0)\mu_\Theta(x_0|s^\mu)q(s^\mu)d\tau dx_0 ds^\mu, \tag{8}$$

where p_τ is the probability density function of τ given x_0. Then p_τ can be factored as follows,

$$p_\tau(\tau|x_0) = \prod_{t=0}^{T-1}\pi(a_t^\pi|s_t^\pi)\rho(a_t^\rho|s_t^\rho)p(s_{t+1}|s_t,a_t). \tag{9}$$

Now taking a derivative of (9) with respect to the parameters θ we note that only μ depends on θ,

$$\nabla_\Theta J_\mu(\Theta) = \int_{s^\mu}\int_{x_0}\int_\tau \nabla_\Theta\mu_\Theta(x_0|s^\mu)R_\mu(x_0,\tau)p_\tau(\tau|x_0)q(s^\mu)d\tau dx_0 ds^\mu. \tag{10}$$

We can now follow the classical policy gradient method [2] and use $\nabla_\Theta\mu_\Theta = \mu_\Theta\nabla_\Theta\log(\mu_\Theta)$, and rewrite (10) as,

$$\nabla_\Theta J_\mu(\Theta) = \int_{s^\mu}\int_{x_0}\int_\tau \nabla_\Theta\log(\mu_\Theta(x_0|s^\mu))R_\mu(x_0,\tau)p_\tau(\tau|x_0)\mu_\Theta(x_0|s^\mu)q(s^\mu)d\tau dx_0 ds^\mu \tag{11}$$

$$= \mathbb{E}[\log(\mu_\Theta(x_0|s^\mu))R_\mu(x_0,\tau)]. \tag{12}$$

We can evaluate the above expectation with the Markov Chain Monte Carlo method. Given K traffic scenes s_k^μ, with M pedestrian initial locations $x_0^{m,k} \sim \mu(x_0|s_k^\mu)$ in each traffic scene, and N sample trajectories $\tau^{m,k,n} \sim p_\tau(\tau|x_0^{m,k})$ for each pedestrian initialization $x_0^{m,k}$ then the Monte Carlo estimate of 11, is the following

$$\hat{\nabla}_\Theta J_\mu(\Theta) = \frac{1}{NMK}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N} R_\mu(x_0^{m,k},\tau^{m,k,n})\nabla_\Theta\log(\mu_\Theta(x_0^{m,k}|s_k^\mu)). \tag{13}$$

In the same manner using (9) we can estimate the gradient of π_β with,

$$\hat{\nabla}_\beta J_\pi(\beta) = \frac{1}{NMK}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{t=0}^{T-1}\gamma^t r_\pi(s_t^{m,k,n},a_t^{m,k,n},s_{t+1}^{m,k,n})\nabla_\beta\log(\pi_\beta(a_t^{\pi,m,k,n}|s_t^{\pi,m,k,n})). \tag{14}$$

And similarly the AV policy's ρ_ξ gradient can be estimated by,

$$\hat{\nabla}_\xi J_\rho(\xi) = \frac{1}{NMK}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{t=0}^{T-1}\gamma^t r_\rho(s_t^{m,k,n},a_t^{m,k,n},s_{t+1}^{m,k,n})\nabla_\xi\log(\rho_\xi(a_t^{\rho,m,k,n}|s_t^{\rho,m,k,n})). \tag{15}$$

Finally the three gradient estimates (13),(14),(15) can be estimated from the same sample trajectories, giving rise to Algorithm 1. Alternatively a possibly more stable alternating training scheme could be used as shown in Algorithm 2

Algorithm 1 Learning μ, π and ρ simultaneously

Initialize Θ_1, β_1, ξ_1 randomly.
Initialize learning rates $\alpha_\Theta, \alpha_\beta, \alpha_\xi$
Set $\mu_1 = \mu(\Theta_1)$, $\pi_1 = \pi(\beta_1)$, $\rho_1 = \rho(\xi_1)$
for $k = 1 \dots K$ iterations **do**
  Initialize empty set $O = \{\}$
  Sample s_k^μ from the dataset
  for $m = 1 \dots M$ iterations **do**
    Sample $x_0^{k,m} \sim \mu_k(x_0|s_k^\mu)$
    for $n = 1 \dots N$ iterations **do**
      Sample $\tau^{k,m,n} \sim p_\tau(.|x_0^{k,m})$ where $a_t^{\pi,k,m,n} \sim \pi_k$ and $a_t^{\rho,k,m,n} \sim \rho_k$
      Add sample trajectory $\tau^{k,m,n}$ to O
    end for
  end for
  Update $\Theta_{k+1} = \Theta_k + \alpha_\Theta \hat{\nabla}_\Theta J_\mu(\Theta)$ using samples in O
  Update $\beta_{k+1} = \beta_k + \alpha_\beta \hat{\nabla}_\beta J_\pi(\beta)$ using samples in O
  Update $\xi_{k+1} = \xi_k + \alpha_\xi \hat{\nabla}_\xi J_\rho(\xi)$ using samples in O
  Update parameters $\mu_{k+1} = \mu(\Theta_{k+1})$, $\pi_{k+1} = \pi(\beta_{k+1})$, $\rho_{k+1} = \rho(\xi_{k+1})$
end for

Table 1: The number of epochs the μ of the presented models were trained for

Model	$P\mu$	$OP\mu$	$P\mu$-D	SPL+ϵ	SPL A.	STPN A	CV
	6	9	2	8	9	2	10

5 Experiments

5.1 Hyperparameters in experiments

The Adam optimizer with a learning rate of $\alpha_\mu = 5 \times 10^{-3}$ for μ and $\alpha_\rho = 3 \times 10^{-2}$ for ρ is used in experiments. The weights of μ are initialized randomly, and the weights of ρ are initialized to 1. A discount rate of $\gamma = 0.99$ is used for all of the models. The presented values are of the pedestrian location distribution models μ showed highest validation performance, seen in Table 1.

5.2 Experiments on the Dense CARLA dataset (D)

We introduce the Dense CARLA dataset (D) a smaller more object-dense dataset consisting of 4 different simulations of 5 scenes, gathered from a drone's perspective. The model $P\mu$-D is trained on the dataset and tested on the regular CARLA dataset. The model $P\mu$-D is trained with the reward R_{STPN}, for a fair comparison the model is compared to $P\mu$ trained on the regular dataset with the reward R_{STPN} for 2 epochs. The results are shown in Table 2. The proposed μ is robust to changes in dynamics from training to testing as seen in the small drop in the number of collisions when comparing $P\mu$-D and $OP - \mu$ in Table 2. The model $P\mu$-D trained on the denser dataset leads to fewer collisions than the base model $P\mu$ and has a higher π-entropy than $P\mu$, likely because the π is less controllable by μ in denser traffic. The distribution of $P\mu$-D is visualized in Fig.3 of the main paper on a dense dataset scene.

5.3 Alternative and Simultaneous training

The AV and the initial pedestrian model are trained simultaneously and alternatively where in the latter case the AV is trained for two epochs for every epoch of training μ. Both models are trained for a total of 14 epochs with the reward $r_\mu = R_{STPN}$. We also report the Avg distance- the average distance travelled by the AV. The *Simultaneous-*μ, ρ has collision rate that is not statistically not different from $OP\mu$ in Table 2. But *Alternative-*μ, ρ has almost twice as many collisions as *Simultaneous-*μ, ρ. Further the alternative *Alternative-*ρ has a similar collision rate when tested with $OP\mu$. This suggests that the AV model learnt by alternative training is poor at collision avoidance. The AV model

Algorithm 2 Learning μ, π and ρ alternatively

Initialize Θ_1, β_1, ξ_1 randomly.
Initialize learning rates $\alpha_\Theta, \alpha_\beta, \alpha_\xi$
Set $\mu_1 = \mu(\Theta_1)$, $\pi_1 = \pi(\beta_1)$, $\rho_1 = \rho(\xi_1)$
for $j = 1 \dots J$ iterations **do**
  Set $\mu_{j,1} = \mu_j$, $\pi_{j,1} = \pi_j$, $\rho_{j,1} = \rho_j$
  for $k = 1 \dots K_\mu$ iterations **do**
    Initialize empty set $O = \{\}$
    Sample $s^\mu_{j,k}$ from the dataset
    for $m = 1 \dots M$ iterations **do**
      Sample $x_0^{j,k,m} \sim \mu_{j,k}(x_0|s^\mu_{j,k})$
      Sample N trajectories $\tau^{j,k,m,n} \sim p_\tau(.|x_0^{j,k,m})$ s.t. $a_t^{\pi,j,k,m,n} \sim \pi_j$, $a_t^{\rho,j,k,m,n} \sim \rho_j$, and add to O
    end for
    Update $\Theta_{j,k+1} = \Theta_{j,k} + \alpha_\Theta \hat{\nabla}_\Theta J_\mu(\Theta)$ using samples in O
    Update $\mu_{j,k+1} = \mu(\Theta_{j,k+1})$
  end for
  Set $\mu_{j+1} = \mu_{j,K_\mu}$
  for $k = 1 \dots K_\pi$ iterations **do**
    Initialize empty set $O = \{\}$
    Sample $s^\mu_{j,k}$ from the dataset
    for $m = 1 \dots M$ iterations **do**
      Sample $x_0^{j,k,m} \sim \mu_{j+1}(x_0|s^\mu_{j,k})$
      Sample N trajectories $\tau^{j,k,m,n} \sim p_\tau(.|x_0^{j,k,m})$ s.t. $a_t^{\pi,j,k,m,n} \sim \pi_{j,k}$, $a_t^{\rho,j,k,m,n} \sim \rho_j$, and add to O
    end for
    Update $\beta_{j,k+1} = \beta_{j,k} + \alpha_\beta \hat{\nabla}_\beta J_\pi(\beta)$ using samples in O
    Update $\pi_{j,k+1} = \pi(\beta_{j,k+1})$
  end for
  Set $\pi_{j+1} = \pi_{j,K_\pi}$
  for $k = 1 \dots K_\rho$ iterations **do**
    Initialize empty set $O = \{\}$
    Sample $s^\mu_{j,k}$ from the dataset
    for $m = 1 \dots M$ iterations **do**
      Sample $x_0^{j,k,m} \sim \mu_{j+1}(x_0|s^\mu_{j,k})$
      Sample N trajectories $\tau^{j,k,m,n} \sim p_\tau(.|x_0^{j,k,m})$ s.t. $a_t^{\pi,j,k,m,n} \sim \pi_{j+1}$, $a_t^{\rho,j,k,m,n} \sim \rho_{j,k}$, and add to O
    end for
    Update $\xi_{j,k+1} = \xi_{j,k} + \alpha_\xi \hat{\nabla}_\xi J_\rho(\xi)$ using samples in O
    Update $\rho_{j,k+1} = \rho(\xi_{j,k+1})$
  end for
  Set $\rho_{j+1} = \rho_{j,K_\rho}$
end for

Table 2: An ablation studying the effect of the prior during the training of μ shows that the μ is robust to changes in the prior during training as $OP - \mu$ and $P - \mu$ trained with the priors OP and P respectively, have indistinguishable collision rates (stdev is 0.02).

	$OP - \mu$	$P\mu$-D
#. collisions	**0.22**	0.19
Avg distance	7.8	7.7
π-entropy	0.23	0.29

Table 3: Training the π and μ simultaneously *Simultaneous* results in metrics similar to those of separately trained models. This is confirmed by testing the *Alternative-*μ,*Simultaneous-*μ against the *baseline AV*, and the *Alternative-*ρ,*Simultaneous-*ρ against *OP*μ

	Alternative		
	μ, ρ	μ	ρ
#. collisions	0.41($\pm$0.03)	0.22($\pm$0.02)	0.42($\pm$0.02)
Avg distance	5.4($\pm$0.1)	7.8 ($\pm$0.5)	5.3($\pm$0.2)
π-entropy	0.18($\pm$0.01)	0.23($\pm$0.01)	0.16($\pm$0.01)
	Simultaneous		
	μ, ρ	μ	ρ
#. collisions	0.21($\pm$0.02)	0.20($\pm$0.01)	0.25($\pm$0.02)
Avg distance	7.7($\pm$0.1)	7.9 ($\pm$0.6)	7.5($\pm$0.5)
π-entropy	0.23($\pm$0.01)	0.21($\pm$0.01)	0.16($\pm$0.01)

Alternative-ρ travels 2 meters less than the other models in Table 3. Altogether this suggest that the AV model does not benefit from alternative training with μ. Interestingly the *Alternative-*μ has the same collision rate as $OP\mu$, showing that the ATS model can improve even if the AV model is lacking behind. The simultaneously trained μ and ρ are comparable in collisions and entropy to the $OP\mu$ and the *baseline AV* model. Of higher interest is the low entropy that *Alternative-*ρ and *Simultaneous-*ρ have when tested with $OP\mu$. This could imply that even the AV can learn to place itself such that the pedestrian agent acts as predictably as possible. Further $OP\mu$ receives higher collision rates with *Alternative-*ρ and *Simultaneous-*ρ than *Alternative-*μ and *Simultaneous-*μ respectively. We hypothesize that this could be because in simultaneous or alternative training it is harder to balance hyperparameters to both μ and ρ.

6 Extending μ to model the pedestrian goal distribution

In the main paper we present the model μ that models the pedestrian initial distribution. Here we present an extension of the model that also allows for learning the pedestrian behaviour model's goal location g^π distribution μ_g. Since μ_g is optimized with the same reward as μ, this should give the pedestrian initial distribution model more opportunities to enforce collisions. The proposed method is to use the same model architecture for μ_g as for μ, but to use a different prior P_g. It should be noted that in initial experiments we tried to model μ and μ_g in the same network, by simply extending μ with an additional fully connected layer outputting the goal distribution. It was quickly noticed that when π's initial position and goal location were undecided the model struggled to learn as most sampled trajectories resulted in no collisions, even when μ was pretrained. It is clear that μ_g should be conditioned on the sample x_0 rather than on the distribution $P\mu$, as the best goal location depends for example on which side of the AV the pedestrian is initialized. To do so the goal prior P_g is conditioned on x_0, y_0, v_0^ρ.

The goal prior $P_g(x|x_0, y_0, v_0^\rho)$ is found by solving a linear system of inequalities. The goal prior is found by solving for the points $x \in g$ for which the constant velocity prediction of the AV and the pedestrian collide within the given time and speed constraints. More specifically, the points $x \in G$

solve for pedestrian speeds s and collision times t the following system of linear inequalities,

$$\begin{cases} |y_0 + v_0^\rho t - x_0 + (x - x_0)st| \leq s_x + s_y \\ 0 \leq t \leq T \\ 0 \leq s \leq \|v_{max}^\pi\|, \end{cases} \tag{16}$$

where s_x, s_y are the side lengths of the bounding boxes of the pedestrian and AV. For points $x \in G$ a collision is possible (i.e. there exists s, t that fulfill (16)), the goal prior is assigned the reciprocate distance travelled by the pedestrian to the collision. Finally the goal prior is normalized. If ρ has an initial speed of 0 then $P_g = P$. To summarize the goal prior is given by

$$P_g(x|x_0, y_0, v_0^\rho) = \begin{cases} P(x) & \text{if } \|v_0^\rho\| = 0 \\ \frac{1}{s_{max}(x)t_{min}(x)} & \text{for } x \in G \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

where $s_{max}(x)$ is the maximal speed s for the point x that fulfills (17) and $t_{min}(x)$ is the minimal time for collision that fulfills (17) for x. When trained on two scenes, and validated on two scenes the presented goal mode μ_g increased the frequency of collisions from 0.1 to 0.7 with 100 epochs on the validation set. Note in this small experiment the SPL-goal model was used without the Human Locomotion Network (i.e. the STPN-goal model), and μ and μ_g were trained with the reward R_{STPN}.

7 List of Mathematical Notations

- α_β- learning rate of the parameters β.
- α_μ- learning rate of μ.
- α_ρ- learning rate of ρ.
- α_Θ- learning rate of the parameters Θ.
- α_ξ- learning rate of the parameters ξ.
- a_t- a vector of actions taken by μ and ρ at timestep t.
- a_t^π- the action taken by the pedestrian at timestep t.
- a_t^ρ- the action taken by the AV at timestep t.
- B^μ- behavioural constraints for pedestrian inital spatial distribution.
- B^π- behavioural constraints for the pedestrian's policy.
- B^ρ- behavioural constraints for the AV's policy.
- β- parameters of the parametric policy $\pi(\beta)$.
- b- bias in the AV model.
- b_t- a car's bounding box at timestep t.
- C- number of channels in μ_Θ.
- c_t- AV's speed sampled from a neural network.
- D- reciprocal temporal mapping of dynamic objects in s^μ.
- D_k- heatmap of pedestrians. The exponential kernel blurred heatmap of pedestrians in D_T.
- D_t- temporal mapping of dynamic objects in s_t^π.
- d_t- AV's distance to the closest external car.
- d_t^x- pedestrian agent's distance to the closest external car .
- δ_t- AV's intersection with the sidewalk.
- ϵ- maximal distance to the goal, to attain the goal-reaching related reward.
- f_μ- additional terms of the loss J_μ that are not related to the number of collisions.
- f_π- additional terms of the loss J_π that are not related to the number of collisions.
- f_ρ- additional terms of the loss J_ρ that are not related to the number of collisions.
- γ- reward discount rate.
- g- acceleration of gravity.
- G- the set of possible goal locations that can lead to a collision assuming constant velocity motion.
- g^π- the pedestrian's goal location.
- h- the cone of initial locations that lead to a collision assuming that the AV moves at a constant velocity, and that the pedestrian has a maximal speed of $\|v_{max}^\pi\| = 3ms^{-1}$.
- I- indicator function, indicating a collision.
- I_g- indicator function, indicating the pedestrian has reached its goal.

- J- policy gradient loss function.
- J_μ- initial pedestrian placement's loss.
- J_π- pedestrian behaviour mode's loss.
- J_ρ- AV's loss.
- K- number of traffic scenes in the dataset/ available in simulator.
- k- iterator.
- λ_*- the scaling of a general reward term R_*.
- λ_a- the scaling of the reward term R_a in r_μ.
- λ_a^π- the scaling of the reward term R_a in r_π.
- λ_d- the scaling of the reward term R_d and R_d^π in r_μ and r_π respectively.
- λ_{dist}- the scaling of the reward term R_{dist}.
- λ_G- the scaling of the reward term I_g.
- λ_g- the scaling of the reward term R_g.
- λ_k- the scaling of the reward term R_k and R_k^π in r_μ and r_π respectively.
- λ_o- the scaling of the reward term R_o.
- λ_ϕ- the scaling of the reward term R_ϕ.
- λ_p- the scaling of the reward term R_p.
- λ_p^ρ- the scaling of the reward term R_p^ρ.
- λ_s- the scaling of the reward term R_s.
- λ_s^ρ- the scaling of the reward term R_s^ρ.
- λ_v- the scaling of the reward term R_v.
- λ_v^ρ- the scaling of the reward term R_v^ρ.
- l_1, l_2- convolutional layers of the neural network μ_Θ.
- L_1, L_2- up-sampled layers of the neural network μ_Θ.
- μ- distribution of pedestrian initial locations.
- μ_g- distribution of pedestrian goal locations.
- μ_Θ- the policy gradient neural network modelling the pedestrian initial distribution in the scene.
- M- number of sampled pedestrian initial locations.
- m- iterator.
- N- number of sampled trajectories.
- n- iterator.
- Ω- the set of allowed values for (τ, x_0, s^μ).
- O- scene occlusion map.
- q- the probability distribution of scenes s^μ.
- P- prior.
- P_g- prior of goal location.
- p- world dynamics and noise.
- p_τ- the probability density function of τ.
- π- pedestrian behaviour model.
- ρ- AV's policy.
- $R = \sum_{t=0}^{T} \gamma^t r(s_t, a_t, s_{t+1})$- a vector of cumulative sums of rewards.
- R_*- A general reward term.
- R_a- A reward term that is 1 if the pedestrian agent collides with the AV.
- R_{coll} reward terms that penalize collisions for the pedestrian agent.
- R_{coll}^ρ the AV's reward terms that penalize collisions.
- R_d- a reward term that encourages the pedestrian to reside on non-zero pixels of D.
- R_d^π- a reward term that encourages the pedestrian to reside on non-zero pixels of D_T.
- R_{dist}- a reward term that encourages the AV to move.
- R_g- a reward term that encourages motion towards the goal g^π.
- R_k- a reward term that encourages the pedestrian to reside on a blurred D.
- R_k^π- a reward term that encourages the pedestrian to reside on D_k.
- R_μ- the one step reward function of μ: the cumulative sum of rewards $R_\mu(x_0) = \sum_t \gamma^t r_\mu(x_t, a_t^\pi, y_t, a_t^\rho)$.
- R_o- A reward term that measures the ratio of pixels of the AV overlapping with the sidewalk..
- R_p- A reward term that is 1 if the pedestrian agent collides with another pedestrian.
- R_p^ρ- A reward term that is 1 if the AV collides with a pedestrian.
- R_{ped}- reward terms that encourage motion in areas frequently visited by pedestrians.
- R_ϕ - a reward term that discourages large unnaturally changes in the pedestrian's pelvis.

- R_ρ- The AV's cumulative discounted reward.
- R_{STPN}- the $STPN$ model reward in Table 2 in the main paper.
- R_s- A reward term that is 1 if the pedestrian agent collides with static objects.
- R_s^ρ- A reward term that is 1 if the AV collides with static objects.
- R_v- A reward term that is 1 if the pedestrian agent collides with an external vehicle.
- R_v^ρ- A reward term that is 1 if the AV collides with an external vehicle.
- $r = [r_\mu, r_\pi, r_\rho]$ - the joint reward vector of the pedestrian initial distribution model μ, the pedestrian behaviour model π and the AV model ρ.
- r_μ- the pedestrian initial distributions' reward function (per timestep).
- r_π- the pedestrian behaviour model's reward function.
- r_ρ- the AV's reward function
- S- scene semantics.
- $s^\mu = (S, D, OP)$- the pedestrian initial location model's state.
- s^π- the pedestrian behaviour model's state.
- s^ρ- the AV's state.
- s_{max} - maximal collision speed for the pedestrian to collide with the AV from location x_0.
- s_t- the AV and the pedestrian behaviour model's state at time t.
- s_t^ρ- the AV's state at time t.
- s_t^π- the pedestrian behaviour model's state at time t.
- s_x- the size of the bounding box of the pedestrian agent.
- s_y- the size of the bounding box of the AV.
- σ_ρ- the AV model's standard deviation.
- t - timestep.
- t_{min} - minimal collision time for the pedestrian to collide with the AV from location x_0.
- T - last timestep of an episode.
- Θ-parameters of μ_Θ.
- τ- the pedestrian and AV's state-action history $(a_0, ..., s_T, a_T, s_{T+1})$.
- v_0^π is the pedestrian agent's initial velocity.
- v_0^ρ is the AV's initial velocity.
- v_{max}^π- pedestrian's maximal possible speed.
- v_{max}^ρ- the AV's maximal possible speed.
- x- a point in the scene.
- x_0- initial position of pedestrian.
- x_t- pedestrian's position at timestep t.
- $x_\perp$ - the otrohogonal projection of a point x in the line $\hat{y}_t$. A point that is on the the AV's future trajectory, and is closest to the point x.
- y_t- AV's position at timestep t.
- $\hat{y}_t$ - the constant velocity prediction of the AV's future motion
- w- learn-able weight in the AV's model.

References

[1] M. Priisalu, C. Paduraru, A. Pirinen, and C. Sminchisescu. Semantic synthesis of pedestrian locomotion. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.

[2] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 1992.